Universitat Politècnica de València

Departamento de Sistemas Informáticos y Computación

Improvements to traffic flow in high pollution scenarios in Valencia

MASTER'S THESIS

Master's Degree in Computer and Network Engineering

Author

José Daniel Padrón Pérez

Advisor

Dr. Carlos Tavares Calafate

Date

June 29, 2021

Abstract

Traffic jams are the main cause of vehicular pollution, which at the same time are one of the main problems in many large metropolitan areas, and municipal administrations are looking for effective methods to improve the quality of life of citizens, especially in downtown areas and pollution-sensitive sites such as hospitals or schools. One of the methods recently adopted is to close these areas to traffic. However, the impact of these methods has not been studied in depth.

In this paper we seek to improve traffic flow in the city of Valencia when traffic is limited for environmental reasons, specifically due to a high pollution scenario. In particular, we analyze the impact of cutting all streets in a major district (*Ciutat Vella*) to avoid pollution in that area. Afterward, we show how our proposed routing algorithm is able to reroute traffic throughout the city without having traffic jam problems associated with this cut. In addition, we determine how the total vehicle emissions in the city vary due to the applied traffic restrictions.

Experimental results show that, even by closing all streets in the district, pollution rates decrease by 2.5-4%, which makes us reflect positively on the applicability and effectiveness of these methods when using our route choice system.

Keywords: Vehicle Networks; traffic management; vehicle pollution; SUMO; ABATIS.

Resumen

Los atascos son la principal causa de la contaminación vehicular, que al mismo tiempo es uno de los principales problemas de muchas grandes áreas metropolitanas, y las administraciones municipales buscan métodos eficaces para mejorar la calidad de vida de los ciudadanos, especialmente en las zonas céntricas y los lugares sensibles a la contaminación, como hospitales o escuelas. Uno de los métodos adoptados recientemente es cerrar estas zonas al tráfico. Sin embargo, el impacto de estos métodos no se ha estudiado en profundidad.

En este trabajo buscamos mejorar la fluidez del tráfico en la ciudad de Valencia cuando se limita el tráfico por razones medioambientales, concretamente debido a un escenario de alta contaminación. En concreto, analizamos el impacto de cortar todas las calles de un distrito importante (*Ciutat Vella*) para evitar la contaminación en esa zona. Posteriormente, mostramos cómo nuestro algoritmo de enrutamiento propuesto es capaz de redirigir el tráfico por la ciudad sin tener problemas de atascos asociados a este corte. Además, determinamos cómo varían las emisiones totales de los vehículos en la ciudad debido a las restricciones de tráfico aplicadas.

Los resultados experimentales muestran que, incluso cerrando todas las calles del distrito, los índices de contaminación disminuyen entre un 2,5 y un 4%, lo que nos hace reflexionar positivamente sobre la aplicabilidad y la eficacia de estos métodos al utilizar nuestro sistema de elección de rutas.

Palabras clave: Redes vehiculares; gestión de tráfico; contaminación vehicular; SUMO; ABATIS.

Contents

Acronyms

ABATIS Automatic Balancing of Traffic through the Integration of Smartphones with vehicles

CH Contraction Hierarchies

CSV Comma Separated Values

DGT *Dirección General de Tráfico*

DLR German Aerospace Center

MLD Multi-Level Dijkstra

OMNeT++ Objective Modular Network Testbed in C++

OSRM Open Source Routing Machine

RSUs Roadside Units

SEPAR Spanish Society of Pneumology and Thoracic Surgery

SNA Social Network Analysis

SNC Social Networking Concepts

SUMO Simulation of Urban MObility

TraCI Traffic Control Interface

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

VANET Vehicular Ad-hoc Network

VSN Vehicular Social Networks

XML Extensible Markup Language

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem statement

Traffic jams associated with the use of private vehicles in large cities are a common phenomenon. In addition to causing stress and other personal discomfort, they also lead to an increase in CO_2 emissions.

Valencia is the fourth city in Spain in terms of traffic congestion, according to data from the ”Traffic Index TomTom 2020” report [1], which analyses traffic jams and traffic volume of 416 cities around the world, 25 of which are from Spain. Data is obtained from more than 600 million drivers who use TomTom technology in navigation devices. All the data provided represents as a percentage how much extra time it takes to complete an average journey that would take 30 minutes without traffic.

The level of congestion in Valencia last year was 17 percent, i.e. a half-hour journey actually took 35.1 minutes on average. In Barcelona, the city with the highest percentage of congestion, with 22 percent, a similar journey took 36.6 minutes on average.

In the quest to reduce CO_2 emissions, some cities have taken actions as for example Barcelona and Valencia, which have restricted traffic in the city center, as well as opting for sustainable urban transport, as the implementation of a network of bicycle lanes that connect the entire city in the case of Valencia.

In terms of health, pollution, according to the Spanish Society of Pneumology and Thoracic Surgery (SEPAR) data [2], generates 10,000 deaths per year in Spain and 7 million in the world, being emissions of polluting gases produced by traffic one of its main causes. Traffic jams are one of the main causes of pollution in large cities, due to the high concentration of vehicles and their stop-and-go traffic.

In conclusion, this project aims to provide an additional solution to complement the measures that have already been taken. Specifically, the areas most affected by pollution, by reducing the amount of gases emitted in these areas. By knowing the pollution rates in each district of the city, drivers can be offered alternative routes to alleviate pollution in these worst affected areas.

1.2 Previous work

This master's thesis builds upon two previous projects, one carried out by Zambrano-Martinez [3] in his PhD Thesis, and another one by Terol Lloret [4] in his Master's Thesis. The first one developed a route management solution based on a route recommendation service called Automatic Balancing of Traffic through the Integration of Smartphones with vehicles (ABATIS), along with a "Traffic Prediction Equation" which can predict future traffic conditions, and consequently improve it in terms of average speed and travel time. This led to a reduction of traffic congestion and to the improvement of traffic flow. The second one extended the first by studying how to reduce traffic in some areas by introducing a fictitious speed penalty. His work evaluated the impact of cutting certain roads into the general traffic of the city, and analyzing how speed reduction affects vehicles on different types of roads.

1.3 Goals

The main goal of this thesis is to propose a novel approach to regulate traffic so that the vehicle route selection is based on environmental criteria. To accomplish this it is necessary to enhance the ABATIS route server (see 3.1) so that it can take different criteria into consideration to compute routes. This criteria can either be based on the pollution data retrieved or on specific sensitive areas we want to protect, such as: daycare centers, schools or nursing homes. At the same time, this Master's Thesis has the following goals:

- Evaluate the variation of pollutants emitted by vehicles when restricting the traffic in a key district of Valencia.
- Evaluate the variation of traffic flow metrics when restricting the traffic in a key district of Valencia.
- Define future works.

1.4 Structure

This Master's Thesis consists of six chapters organized as follows:

- Chapter 1 defines the aim of the project, while also putting it into context.
- Chapter 2 presents the state of the art, where references are made to different related works that are similar to our own.
- Chapter 3 provides an overview of the tools used for simulations and for reaching the thesis goals.
- Chapter 4 presents the proposed approach to address the target problem; likewise, it details the models and modifications done to the project.
- Chapter 5 presents experimental setup and the results obtained, including discussion.
- Chapter 6 presents the main conclusions of the Master Thesis, and the future work lines that can be extracted from this project.

Chapter 2

Related works

In this chapter some academic works related with this project are analyzed, even though their main goal is not specifically related with the analysis of the level of diverse pollutants when applying traffic restrictions.

2.1 EcoTrec

In [5] Doolan and Muntean present a vehicle routing solution to reduce their carbon emissions without significantly affecting travel times. The model is called EcoTrec, and it is based on a Vehicular Ad-hoc Network (VANET) where vehicles exchange messages related with traffic and road conditions, such as average speed on the road. After getting the necessary information, a fuel efficiency model is built and, due to this, less greenhouse pollutants are emitted while maintaining low traffic congestion levels.

In figure 2.1 we can observe their architecture model, which consists on the EcoTrec routing engine, and on three different models: vehicle, road and traffic.

On the one hand, the EcoTrec engine has the aim of calculating the most fuel-efficient route by looking at the efficiency of individual road segments considering different factors: road condition, traffic condition, and the weight of the road.

On the other hand, the models have the following functions:

- The vehicle model is built and updated by each vehicle based on their GPS sensors, speedometer and accelerometer.
- The road model is maintained by a central server and updated with information of nearby roads.
- The traffic model is based on the vehicle's local traffic conditions.

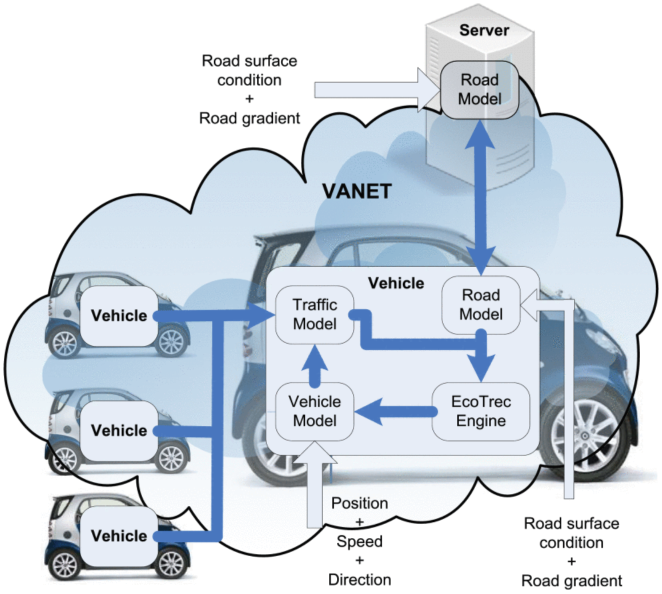

Figure 2.1: EcoTrec system architecture.

