Costa Rica Institute of Technology

Optimization of Traffic Simulation using GPS Navigation Records

by

Carlos Gamboa Venegas

Advisor:

Esteban Meneses, PhD

A thesis submitted in partial fulfillment

for the degree of Master of Science

Master of Computer Science

School of Computing

May 2021

Instituto Tecnológico de Costa Rica
Escuela de Computación
Unidad de Posgrado

ACTA DE APROBACION DE TESIS

Optimization of Traffic Simulation using GPS Navigation Records

Presentada por el estudiante

Carlos Gamboa Venegas

TRIBUNAL EXAMINADOR

ESTEBAN MENESES ROJAS (FIRMA)
Digitally signed by ESTEBAN MENESES ROJAS (FIRMA)
Date: 2021.05.22 17:41:56 -06'00'

Dr. Esteban Meneses Rojas
Profesor Asesor

Firmado por RAFAEL MAURICIO ARROYO HERRERA (FIRMA)
PERSONA FISICA, CPF-02-0474-0628.
Fecha declarada: 25/05/2021 06:02 PM
Esta representación visual no es fuente de confianza. Valide siempre la firma.

Dr. Mauricio Arroyo Herrera
Profesor Lector

STEFFAN GOMEZ CAMPOS (FIRMA)
Digitally signed by STEFFAN GOMEZ CAMPOS (FIRMA)
Date: 2021.05.26 11:22:09 -06'00'

MSc. Steffan Gómez Campos
Lector Externo

LILIANA SANCHO CHAVARRIA (FIRMA)
Firmado digitalmente por LILIANA SANCHO CHAVARRIA (FIRMA)
Fecha: 2021.05.26 16:26:25 -06'00'

Dra. Lilliana Sancho Chavarría
Coordinadora
Unidad de Posgrado, Escuela de Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

14 de mayo, 2021

Declaration of Authorship

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Abstract

A traffic simulation is a tool that permits constructing a virtual environment based on a real one, with the objective to perform analysis about the actual conditions and more important, apply changes to the virtual scene or to the driving rules to generate new scenarios and test solutions. However, the problem we found with simulations it that with incorrect parameters may not represent the traffic conditions we are looking for.

In this work, we propose a method to calibrate the traffic simulations using data available for transportation in Costa Rica. This data comes from Global Position System (GPS) navigation records. The calibration algorithm search to represent those actual traffic conditions in a virtual environment, and after that, propose and design solutions to ease the complicated traffic situations.

This thesis reflects the work of months to design and implemented an algorithm to calibrate simulations of five sectors of the country where we found difficult traffic conditions. The algorithm calculates a Measure of Performance to compare data from the simulation and the GPS records, and it searches iteratively for the best parameters. In the end, it validates the best solution found with a statistical test.

As results, we achieved to calibrate the simulations for the five studied sectors, reaching a configuration of input parameters that reflects the traffic conditions extracted from the GPS records, as a portrait of the real-life conditions of the locations.

The impact and applications of this work are plenty. For the computing part, we can dig more profound in using more techniques of calibration, and also exploit the data available for more general works. Moreover, it can become in a significant resource for analysis and decision making in urban mobility studies.

Acknowledgements

First, I want to thank my wife Juliana for give me her love and help, for encouraging me to work every day. And of course, to bear my stress better than me.

Thanks to Mami and Papi for supporting me every day of my life. There are not enough words to thank them, and I will never have enough money to pay them everything they have done for me.

Special thanks to Esteban, for being such a patient advisor, for never give up on me, and remind me every week my desire to finish this work.

To my colleagues at the laboratory of advanced computing, for always give me that feedback to present the best work I can.

Finally, I want to thank to life and time, to books and technology, to my eyes and my hands. I couldn't have done this thesis without them.

Dedicated to my wife, and all these years together.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Background

Costa Rica has a significant delay in road infrastructure of over 30 years. This is a reason the Gran Area Metropolitana (GAM), the principal urban and industrial zone in the country, has several problems of traffic congestion and public and private transportation, among others related to large amount vehicles on the streets. A situation that increases during rush hours and affects much of the population.

To solve such problematic, we can draw upon an Intelligent Transportation System (ITS), a tool that gathers several types of traffic and social data for analysis and decision making, to generate temporal or permanent solutions. Despite Costa Rica doesn't have an ITS, in the last years the government has worked to create and improve infrastructure in the most important points of the GAM and prevent difficult traffic situations. However, smaller road sectors receive little interest because when they present minor complications they appear normal, but during rush hours the congestion affects the traffic flow and they turn more problematic, needing a more elaborated approach to solve their specific cases. That is why simulations are extremely important.

A traffic simulation is a tool that can construct a virtual environment based on a real one, with the objective to perform analysis about the actual conditions and more important, apply changes to the virtual infrastructure or to the driving rules to generate new scenarios and test solutions. Although this sounds great, there is a problem, it is possible to misunderstand simulations, due to initial parameters

of the simulation may not represent a genuine state of the scenario. Meaning, the input data necessary to perform an accurate and adequate simulation needs to be adjusted. This adjustment is reached with methods to optimize simulations, commonly calibration, validation and verification.

In this work, we proposed a method to calibrate the traffic simulation using Global Position System (GPS) navigation records, collected from Waze, a commercial mobile application that Costa Rican drivers use day to day to navigate in the country.

The whole calibration process requires select location for study them, preprocessing the GPS navigation data, process and adjust the network files of the virtual roads for the simulation, design the calibration method and its implementation as a software tool to perform experiments, evaluate results, and propose solutions for each location to analyze their impact in the virtual scenario.

The following are contributions of this work:

- A method to calibrate traffic simulation using GPS navigation data.
- A software tool that implements the calibration method to simulate and verify problems in road infrastructure in Costa Rica.
- Analysis of the impact of proposed solutions for the traffic problems using the calibrated simulations.

1.2 Problem

Costa Rica has around 1.142.184 of vehicles with a density of 231 units per every thousand people [39]. In the center of the country, it lies the most important urban zone called GAM where around the 60 percent of the population live and reaches around 4 percent of the total extension of the territory, according the National Institute of Statistic and Census (INEC) [19].

The GAM is the center of the economy and industry, and where most people work, which makes its roads heavily transited. The exact amount of vehicles driving in the GAM per day is unknown, however there is an average of at least 5.000 heavy vehicles each day, according to PEN study about mobility and transportation of

FIGURE 1.1: La Salle crossing, La Sabana: OpenStreetMap view vs network simulator view.

Costa Rica during 2018 [39]. Same study where they explain how the large amount of vehicles driving in the GAM have a direct consequence on travel time of people and product delivery, resulting in loss of competitiveness and quality of life of the population.

In recent years, the government has paid special attention to this topic, trying to improve infrastructure in the zones with more traffic congestion. But, in some places it is quite difficult to change the physical infrastructure or to build new roads. Figure 1.1, for example, shows there is no space for more lanes and there is a very complicated infrastructure to redesign in the sector. Therefore, we need a more focused approach to determine the actual cause of traffics jams and present solutions.

Consequently, we leverage simulation tools to recreate those complex environments, analyze the involved variables, make decisions about modifications or new driving rules, and test the solutions proposed. For example, in the location previously noted 1.1. However, there is a problem, a simulation uses default parameters and sometimes we can adjust them with straightforward tests, despite that, we can't be totally sure that the simulation represents a valid scenario, to use them to simulated more complex situations and solve specific problems.

The problem lies in that we need to have the certainty to use traffic simulations that are valid and trustworthy. Otherwise any suggestion to the traffic situations can present enough evidence to support. We have the possibility to calibrate and validate simulations using the GPS navigation data available from smartphone applications. It is a topic where exists a lack of alternatives to calibrated traffic simulation using this type of information.

With this thesis we will have a software tool to calibrate and adjust the simulations, getting enough data to analyze and have evidence to support our suggestions about the changes required in road infrastructure, and driving rules or politics to improve the traffic conditions of some sectors of the GAM.

We formulate the **hypothesis** for this work: it is possible to calibrate traffic simulations using GPS navigation data with a statistical significance level (Type I error probability) of 0.1.

1.3 Justification

Simulations are very important to understand the behavior of traffic and it is a fascinating alternative where an ITS does not exist. Especially in Costa Rica, where urban mobility is a common issue, and the government has increased infrastructure projects to mitigate the impact on the principal and most congested roads and junctions of the country.

Without an ITS, we have insufficient sources for traffic data to perform calibration and validation of simulations and its posterior analysis. Using GPS navigation data records, we took advantage of this valuable opportunity to calibrate those simulations and create tools to make better decisions based on accurate results.

The primary innovation of this thesis is the use of GPS navigation records collected from mobile application users to calibrate traffic simulations of problematic traffic zones in Costa Rica. As an alternative approach to the calibration of simulation using data from ITS, data that is impossible to collect due the lack of technology in the country.

The impact of this work is the creation of a complete pipeline to calibrate simulation of locations with traffic problems in rush hours and then take those calibrated

simulation to propose solutions and analyze the effect of the changes. Those infrastructure improvements or new driving rules help in decision making and related traffic policies.

Calibrated simulations can help to adjust traffic lights timing as well, anticipate the impact of accidents, select the best location of the pedestrian crossings, understand the impact of public events, and many other. In addition, larger simulations can enhance the design of Smart Cities and a lot more possibilities.

1.4 Objectives

The main objective of this work is to evaluate the application of GPS navigation data to calibrate traffic simulations.

1.4.1 Specific Objectives

As specific objectives, we have:

1. Select adequate scenarios for study based on preliminary simulations and established guidelines.
2. Design an optimization method to calibrate and validate traffic simulations using GPS navigation records.
3. Create a software tool to implement the method to calibrate simulations.
4. Analyze the impact of proposed solutions to the traffic flow problems of the selected locations.

Chapter 2

Background

2.1 Intelligent Transportation Systems

It is well known that traffic congestion leads to several problems that affect the environment and the quality of life of the population of a city. These problems include air and sound pollution, increment in fuel consumption, delays in commute time and emergencies, and of course, traffic accidents, being these another cause for traffic jams. Such problems can create a continuous loop, a loop we need to stop or minimize their negative impact on the society.

Zhang *et. al.* [52] explains three strategies for reducing traffic jams. First one is to implement new transportation policies, related to public transport, exclusive lanes, restriction in schedule or license plate, some others. Second, relates to constructing additional infrastructure and improving the existing one. And last strategy says that it is possible to optimize the existing transportation system by analyzing the data gathered from different tools such as Global Positional Systems platforms, injections loops, video cameras and many others.

We can complement those three strategies with an Intelligent Transportation System (ITS) [15], which is a group of technological methods and tools to collect, process and analyze data from road infrastructure, to provide information to users and transportation system operators to make better decisions. These applications aim to be an efficient approach to improve the performance of transportation systems, reinforcing safety and security for travels, providing alternative routes to

avoid and mitigate traffic congestion, and as secondary effect, reduce air and noise pollution and increase energy efficiency.

A robust ITS can incorporate some fundamental components, such as advanced transportation management systems, advanced traveler information systems, advanced vehicle control systems, business vehicle management, advanced public transportation systems, advanced urban transportation systems, also can include intelligent vehicle systems, commercial vehicle operations programs, and many others. [52] [30]. Systems that need a source of information to perform their purpose.

Right now there is an explosion of data, available for almost every science discipline and for a huge diversity of researches, this amount of accessible information can help the development of better ITS's, changing the common use of technology-driven systems to a new data-driven techniques.

Zhang's team notices the relevance of a Data-Driven Intelligent Transportation System (D^2ITS), fed by large amount of data collected from multiple resources such as video cameras, multi-sensors like injections loops, laser radars, and GPS [31], data that can generate new information, prediction tools and services that can be included into an ITS. GPS navigation records are just one example of how to use data, providing an instrument to analyze and predict the behavior of users, drivers and people who take the public transport, and taking advantage of the real-time positioning information to trace vehicles and many other application.

To take advantage of D^2ITS we require data. It is possible to have poor data because of the lack of technology or disinterest of the government in implement those systems. Even so, we can use other approaches, such as traffic simulations, being a low-cost tool to replicate a variety of transportation events from real scenarios in a virtual world, and the use of data available from current ITS's can enhance the different simulations of these virtual environments.

2.1.1 Management and Analysis of Data

Data for ITS could come from multiple sources. The literature explains some ITS's use only two of them, other show the use of over two sources and in a more complex way. Amini [2] refers sources of data in transport are still limited to

data from mobile location, probe vehicles, smart cards and some information from social networks.

An OECD study [32] shows an ITS that makes use of data from cameras and micro-controllers, extracting information from sensor in an interconnected network. Also, authors mention more sources of data including Global Positioning Systems (known as GPS), sensors and devices such as accelerometer for motion, radar lasers, metadata from mobile devices, card to access public transportation and many others.

When data is generated so rapidly, with a lot of variety and in high volume, we can talk about Big Data. This data is the fuel for the new ITS's, a new era of systems driven by data. Right now, collected data is not a challenge, but to analyze them and extract useful information. That is how researches develop new platforms to deal with the outrageous amount of data, to integrate the tools and algorithms to gather relevant information from them.

Amini [2] divides the application of big data in ITS in three groups: i) urban planning, which studies the mobility pattern and travel demand using location data; ii) transportation operation, focused on travel prediction time, incidents detection and dynamics rerouting; and iii) safety, to predict crashes, understand driver's behavior and study critical situation in road infrastructure. In their work, they build a platform to provide a simple way to execute machine learning algorithms to analyze the real-time data coming from the real-world data sources. A solution that is implemented in a low cost virtual environment such as a simulation.

In their work [51], Zeng summaries an architecture of ITS that gathers information from video cameras, speed readers and sensors, GPS and radio-frequency identification, to create more complex layers and implement systems like traffic guidance, video surveillance, vehicle information and others. Then combining parallel analysis of big data and intelligent control creates useful applications to the final user of the complete system.

Similar to Zeng's research, Khokale and Ghate proposed big data architecture [21], but this time the author explains advantages to use big data in ITS. It is useful to handle the large amount of traffic monitoring data, big data can improve efficiency of transportation, big data can improve safety, and finally, can help to control vehicle identification (detect fake vehicles).

2.1.2 Data-driven Applications

We need to mention here some uses of data in traffic analysis and control applications. We have big data for urban planning as a tool to help to make better decisions [42]. Let's say we have a lot of mobile phones from people in the road, this can be feed the public transport systems to improve its own quality of service. Thinking about future, smart cities on real-time or not-real-time can enhance the analysis of the resultant big data flow and mathematical models. Data can empower the Smart Cities implementation of real-time data to create simulations visualization of vehicle travel time and queue length in roads, travel times of public transport, also, energy consumption and CO_2 emissions [43].