In terms of results, their solution showed an average reduction of fuel consumption comparable to other algorithms, while maintaining the traffic flow conditions. Contrarily, for traffic congestion situations, their solution is not appropriate because vehicles calculate alternative routes by their own, without considering neighbor's routes, which leads to maintaining the congestion levels. In addition, they do not measure the different pollutant metrics (only CO_2), so they are not as explicit as we are in our work.

2.2 *i*MOB

*i*MOB is an **i**ntelligent urban **MOB**ility management system that was proposed by Akabane *et al.* [6]. The project is based on applying the Vehicular Social Networks (VSN) concept and analyzing the impact of using Social Network Analysis (SNA) and Social Networking Concepts (SNC) measures for urban traffic flow. At the same time, its architecture is based on a three-level system (classified from bottom to top): environment sensing, vehicle ranking mechanism, and altruistic rerouting decision. The main features of each level are described below.

- Environment sensing is responsible of obtaining the local awareness through vehicle crowdsensing. It enables users of the VSN to solve problems in collaboration with each other. For example, by exchanging environmental sensing data about the road traffic conditions, VSNs can offer details about the real time road conditions, which will be transmitted in order to help in the decision process of the urban mobility management system.

- The vehicle ranking mechanism selects the best located vehicle in the network based on the communication links between them. Then, the selected vehicle is responsible for the information aggregation and knowledge generating processes.

- An altruistic rerouting decision performs the vehicle rerouting strategy in a collaborative way in order to avoid the congestion spot. To this end, two concepts of social networks are applied, namely social interactions and virtual community. In order to do that, the chosen alternative route is shared among surrounding vehicles. The main goal is to divert the maximum number of vehicles from traffic congestion along their route.

Paper results show that the use of social interactions and virtual community in the vehicular environment helps to reduce the average travel time and CO_2 emissions; nevertheless, they do not study other pollutants neither average speed nor traveled distance.

2.3 SmartFlow

SmartFlow is a VANET protocol proposed for Vehicle to Infrastructure (V2I) communications by Khan and Koubaa [7]. The main goal is to avoid long waiting delays due to traffic signals and, for this purpose define state a protocol that can detect congestion and suggest vehicles the speed for circulating and for crossing traffic signals before they are red. All this can be done by frequently sending beacon messages to other vehicles and Roadside Units (RSUs).

The protocol system model is three-legged, including: vehicle model, road model and vehicular mobility model. Vehicle model assumes that all vehicles are intelligent and able to communicate to each other and with the RSUs through a wireless transceiver. The road model aims at transmitting the characteristics of each road segment (road length, speed limit, number of lanes, number of RSUs, etc.) to vehicles. The vehicular mobility model is the Car-following model, which primarily enforces the rule of keeping a safe distance ahead.

In terms of results, the protocol performs well compared to others in the paper. In contrast, unlike us, this system can only be implemented when vehicles are 100% autonomous, meaning that it is not applicable for the present day. We also differ from them because we measure environmental parameters and evaluate the behavior of our model based on these parameters.

2.4 REACT

In [8] Gomides *et al.* present REACT, a traffic management solution with the aim of minimizing vehicle congestion in smart cities. For this purpose, they suggest a VANET based on a Vehicle to Vehicle (V2V) network architecture, where vehicles behave as intelligent agents as they are able to process and transmit information between each others. Communication is rooted on two phases: request and response. For the request phase, vehicles must ask for information about the displacement in neighbor road segments. For the response phase, vehicles present on these roads must reply with their displacement analysis.

This is a good solution to reduce communication overhead and also their results show a good performance in terms of travel time. Nevertheless, this approach seems to be optimistic, because they do not use real traffic traces for the experiment.

2.5 dEASY

Akabane *et al.* presented a novel distributed traffic management system called dEASY (**d**istributed v**E**hicle tr**A**ffic management **SY**stem) [9]. This is an infrastructure-less system based on a three-layer architecture which consists of: environment sensing and vehicle ranking, knowledge generation and distribution, and knowledge consumption. Their main characteristics are described below.

- The environment sensing and vehicle ranking layer collects raw traffic data in real time, such as position and speed, so that it becomes possible to extract from it an accurate knowledge about the traffic conditions of the roads.
- The knowledge generation and distribution layer deals with processing and aggregation of the raw data. In this layer, the individual raw data provided by the environment sensing and vehicle ranking layers are gathered and grouped until new knowledge comes out. In addition, this layer also delivers services to customers, e.g. route suggestion.
- The knowledge consumption layer provides a knowledge-based decision. This decision can have two approaches: selfish (decisions are made to seek your own benefit), and altruistic (decisions are made to seek the benefit of the overall system).

Results show that the system performs well in terms of travel time and CO_2 emissions. However, their approach is not as explicit as ours because they only measure CO_2, failing to study other important pollutants such as microparticles or nitrogen oxide emissions.

Chapter 3

Simulation tools

To carry out this project we have used the Simulation of Urban MObility (SUMO) traffic simulator [10], the Objective Modular Network Testbed in C++ (OMNeT++) network simulator [11] , and our route server named ABATIS [12]. These elements are combined to achieve the desired functionality, as depicted in figure 3.1.

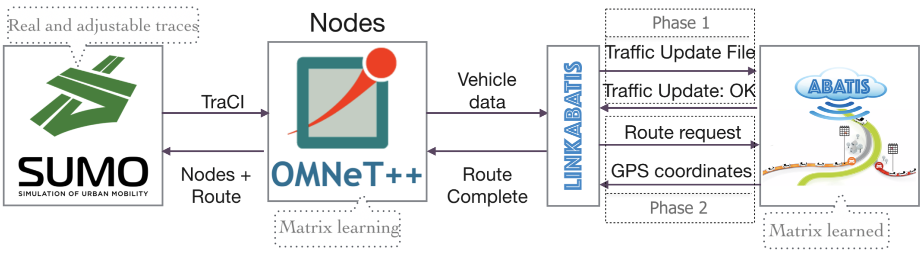

Figure 3.1: Overview of our simulation framework.

Notice that Traffic Control Interface (TraCI) provides the required bonding between the OMNeT++ network simulator and the SUMO traffic simulator. Additionally, LinkABATIS provides the necessary interface between OMNeT++ and the ABATIS route server; concerning INET, it provides the essential network libraries and models of the OMNeT++ environment.

In this chapter we are going to explain these different tools, and how they are related, highlighting their main features in our target context.

3.1 ABATIS route server

ABATIS [12] is a route server based on the Open Source Routing Machine (OSRM) [13], and it is implemented in C++, being open-source. As input data it uses the files generated by OpenStreetMap, both in PBF format or in OSM XML. Its functionality can be divided into two essential parts: the retrieval of routes by the client, and the tools that are used to generate these routes.

On one hand, with respect to obtaining the routes, this is done through an HTTP GET request such as the one below:

```
http://server_ip:5000/route/v1/driving/long1,lat1;
long2,lat2.json?steps=true&overview=full&geometries=geojson
```

Where $server\_ip$ is the IP address of the ABATIS route server, 5000 is the port used to establish the TCP connection, $long1$ and $lat1$ are the longitude and latitude (in decimal degrees) of the starting point, and $long2$ and $lat2$ are the longitude and latitude (in decimal degrees) of the target point. The response to this request is provided using the JSON format, and it has a structure similar to the one present in figure 3.2.

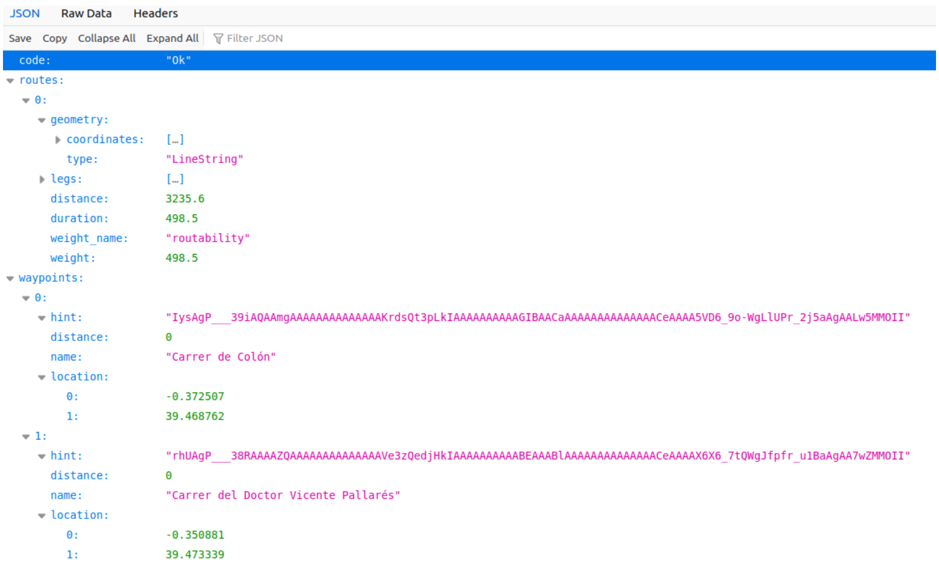

Figure 3.2: Examples of an ABATIS response.

This example refers to a route between "Carrer de Colón" and "Carrer del Doctor Vicente Pallarés", which has a distance of $3235.6m$, an estimated travel time of $498.5s$ and a weight of 498.5. Notice that, if we vary the weight of the roads, it affects the weight of the route; also, if there is an alternative route with a lower weight, the server will choose it.