Another use of data is the prediction of traffic flow using deep networks trained with greedy layerwise unsupervised learning algorithm. Data collected from different detectors are aggregated to get the average traffic flow of studied freeway. Using root mean square error, they evaluate the performance of the proposed model trying to predict the traffic flow in the road for periods of 15 min, 30 min, and 45 min and 60 min. [28]. Similar work of prediction of traffic flow using data is also implementing using deep learning [4], they implemented a workflow data collected from sensors, aggregate flow, average speed, occupancy. Using specific software tools such TensorFlow plus Keras, and indicators like mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE) to analyze the prediction results.

In this field, data collections and big data are used for processing information from sensors counting, where data comes from main roadway in both directions, trying to optimize the vehicle generation using algorithms such as Nelder-Mead Simplex, Tabu Search and Genetic Algorithms [7]. We will explain some of these algorithms later.

Another important application is the safety monitoring of real-time data for crashes prediction, congestion measurements, data from Microwave Vehicle Detection System (MVDS). Here is crucial the real-time congestion monitoring. Researches found a way to measure the real-time congestion based on Big Data, determined as a more desirable approach to identify the congestion pattern in both the temporal and spatial dimensions. [44]

2.2 Traffic Simulations with SUMO

Simulations help to emulate traffic situations, it is a particular and special technique to show weak points in the street network of a location or to predict traffic conditions. It is useful when there is no a clear traffic flow and its behavior is complex. The ultimate aim of a traffic simulation is to create a virtual environment to understand some behaviors difficult to capture and analyze from real scenarios.

Nowadays, data analysis and hardware advanced technologies have intensified the use of traffic applications to increase the safety and security of the people, the efficient use of energy, improving user navigation systems, and planning road infrastructure. Many of these problems are complex and can scale significantly, therefore we can rely on traffic simulations models, that can be more accurate and dynamic that the common analytical methods used in the last years to present solution for the mentioned problems. [14]

A crucial part of the study of traffic and transportation problems is the traffic flow dynamics. It is essential to include this field on simulation models and simulation software because is a classic feature to generate virtual traffic situations, determine optimal routes, optimize logic for traffic lights, provide information for advance control of traffic in ITS, and reveal environment effects of traffic operations as fuel consumption and CO_2 emissions, both topics of real concern in our society. [46]

Traffic flow dynamics models can be separated into three categories [6] [46]:

- Macroscopic model describes the collective state of traffic, the time-space evolution of the variables local density also refer as volume, speed and flow. Model based in the continuum traffic based theory.
- Microscopic model describes the behavior of the dynamics of individual vehicles in the traffic flow. Here is common to define models to lane change, acceleration and breaking, driver aggressiveness, etc. This is the most interesting model for study in the last years.
- Mesoscopic modeling of traffic flow consist in a model less demanding of data and computationally more efficient that the other two models, but combining aspects of both of them.

- Submicroscopic: each vehicle and also functions inside the vehicle are explicitly simulated, e.g. gear shift [27]

Simulation software is challenging to develop, it can implement several mathematical models, and we can find open-source and commercial software. Regardless of this, its interest has grown significantly in the academic, exploring more a more the use of this software to solve real problems and improve life for humanity. But its use need to be taken seriously, with most relevant aspect of simulation is to know if they are accurate, showing that the simulation model is close enough to the actual system (real-life traffic conditions). This can be found using the model validation and calibration, an iterative process where each step execute algorithms to calibrate parameters and verify results. We will talk about this later in the chapter.

2.2.1 SUMO

SUMO (Simulation of Urban MObility) [24] is a traffic simulation software created to simulate the traffic in a city. Specifically, to understand the underlying model used to simulate behaviors, compare features like speed simulation or the capacity to represent reality from other models. Created in the Centre for Applied Informatics at Cologne, Germany, has been a popular tool to simulate and study traffic flow models, with additional tools to simplify the process, converting to different formats and creating routes to describe city transportation environment.

Implements a microscopic and multi-modal simulation, meaning that not only cars can be modeled but also motorcycles, pedestrians and public transport. Introduces simulation of sensors, different measurements and models, multi-lane streets, different vehicles types, pedestrians, bicycles, public transport, detectors and many other features. [27]. SUMO models road junctions with traffic lights, implementing right-of-way rules and its variants. Includes a model for demand of traffic that uses an Origin-Destiny matrix to control demand and adapt to simulation circumstances.

In model car dynamics every step of the simulation represents one second, modelling the traffic flow microscopically, where each vehicle dynamic is modelled individually within a network, having each own speed and location. The simulation model that SUMO implements for the car-driven behavior is the Gipps-model

extension, a model capable of display key features of traffic such as free and congested flows. It is also collision-free model with the principal reason to avoid artifacts arisen by intrinsic flaws of the model.

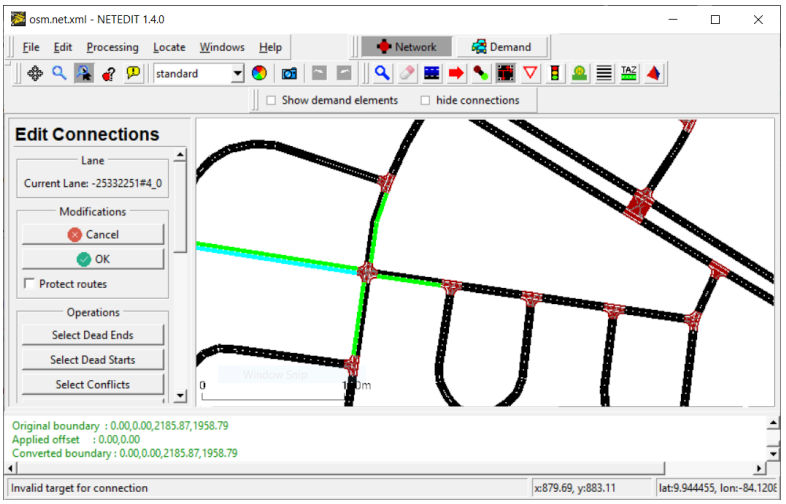


FIGURE 2.1: NETEDIT window: interface to modify the network file, traffic demand and additional elements.

The entire solution includes two GUI applications, NETEDIT 2.1 provide a graphical network editor to create, analyzing and edit network files, allowing the set-up of the complete environment for the simulation. User can change roads, lanes, junctions, include routes for cars, adjust traffic lights timing, add bus stops, set right of ways and forbidden turns, and much more. SUMO-GUI is the principal tool to visualize the current simulator, in here is possible to pause or adjust speed simulation, visualize cars in different colors by its speed, trace cars and visualize the selecting routes. The results of the simulation can be statistical information for general behavior or specific data for each car of segments in the network.

The program works with XML files, the initial input is formed by the three most important elements: network data, traffic demand and additional traffic infrastructure, those files are created directly from the NETEDIT. A required configuration file contains the three files and additional parameters for the simulation. The details of these parameters can be found in SUMO documentation [45]. This tool runs in a standalone computer using the terminal command, or the GUI-only version to visualize the simulation.

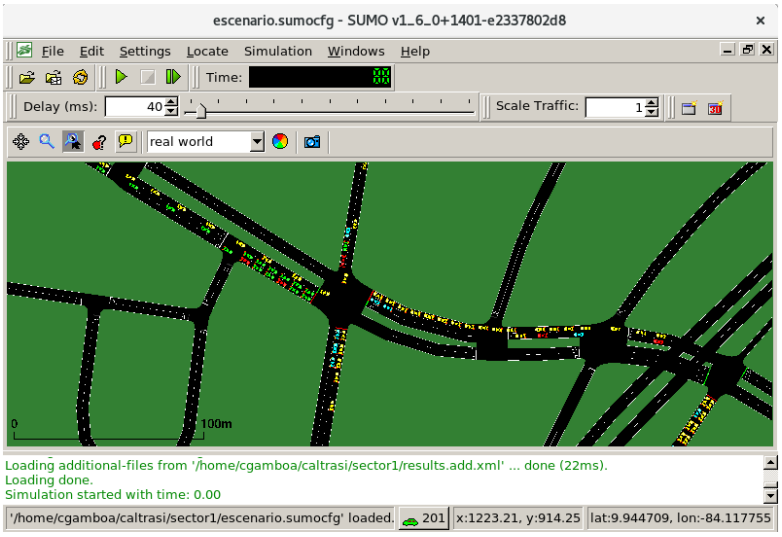


FIGURE 2.2: SUMO-GUI: graphical visualizer of simulation

2.2.2 Related and Recent Work

Recent development and SUMO applications [22] are presented in the study. Where SUMO simulates vehicular communication to study the effect of vehicle-to-vehicle and vehicle-to-infrastructure communication where a combined simulation of traffic and communication is necessary, traffic lights algorithms to make traffic lights capable to adapt to current traffic situations, evaluation of traffic surveillance systems to develop surveillance technology and use image processing of simulated areas to predict weather that probably trigger critical traffic situations. Finally, similar to our work SUMO simulates route choice and dynamic navigation.

Traffic flow generation using Origin-Destination matrix [50] with data from induction loop measurements available from traffic authorities, and then it uses the DFROUTER tool (a SUMO tools to reroute vehicles), along with a heuristic, to generate an O-D matrix for traffic that resembles the real traffic distribution. Simulation results validated against real data.

Flow is another work where they used reinforcement learning with SUMO [20] to analyze traffic dynamics and perform optimization. Flow provides users with the ability to easily implement, through TraCI's Python API, hand-designed controllers for any components of the traffic environment such as calibrated models of human dynamics or smart traffic light controllers. Together with the dynamics built into SUMO, Flow allows users to design rich environments with complex

dynamics. A central focus in the design of Flow was the ease of modifying road networks, vehicle characteristics, and infrastructure within an experiment, along with an emphasis on enabling reinforcement learning control over not just vehicles, but traffic infrastructure as well.

A very interesting work is the calibration of the car-model is actual parts of American roads by [23] In this research they calibrated travel times compared with data from simulated traffic detectors. They managed to reduce the simulation error from 40% to a 15%.

Paternina et al [36] proposed to use the advantages of artificial intelligence-based techniques such as reinforcement learning and artificial neural networks, in order to propose a global optimization approach that can be coupled with discrete-event computer simulation models to efficiently resolve practical problems.

2.3 Simulation Optimization Techniques

A simulation model is the study of a mathematical model using simulation. By running the simulation model with specific values of the input variables, we can examine a system behavior. We can define a simulation experiment as one or several tests in which meaningful changes are made to the input variables of a simulation model to observe and identify the reasons for changes in the output.

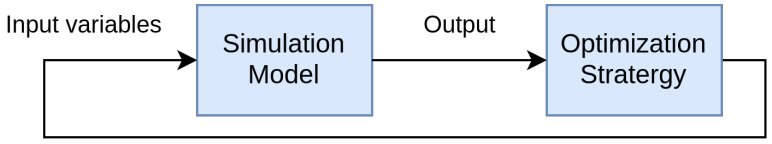


FIGURE 2.3: Simple simulation optimization diagram

Simulation optimization is the process of finding the best input variable values from among all possibilities without explicitly evaluating each possibility. In the image 2.3 we view a simple simulation optimization diagram, where an input is given to the simulation model, then output goes through the optimization strategy process, where it is hoped that the best input parameters are chosen to enter again to the simulation model, eventually, a stop condition will be triggered where we perform enough simulation experiment or solution is good enough. At the end, the objective of simulation optimization is to minimize the resources spent while maximizing the information obtained in a simulation experiment. [8]

There are at least six categories of optimization simulation methods, some authors name them different. However, there is a clear line where we can separate them. We extracted the next categorization from [8] and [1], Amaran collects fairly good amount of literature and updates the categories of optimization algorithms. Our summary review is as follows:

Discrete optimization (statistical methods): are methods working on finding optimal settings for variables that can only take discrete values. Some of the more popular algorithms are: finite parameter spaces, ranking and selection and, multiple comparison procedures.

Gradient Base Search Methods: or stochastic approximation methods are one of the oldest methods for simulation optimization. They attempt to descend using estimated gradient information, meaning that try to estimate the response function gradient ($\bigtriangledown f$) to assess the shape of the objective function and employ deterministic mathematical programming techniques.

Response Surface Methodology (RSM): is a procedure for fitting a series of regression models to the output variable of a simulation model (by evaluating it at several input variable values) and optimizing the resulting regression function.

Direct Search Methods: these methods are a sequential examination of trial solutions generated by a certain strategy (as describes Hooke and Jeeves.) As opposed to stochastic approximation, direct search methods rely on a direct comparison of function values without attempting to approximate derivatives. Meaning that depends on some sort of ranking of quality of points, rather than on function values per se.

Random (Heuristic) Search Methods: are part of direct search methods, are used to find a way to organize the search process to not search over all possible solutions, it turns in a low-cost search that is likely to discover a good, or near-optimal solution. Many of these techniques balance exploration with exploitation thereby resulting in efficient global search strategies. The heuristic as a rule-of-thumb may not guarantee convergence and optimality, making heuristic methods vulnerable to falling into local optima.

Model Base Methods: these are methods attempt to build a probability distribution over the space of solutions and use it to guide the search process. Three

specific methods are: estimation of distribution algorithms, cross-entropy methods, and Model Reference Adaptive Search (MRAS). [1]

2.3.1 Random Search (Heuristic Search)

Heuristic methods are usually rather problem specific, and often are based on simple common-sense ideas inspired by, or tailored to, the type of problem being solved. They are most often applied to the computationally intractable NP problems, simply because otherwise the best (most efficient) methods we know of for solving these problems exactly (or optimally) can take an exponential amount of computation time.

The more general heuristics methods are:

Hill Climbing: it is the greediest method, the idea of this algorithm is just not to accept a new solution unless it is better than the last best solution found. This is an intensive and pure search, with no space for exploration. Therefore, the algorithm is more likely to end up with a local optimum and, it can be very sensitive regarding the starting point.

Genetics algorithms (GA): are methods for search strategy that employ random choice introducing the concepts of mutation and selection, to guide a highly exploitative search, balancing exploration of the workable domain and exploitation of "good" solutions.

In general, a genetic algorithm is analogous to biological evolution, it works by creating a population of strings and each of these strings are called chromosomes. Each of these chromosome strings is basically a vector of a point in the search space. New chromosomes are created by using selection, mutation, and crossover functions. The selection process is guided by evaluating the fitness (or objective function) of each chromosome and selecting the chromosomes according to their fitness values (using methods such as mapping onto Roulette Wheel). Additional chromosomes are then generated using crossover and mutation functions. The cross over and mutation functions ensure that a diversity of solutions is maintained.

Evolutionary strategies (ES): similar to GA, ES are algorithms that imitate the principles of natural evolution as a method to solve parameter optimization problems.

Ant colony optimization: is a heuristic method that has been used for combinatorial optimization problems. Conceptually, it mimics the behavior of ants to find the shortest paths between their colony and food. Ants deposit pheromones as they walk; and are more likely to choose paths with higher concentration of pheromones.

Related to ant colony is swarm intelligence, which is a field that studies the emergent collective intelligence of groups of simple agents. In groups of insects, such as ants and bees, that live in colonies, an individual can only do simple tasks on its own where- as the colony's cooperative work is the main reason in determining the intelligent behavior the colony shows.