On the other hand, in relation to the tools used to generate the routes, ABATIS implements the following ones:

- **osrm-extract** obtains information of the area that we extract from OpenStreetMap. In our case, it obtains the information of the city of Valencia. With this data it creates three files necessary for route calculations: an .osrm file with the essential data to calculate the routes; an .osrm.restriction file with the traffic restrictions of the streets; and an .osrm.names file which contains the names of all the roads.

- **osrm-prepare** generates from the files created by osrm-extract, which consist of the files needed for the calculation of routes, such as: .osrm.edge, .osrm.fileIndex, .osrm.hsgr, osrm.nodes, .osrm.ramIndex, and .osrm.timestamp.

- **osrm-customize** is used to modify the weight of particular tracks, allowing us to update the map according to our needs and interests.

- **osrm-datastore** complements the previous one, since it is the one that allows us, without having to stop and re-execute the server, to store the modifications that we have indicated.

- **osrm-routed** is carried out by the ABATIS route server, and it allows us to specify two different types of processing: Contraction Hierarchies (CH) and Multi-Level Dijkstra (MLD). CH is best suited for use cases where the query performance is key, especially for long-distance matrices. MLD is more suitable for use cases associated with shorter distances, where query performance still needs to be very good, and live updates of data must be performed, e.g. for periodic traffic updates. In our case we will use MLD because we will use live updates of traffic data.

The connection between all these tools is shown in figure 3.3. As we can see, when the server starts, it uses the whole bunch of tools, being this a costly process in terms of time and memory. Contrarily, when it is already started, and because our approach requires periodic traffic updating, instead of restarting every single time, it simply executes the osrm-customize tool to update weights, and the osrm-datastore tool to store these values. After this process, an updated route that follows our criteria is obtained.

3.2 SUMO traffic simulator

Developed in 2001 by the German Aerospace Center (DLR), and made available as Open Source software in 2002, SUMO [10] is a highly portable, microscopic, continuous multi-modal traffic simulation package designed to handle large networks. Its features include, apart from those mentioned, communication with

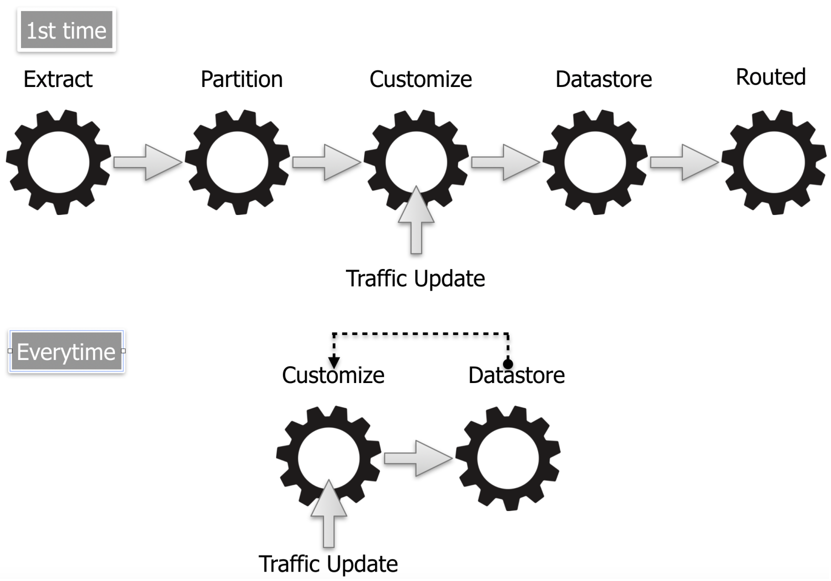

Figure 3.3: ABATIS working routine.

network simulators such as OMNeT++ or ns-3, and the ability to import road networks in formats such as OpenStreetMap or OpenDrive.

SUMO is basically a microscopic traffic simulator where each vehicle is explicitly represented, defined by an identifier, departure time, and the route to follow. Additional details such as speed, position, or arrival parameters can be added if desired. Besides this, it is also possible to assign noise or vehicular pollution variables, as it will be detailed for our vehicular emission models (see section 3.2.1). We can see an example of its graphical interface in figure 3.4.

In terms of vehicular traffic definition, this can be generated using different sources. In our case, where we address a large-scale scenario, Origin-Destination (O-D) matrices is the preferred method, because these matrices describe the movement of traffic indicating the number of vehicles in an area, and when trips actually start.

Finally, SUMO is capable of generating multiple output files for each simulation. It provides data from induction rings to the positions of each vehicle at all times. At the same time it can give more complex data, such as: information of traveled time, distance, and pollutants emitted.

3.2.1 Emission models

SUMO has different tools and models, but in this section we will focus on the one used to reach our objective: the emission model. It includes four models: HBEFA

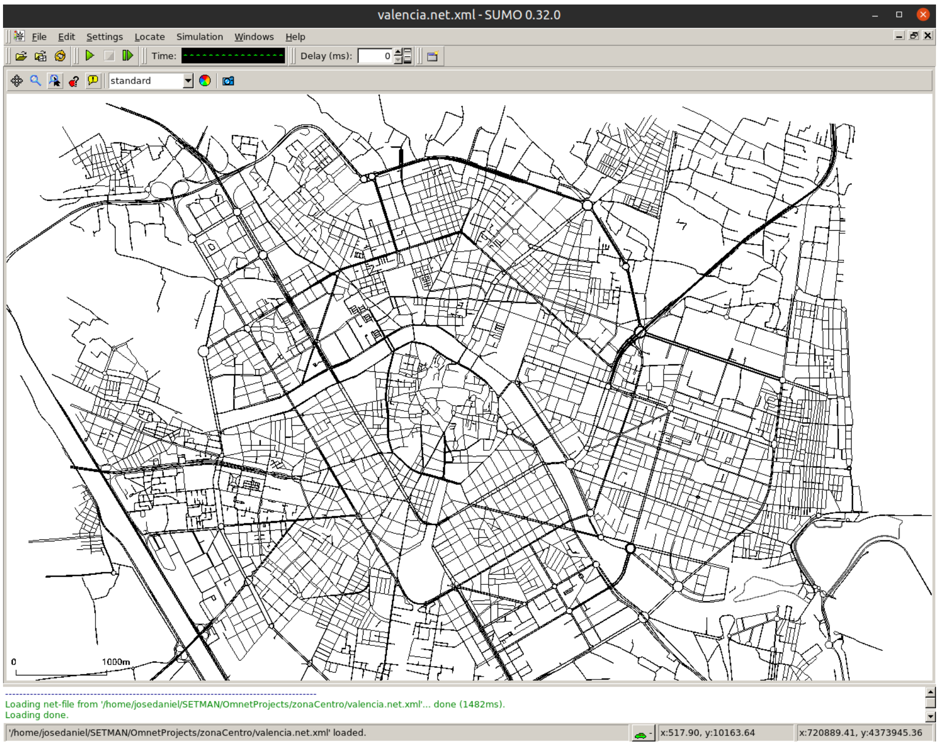

Figure 3.4: SUMO graphical interface.

v2.1-based, HBEFA v3.1-based, PHEMlight, and Electric Vehicle Model. In our case we will use the HBEFA model [14], specifically version 3.1 (HBEFA3). This model is based on an open source emissions database, while PHEMlight is closed and commercial, and the Electric Vehicle Model was created by Kurczveil *et al.* [15].

Within the HBEFA3 model there are different emission classes for vehicles, going from heavy duty emission to passenger and light delivery. In our case we will opt for the second one because it represents a better approach for an overall view of traffic. Inside it are the classes that interest us, such as: PC_G_EU4 (gasoline driven passenger car Euro norm 4) and PC_D_EU6 (diesel driven passenger car Euro norm 6).

The model also measures different types of pollutants, which are: CO_2, CO, HC, NO_x and PM_x. These will be useful to us in order to detect the variation of emission levels when applying our solution.

3.3 The OMNeT++ network simulator

OMNeT++ [11] is an extensible, modular, and component-based C++ simulation library and framework, mainly for building network simulators. The concept of "Network" is understood in a broader sense, including wireless and wired communication networks, chip networks, queuing networks, etc. Domain-specific functionality, as well as support for sensor networks, ad-hoc wireless networks, and Internet protocols, is provided by model frameworks, developed as independent projects, such as INET.

Since its beginning, OMNeT++ was designed to support large network simulations. To this aim, its design follows the next criteria [16]:

- Models need to be hierarchical, and built from reusable components as much as possible.
- Software should facilitate the visualization and debugging of simulation models.
- Software itself should be modular, customizable, and should allow to embed simulations into larger applications.
- Provide open data interfaces, to generate and process input and output files with commonly available software tools.
- Offer an Integrated Development Environment to facilitate model development and results analysis.

Regarding the first item in the above list, OMNeT++ provides a component architecture for modules, but first we should know what is a model. A model consists of modules that communicate using message passing. Active modules, also known as simple modules, are programmed in C++. They can be grouped into compound modules (number of hierarchy levels is not limited), and then assembled into larger components and modules using a high-level language (NED). These communicate with each other through gateways both within the composite modules and outside. The network module is the system module, which is a special compound module type without gates to the external world. Finally, the main utility of joining certain modules is to give them common characteristics and parameters, which can facilitate programming. An illustration of the model hierarchy can be seen in figure 3.5.

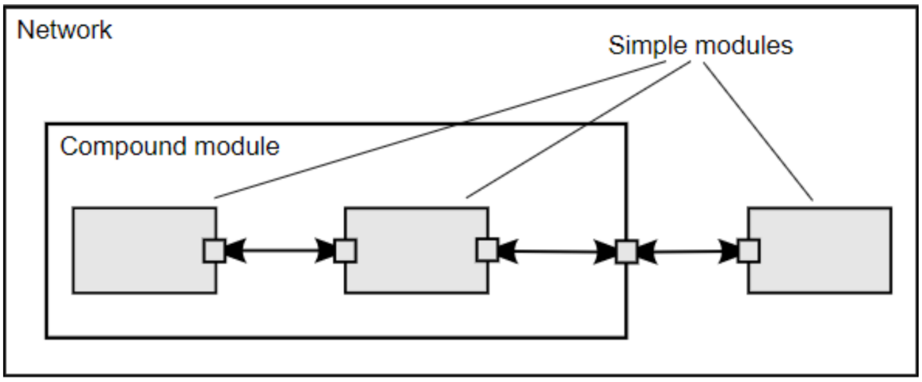

Figure 3.5: OMNeT++ model hierarchy.