Tabu search: widely and successfully used in combinatorial optimization, it is an interactive process with the same capability of SA for escaping the local optima. The neighborhood length is very important and consist of a modified neighborhood search procedure that employs adaptive memory to keep track of relevant solution history (tabu list), which is updated in every iteration and allow the method to go beyond local optimality to explore promising regions of the search space.

Scatter search: Scatter Search uses adaptive memory in storing best solutions, as well as Tabu search. This method differs from other evolutionary approaches, such as Genetic Algorithms, by using strategic designs and search path construction from a population of solutions as compared to randomization (by crossover and mutation in GA).

Simulated annealing (SA): a stochastic search method commonly used to solve the deterministic optimization problems and combinatorial problems in traffic assignment [35]. The concept of annealing comes from thermodynamics, which deal with how a liquid substance is slowly cool down into a solid to produce a stronger, more stable product. Using simulated annealing as an optimization tool is because of the work of several researchers who were actually working in different disciplines at different times.

The method is a variation on conventional iterative improvement methods that begin with an initial feasible solution, repeatedly generate and consider changes in the current configuration, and accept only those that improve the objective function. This improvement mechanism has a probabilistic factor, in which non-improving moves are occasionally made, and therefore offers chances to avoid

getting stuck in the local optima, while keeping track of the best overall solution, hoping to arrive to a global optimum. [48]

To avoid the characteristic convergence to a local optimum that typifies deterministic local heuristic methods, simulated annealing methods probabilistically accept configurations that temporarily deteriorate the quality of the system being optimized. An acceptance probability is computed, based on the change in the objective function and a temperature parameter.

As the temperature is appropriately reduced (this is called an annealing schedule or a cooling schedule), fewer non-improving moves are accepted; thus, a coarse global search evolves into a fine local search for optimality, and the probabilistic jumps provide avenues to avoid sinking into non-global optima. [9]

Implementation of simulated annealing requires choosing parameters of the initial and final temperatures, the cooling schedule, and number of function evaluations at each temperature.

2.3.2 Simulation Calibration

The reliability of a simulation, seen as an experiment of the real system through its model, will depend on the ability to produce such a simulation model that represents the system behaviour closely enough. As we already know, simulation optimization is the process of finding the best values of the input variables of a model from among all possibilities without explicitly evaluating each possibility.

The process of determining whether the simulation model is close enough to the actual system is usually achieved through the validation of the model, an iterative process involving the calibration of the model parameters and comparing the model to the actual system behaviour and using the discrepancies between the two, and the insight gained, to improve the model until we get the desired or acceptable accuracy.

The goal of calibration is minimizing the difference between reality, as measured by a set of observed data, and the model results, described by another set of data that has been produced or constructed from the simulation model. This is done mostly by adapting the parameters of the simulation until some minimum (best fit) has been reached [3]. Calibration then, becomes an optimization problem [8]

[48], where the validation process estimates the differences between the simulation variables using the parameter set resulting from calibration, to obtain a desired confidence level where the model and its results are reasonable for the objective it was developed for.

As part of the final simulation optimization process, we have the model verification, calibration, and validation, crucial steps in the development of a valid simulation model. [33] The calibration and validation of a simulation tool (with a set of parameters) create that process of comparison on an appropriate scale of the simulation results for chosen variables with a set of observations of the variables. [14]

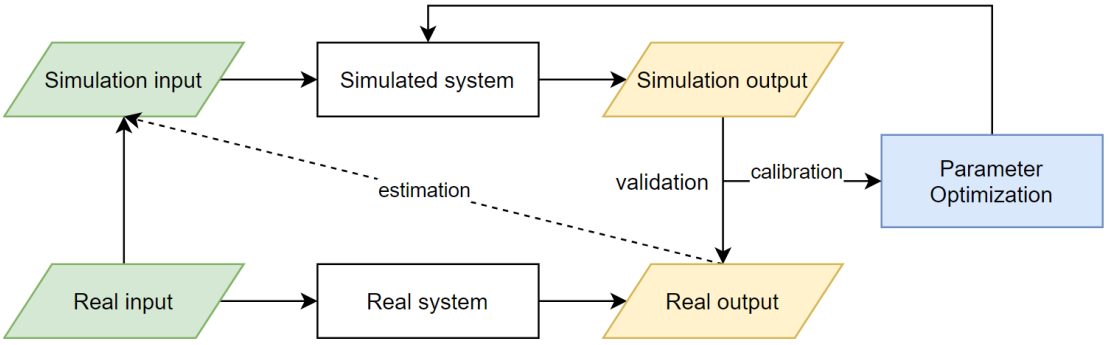


FIGURE 2.4: Diagram for real systems and simulation system optimization flow

Figure 2.4 shows a diagram representing the relation of real and virtual systems, Validation intends to determine how well a simulation model replicates a real system. In calibration, the outputs of the simulation and the real system are also compared, but the parameters of the simulated system are optimized until the difference between both outputs is minimal or at least meets specific minimum requirements. The fitting process is generally known as model calibration. As for calibration, during the validation of a simulation tool, predictions from the simulation model are compared to observations from reality, but a data set different from the data set used for calibration should be utilized. Unfortunately, calibration and validation against suitable observed data are not commonly practiced in the field of traffic simulation.

A crucial point in the calibration step is to collect the measures of performance that will allow comparison of simulation results with the observed current reality. This Measure of Performance (MoP) will define the application of the simulation tool and its objective. Taking into account that the calibration and validation of

the model should focus on the specific site and traffic situations to be covered in the simulation study. [14]

The objective of the traffic simulation optimization of this study is to modify the network between original and future scenarios to improve the MoP. Typical MoP's can be categorized by multi-valued collective variables such as flow, density, speed, queue length, multi-valued individual variables like travel times, trajectories, individual travel times, and single-valued waiting time, last decile of queue length or manual observation.

The choices of the appropriate methods and their application to the validation of traffic simulation models depend on the nature of the output data. Single-valued MoPs are appropriate for small-scale applications in which one statistic may summarize the performance of a system. Multivariate MoPs capture the temporal and/or spatial distribution of traffic characteristics and thus are useful to describe the dynamics at the network level. It may also be useful to examine the joint distribution of two MoPs (e.g., flows and travel times) to gain more information regarding the interrelationships of MoPs [5].

2.3.3 Related Work on Traffic Simulations

Li et al [26] describe the process of traffic optimization and organization, both processes where implemented using the combination of static channelized of road junction and the signal optimization. For the signal optimization they used a model called Simultaneous Perturbation Stochastic Approximation, an approach that try to approximate the gradient of the objective function through finite differences. With important reduction in computational cost compared to traditional stochastic approximation methods.

The article presents a case study where they analyze one sector of 1.68 km with 7 signal intersections, and during 7:30 am and 8:30 pm during the rush hour in the morning where average speed of vehicles is around 10 and 15 km/hour, using data from historical database of the traffic flow in the road. The final results show that the signal optimization method for the simulation show that the proposed model and method were effective and feasible, increasing the average car speed after the optimization.

Similar work what done by Celick and Karadeniz [10], where using SUMO simulator, as well as we did, they try to optimize the traffic flow, depending on the traffic density, their approach take an intersection and examines all the lanes on each side and processes based on the lane where the longest tail is located, then it develops a real-time traffic light optimization system to set the new lights configuration, creating a smart intersection system.

The most important conclusion is that for traffic light optimization the real-time analysis and change method gives better results than fixed time and green wave method that is based on the principle that majority of cars which pass on green light encounter green light again at the traffic lights on the next intersection

A similar study develop in this thesis is the work made by Flitsch et al [16]. Using historical data collected from sensors and aggregating this data, they recalculate the traffic situation of a road sector. They perform a calibration of a simulation of the Austrian road network. They applied two approaches: off-line calibration and on-line calibration. With the main idea of the calibration of improving the traffic volume on different routes based on time variations data.

The off-line calibration is used to develop a realistic simulation and to refine the demand model. Optimizing the routes and starting time for the vehicles in the SUMO routes file. The on-line calibration is used to adapt the simulation to real-time traffic situations. Currently, it is only based on the comparison of simulated and real-world vehicle detector loops. Experiences showed that the offline calibration process is too slow in a microscopic simulation. And also too slow for on-line calibration, where the simulation should keep pace with real world traffic situations.

Data they used were collected from vehicle detector loops (VDL), vehicles with sensors that provide floating car data (FCD) to get the real-time-traffic information, and Bluetooth-data for traffic information systems and traffic management. They refer the webcams as another possible traffic information source. However, an application for vision based object recognition is required for the data analysis. For validity checks on real-time data, knowledge of locals and experts is a valuable resource.

The general procedure of their the work starts selecting the target area for the calibration and validation. And the quality of data is defined before selecting the type of traffic data to be used. This is because sometimes only specific parts of a

road network need to be calibrated or validated. Then the data type is selected, for then check the availability and plausibility of data for the selected location. Plausibility check includes to validate a potential change of the road network where open street maps or other maps may be used as a reference. Changes in the road network do not only concern the calibration and validation, but also route selection and trip file. Finals steps involve checking all different types of data and perform plausibility test. And based on this information, SUMO may calculate an average speed and traffic volume. In their work as last results, they build a procedural model to calibrate and validate the simulation each time revising the data available and checking for plausibility.

2.4 Parallel Computing

In computing, serial programs are programs written to run on a single core computer, often do not take advantage of multi-core machines. We can do this if we can turn them into parallel programs, meaning rewrite some parts of the code to parallelize tasks, making them able to run at the same time in several cores and of exploit the multi-core infrastructure. [34]

Is it important to understand the available parallel hardware to benefit from it. We found two types of parallelism. Shared-memory systems and distributed-memory systems. In shared-memory system, is possible for each core to access each memory location. They communicate accessing shared data structures. In distributed-memory systems, each processor has its own private memory, communicating with other processors through interconnection networks, usually sending messages or with special functions to access memory of the other processors.

Parallelism with PyMP package[25], a Python package to implement OpenMP functionality that is shared-memory programming, with characteristics like minimal code and high efficiency, and of course multiprocessing.

With PyMP at the beginning, the memory of the children is not copied, but referenced. Once the process writes to the memory is when the own memory region is created. Keeping the processing overhead low but not such as original OpenMP implementation. When the parallel region is left, all child processes died synchronizing data structures via shared memory or a manager process, and only the original process survives.

```
import numpy as np
ex_array = np.zeros((100,), dtype='uint8')
for index in range(0, 100):
    ex_array[index] = 1
    print('Position assigned! {} done!'.format(index))
```

LISTING 2.1: Python code for sequential for-loop

In listing 2.1 it is shown an example of a classic loop, an array of one hundred positions is created with zeros, the loop goes over each index and set the value to 1. This a is classic sequential code.

The parallel version in listing 2.2, shows the code where `pymp.Parallel(4)` creates four parallel threads in `p` variable to execute the for-loop using `range`, this will divide the 100 execution in 4, doing each thread 25 indexes.

```
import pymp
ex_array = pymp.shared.array((100,), dtype='uint8')
with pymp.Parallel(4) as p:
    for index in p.range(0, 100):
        ex_array[index] = 1
        # The parallel print function takes care of
        asynchronous output.
        p.print('Perfect! {} done!'.format(index))
```

LISTING 2.2: Python code for parallel loop using PyMP

In R parallelism is implemented using package 'doParallel' and 'foreach' [47]. The final implementation is just necessary to change from %do% that evaluates the expression sequentially, to %dopar% that evaluates it in parallel. Example code is found in the code snippet 2.3

```
library(foreach) #load library

#implements the SEQUENTIAL execution of the sum
x <- foreach(a=1:1000, b=rep(10, 2)) %do% {
  a + b
}

#implements the PARALLEL execution of the sum
x <- foreach(a=1:1000, b=rep(10, 2)) %dopar% {
  a + b
}
```

LISTING 2.3: Use of foreach in R

Combination of these packages provide parallel execution, meaning we can execute code tasks and repeated operations on multiple processors/cores in personal computers, or on multiple nodes of a cluster [49]. It's important to clarify that 'foreach' principle belongs to the data parallelism model single instruction multiple data (SIMD) [34]) rather than to the task parallelism model (different codes) [41].

Summarizing, we can declare that parallel computing comes to doing three things: splitting the problem into pieces, executing the pieces in parallel, and combining the results back together. Both implementation in Python and R help with these tasks to take advantage of the multi-core resources.

Chapter 3

Calibration Technique for Traffic Simulation

Solution Overview

Study traffic flow in congested sectors in Costa Rica is difficult, the absence of ITS limits the use of data to analyze traffic situations. National government monitors major roads, while the majority of secondary roads are responsibility of local government. This causes coordination and communication problems, adding that decision regarding new infrastructure or vial changes and politics takes time. It is necessary a way to help decision making and address traffic problems in roads.

The principal goal of this work is to create a tool to calibrate a traffic simulation in SUMO, being able to represent the real-world situation of the traffic on specific road sectors in a determined period. This calibration is done with GPS records from a Waze application, these data is used to compare the reported speed with results from simulation, using an optimization algorithm we can adjust the traffic flow to represent the actual traffic conditions as close as possible.

The complete process involves four parts, first the wrangling of the data coming from GPS records, second the processing of the output results from SUMO simulations, follow by the design and programming of the algorithm to calibrate the simulation, and finally the execution of experiments to obtain the final calibrated simulation parameters.

The analysis of the results involves the study of the effect of changes applied to the sectors in order to improve the traffic flow in problematic areas, the sectors were selected using personal experience and observation, and traffic solution were proposed using common road alternatives that ease the traffic conditions.

Two extra steps in this research are the sensitive analysis that provides information about the input parameters that produce more changes in the traffic flow, and the analysis of parallel code performance to illustrate the importance in the parallel execution of experiments to reduce time to implement solutions.

3.1 Selection and Preparation of Sectors

Rush hour in Costa Rica is problematic, several locations of the GAM show heavy traffic and congestion affecting a substantial amount of people each day. Selecting a road sector to analyze their traffic flow is not as easy as it sounds. Government is already working in some streets, for example Circunvalación, the principal by-pass in San José is being completed with a brand new north segment, and some secondary roads are suffering changes such as more lanes and new signaling. Considering that, we try to focus on five sectors that present a special case and, at the moment, they are not receiving explicit attention from the corresponding entities.

We chose five road sectors from different areas, regarding the importance of the location, the impact of current traffic congestion, and the feasibility to create new infrastructure. From the study called Congestion of Vehicular Flow of GAM by CFIA [12] we considered the reasons for traffic congestion together with some ways to address the road congestion to select the scenarios. Summary these reason as the traffic demand in rush hours, the limited road spaces, the elevated cost of road infrastructure to ease traffic flow in rush hours, and to deal with these conditions we considered the impact on junctions, the traffic light coordination timing and the priority to public transportation. We describe those five sectors in the following paragraphs.