3.3.1 INET

Within OMNeT++ we ca find INET [17], an open-source model library for the OMNeT++ simulation environment. It provides protocols, agents and other models in order to work with communication networks. These protocols and models can be: models for the Internet stack (TCP, UDP, IPv4, IPv6, etc.), wired and wireless link layer protocols (Ethernet, IEEE 802.11, etc), VANET protocols, and many others. To be precise, in our project we will focus on the use of the VANET protocols.

INET is built around the concept of modules that communicate by message passing. Agents and network protocols are represented by components, which can be freely combined to form hosts, routers, switches, and other networking devices. New components can be programmed by the user, and existing components have been written so that they are easy to understand and modify.

TraCI

As a part of INET we can find TraCI [18], an open source API with the goal of communicating SUMO and OMNeT++. It is based on a client/server TCP communication architecture where both endpoints exchange messages; in particular, SUMO acts as a server, while OMNeT++ acts as a client. The message format is shown in figure 3.6.

Once the TraCI TCP connection is established, the network simulator, in this case OMNeT++, controls the SUMO vehicle simulator. This enables having mobility updates for each vehicle, which will affect the city-wide mobility thanks to the information generated by the vehicular network. To ensure synchronization between the simulators, the network simulator periodically sends a command to the traffic simulator containing the actual time of the simulation, plus a subsequent

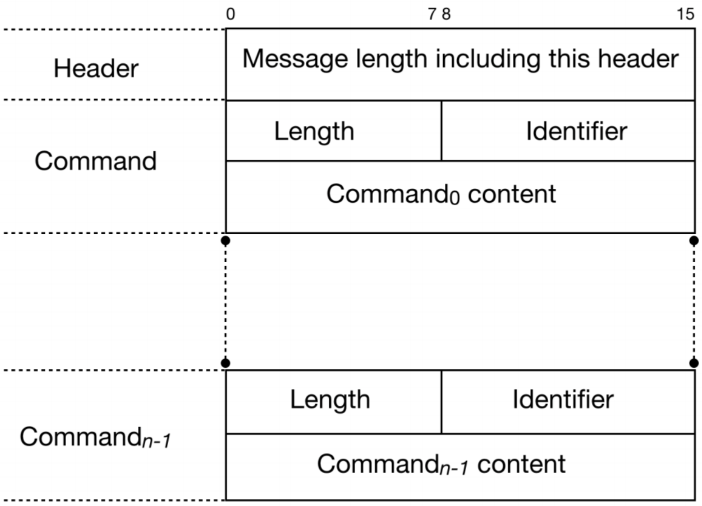

Figure 3.6: TraCI message format.

time step of the simulation. The vehicle simulator performs the next step of the simulation and sends the positions of the vehicles back to the network simulator. This is possible because SUMO is one step ahead of OMNeT++.

3.4 LinkABATIS

The linkABATIS [19] application is a key element of the project. It has been fully developed in Python and, among other functions like generating the necessary SUMO files for simulation, it is in charge of the communication between the ABATIS route server and the OMNeT++ network simulator. As it is known, the OMNeT++ simulator is not capable of interpreting geographic coordinates. LinkABATIS, before making the HTTP GET request to ABATIS, does a translation from edges to coordinates and, when it has the response from the server, it will do the reverse process in order to maintain a bidirectional communication between the simulators and the route server. This process is related with the Phase 2 of the communication, as we can see in figure 3.1. It can be observed in more detail in figure 3.7.

Regarding Phase 1 of the communication process, linkABATIS creates a Traffic Update file in order to change the values of the weights of the streets following our environmental criteria, which we will describe in the next chapter. This file is stored as Comma Separated Values (CSV), and it includes the information of the source node, destination node and the updated speed in kmph.

After creating the CSV traffic update file, linkABATIS will check if the server

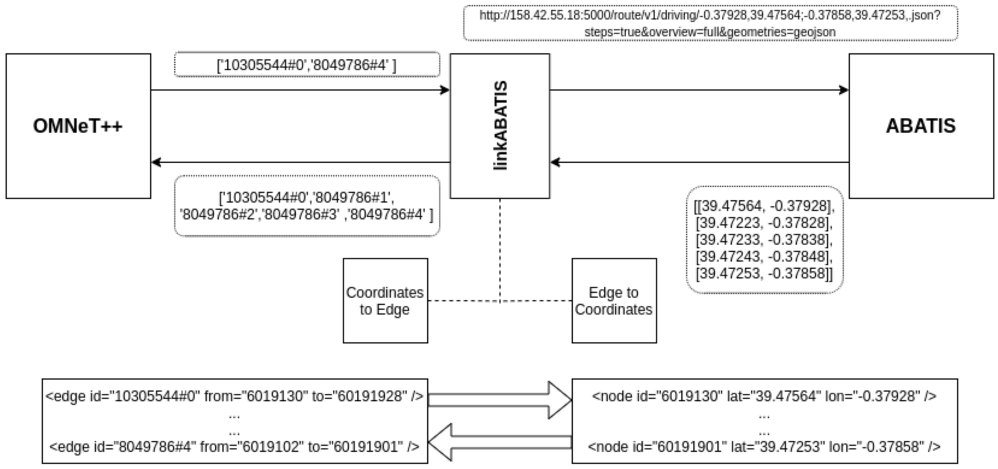

Figure 3.7: OMNeT++ – linkABATIS – ABATIS communication process.

is alive by pinging it and, if so, it will send the file. Then, as we saw in section 3.1, it will execute the osrm-customize and osrm-datastore tools in order to obtain the updated routes.

Chapter 4

Proposed solution

In this chapter we will focus on presenting our proposed solution to address the problem of traffic flow in high pollution scenarios in the city of Valencia. To this end, we have made use of the simulation tools presented in the previous chapter. These are related through the flowchart presented in figure 4.1, which we will proceed to describe below.

As we can observe, linkABATIS is at the top of the diagram. First, it is responsible for reading the real traffic traces obtained by Zambrano-Martinez *et al.* in [20], and then store them in an SQL database. After that, two simulations can be performed: a default one and one in which we apply our solution for high pollution scenarios. For the first case, linkABATIS will directly generate the necessary files for SUMO. For the other case, linkABATIS will apply our echo algorithm, which will be discussed later. After being applied, linkABATIS will generate the necessary files for SUMO. At this moment we will have everything ready to start the simulation.

For simulating we must first run SUMO, which at the same time initialize the TraCI server. Then, after configuring OMNeT++ parameters, we can proceed to execute it. On one hand, when the simulation is finished, SUMO will give us the vehicle's data emissions in an Extensible Markup Language (XML) format, which we will then convert into a CSV file with the help of SUMO tools. On the other hand, OMNeT++ will generate CSV files with the traffic data. Finally, we then combined the files with Python scripts developed for this project, and graph the results using Matlab.

After having a global vision of the simulation flow required for this thesis, we will now discuss in detail certain key aspects such as: how to cut streets (minimize traffic flow), how to generate and measure the traffic parameters, how to achieve the objective of measuring the pollution levels, and how to obtain the final results.

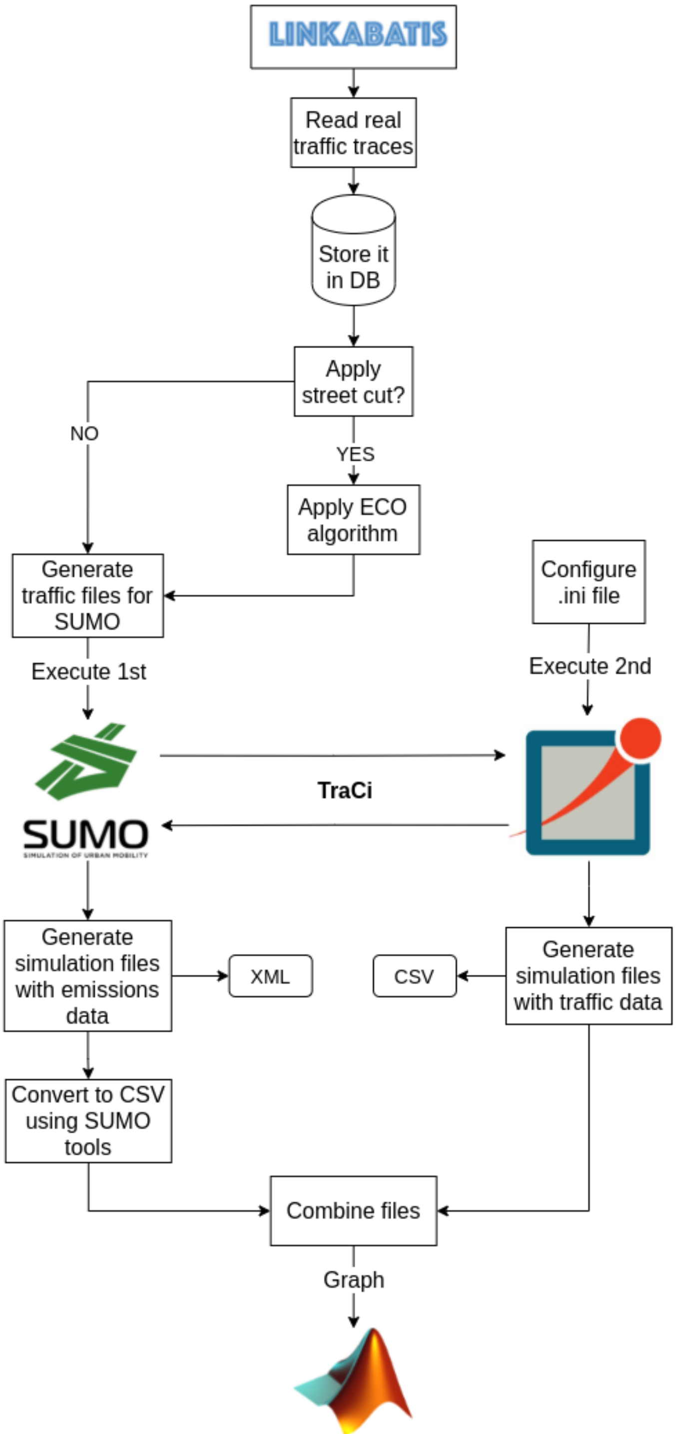

Figure 4.1: Simulation flowchart.