Sector A is the main junction in Plaza Mayor, Rohrmoser visible in Figure 3.1, it presents traffic congestion in all directions, specially in the way north to south. North street comes from an uphill and a bridge and stopping in a traffic light aggravates the jam at the north. The traffic light is necessary because it is a junction with around 5 routes and 8 turns. In this sector we are looking forward

to reduce that north-south jam, showing the impact of some simple changes. The districts at this location are Uruca and Pavas. This information is important because is part of the filtering process to extract the involved road segments.

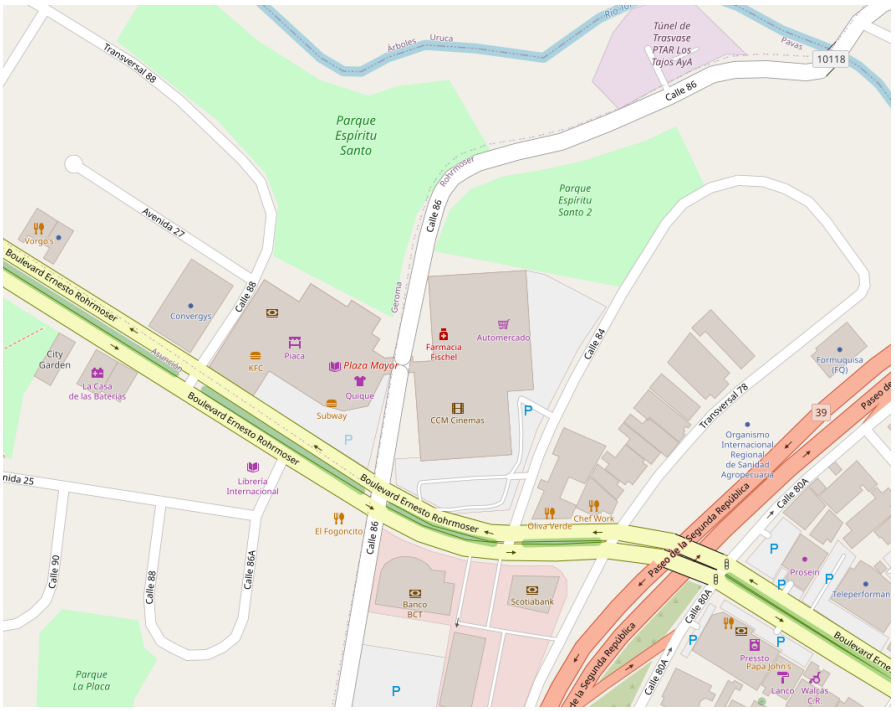


FIGURE 3.1: Sector A map: Plaza Mayor junction

Sector B is an entrance from highway San José-Cartago to Taras, shown in Figure 3.2. This sector is important because around 7.400 drive through this highway back during rush hours [29], the highway entrance to Taras is just one lane crossing the two lanes highway Cartago-San José, after that just 350 meters towards Taras there is a three-way junction without traffic light, just a stop sign in the lane from east, causing heavy traffic because most vehicles coming from north want to turn east and rules of the road and courtesy are rarely applied by drivers, the urgency to get home and the rush hours do not facilitate the situation. Here the districts involved are San Nicolas and El Carmen.

Sector C at Invu Las Cañas and junction with train rails is shown in Figure 3.3. The particular situation in this sector is a train rail cross just in the middle of two four-ways junctions, causing a complicating scenario. From the south of the main street cars turn right but immediately can turn right again, continue straight or worse, turn left at point 2 in the figure. Also, vehicles coming from point 2 usually turn left to point 1 because it is the main exit from that sector. Involved districts in this location are Desamparados and Rio Segundo.

FIGURE 3.2: Sector B map: San José-Cartago entrance to Taras and three-way junction

The Sector D is located in San Pedro from UNED (west) to Sabanilla (east): this sector is simpler to explain but difficult to understand and analyze, as its shown in Figure 3.4 the main traffic flow goes from the west (1) to the east (2), with a second heavy traffic coming from north in point 3 and south in point 4, others routes combine together to generate a complicated situation in central points, meaning lots of heavy traffic and congestion on the way to Sabanilla downtown. Here we found three districts involved San Pedro, Mercedes and Sabanilla.

Sector E is the main entrance to Barva from Heredia visible in Figure 3.5: similar than previous case, the traffic flow from 1 to 2 is the main problem, multiples entries and exits complicate the situation on the way to Barva, adding the public

FIGURE 3.3: Sector C map: railroad crossing between two complicated four-way junctions

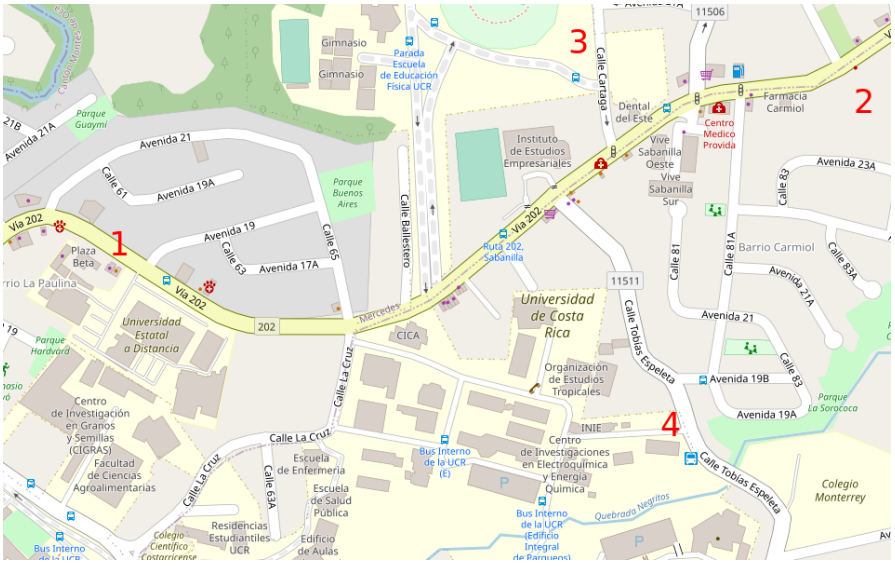


FIGURE 3.4: Sector D map: multiple traffic lights route San Pedro to Sabanilla

transportation because the main street does not have the respective bus bays on each bus stop, conditioning the general vehicles speed to the bus pace and the time to getting people on and off the bus.

For each sector we use the SUMO plugin called WebWizard to download the network files in OpenStreetMaps format (osm extension file). We revised the original

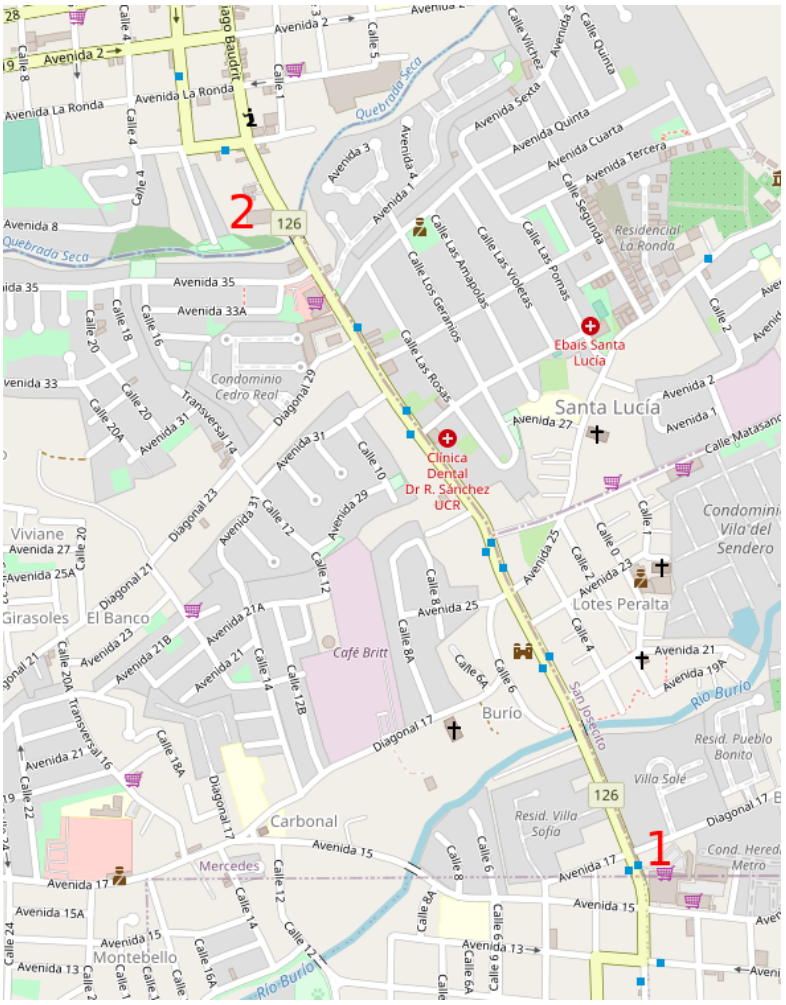


FIGURE 3.5: Sector E map: main road from Heredia downtown to Barva

network files to eliminate errors like extra segments, missing segments, missing traffic lights, incorrect turn rights, etc. And wrote two bash scripts using SUMO tools to convert the network file with a more appropriated structure, changing the segments identifiers to allow just numeric ids, avoid u-turns, and eliminate the unnecessary data of polygons from buildings and other structures. And also, we created the initial SUMO configuration files.

Once those files are ready for one sector, we run the first simulation as an initial check to generate the first results and make sure the network is not corrupted. Also, some preliminary results can help to visualize and understand how simulation recreates the traffic flow in the selected sector using random routes and flow, this gives us hints to set the initial parameters of the experiments.

3.2 Wrangling and Preparation of Data

3.2.1 Data from Waze reported events

The data used to calibrate the simulation are GPS navigation records from the commercial mobile application called Waze. A collaboration among Programa del Estado de la Nación (PEN), Ministerio de Obras Públicas y Transportes (MOPT) of Costa Rica and Waze allows us to use data coming from jams and incidents reported since 2018. Even though, the raw data is not available for this study, we used different data layers provided for a previous team of researches that worked on the preparation of this raw data [13, 18, 38], cleaning and organizing the records in a data structure of R programming language called data frame, and saving them on an RDS file. After that, we can perform our filters to select the GPS records from the timeframe and the location covered.

The new data contains records with the following variables: *city, length, speed, anno, hour, delay, line, startNode, month, dayWeek, endNode, roadType, street, day*. We need only *speed* and *line* information for each record, but the filters are executed according to the other variables.

The first filter will take records from the weekdays and the 17-hour of each day, after that it will select two variables, *line* that contains the geometric information required to intersect with spatial information from SUMO and other sources; and *speed* that is in the average reported speed of the jams in that moment. With that, we took only one hour records during the rush hour in the work days. Focusing the study in a specific time frame to try to avoid errors caused by the unstable conditions of the beginning and the ending of the rush hour. We assume this is the steady-state of the system.

Next step is to convert those records into spatial data. The spatial data structure includes points, lines, polygons and grids; each of them with or without attribute data [37]. Then, using the dataset *RedVial* the algorithm takes the district IDs to extract the 100 meters roads segments for each district , dataset prepared for the previous studies of Gomez and Cubero to [18] [13]. Finally, it intersects the GPS records with the road segments to once again reduce the amount of data and canalize only the required information.

At this moment it aggregates the data and gets the statistics of speed for each segment, resulting in a new data frame with the segments of road organized by id and the respective average speed reported for one hour, specifically 17-hour (5 p.m.). This information is saved in a csv (comma-separated values) file that is going to be used during the calibration process to compare with the speeds resulting from the simulation.

3.2.2 SUMO simulation output files

For this work, we required the information from a specific output format called as Lane- or Edge-based traffic measures. The returned values in this output describe the situation within the virtual road network in terms of traffic measure indicators. These values are macroscopic because they are generated by lanes or edges, referring to an edge as a road segment in one-direction.

SUMO requires additional information in order to generate the previous output. This information is a set of configuration parameters to ask for the values to be generated for each sector. In 3.1 we indicate we want data from each edge using label *edgeData*, then we set the attributes with the suffix of the filename, the *trackVehicles* in true to performed speed aggregation for over all vehicles, exclude empty edges and the interval frequency that is the period to make the aggregation, we used in all experiments 500 second intervals. For our study, we focus only on the speed data.

With those parameters set, we expect at the end of each run the resulting file containing the information for each interval and each segment of the network. In the listing 3.2 we view how a first interval starts at 0.00 seconds and ends at 500 seconds, this is the first aggregation period, now each edge is included with the respective information, the average travel time, the average waiting time, and the more important variable of this work: the mean speed in the edge; the remaining data is not used at all.

On every simulation experiment we are simulating one hour of traffic flow, but the traffic flow during the entire hour is not stable, even when we can generate a continuous flow; that is why we separate the generation of data in intervals to discard the first and last interval where the state of the simulation is not stable. At the beginning of the simulated period, in the first intervals, cars are still

```
<additional xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
    " xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/
    additional_file.xsd">
    <edgeData id="001" file=".output.meanedge.xml" trackVehicles
    ="true" excludeEmpty="true" freq="500"/>
</additional>
```

LISTING 3.1: Structure of the additional file as input for simulation

```
<interval begin="0.00" end="500.00" id="001">
    <edge id="-25399903" sampledSeconds="2054.32" traveltime="
    42.80" waitingTime="1269.00" speed="2.50" departed="0"
    arrived="0" entered="48" left="48" laneChangedFrom="0"
    laneChangedTo="0"/>
    <edge id="-474561682" sampledSeconds="500.63" traveltime="
    3.88" waitingTime="0.00" speed="12.48" departed="0" arrived=
    "0" entered="129" left="129" laneChangedFrom="0"
    laneChangedTo="0"/>
    ...
    ...
</interval>

<interval begin="500.00" end="1000.00" id="001">
...
</interval>
```

LISTING 3.2: Structure of output file of the simulation

spawning at the origin edges, meaning the involved edges don't have any flow. In the last intervals cars are not spawning as well, at contrary they all are reaching destination, again, this causes a low or total lack of flow in the concerned edges.

One more step of aggregation is done by taking the middle interval and perform an average of the speed by each segment. These results are saved in the **iter.xml** file, this file is written in Open Street Map format, containing each node with its latitude and longitude values, and each segment of the road with the respective and already calculated average speed from the simulation. The format structure of the **iter** file is shown in code snippet 3.3, the *way* label indicates a spatial object represented as a line in the map 2D view, this line contains the respective edge in the original network and its average speed. During simulation, this file is used to intersect GPS records and generate the comparison files we need to calibrate the simulation.

In SUMO documentation [45], we can have all details about the configuration parameters and the output that SUMO can generated for edge or lanes.