4.1 Street cut generation

Proceeding with the cut of particular streets, we have redesigned the algorithm presented by Terol Lloret in [4] to fulfill our requirements. This algorithm is able to tailor to any traffic restrictions defined by city administrations based on environmental criteria. It can be observed below, and it is important to note that, for the description of the algorithm, we consider the ABATIS route server is active and running.

Algorithm 1: Proposed ECO algorithm to apply traffic restrictions.

Input: OSM file, Database, CSV file with streets name, α value
Output: Traffic update file, Updated DB

1 Open database;
2 streetsList = csvToList(CSV file with streets name);
3 streetsIdList = findStreetId(streetsList);
4 **for** *streetId* ***in*** *streetsIdList* **do**
5   originalSpeed = getOriginalSpeedFromDB(streetId);
6   newSpeed = orginalSpeed / α value;
    `/* ECO function */`
7   updateSpeedInDB(newSpeed, streetId);
8 **end**
9 Save changes in Database;
10 Create Traffic Update File;
11 Send copy to ABATIS server route;
12 Execute ABATIS Customize;
13 Execute ABATIS Datastore;
14 **for** *i* < *streetsIdList length* **do**
15   streetId = streetsIdList[i];
16   newRoute = requestNewRouteToABATIS(streetId);
17   updateRouteInDB(newRoute);
18 **end**
19 Close database;

Having a close look to Algorithm 1, we can observe it is divided in three main functionalities: it first deals with the updating of speed values in street segments, second with the ABATIS part, and third it requests the new routes and updates them in the database. Having said this, we will proceed to explain these functionalities in more detail.

The first functionality goes from line 1 to line 9. At first, we read the input CSV file with the name of the streets, and then convert it into a list. Afterwards, we apply a function for this list that searches on the database, and converts each name into one or various edge ids, depending on how many segments each street has. With the aim of applying our equation 4.1, for each edge, an iteration over all the elements of the list is done. This equation changes the traffic weights of the different street segments by diving the default velocity (V) by factor α. This is done in order to artificially reduce the speed of the street segment, and consequently simulating a street cut. Finally, all the speed values are updated and saved into the database.

$$f(\alpha) = \frac{V}{\alpha} \tag{4.1}$$

The second functionality goes from line 10 until line 13. It deals with the creation of the aforementioned Traffic Update File (see 3.4) and subsequent delivery to ABATIS. Finally, the Customize and Datastore tools are executed into ABATIS, in order to change the weights and have new routes based on our criteria.

The last functionality goes from line 14 until the end, and deals with the actualization of the route for each vehicle that goes through the selected streets. For this purpose, it requests to ABATIS the route, and then converts into edges, and update the route in the database.

4.2 Traffic parameters assessment

In order to evaluate the traffic-related parameters, we divide this section into two subsections: one where we will explain the details on how we generate the traffic to perform the simulations, and another one where we will evaluate the format of the obtained results, as well as the different traffic parameters obtained.

4.2.1 Traffic generation

As it was mentioned before, after applying the street cut algorithm, thanks to LinkABATIS, the necessary files to do the simulation in SUMO are created. These files are two: one file with the information of vehicles (.add file), and another one with the routes information (.rou file). Both can be seen in Listing 4.1 and 4.2, respectively. We will now proceed to describe the content of each file more in detail.

On the one hand, the .add file consists on three different elements: vType, routeDistribution and vehicle. First, the vType element is responsible of creating different vehicle models; in our particular case, it deals with the creation of the different emission class models, which we will cover in more detail in the next subsection (see 4.3). Second, the routeDistribution element is in charge of organizing the different route sub-elements. These route sub-elements refer to the routes inside the .rou file, so their goal is to be a link between both files. Finally, the vehicle element represents the vehicle who is going to be simulated. It has an id, a type (vType defined before), and a route (present in the routeDistribution and .rou file), among others.

On the other hand, the .rou file is formed by different route elements. These elements consist of an id and edges. The id corresponds to the refId of the route sub-element of the .add file, and the edges correspond the different street segments that conform the target route.

Listing 4.1: Example of an .add file.

```
<additional>
    <vType id="gasoline_b" emissionClass="HBEFA3/PC_G_EU3"/>
    <vType id="gasoline_c" emissionClass="HBEFA3/PC_G_EU5"/>
    <vType id="gas_oil_b" emissionClass="HBEFA3/PC_D_EU4"/>
    <vType id="gas_oil_c" emissionClass="HBEFA3/PC_D_EU6"/>
    <routeDistribution id="dd_4823219">
        <route refId="4823219_to_191715080#2" probability="0.33"/>
        <route refId="4823219_to_191715084#0" probability="0.33"/>
    </routeDistribution>
    <vehicle id="emitter_dd_4823219_0_0" type="gasoline_b" depart="0.0" departLane="0" departPos="26" departSpeed="5.56" route="4823219_to_191715080#2"/>
    <vehicle id="emitter_dd_4823219_1_0" type="gasoline_b" depart="0.0" departLane="1" departPos="26" departSpeed="5.56" route="4823219_to_191715080#2"/>
    ...
    ...
</additional>
```

Listing 4.2: Example of a .rou file.

```
<routes>
    <route id="4693573#1_to_9281469#1" edges="41197570#6 10380715#0 147061059#0 147061059#1 41058459#8 41058459#9 9281469#0 9281469#1"/>
    <route id="4693573#1_to_52092239" edges="41197570#6 10380715#0 10380715#1 10380715#2 10380715#3 10380715#4 10380715#5 119441497 11993589#3 52092239"/>
    ...
    ...
</routes>
```

4.2.2 Traffic data measurement

As already mentioned above, when the simulation is finished, two results are obtained; one in SUMO and one in OMNeT++. In this subsection we will cover the second one, as it is the one that contains the data we will use for traffic parameters evaluation.

OMNeT++ generates six different files, three of which are OMNeT++'s own files, and the other three which we create using the TraCI API. Having said that, we will now only take into account the CSV files generated through the TraCI API. From these CSVs we will focus on the one that provides us different data for each vehicle, such as: travel time, distance traveled and average speed. All the parameters contained in the CSV file are shown in the table 4.1.

Table 4.1: Values of OMNeT++ CSV file.

id vehicle	**departure time**	**arrival time**	**travel time**	**distance**	**total speed**	**average speed**

We now proceed to describe the content and units of measurement of some of them. The departure time is measured in milliseconds, and it indicates the simulation time when the vehicle departs. Arrival time has the same characteristics as departure time, but indicates the simulation time when the vehicle arrives at its destination. Travel time is the result of subtracting arrival time minus departure time. Distance is measured in meters, and it is the total distance covered by the vehicle during the simulation. Total speed is measured in meters per second, and it has the functionality of being the key to calculate average speed. Average speed is the result of diving total speed by travel time (in seconds).

4.3 Environmental parameters assessment

To address the assessment the environmental parameters we have divided this section into two parts: one in which we describe how we generate pollutants, and the data used for that purpose, and another one where we evaluate the format of the obtained results, as well as the different pollutant parameters obtained.

4.3.1 Pollutants generation

As it was discuss before, for our solution we use real traffic data in order to have a better approach. For this reason, we also use actual statistics for the traffic emissions model. These statistics are retrieved from the Spanish's Department of

Traffic or *Dirección General de Tráfico* (DGT) on their 2019 annual table statistics [21]. From it, we will take only the data related with passenger cars from the province of Valencia. We will now proceed to detail the content of our data.

In terms of number of passenger cars, Valencia has registered a total amount of 1,278,936 cars, and these can be organized according to two different criteria: the type of fuel, or an environmental label.

On the one hand, according to the type of fuel, the distribution can be seen in table 4.2, but this type of distribution does not fulfill our requirements because it does not distinguish an Euro4 norm car from an Euro6 norm, whose emissions are drastically lower.

Table 4.2: Vehicles distribution in Valencia according to fuel type.

Fuel	**Number of vehicles**	**Ratio (%)**
Gasoline	570,971	44.64
Diesel	704,918	55.12
Other	3,047	0.24

On the other hand, according to an environmental label [22], the arrangement can be observed in table 4.3; similarly to the previous one, this arrangement does not meet our requirements, because it does not differentiate diesel from gasoline, which is an important issue in terms of pollutant emissions.

Table 4.3: Vehicles distribution in Valencia according to an environmental label.