3.3 Implementation of Calibration Algorithm

We develop the solution in Python and R languages. Python was chosen to facilitate the programming and take advantage that SUMO is written in Python. R was chosen to reuse the existing code elaborated by Cubero *et al.* [13] as base for the spatial data processing, and specifically the parallel code execution to intersect road networks.

The figure 3.6 shows a simple diagram of the process of our simulation optimization solution. It starts with preparing the input parameters that are the period time for vehicles insertion in specific routes of the sector. SUMO runs the simulation and generates an output that is the aggregated data of the road segment by a time interval. This output is used in combination with the GPS records to compare them and calculate the MoP of the data, starting the calibration algorithm to find new parameters to simulate again. Every iteration the algorithm will perform the same procedure, generating new parameters, executing simulation and calculating the MoP to verify and validate the alternative solution. The process stops when the indicated iterations are performed. The final step is the statistical test to validate the best found solution, to decide if it is truly useful.

3.3.1 Simulation Input

The input parameters of the traffic simulation are the vehicles flows created manually and are indicated in the *routes.rou.xml* file. The listing shows the attribute of each route configuration. As shown, each flow (car route) contains an initial

```
...
<way id="1">
    <nd ref="276218495" />
    <nd ref="6" />
    <nd ref="7" />
    <nd ref="8" />
    <nd ref="9" />
    <nd ref="7304134979" />
    <tag k="highway" v="residential" />
    <tag k="speed" v="3.44" />
    <tag k="edge" v="808111621" />
</way>
...
```

Listing 3.3: OSM file for each simulation experiment

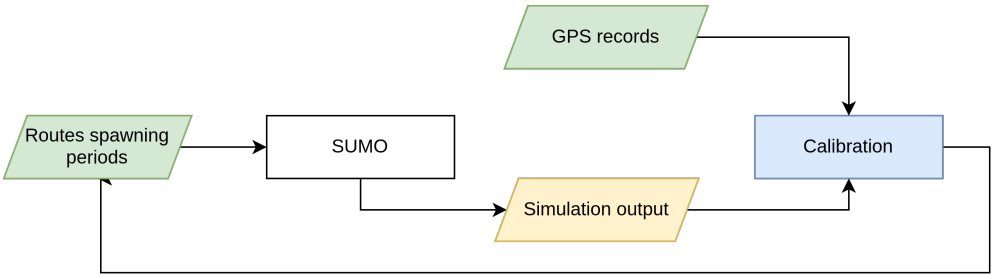


FIGURE 3.6: General workflow of the solution

node (from) and final node (end), those are set manually according to the study and visualization of real conditions of the sector where more traffic flow is created. Attributes like *departLane*, *departPos*, *departSpeed* are set to random values to include variability, but the most important attribute is **period**. This parameter is the spawning time in seconds between vehicles in the routes from start to end nodes. Being able to generate lots of traffic flow if those values are tiny, but may cause deadlocks in the roads.

```
1
2 <routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/
    routes_file.xsd">
3     ...
4     <flow arrivalLane="current" begin="0.00" color="magenta"
    departLane="random" departPos="last" departSpeed="random"
    end="1800" from="808111836" id="5" period="8.0" to="
    808111145" type="veh_passenger" />
5     ...
6 </routes>
```

The above described is the way we chose to generate heavy traffic flow. However, there are two more options that are important to mention.

One approach is to have a value that for each segment shows a probability to be selected as an initial node, as a final node or intermediate node for a specific vehicle path. This will create a more random behavior, but more difficult to control because the created paths may not represent a normal route of a vehicle in the current location. We can increase the weights of the segments we believe are origins, end and part of traffic routes. But this is more complex than that, because that information fed the randomtrips.py script that is used to generate the flow and is almost impossible to create a specific behavior. This randomly state may sound good for this study, but the calibration process of the random trips can take

a long time. Indeed, this was the first option we tried, but simulation was taking too long to execute and the algorithm wasn't obtaining good results, not because the design of it, it was because randomizing was generating so many variants that were not possible to control.

Another option is to create the Origin-Destiny matrix. This procedure involves a more detailed study of the routes and it is recommended for analysis of much bigger scenarios. Eventually, we decided not to use it, because we are focusing in much smaller sector of the city.

The option we chose allows us more control of the routes, at least for the amount of them and their impact in the sector. This is important because we can focus on paths that commonly create most of the traffic problems.

To complete the input files of the simulation, a configuration contains the filenames road network, the routes flows, and extra additional files mention in section 3.2.2 required for configuration to set the aggregated output by lanes.

3.3.2 Simulation Run and Calibration Algorithm

The algorithm chosen and designed for this work was Simulated Annealing (SA). This algorithm will take an initial set of parameters to run simulation once and to create the Initial Solution, set also as the Best solution, that will work as a comparison start point. Then the algorithm will chose a combination of parameters based on the initial input to generate an alternative solution, either worse or better, the value of this solution will be compared with the previous best solution, and the best of both will be set as the new Best Solution. On each iteration, the algorithm will try to select a better solution.

For this kind of traffic simulation, a solution is a set of values that describes the average speed by segment from GPS records and from simulation. We can see those two sets of values as two vectors, which we can compare and test if they are similar. That is what we need close enough vectors to affirm that the simulation is representing a real-situation of traffic congestion in the studied sector.

To compare those two sets of values and obtain a measure of the solution, first we calculated the Measure of Performance (MoP) using Root Mean Squared Error(RMSE) that gives us a value of the distance of the vectors. This value is in the

same units of the values we are comparing, how we are using meters per second, the RMSE value in meters per seconds. With this result, we have an idea how close those vectors are and we can put a single number on each solution. The final aim of the calibration method is to reduce that distance, trying to get the best similar values as possible and the lowest RMSE close to 0.

At the end of the process, the statistical test Paired Sample T-Test is calculated to determine if the best solution we found is statistically relevant, and we can be sure that calibration algorithm did a good job.

More specifically the implementation of the algorithm is as follows:

First, we run the simulation method to create an output file for each parallel run, the average speed for every segment is collected in a new file in a spatial format including the speed for each road segment.

Then the algorithm executes R code to read the previous results and intersect them with the spatial lines data from the GPS information we already have for the sector, creating a single csv file with the speed for each segment. This process is required because from SUMO results we don't have the possibility to know what road segment correspond to the GPS data, due that SUMO data doesn't include the spatial id for segment and identifiers are different, that's the reason the spatial intersection needs to be done in R language.

After completing that process, code return to Python to calculate the MoP, using specific libraries to obtain the RMSE value. This solution is the initial solution of the system, it is feasible but not optimal. The final decision is to pick the best solution each time, selecting the minimum RMSE within the best value and the calculated.

Here is when we start the iterative algorithm. Based on the last solution parameters, SA will choose several neighbors indicated as a parameter. For each neighbor the program will run a simulation, intersect the results with the GPS records and get a RMSE value, each time will be compared with the last best solution found and changing it if a better solution is found.

Neighbors are chosen using the property of temperature of the SA. The nature of the algorithm is that a variable Temp resembles the temperature in the origin simulated annealing application on metallurgy. This variable starts at a high value and each iteration is reduced, similar to the cooling process. So, the higher

```
SA():
    run_Simulation() #SUMO
    process_Simulation_Results()
    intersect_GPS_records() #implemented in R
    actual_Solution = calculate_MoP()

    #start loop on cooling process
    for temp in temperatures:
        best_Solution = actual_Solution
        all_neighbors = calculate_neighbors()

        #loop through neighbors
        for neighbor in all_neighbors
            run_Simulation() #SUMO
            process_Simulation_Results()
            intersect_GPS_records() #implemented in R
            new_Solution = calculate_MoP()

            if new_Solution < best_Solution
                best_Solution = new_Solution

        change = actual_Solution - best_Solution
        if (change > 0):
            actual_Solution = best_Solution
        else:
            if probBest(change, currentTemp):
                actual_Solution = best_Solution
```

LISTING 3.4: General structure of the Simulated Annealing algorithm

temperature, the more will be the change in the input parameters of the simulation, that is, the period of the flows in the routes.

On each iteration we expect we get closer to the best solution, selecting every time the lowest RMSE. The cooling factor each time reduce the temperature change in the neighbors, meaning the change in the input parameters is less every time, and also reducing the chances to get lots of different neighbors and trying to converge in an optimal global solution.

The algorithm performs the paired sample t-test to determine if both resulting vectors of speed are statistically similar, resulting in successful calibration of the parameters or not.

3.3.3 Goodness-of-Fit and Statistical Test for Validation

To determine if calibration algorithm is doing a good job, we required an adequate way to measure statistically and validate the results. Commonly use in these cases

is the Goodness-of-fit (GoF) measures, that can be used to evaluate the overall performance of a simulation model [40]. Among this method we found the Route mean squared normalized error or route mean square percent error [5]. Root Mean Square Error (RMSE) measures how much error there is between two data sets. In other words, it compares a predicted value with an observed or known value. The smaller an RMSE value, the closer predicted and observed values are [14] .

In the equation 3.1 we see how this calculation takes the difference for each observed and predicted value, square them and then divide the sum of all values by the number of observations.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(Y_n^{sim} - Y_n^{obs})^2} \tag{3.1}$$

To show statistical relevance, we test the final solution with the calibration algorithm with a two-sample paired T-test. This test is commonly used to test the difference (d0) between two population means and determine whether the means are equal. In this research, we are testing if the two resulting vectors from the simulation and the GPS records are similar, meaning the difference of paired values is close to cero.

Hypothesis testing:

$$H_0 : speeds_{GPS} = speeds_{sim}$$

$$H_1 : speeds_{GPS} \neq speeds_{sim}$$

Contrary to the traditional hypothesis testing that wants to reject the null hypothesis, we are trying to accept the null hypothesis H_0, meaning the mean for the speed are similar, the two final vector of the speed segment from the GPS records and simulation should be as close as possible.

As we indicated in the hypothesis we are setting a significance level (α) of 0.1. This means that we expect in all experiments got a p-value greater than α to accept the null hypothesis and verify the simulation is generating traffic flow sufficiently similar to the evidenced by the GPS data.

3.4 Design of Experiments

The complete calibration algorithm required a configuration file to indicate of the initial values of the parameters for the Simulated Annealing, the processing of the spatial data and the initial routes for the simulation, with the range of the initial values to set the spawning time for each route.

This configuration file includes the edgesId being the identifiers of the segments of 100 meters that will be intersected with the simulation output. The edgeSelectionList are the identifiers of the edges in the simulation's network, these are the edges to be measured to calculate the average speed. The parameters of the simulated annealing algorithm are the temperature (initialTemp), the amount of neighbors generated from each parameters configuration (numNeighbors), the number of routes values we are going to change to generate a new neighbor (changePositions), the amount of levels for temperature cooling (numTemp), and their reduction factors given in a vector(factors). And, a final section where the information about the route, where each route includes a start node and end node, both are road network identifiers (Open Street Map format), a third value that is a list of intermediate node, if required, and the last is a color to be able to identify each route in the during visualization.

These are maybe the most important parameters for the simulation, the others are merely file paths to set the file location and the commands to execute the sumo program with each configuration.

Once in the heart of the experiment, the algorithm will run the simulation that will run the routes spawning behaviour during 1800 seconds (30 minutes), it will discard the first and last 500 seconds intervals, which ares are initial and final states where the simulation is not balanced, meaning the involved segments are with little traffic flow. At the end of the simulation, the effective time is around 15 to 20 minutes of constant traffic flow in the sector. That is the result of a balance state that we expect in the rush hour in a real environment.

From here the calibration algorithm does its core job, they will compare the GPS records with the SUMO results and will try to optimize those the spawning times to generate the traffic flow we are looking for. For each experiment we set the initial parameters and we run the algorithm to get an ultimate solution that is optimized.

```
#segments ids (shapes)
edgesTest = [249042,249043,...,249049,249050]

#edge sim ids (sumo)
edgeSelectionList = [808111920, 808111919, ...
    ,808111621,808111620]

#Simulated Annealing Parameters
initialTemp = 15
numTemp = 5
factors = [0.6, 0.7, 0.7, 0.7, 0.8]

#neighbors generation
numNeighbors = 50 #10
changePositions = 3
periodMin = 20
periodMax = 25

#routes
["837481745", "837481830", "", "red"], #2
["837481745", "837481708", "", "green"], #3
.
.
.
["837481783", "837481830", "", "red"] #2
```

LISTING 3.5: Configuration file of the calibration algorithm

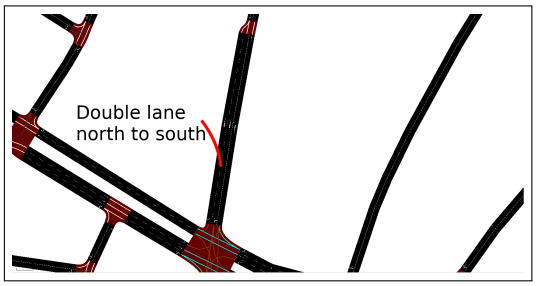
The final data we get is a specific configuration of simulation parameters that generated a solution that is almost a mirror of real-life conditions.

Once the simulation in each sector is optimized, we proceed with the implementation of changes proposed for the sector, trying to improve traffic flow by reducing the traffic congestion and increasing speed in some specific segments. For the final aim to quantity the improvement in those road segments.

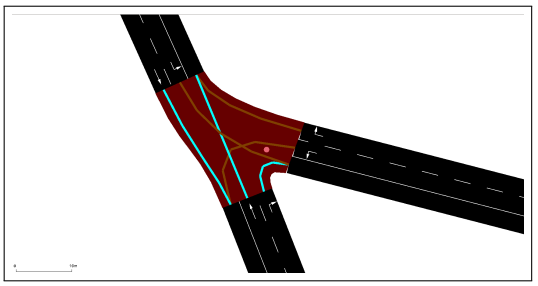
We did this to the five sectors. Some need the creation of one more lane, others instead, need new traffic rules and signals, also the creation of exclusive sections for buses. We describe these changes in detail.

Figure 3.7(a) shows sector A solution implementation, it needs a secondary lane to turn right in north-west direction. This will allow to reduce the traffic flow in the way north to south and the turn north to east, allowing more vehicles to turn, because vehicles going to west have their own way.