Environmental label	**Number of vehicles**	**Ratio (%)**
CERO	1,107	0.09
B	432,898	33.85
C	408,915	31.97
ECO	16,882	1.32
Without distinction	329,160	25.74
Unknown	89,974	7.03

Our solution is to combine the two distributions, dismissing the non-relevant values, so we will not consider 0 emissions vehicles neither hybrid (ECO) vehicles. For the unknown ones, and the ones with no distinction, we will assume they follow the distributions of percentages of diesel and gasoline cars of each tag. All in all, we will only consider four types of vehicles for our simulation: gasoline with B or C tag, and diesel with B or C tag.

Now for SUMO, as mentioned before in section 3.2.1, we have different emission classes for the model HBEFA3. In particular, the classes we will use are:

PC_G_EU3, PC_G_EU5, PC_D_EU4 and PC_D_EU6, which correspond to gasoline passenger cars of Euro3 and Euro5 norm, and to diesel passenger cars of Euro4 and Euro6 norm. These are selected according to the criteria of the DGT when assigning the environmental tag. The final relation between SUMO classes and DGT data can be observed in table 4.4, which is the one we used for generating the traffic. Finally, for our experiments, we have a 45% of gasoline cars (23% of B tag and 22% of C tag), and 55% of diesel cars (28% of B tag and 27% of C tag).

Table 4.4: Our estimation of the vehicle's distribution in Valencia according to type.

Vehicle type	Emission class	Volume (%)
Gasoline B	PC_G_EU3	23
Gasoline C	PC_G_EU5	22
Diesel B	PC_G_EU4	28
Diesel C	PC_G_EU6	27

4.3.2 Pollutants measurement

As already mentioned before, when the simulation ends two results are obtained, one in SUMO and one in OMNeT++. In this subsection we will cover the first one, as it is the one that contains the data we will use to assess environmental parameters.

SUMO generates two different XML files: a vehicle route file, which contains information about which route a vehicle took; and a trip info file, that includes information about each vehicle's travel info and pollutants generated. That being said, we will now only take into account the second file. Then, we will convert it into a CSV file using a script in Python provided by the SUMO tools. Finally, from it we will extract emissions, along with departure and arrival lanes (this is in order to put it in context on a map).

The pollutants measured, as stated earlier in section 3.2.1, are microparticles (PM_x), hydrocarbons (HC), nitrogen oxides (NO_x), emissions of carbon dioxide (CO_2), and carbon monoxide (CO). All the results obtained are expressed in milligrams (mg). With the measure of these contaminants we can reach our goal of verifying whether our solution is good for the environment or not.

4.4 Final results approach

When the simulation is finished, and all the files we want have been generated and processed, it is time to combine them in order to have an overall view of the results. When combining these files, we generate a single CSV file containing the following columns:

1. vehicle id
2. departure time
3. arrival time
4. travel time
5. distance
6. total speed
7. average speed
8. CO_2 emissions
9. CO emissions
10. HC emissions
11. NO_x emissions
12. PM_x emissions
13. vType
14. departureLane
15. arrivalLane

Then, with a Python script specifically developed for this purpose, we delete vehicles that do not have the same id inside the CSV. This is done because, when simulating, the total number of vehicles present in the simulation is usually not the same and, when interpreting results, this does not allow for a fair comparison. Therefore, instead of eliminating vehicles randomly, we eliminate vehicles that are not in both files, so that we obtain the same vehicles, and we can effectively check whether our solution is valid or not.

After cleaning the data by relying on a new Python script, we create another CSV file to analyze the most frequent source and destination points for each case. This file contains 6 columns, which are: coordinates of the origin (latitude and longitude in two columns), the weight of that coordinate (number of times it is repeated), coordinates of the destination (latitude and longitude in two columns), and the weight of that coordinate. Finally, when all this process is done, we proceed to import the CSVs into Matlab in order to present the results graphically.

Chapter 5

Experiment and results

To examine the effectiveness of the proposed solution, a key experiment has been carried out to assess the consequences of minimizing the traffic flow in the city center of Valencia, specifically in the district of *Ciutat Vella* (Old Town). It is one of the districts with the highest traffic flow levels in the entire city, as well as being the oldest, making it culturally and historically important. To contextualize *Ciutat Vella*, it is shown in figure 5.1. It has a surface of 1.69 km^2, and a population of 25,788 inhabitants, which leads to a density of 15,259 hab/km².

Figure 5.1: District of *Ciutat Vella* (highlighted in pink).

For carrying out the study, we create a set of vehicles that depart from various points of the city, in order to have more representative data. These sources points are depicted in figure 5.2, and, as it can be seen, are well distributed throughout the city, so that they are representative of real situations. According to the traffic data information that Zambrano-Martinez *et al.* obtained in [20], in our experiment 20,000 vehicles were circulating during a 15-minute period.

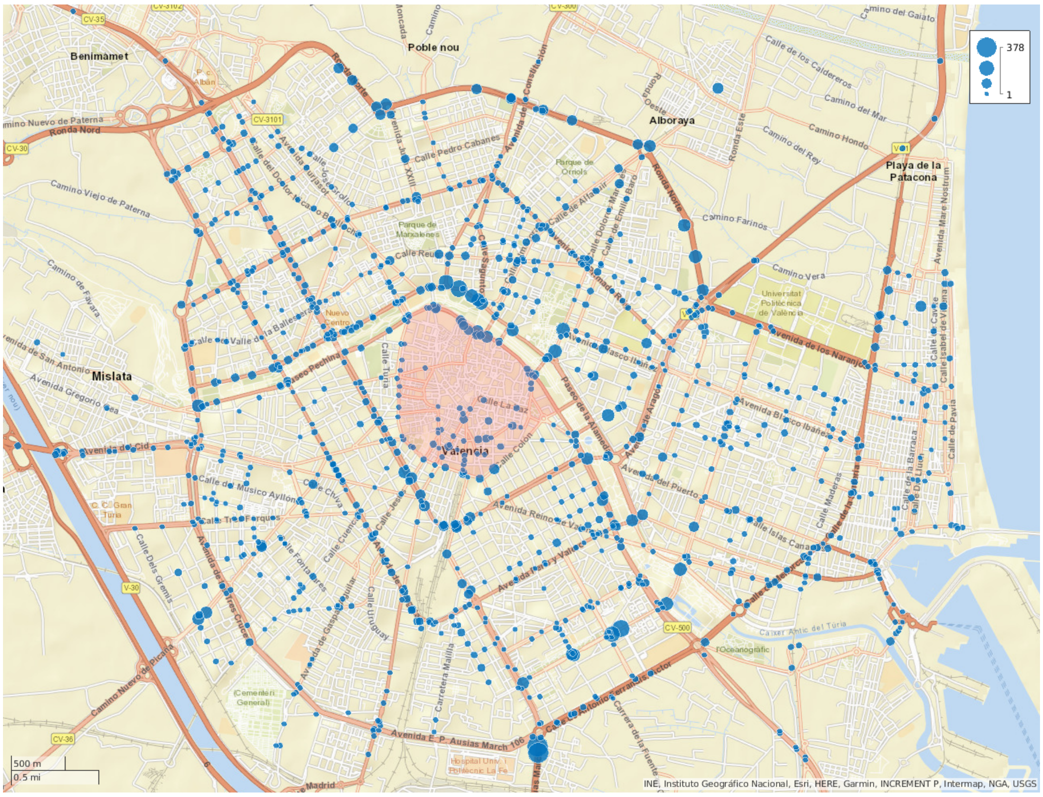

Figure 5.2: Simulation traffic sources with *Ciutat Vella* highlighted in pink.

Finally, to describe in detail our experiment and to evaluate its results, we have divided this chapter into two sections. In the first one we will present the experiment in detail, and explain why this particular area has been selected. In the next one, we will go deep into the the results, measuring the overall impact on traffic from both macro and micro perspectives, detailing what option vehicles had when traversing one particular street, also detailing the measured differences in terms of path length, speed, time and pollutant emissions.

5.1 Experiment overview

As mentioned before, our experiment is focused on regulating the traffic flow of the district of *Ciutat Vella*. In order to do this, and based on the experimental results achieved by Terol Lloret [4], if we increase the value of α for a particular street to a value somewhat high, for example α = 100, the maximum allowed speed of the street will be close to zero (street cut), and the traffic flow on that street will drop dramatically, with the number of vehicles passing through that street tending to zero.

Taking the aforementioned issues into consideration, we want to free the city center of vehicles, thereby joining the global trend that is taking place in large cities such as Madrid, Barcelona, London, etc., to obtain a city center with low levels of pollution. To achieve it, what we will do is to adjust the α parameter to 100 in all of the streets that are in *Ciutat Vella*. The value of one hundred for α is selected based on the assumption in the title, "a high pollution scenario". This is a notable reduction factor that will cause street weights to vary greatly, thus diverting vehicles that previously traveled on those roads to other roads, unless they have no other option because their final destination lies on those particular streets.

5.2 Results

5.2.1 Macro perspective

In terms of experimental results, table 5.1 shows that the average speed of the vehicles, distance, and travel time do not have significant variations in the overall numbers. In fact, we found that the average travel time by vehicles did decrease slightly, which leads us to think that ABATIS selects routes where vehicles go through less traffic lights or prefer large avenues.

On the other hand, we verify that pollutants such as CO_2, CO, HC, NO_x and PM_x, decrease by about 2.5-4% in the city center cutoff situation, which is directly related with travel time, that also decreases by 3.05%. We can conclude that these slight variations do not represent significant improvements in traffic flow in the city since most of the traffic is concentrated in other districts, and there are several alternative routes; however, they do have a positive impact in terms of pollutant emissions.

Figure 5.3 shows a box and whisker plot with the distribution of the average speed of the vehicles in Valencia during that 15-minute period used for our study. As shown, in a cutoff situation, the average speed is slightly higher, but both

Table 5.1: Average traffic data in Valencia when cutting off the streets of *Ciutat Vella*.