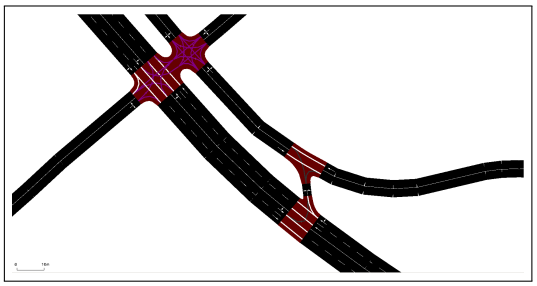
In sector B we are trying to reduce the impact of the triple junction, the solution here is to reorganize completely the turns at the junctions and adding traffic light



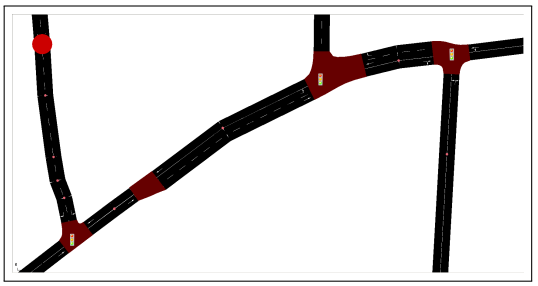
(a) Solution for sector A: double lane for direct right turn



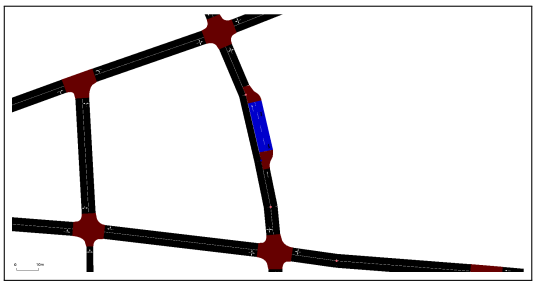
(b) Solution sector B: traffic light in junction allowing right turns in red



(c) Solution sector C: early entrance in the way from east to north



(d) Solution sector D: adjusted timing in 3 traffic lights



(e) Solution sector E: creation of off-line bus stop

FIGURE 3.7: Images of the proposed solutions for every sector

for control 3.7(b). Proving direct flow from north to south using traffic light, timing the way north to east, meanwhile allowing free right turns on red for the other two routes.

In sector C we have a problem in the double junction - train rails crossing sector, the solution we test here is the creation of an early entrance to the north sector as shown in the image 3.7(c), avoiding that vehicles have to stop and obstruct the junctions to turn right in the train rails crossing. Also, this will try to reduce the traffic congestion on the way from east to west.

Sector D is much more complicated, there is no space for new infrastructure, the street width is just the correct for the current state 3.7(d). The decision here is to set new traffic light timing according to the traffic flow. This approach can be more intelligent, it is even the preferred topic of several papers. The idea here

is to adjust timings to allow more flow in the straight ways while allowing right turns in red.

For sector E we attack the main reason for traffic delays: buses. We expect that, with the creation of around 5 off-line bus stops, for example in 3.7(e); can increase the speed of the vehicles on the way south to north, reducing the traffic congestion in the specific main road segments, which are really important because secondary ways enters to the main street and will compete for the road increasing the traffic congestion problematic.

After performing the calibration of original states, we created a new network file with the respective changes. Once again, the simulation without the calibration algorithms is run to gather data of the speed values of the same segments with the new changes. These values are collected and compared with the last optimized results from the simulation. Here we want to detect an increment on the speed of specific sector, performing a measure of distance for both results and running a T-Test to check the statistical relevance.

3.5 Sensitivity Analysis

We are calibrating input parameters of a simulation, that are basically periods of time (in seconds); to spawn vehicles in specific routes with the aim of generate a general traffic flow in a road sector that represents the real-life conditions.

The sensitivity analysis can be used to identify which input parameters really need to be calibrated and where sensitive to the inputs. And may be useful to identify elements of a modeling process and the regions of the inputs that are most responsible for producing an acceptable model. [14]

In this to study we looked for variations in the output based on changes in the input parameters of the optimal solution. Remember that these parameters are the period in seconds to spawn vehicles on each flow (route/path set for each sector), then the variations in the input is an set amount of seconds, which we are going to change according to the results.

We run simulations applying the changes for every input parameter once at the time to obtain a new value of RMSE and compare it with the optimal value

found during the calibration. If there is no change, we increment the variation in seconds and run simulations again until we get a RMSE greater than optimal and a difference of more than 1 m/s (3.6 km/h)

This easy but straightforward approach can show us what flow routes have the more impact on the traffic situation of the studied sectors. Making possible the future design of new solutions to reduce the traffic congestion and pay more attention to those problematic routes in future studies.

Chapter 4

Analysis of Results

4.1 Experimental Setup

The final experiments were completed in the cluster Kabré [11] hosted by the National Center for High Technology (CeNAT) of Costa Rica. We used one of the four Andalan nodes, recommended for sequential tools that need processing power and also can deal with some parallel problems as the ones we worked on.

The cluster operative system is CentOS Linux 7 (Core), using SLURM Workload Manager as the management and job scheduling system. For network editing and small simulation we used a Fedora 26 (Workstation Edition) machine, Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz, 4 cores, 2 threads per core and 16 GB of RAM memory.

Two of the four Andalan nodes feature an Intel Xeon, each one with 24 cores @ 2.20 GHz, 2 threads per core, and 64 GB of RAM. A third node features an Intel Xeon with 16 cores @ 2.10 GHz, 2 threads per core, and 64 GB of RAM. A fourth node features 10 cores, 2 threads per core, @ 2.20 GHz and 32 GB of RAM. The fourth and last node has 24 cores @ 2.40 GHz, 2 threads per core, and 128GB of RAM.

The following is a list of programming languages with the libraries used, as well as the clusters modules and the software utilized for this work.

Programming Languages

- Python version 3.8.5, installed as a Conda (v 4.8.3) environment in the cluster
  - numpy, subprocess, sklearn, pandas, PyMP version 0.4.3, among other commonly used libraries.
- R version 3.5.1
  - sf, sp, raster, dplyr, foreach, parrallel, doparallel, metrics, rgeos, rgdal

Cluster modules

- Geo-spatial Data Abstraction Library gdal v2.3.1
- Geometry Engine - Open Source GEOS version 3.6.3
- PROJ library version 5.1.0

Open Source Simulation and Edition Software

- Eclipse SUMO GUI Version version 1.6.0
- Eclipse SUMO netedit Version version 1.6.0
- SUMO OSMWebWizard

To execute one experiment, a SLURM script activates the Conda environment for Python, loads the required modules, and executes the main script. This script records the execution time using a simple difference between saved time values before and after running the Python script.

The complete code solution is in the github project named CalTraSi as **Cal**ibration of **Tra**ffic **Si**mulation: https://github.com/carlosgamboa/caltrasi.

4.2 Calibration Results

Sector A (represented in figure 3.1) has 15 routes to create heavy traffic in the main junction and the way north-south we are interested in. Three routes from west going to north, east and south. Another three routes coming from south going to west, north and east. Three more coming from the north to west, south and east. Three routes again from east to north, west and south and finally, three routes coming from south highway (Calle 80A in the map) going to east, north and west. More details about the intermediate nodes can be found in the configuration file, along with the network file of the sector.

Every iteration the algorithm calculates 50 neighbors from each solution changing 3 values at the time and, and iterating over five temperature levels. The initial value of the period of the input parameters is a random number from 20 to 30 seconds, with which we obtained an initial RMSE of 4.555. This value is given in units of meter per second, if we converted to km/h we get 16.398 km/h and we can visualize better the error. Once the calibration algorithm ends, we obtained the final RMSE of 0.629 m/s (2.26 km/s). A value less than the initial 4.555 and less than 1 m/s, that is what we are looking for.

The final two-sample paired T-test gives us a p-value of 0.685, with α equals to 0.1 the H_0 can't be rejected, of contrary we accept H_0 having statistical evidence that indicates the two vectors of speeds are similar. Supporting and validating the calibration algorithm results.

Sector B (use figure 3.2 as reference) only has 8 routes, three routes from north highway, going to south of the same highway and two routes going to the east (via 219) and south (via 236), one route in the highway going from south to north to increase traffic in the junction with via 219, two routes form via 219 to north and south, and two last routes going from south in via 236 to east and north. Those routes increment the traffic flow in the junction in study.

The algorithm runs with 5 levels of temperature, selecting 40 neighbors for each solution found and changing 3 values of those 8 parameters at the time to choose a new combination of parameters. The initial value of the RMSE with periods from 15 to 30 seconds is 11.422. The largest initial RMSE obtained in all our experiments. The calibration algorithm runs until we obtained a final RMSE of 0.973, a huge difference compared with the initial value and less than 1 m/s

threshold we are always looking for. To validate results the p-value of the T-test is 0.813, again, we accept the H_0 with α 0.1 indicating both resulted vectors are statistically similar.

Sector C (see figure 3.3) has 7 routes, because is a much smaller sector, but not a simple one. On the contrary, it is one of the more complicated due to the double junctions that enclose the train rails. Two routes coming from north, one goes to the south, the other one to north-east to Calle las Americas. From south, one route goes to north and one to north-east. From north-east the only route is going to south, there is no reason to represent other routes from north because they use a different road. Finally, two routes coming from Calle Pasito to north and south. All of these routes have a direct impact on the two main junctions.

For these simulations, the periods start with a random value from 30 to 40 seconds, the initial RMSE is 7.652. To reduce that value, we generated 60 neighbors for each solution, changing 2 positions at the time because it is only 7 routes, and we use 4 levels of temperature to obtain a final RMSE of 1.175. This datum is important because we expect a value less than 1 m/s, but the sector presents special conditions that complicates the simulation and those two main junctions sometimes generated a vehicle deadlock, a condition where cars block each other and get stuck waiting for others to move, but nobody moves at the end. The simulation controls this behavior setting a value to teleport vehicles to another position in their path and avoid such deadlocks.

All of this cause that calibration could not be so close, even the value obtained is close to 1, the statistical test shows a p-value of 0.104 slightly above the α of 0.10 selected for our hypothesis. This is sufficient to accept again the H_0 indicating both resulted vectors are statistically similar each other, supporting our calibration algorithm.

For the last studied sectors, we present two important cases of heavy traffic in the more complex conditions. A main road moves a high traffic flow to a specific direction while other routes fight each other for right of way, hindering the constant flow and reducing the speed in the majority of the segments, as well generating a chain of situations that impact negatively the general traffic conditions.

Sector D (see figure 3.4) is a clear example of mentioned above. In rush hour at 5pm the main traffic flow moves from west to east, with 3 traffic lights this sector is completely stressful to drive through to it. We can have lots of routes here, for

the study we create 15 routes. To summary those we can mention three routes coming from west, four coming from east to different directions, three routes from north, two coming from Calle la Cruz and two more from Calle Tobias Espeleta. Details can be found is the configuration file, at the moment we can focus on the amount of routes that create a lot of traffic, even so, initial results evidence we need to calibrate those spawning times.

With periods from 15 to 25 seconds, we obtained an initial RMSE of 5.215. Simulation run with five levels of temperature, selecting for each solution 60 neighbors and changing 3 values at the time. The final RMSE was 0.687, an adequate result. The statistical p-value gives us 0.144, slightly above 0.1 but enough to accept the H_0 to indicate both results vector are similar. With these results, the calibration process with this complicating scenarios is satisfactory.

Last sector, sector E as shown in figure 3.5 has a new variable, here the buses have the more negative impact in the traffic conditions. Here 16 routes where created manually using observations of the commonly paths used by vehicles. Bus routes, however, have only one direction and there are several bus destinies that need to share the main route (route 126 in the map). East and west are the origin of two spawn points each one, having a total of 6 spawning points, is one of points in the country that clearly illustrates the worst traffic conditions during rush hour.

Initial RMSE of 4.040 was obtained with initial periods from 20 to 25 seconds. Simulation run with 5 levels of temperature and selecting 50 neighbor per solution, changing 3 values each time. With that, calibration algorithm results in a final RMSE of 0.624, and with a p-value of 0.770 showing statistical relevance to accept H_0 to indicate that the speed vectors are similar each other.

The table 4.1 summaries the simulation results of the initial RMSE, the final RMSE once the calibration is run and the final p-value of the statistical test.

Sector	**Initial RMSE**	**Final RMSE**	**p-value**	n-size
A	4.555	0.629	0.685	14
B	11.422	0.973	0.813	8
C	7.652	1.175	0.104	7
D	5.215	0.687	0.144	15
E	4.040	0.624	0.770	15

TABLE 4.1: Measure of Performance and p-value calculation of the studied sectors

However, there is a crucial aspect here, the α of 0.1 is a little lax in terms that we can have more probability of getting Type II Error. We created a secondary scenario where we incremented the α value to 0.5 to reduce the Type II Error by increasing the power of the test. If we increase the significance level α from 0.1 to 0.5 we are increasing the Type I error, and we are reducing the Type II error. In this case, we accept most of the sectors. Resulting in just two sectors C and D rejected with a p-value clearly less than 0.5 has show in table 4.1.

The small sample size and the high data variability can be a reason the sector C and D are not passing the test. With a larger sample sizes allow hypothesis tests to detect smaller effects. To change this, we need to increase the amount of segment being evaluated, but we need to beware of this, because it might be increase the difficulty to control the calibration results. For variability we can take the data from specific periods of time to reduce variability and used them separated instead of calculating the average speed by segments. These options can be worth it to be studied in the future.

4.3 Evaluation of Proposed Traffic Solutions

To evaluate the proposed traffic solutions and test if they produce a positive impact on traffic conditions, we run a new simulation using the initial parameters of the optimized simulation and the changes applied in the network road and/or traffic rules; we take the resulting speeds for each involved road segmented and we compare them with the result of the optimized simulation. Showing the differences for each speed per segment and calculating the speed increment rate to quantify the impact of the proposals.

Beginning with sector A, this sector focused on 11 road segments that we chose strategically to test the more specific traffic conditions and have more control of the situation. Figure 4.1 shows the values of speed in the selected segments with for the optimized simulation and the solution proposed. Segments 5, 6, 7 and 8 (see figure in A) are the segments closer to the main junctions where we created a new lane to turn right and try to reduce the amount of vehicles going to south and east. For segment 5 the difference is not that big because is the closest segment to the junctions where cars need to stop on the traffic light, however, the other three segments show a difference of more than 14 m/s (50 km/h) in average. This means that cars coming from north have good speed and cars that need to turn right have complete free right-of-way to do it.

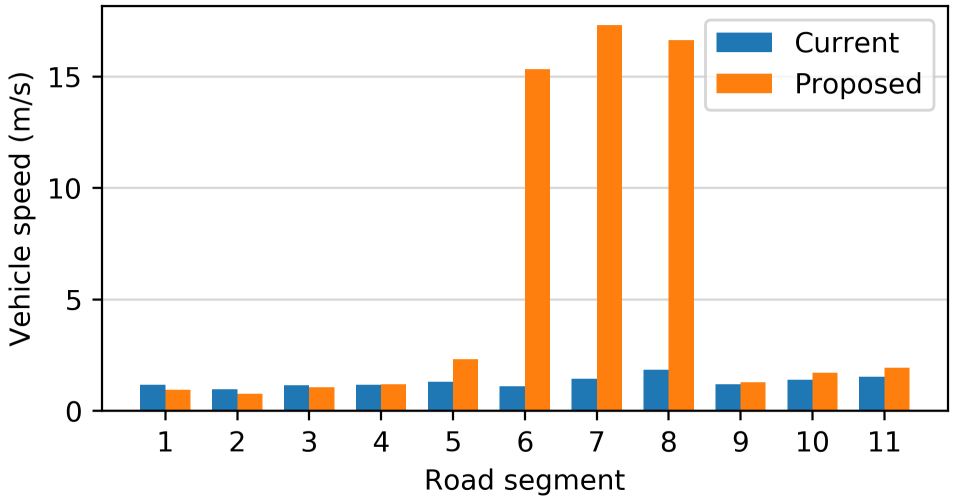


FIGURE 4.1: Comparison of segments speed in sector A

To understand better the speed increment we calculate the rate or speed up in table 4.2. Here see the increment of 13 times in segment 6, and 9 times in segment

8. This is an exceptional result, now we evidence that reliable simulation shows that a new turning lane can increase considerably the speed of cars reaching to the main junction from the way north.

Road segment	SegmentId	Speed increment rate
5	266373	1.78
6	266374	13.98
7	266375	12.07
8	266376	9.04

TABLE 4.2: Speed change in sector A with implemented solution

The proposed solution in sector B is a traffic light and free turns to right during red light on the main junction. In the chart 4.2 we can notice the increment of speed in sectors 6, 7, 8, 9, 12, 13, 14 and 15 compared with speeds of the current solution. Table 4.3 shows the rate of increment where segments 15 has the highest value. However, in the chart we can see a reduction of speed in segments 2, 3, 4, 5, those segments are the roads coming from south towards the junction, and we see this behavior in segments 10 an 11 as well. From image A.2 in appendix A for sector B we see segments 2, 10 and 11 are the closest to the junctions. Here the traffic light controls the flow making vehicles stops during a specific time period, although there is a straight way to continue to the right, the traffic lights reduced the average speed on those road segments.

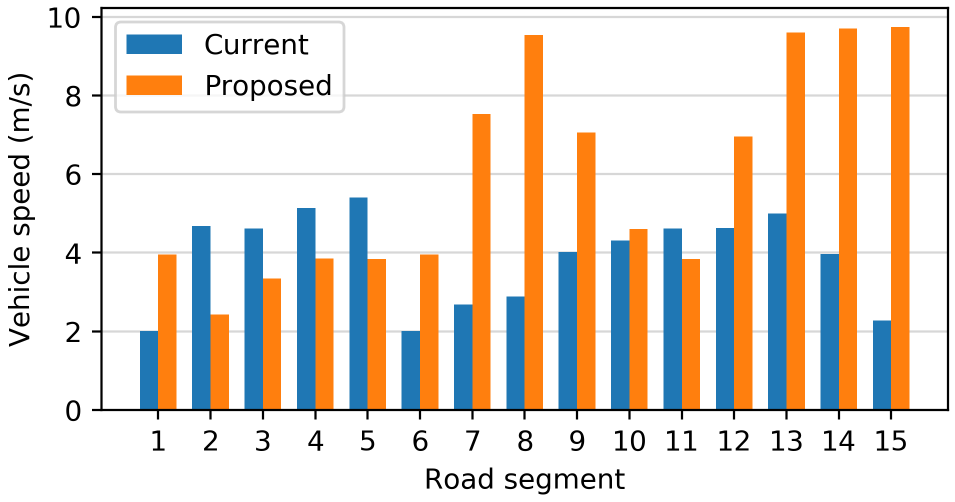


FIGURE 4.2: Comparison of segments speed in sector B

In sector C we pay more attention to segments 3, 4, 5, 6, 7 and 8. We present a simple solution, creating an early entrance from south to north avoiding the

Road segment	SegmentId	Speed increment rate
6	320604	1.96
7	320605	2.80
8	320606	3.29
9	320607	1.75
12	320610	1.50
13	320611	1.92
14	320612	2.44
15	320613	4.27

TABLE 4.3: Speed change in sector B with implemented solution

junctions in the trail rails crossing. With this, simulation shows (4.3) a speed increment in the segments 3, 4, 5, 6 reaching the junction, and increment in the segments 7 and 8 that are the exit from north to the main street. This solution seems to reduce the traffic congestion in the junctions, or at least we can visualize an improvement in the exit speed from the north section.

Those speed increment are the highest results found in this work. Table 4.4 displays the increment rates where the minimum rate is 2.66 times in segment 3, the farthest from the junction, and segment 5 and 6 that are the closest to the junctions shows the highest rate. This represents extremely important benefits in the traffic speed of the sector and suggests the reduction of the traffic jam.

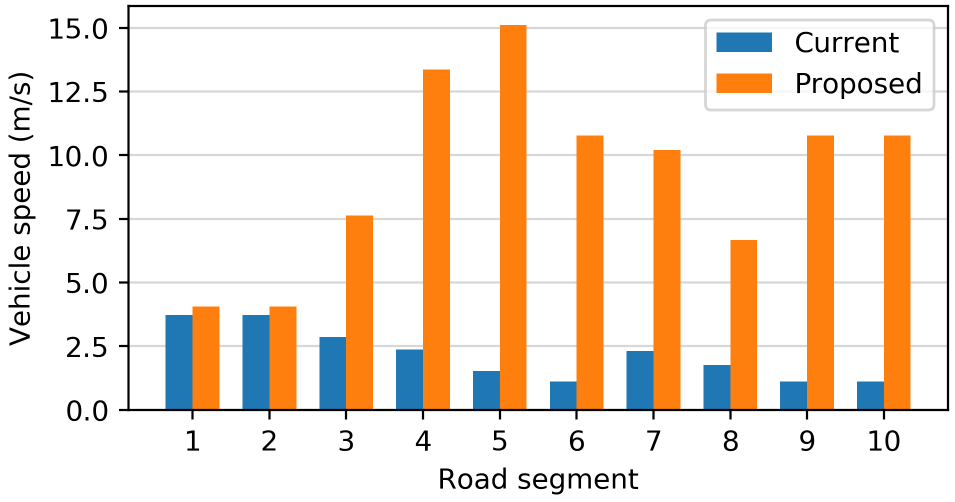


FIGURE 4.3: Comparison of segments speed in sector C

Sector D is one of the more complicated. Here there are a lot of routes and we have several limitations about the structural road changes that we can propose. The simplest solution is to adjust the timing of the existing traffic lights at the

Road segment	SegmentId	Speed increment rate
3	236680	2.66
4	236681	5.61
5	236682	9.90
6	236683	9.72
7	243507	4.42
8	243508	3.79
9	243765	9.72
10	243766	9.72

TABLE 4.4: Speed change in sector C with implemented solution

east on the map (3.4). That solution involves set a higher duration for routes in the main road, meaning the way from west to east will have more time to go through. Secondary turn time is reduced or adjusted according to the real time at the moment. In the chart 4.4 we see the values of the segments 1 and 2 that are the road segment closest to the traffic lights, here the average speed present almost no change, however, this can be caused to the natural breaking of vehicles approaching to the junctions. After that, in the other segments the speed increments, presenting the most important differences from segment 4 to 11.

This increment rate is displayed in the table 4.5 where the majority of segments show an increment of approximately the double of speed. This behavior is clear evidence that simulation shows a benefit in the speed of vehicles in the segments approaching the traffic lights, implying an improvement of the traffic jam.

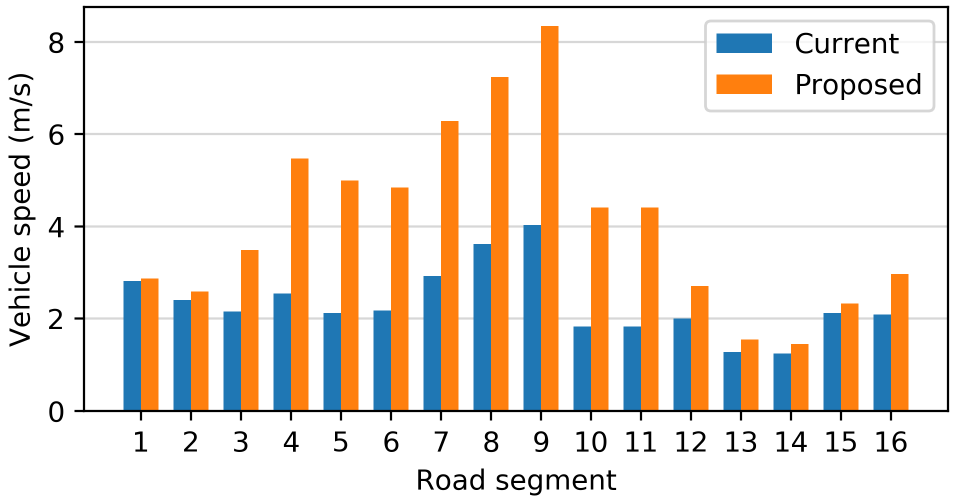


FIGURE 4.4: Comparison of segments speed in sector D

Road segment	SegmentId	Speed increment rate
3	291287	1.61
4	291288	2.15
5	291289	2.35
6	291290	2.22
7	291291	2.15
8	291292	2
9	291293	2.06
10	294134	2.40

TABLE 4.5: Speed change in sector D with implemented solution

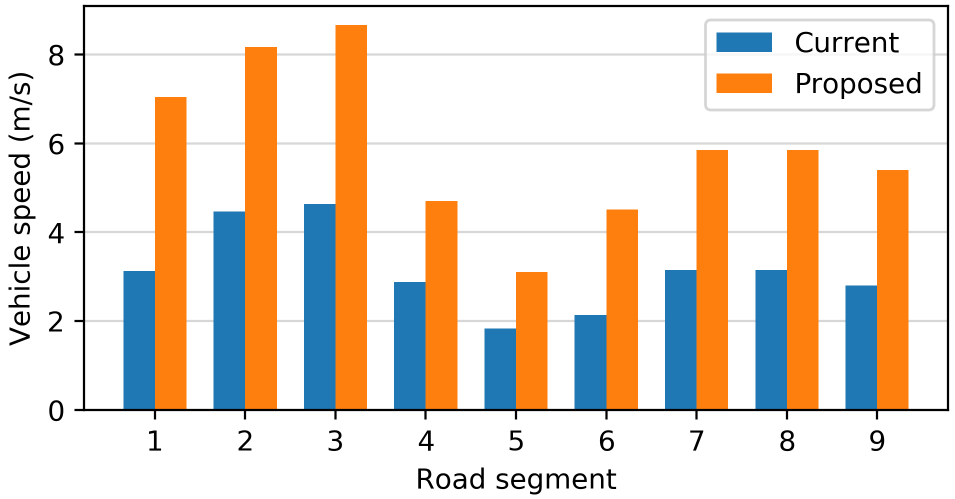


FIGURE 4.5: Comparison of segments speed in sector E

A new approach is presented in sector E. Knowing that buses cause the major impact in the vehicle speed in the main road, we decide to create 5 bus stop bays, allowing the bus leave the main lane and wait in the dedicated space while boarding passengers. This rule allows the vehicles in route continue with good speed avoiding heavy breaking and waiting time behind the stopped bus. This sector is large containing lots of segments, however, be focused in 9 of them, the central ones, that have more secondary lanes that arrive to the main road and also affects the traffic flow.

Simulation results with the proposed solution in chart 4.5 show an increment on the average speed of vehicles in all segments compared with the optimized solution found. Even thought this increment rate is mostly less than two times, as indicated 4.6. From the visual study of the simulation (4.6) we can observe that effectively bus rules reduce the waiting time, but this causes that more cars continue in the

Road segment	SegmentId	Speed increment rate
1	249042	2.25
2	249043	1.82
3	249044	1.86
4	249045	1.63
5	249046	1.69
6	249047	2.11
7	249048	1.85
8	249049	1.85
9	249050	1.92

TABLE 4.6: Speed change in sector E with implemented solution

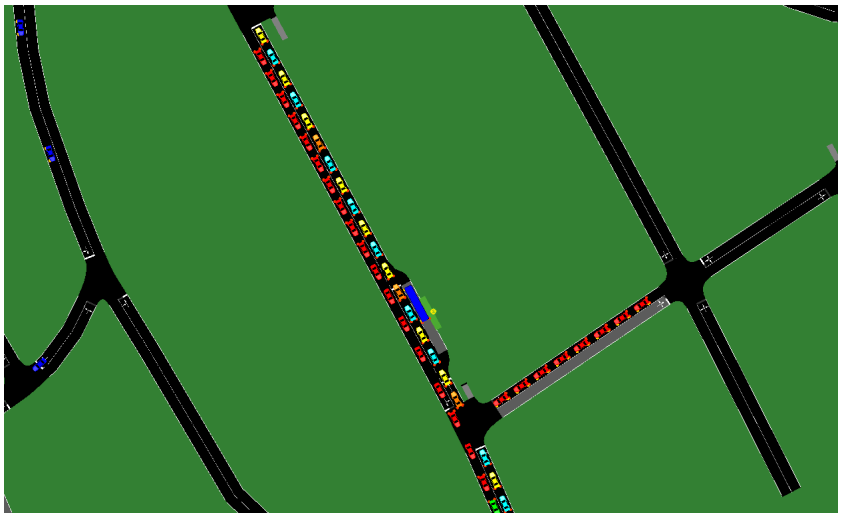


FIGURE 4.6: SUMO simulation view: bus stopping at bus-bay creating continuous flow of cars in the main road

route, generating more and more traffic. This is why speed increments, but not as much as other sector is the study show.

4.4 Sensitivity Analysis

The objective of sensitivity analysis is to detect those specific routes that are more sensible to changes and change the best RMSE that results in an optimal configuration of the spawning periods. In this section we present tables of results with the route ids and the numbers of seconds added to the optimal value that changes the new RMSE in more than 1 m/2, making the final solution not optimal. We verify this running the same two-paired t-test we used in the calibration process to evaluate the statistical relevance of the solution.

Route id	**Time change (sec)**
8, 9	-1
3	1
13	3
13	4
11	5
13	6

TABLE 4.7: Modified route for sensitivity analysis in sector A

Staring with sector A, table 4.7 shows the most sensible routes. The minimum value change is 1 second, where routes 8 and 9 affect the solution when their period changes -1 seconds, and route 3 affects the solution when 1 seconds is added, this means that those routes are extremely sensible and with minimum changes the traffic flow in the sector is affected and simulation can't represent the real scenario observed from the GPS records. Route 3 goes from south to north. 8 and 9 comes from west to north and east. Implies that those three routes can be the main cause of traffic flow in the sector. An important point here in the route 13, it starts changing the solution when we add 3 seconds and so on. That route is coming from south highway entering the main road and going to north, this delay in spawning time is causing a low traffic flow reflecting in the average speed in the roads segments and again not representing the real traffic conditions.

In table 4.8 we found route 6 as more sensible when we subtract -4, -3, and -2, and then route 5 is sensible when we start adding seconds. From this table, we can suggest that those two routes are crucial for the traffic congestion. If the period changes eventually the flow changes and the simulation can't represent the high traffic conditions we are looking for.

This route 5 is going from south to east affecting directly the main junction in study, and route 6 is going from east to north, this is the route affecting directly segments 11, 12, 13, 14 and 15, that we can see in the sector B figure in appendix A.