Metrics	Default situation	City center cut	Difference
Speed	32.14 Km/h	32.52 Km/h	1.18 %
Distance	840.31 m	838.83 m	-0.17 %
Time	156.59 s	151.81 s	-3.05 %
CO_2	375.36 g	366.29 g	-2.42 %
CO	7.35 g	7.14 g	-2.91 %
HC	45.85 mg	44.71 mg	-2.49 %
NO_x	796.36 mg	764.06 mg	-4.06 %
PM_x	11.85 mg	11.50 mg	-2.90 %

are close to 32 km/h. It may seem that maximum speed points for this figure are somewhat high, and this is due to the source points of the vehicles in the simulation, because, as we can see in figure 5.2, some of them start from locations outside the inner part of the city, and then arrive at it, passing before through highways.

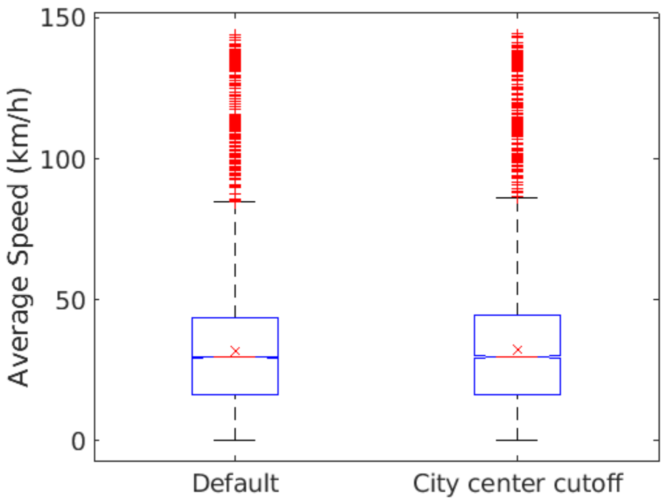

Figure 5.3: Box plot for the average vehicle speeds in both experiments: default situation (left), and city center cutoff (right).

Figure 5.4 shows a box plot that represents the distribution of the total distance traveled by the vehicles during the simulation period. Taking a closer look, we can notice that differences between both are very limited; in fact, minimum, maximum, and mean values for each experiment are mostly identical. These results lead us to consider that the routes which our algorithm chooses for the vehicles in the city center cutoff situation is optimal, thereby avoiding pollution-protected areas without a significant penalty on overall traffic performance.

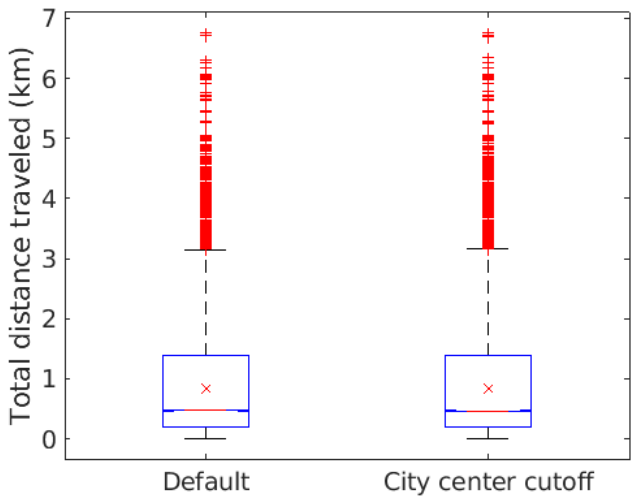

Figure 5.4: Box plot for the total distance traveled in both experiments: default situation (left), and city center cutoff (right).

The graph shown in figure 5.5 is related to the distribution of the travel times carried out by the vehicles in both experiments. By analyzing it, we can notice that differences between minimums are mostly non-existent; however, this does not occur for the mean and maximum values, where for the cutoff situation the maximum is a bit higher, and mean is slightly lower. This leads us to consider that the routes our algorithm is selecting do not affect the traffic flow in a way that generates unexpected traffic jams.

5.2.2 Micro perspective

Having analyzed results from a global perspective, we now take a closer look at the results for this same scenario by taking a representative sample. In this case, a vehicle that has the route *Pérez Galdós* avenue - *Balmes* street is selected. To this end, we will analyze the difference between the routes when our algorithm is applied, and how this translates in terms of traffic-related measures and pollutants emitted.

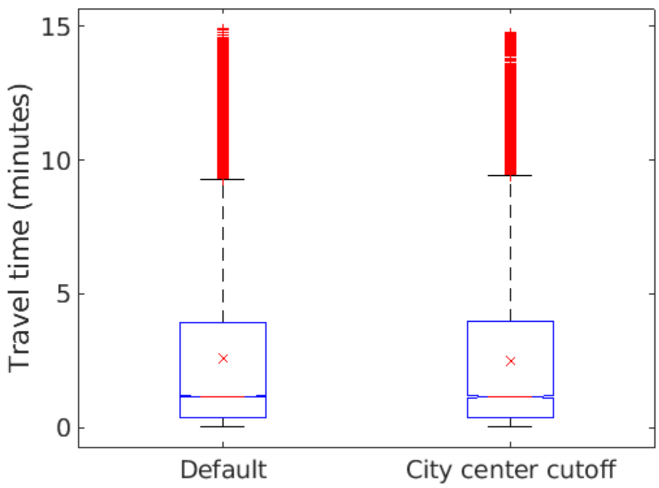

Figure 5.5: Box plot for the travel times in both experiments: default situation (left), and city center cutoff (right).

On the one hand, in a normal situation, as shown in figure 5.6, a vehicle starting at *Pérez Galdós* avenue (red mark) and heading to *Balmes* street (green mark) would go through the path depicted in the figure, which traverses various residential streets such as: *Calixto III* and *Juan Llorens*. This route is costly in terms of time and pollutants, because the car accelerates and brakes many times, which is counterproductive for emissions. Anyways, the vehicle arrives at its destination correctly because it's inside *Ciutat Vella*.

On the other hand, when our algorithm is activated, traffic is cut in the streets of the neighborhood of *Ciutat Vella* to avoid pollution in that area, so the route varies as now *Balmes* street cannot be accessed (it is inside *Ciutat Vella*). As can be seen in figure 5.7, the new route goes through the *Pérez Galdós* avenue (red mark), some residential streets (*Maestro Guerrero* and *Azcárraga*), *Gran Vía Fernando el Católico*, and finally crosses the entire *Lepanto* street, and leaves the vehicle a few meters from the destination (green mark). This route is not as costly in terms of time and pollutants, because the car accelerates and brakes less often. This is because most of the route is along an avenue, or in a straight line along a street.

The overall data collected by the simulation is shown is table 5.2. As can be observed, the restrictions applied to the city center do not affect the route of this particular vehicle. In fact, our algorithm improves its results. As can be seen, in terms of traffic parameters, a higher average speed is achieved, with an increase of 13.09%, and a reduction in travel time of 11.51%, while the distance traveled

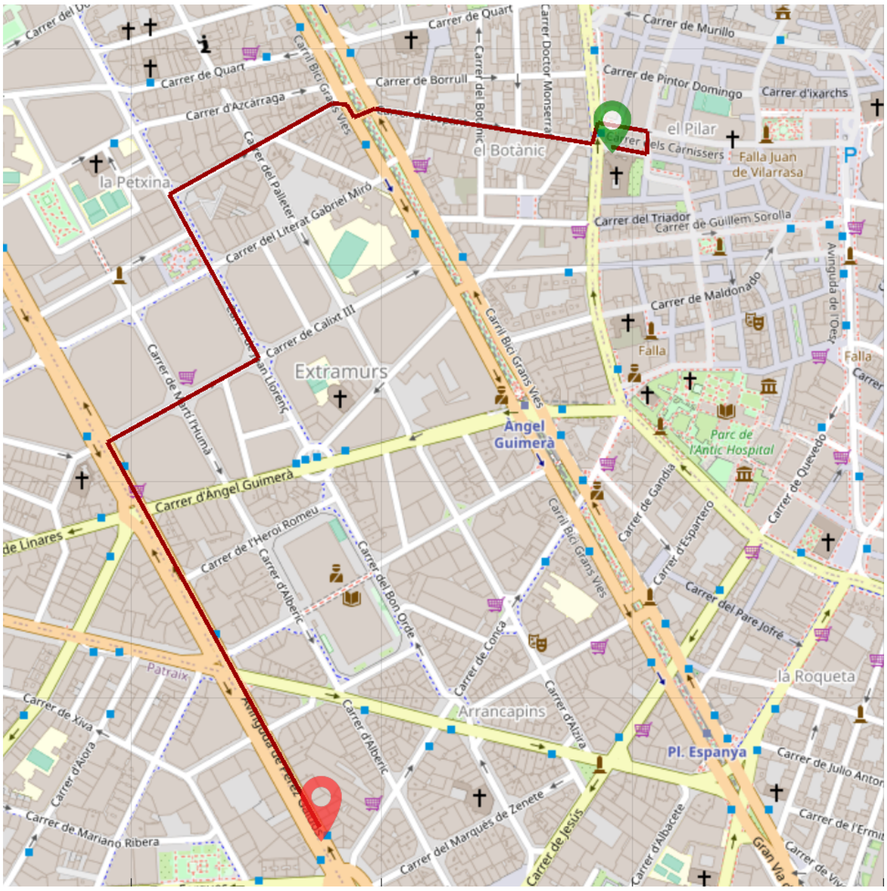

Figure 5.6: Default Route *Pérez Galdós* Avenue - *Balmes* Street.

remains similar.

Moreover, in terms of pollutant emissions, there is an improvement between the default route and our route selection. Four out of five pollutants have reduced from 12% to 18%, but there is a huge improvement in terms of CO_2 emissions, which have been reduced by an 80.99%. This could be due to the fact that it is a diesel car, and its combustion engine generates more CO_2, but by accelerating and braking fewer times, the car revs less, and therefore there is a drastic drop in this pollutant.

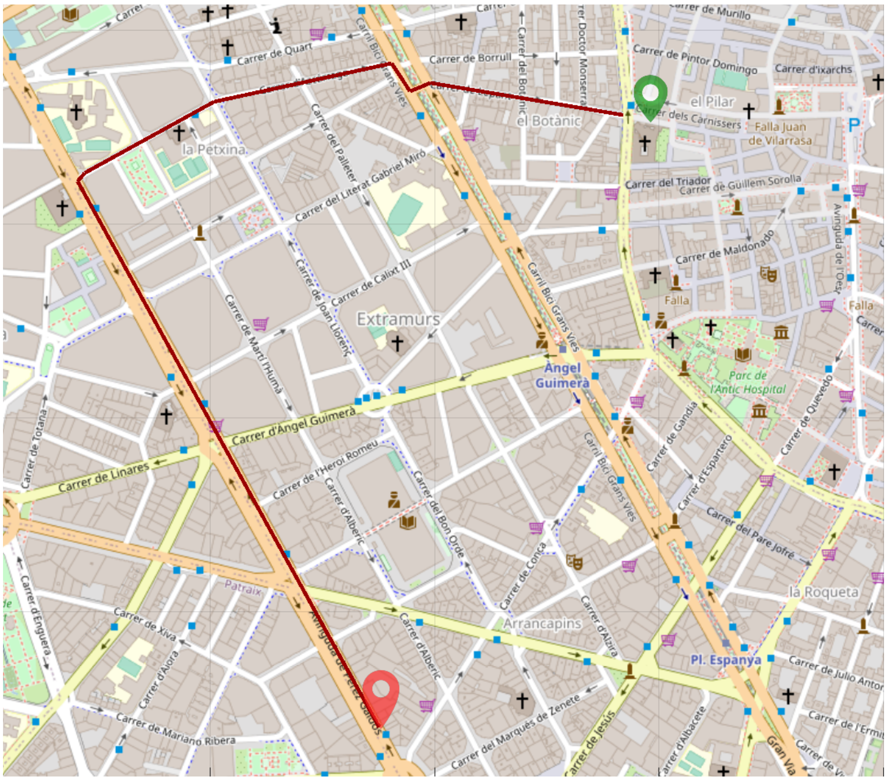

Figure 5.7: ReRoute *Pérez Galdós* Avenue - *Balmes* Street.

Table 5.2: Vehicle data when cutting off the streets of *Ciutat Vella*.

Metrics	**Default situation**	**City center cut**	**Difference**
Speed	26.96 Km/h	30.49 Km/h	13.09 %
Distance	1220.92 m	1220.77 m	-0.01 %
Time	165 s	146 s	-11.51 %
CO_2	303.35 g	57.68 g	-80.99 %
CO	52.97 mg	46.33 mg	-12.54 %
HC	11.98 mg	10.49 mg	-12.43 %
NO_x	611.91 mg	508.79 mg	-16.85 %
PM_x	10.45 mg	8.53 mg	-18.37 %

Chapter 6

Conclusions and future works

6.1 Conclusions

In this Master Thesis we have described the problems faced by large cities with increasingly high rates of air pollution, and how some of them are trying to remedy such problems by selectively restricting traffic in key areas. In this sense, we have used a novel algorithm that is able to smoothly deal with the traffic distribution in high pollution scenarios, especially when trying to minimize pollution values in key streets/avenues of certain districts.

In the same way, the different tools and components used in this work have also been detailed, and the necessary modifications introduced have also been explained. Having clarified the use of OMNeT++ as a vehicle network simulator, SUMO as a vehicle traffic simulator, and ABATIS as a route server, we have proceeded to explain the interaction between the three elements, and the applications we have used for this purpose, such as TraCI.

Moreover, the most important tool of the project has been linkABATIS, used for the connection between the two simulators and the route server and to generate the simulation files. How they interact has been described, and the algorithm implemented to reduce traffic has been detailed. The flowchart to perform a simulation has also been described, as well as the details of the generation of the simulation files and the analysis of the simulation results.

We have focused on a case in which the efficiency of cutting the streets of the district of *Ciutat Vella*, in the center of the city of Valencia, is studied. It has been determined that, in general, this cutoff does not affect traffic circulation parameters such as average speed, distance traveled, and travel time, nor does it affect pollution-related parameters, such as CO_2 emissions and microparticles (PM_x), among others. In fact, it is observed that the emission of pollutants decreases between 2.5 and 4%.

This leads us to conclude that, for this case study, the closure of the center does not affect the general traffic of the city if the routes recommended by our server are followed. In addition, it makes us reflect positively on the applicability and effectiveness of these methods when using our route recommendation system. On the other hand, for simulation reasons, we have that the cars that had as destination some place located in the center, have had to stop at the nearest area, as we have seen in the particular example, but with an improvement in terms of pollutants emitted.

6.2 Future works

As future work, we plan to continue analyzing the impact of more generalized traffic cuts, and in different situations, both for pollution and other types of restrictions (holidays, major events, etc.). In this sense, we plan to implement a solution capable of balancing pollution throughout the city, to avoid reaching critical values.

Likewise, our objective is to obtain a more complex equation to apply the traffic restrictions due to pollution. This means that alpha will not be enforced as a parameter without dependence, but as a part of another equation that depends on the various pollutants, weighted according to their impact on health, i.e. 1g of CO_2 emitted is not the same as 1g of PM_x; therefore, this effect has to be accounted for in the equation.

Bibliography

[1] TomTom, "Traffic congestion ranking: Tomtom traffic index," Jan 2021, last visited: 2021-05-18. [Online]. Available: https://www.tomtom.com/en_gb/traffic-index/ranking/

[2] S. S. of Pneumology and T. Surgery, "Air pollution causes 10,000 deaths a year in spain, many more than the 1,700 caused by traffic accidents," June 2019, last visited: 2021-05-18. [Online]. Available: https://www.separ.es/node/1505

[3] J. L. Zambrano-Martinez, "Efficient traffic management in urban environments," Ph.D. dissertation, Universitat Politècnica de València, 2019.

[4] M. Terol Lloret, "Solución para la gestión de tráfico de valencia en base a criterios ambientales," Master's thesis, Universitat Politècnica de València, 2021.

[5] R. Doolan and G.-M. Muntean, "Ecotrec—a novel vanet-based approach to reducing vehicle emissions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 608–620, 2017.

[6] A. T. Akabane, R. Immich, E. R. M. Madeira, and L. A. Villas, "imob: An intelligent urban mobility management system based on vehicular social networks," in *2018 IEEE Vehicular Networking Conference (VNC)*, 2018, pp. 1–8.

[7] Z. Khan and A. Koubaa, "Smartflow: An adaptive congestion avoidance protocol for smart transportation systems," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 1535–1540.

[8] T. S. Gomides, R. E. De Grande, F. S. Souza, and D. L. Guidoni, "A traffic management system to minimize vehicle congestion in smart cities," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 1439–1444.

[9] A. T. Akabane, R. Immich, L. F. Bittencourt, E. R. Madeira, and L. A. Villas, “Towards a distributed and infrastructure-less vehicular traffic management system,” *Computer Communications*, vol. 151, pp. 306–319, 2020.

[10] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, “Sumo (simulation of urban mobility)-an open-source traffic simulation,” in *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, 2002, pp. 183–187.

[11] A. Varga, “The omnet++ discrete event simulation system, the european simulation multiconference,” *Prague, Czech Republic*, 2001.

[12] C. T. Calafate, D. Soler, J.-C. Cano, and P. Manzoni, “Traffic management as a service: The traffic flow pattern classification problem,” *Mathematical Problems in Engineering*, vol. 2015, 2015.

[13] D. Luxen and C. Vetter, “Real-time routing with openstreetmap data,” in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS ’11. New York, NY, USA: ACM, 2011, pp. 513–516. [Online]. Available: http://doi.acm.org/10.1145/2093973.2094062

[14] INFRAS. (2021) Handbook emission factors for road transport. [Online]. Available: http://www.hbefa.net/

[15] T. Kurczveil, P. Á. López, and E. Schnieder, “Implementation of an energy model and a charging infrastructure in sumo,” in *Simulation of Urban MObility User Conference*. Springer, 2013, pp. 33–43.

[16] A. Varga and R. Hornig, “An overview of the omnet++ simulation environment,” in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, 2008, pp. 1–10.

[17] INET. (2021) Inet framework. Last visited: 2021-3-06. [Online]. Available: https://inet.omnetpp.org/

[18] A. Wegener, M. Piorkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, “Traci: An interface for coupling road traffic and network simulators,” *Proceedings of the 11th Communications and Networking Simulation Symposium, CNS’08*, 04 2008.

[19] J. L. Zambrano-Martinez, C. T. Calafate, D. Soler, L. Lemus, J.-C. Cano, P. Manzoni, and T. Gayraud, “A centralized route-management solution for autonomous vehicles in urban areas,” *Electronics*, vol. 8, p. 722, 06 2019.

[20] J. L. Zambrano-Martinez, C. T. Calafate, D. Soler, and J.-C. Cano, "Towards realistic urban traffic experiments using dfrouter: Heuristic, validation and extensions," *Sensors*, vol. 17, no. 12, p. 2921, 2017.

[21] DGT. (2021) Parque de vehículos, tablas estadísticas 2019. Last visited: 2021-20-06. [Online]. Available: https://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/parque-vehiculos/tablas-estadisticas/2019/

[22] ——. (2021) Distintivos ambientales dgt para vehículos. Last visited: 2021-20-06. [Online]. Available: https://www.puntosdgt.es/distintivos-ambientales-dgt//