Route id	**Time change (sec)**
6	-4
1, 6, 7	-3
1,6	-2
5	-1
7	1
5	2
2, 5	3
0	4
4, 5	5
3, 4, 5	6

TABLE 4.8: Modified routes for sensitivity analysis in sector B

In sector C we found at least 6 routes very sensitive to change: routes 1 to 6. A sector where we defined 7 routes, almost all routes are sensitive parameters. This can be due the complexity of the street layout and the delicate of the real traffic conditions.

Route id	**Time change (sec)**
4, 6	-4
0, 5, 6	-3
1, 2, 3, 4, 5	-2
0, 4, 6	-1
1, 2, 3, 4, 5, 6	1
1, 2, 3, 4, 6	2
0, 1, 4, 5, 6	3
2, 6,	4
1, 2, 3, 4, 5	5
0, 3, 4, 5	6

TABLE 4.9: Modified routes for sensitivity analysis in sector C

In Sector D is peculiar, only route 2 affects the optimal solution after adding more than 6 seconds. Meaning this route is the one that has a real impact in the traffic flow. This route in the main route that generate vehicles going from west to complete east, of course, it has the biggest impact. A future approach to calibrate and propose a solution for the real traffic problems must focus in this specific vehicle way.

Route id	**Time change (sec)**
2	6
2	7
2	8
2	9
2	10

TABLE 4.10: Modifies routes for sensitivity analysis in sector D

Route id	**Time change (sec)**
0,1,2,4,6,7,8,10,11,12,13,15	-4
0,1,2,4,5,6,8,9,10,11,12,13,14,15	-3
0,1,2,3,4,5,6,7,8,9,14,15	-2
1,2,3,4,6,7,10,11,12,15	-1
0,1,3,4,5,6,9,11,15	1
0,1,5,6,8,10,12,13,14	2
0,2,3,5,7,8,9,10,11,14,15	3
0,1,2,3,4,5,6,7,8,13,14,15	4
0,1,2,4,6,8,9,10,11,12,13,14,15	5
0,2,4,5,6,7,11,12,13,14	6

TABLE 4.11: Modifies routes for sensitivity analysis in sector E

Finally, sector E is chaotic, almost every route is susceptible to changes, even the minimum change of 1 seconds alters the final solution and 10 routes reflect this behavior. What we can search here is the route that is less affected, but in the shorter range from -2 to 2 there is no clear route that be less involved. This can be a reason why the calibration algorithm was more complicated to set up for this sector.

4.5 Parallel Execution Performance

To analyze the performance of parallel code, we executed two experiments, first we test the efficiency of R code to process the GPS records. The parallel R code is used to intersect the spatial information of the reported traffic jams with the street segments of 100 meters each from data set *Red Vial Nacional* [38]. Table 4.12 shows the amount of road segments existing per sector and the jams reports that need to be intersect with.

Sector	Segments	Jams
C	749	128845
D	1110	325704
E	903	328123

TABLE 4.12: Size of data frames of 100 meter segments and reported jams

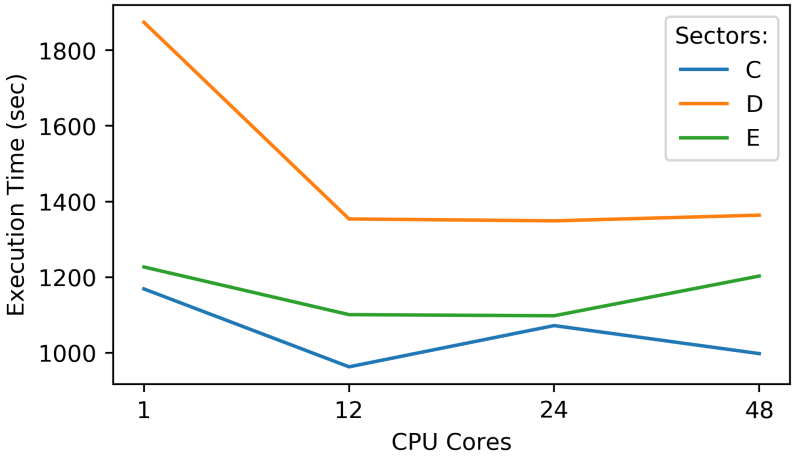


FIGURE 4.7: Execution time of pre-processing GPS data with 4 core configurations, for sectors C D and E

We used different configuration of cores to observe the time change, which results are displayed in figure 4.7. Clearly with one core the sector D takes the more time to execute. Later, with 12 cores the time reduce significantly around 520 seconds (8.66 minutes) and then with 24 and 48 cores the time remains almost constant. This behavior might be because a single processor at the server has 12 cores, even though the machine has 2 sockets, the optimal configuration is running just with one processor (one socket), with 24 cores the benefit is the same. Sector C presents a small improvement in execution time with 12 cores, increasing with 24 and reducing again with 48, at the end this does not represent a positive impact and is the sector with the lowest execution time of all the experiments. Finally,

sector E is constant, the time reduction with 12 and 24 cores is minimum around 2 minutes, and with 48 is almost the same as 1 core.

In conclusion, the preparation time of the GPS records with parallel code depends on the amount of jams reported in the gathered data and the amount of road segments by each district involved in the studied sector. We observed better results with networks with more segments.

	Speed Up (cores)		
Sectors	**12**	**24**	**48**
C	1.21	1.09	1.17
D	1.38	1.38	1.37
E	1.11	1.11	1.01

TABLE 4.13: Speed up of pre-processing GPS data algorithm written with R language

Last table 4.13 illustrates the speedup of the parallel implementation. Only sector D shows more speed up compared with the other two sectors. With 12 cores we can found an important speedup, however, none of those 3 values are so meaningful as we expect in a parallel code implementation.

Now, for the calibration algorithm, here we have Python PyMP library for OpenMP-like functionality for Python. In the calibration algorithm, we run the simulation with each parameters configuration several times to aggregate the average information of speeds. We tested the maximum of 5 times in parallel, 2 times in parallel, or just 1 time run. We obtained each execution time, but more important we calculate the speedup.

Table 4.14 shows the results for the sector C, where we can see and increment in performance of more than double when we set the algorithm to use 5 cores, speedup with 2 cores is not much. From here, we could test the algorithm with more than 10 cores running 10 times the simulation, but for simplicity reason we decided 5 times is enough to get statistical relevant aggregated data.

Ranks	**Execution Time (sec)**	**Speed Up**
1	7784	1
2	6528	1.19
5	3623	2.14

TABLE 4.14: Execution time and speedup values of the calibration algorithm for sector C

Chapter 5

Conclusions

5.1 Summary

In this work we complete satisfactory to all objectives:

- We selected adequate scenarios for study based on preliminary simulations and established guidelines. Every sector has its own traffic conditions and particularities, and proposed solution to ease their problematic conditions.
- We designed an optimization method to calibrate and validate traffic simulations using GPS navigation records, comparing the simulation results of speed per segment with the data gathered from events reported in GPS navigation platforms.
- We successfully created a pipeline of computational tools to implement the method to calibrate simulations for all the studied sectors and adaptable for future locations. We can describe the results from simulation and GPS records as statistically similar. Supporting the successful calibration of the simulation.
- We analyzed the impact of proposed solutions to the traffic problems. Creating new simulations to measure the speed in the involved road segments, and we observed improvements in over 3 segments in each sector. These are exciting results, because we are quantifying the traffic flow improvement, at least in the studied segments.

5.2 Contributions

- In this thesis document we present the study and design of a calibration strategy for traffic simulation using as the GPS records from a commercial navigation application. Maybe the only high-volume data source we can obtain in the country to analyze the traffic conditions.
- We created a software pipeline to simulate traffic road sectors where there is necessary to study and understand problematic conditions and proposed solution and take decisions.
- We quantify the solution improvement showing the increment of speed in specific roads segments, revealing that is it possible to ease the traffic congestion.

5.3 Limitations

- Maybe the principal limitation we have is the only source of data, we would like to have a national ITS to have access to more traffic data from sensors and video cameras for example. That will help the analysis because we can have more certainty about the actual traffic conditions.
- GPS records data present another limitation, that we only have reported events from jams, this jams represents and specific conditions where vehicles go to slow because of the traffic congestion. With this data, we don't know about conditions where a road is empty or have a fast and constant flow. At the end, in this work we simulated the worst traffic scenarios during a specific hour of the day.

5.4 Recommendations

- Out first recommendation will be to analyze the sector in different rush hour. It is known that morning and evenings behave differently by different factors, for example the change in start time and end time of rush hours for seasons or special events such as epidemiological crisis [38].

- The use of a faster and more reliable file format for GPS records, using R not optimized data format used can affect the processing time of data, with a more optimized format such as Parquet [17] we can reduce data wrangling time and improve the testing and implementation of experiment.

- The last recommendation for future work of this research, or similar traffic simulation studies, is to understand in the best way creation of traffic flow and that possibilities for the simulation. Because this aspect can be the more randomly created in real life, we never are sure of the amount of vehicles coming from a specific road, and for an undetermined reason the current flow can change as the drivers decide. That is why a crucial part of the traffic simulation studies must be focused on the sources and destinies of traffic.

5.5 Future Work

- An important question for future studies is to determine impact on new locations, study and analyze those sectors and proposed changes to solve the traffic congestion. The importance of this work is that we designed a general pipeline to be opened to study new road sectors as long as we considered the tool limitations.

- Future research could focus on simulation of public transportation and changes in traffic rules and their acceptance by the drivers. This can be a crucial aspect of design new rules to respect the most important public transport in CR, buses. Giving new guides to create bus-bays, rules to follow their right-of-way and measure the efficiency or not of those solutions.

- To study the parallel code performance, we can this on implement a real parallel Simulated Annealing algorithm to improve the time to execute the simulation, mainly focusing on the parallel execution of solution by neighbor. Also, and important contribution would be a test of the current implemented algorithm against other heuristic alternatives and possibly some Machine Learning options.

Appendix A

Segmented Maps

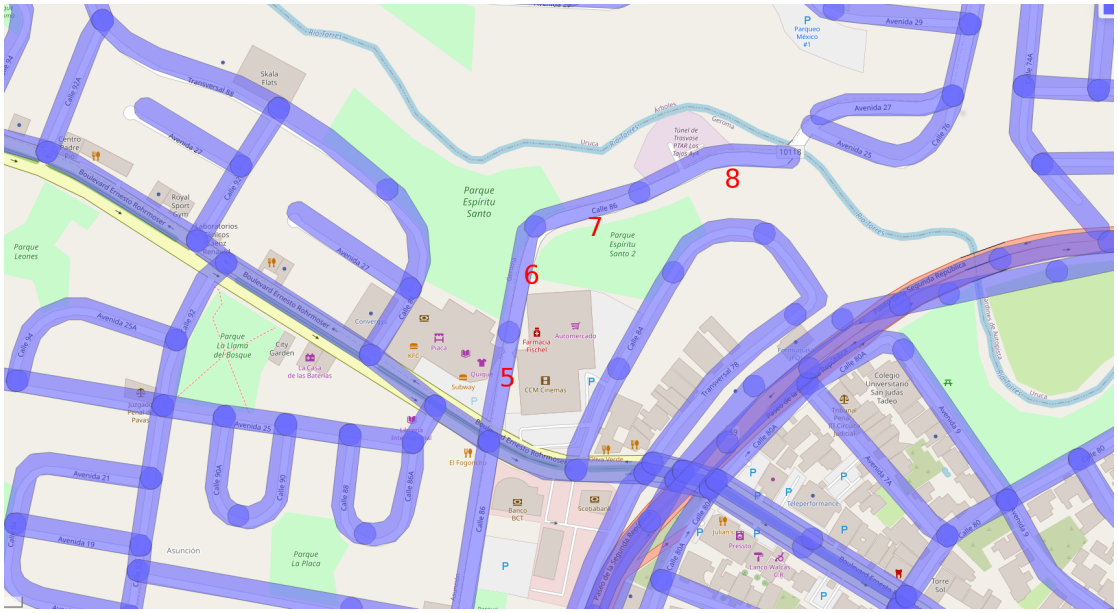


FIGURE A.1: 100-meter segment map of sector A

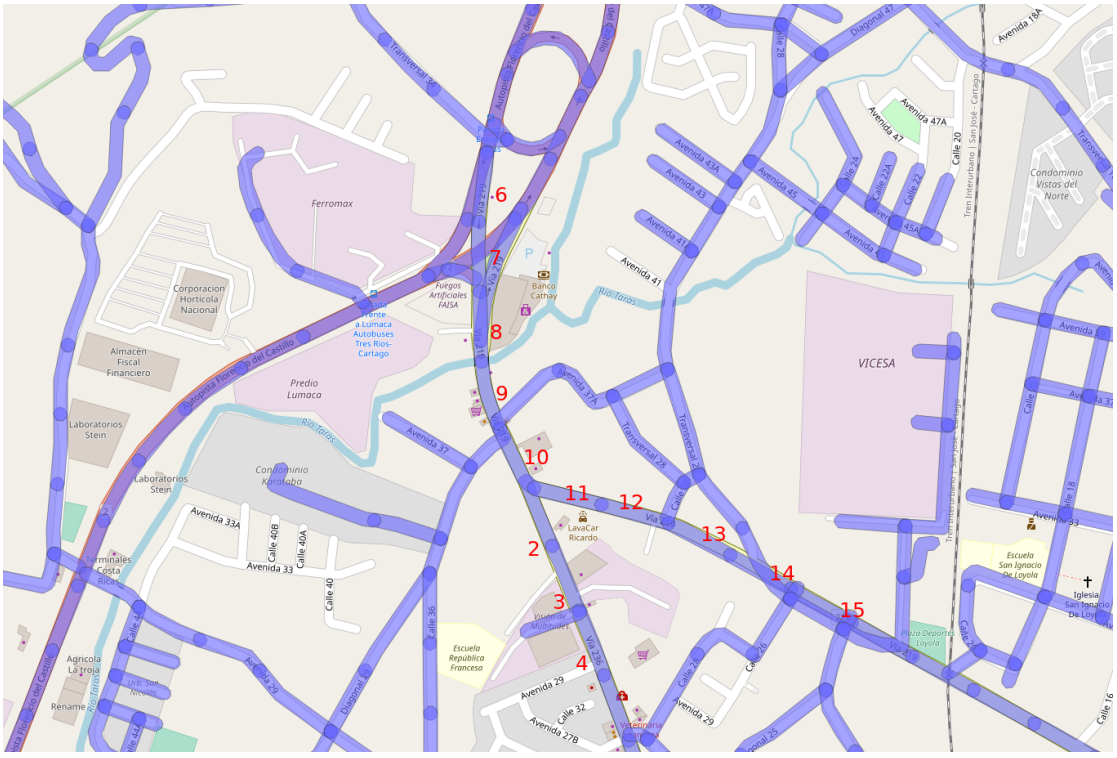


FIGURE A.2: 100-meter segment map of sector B

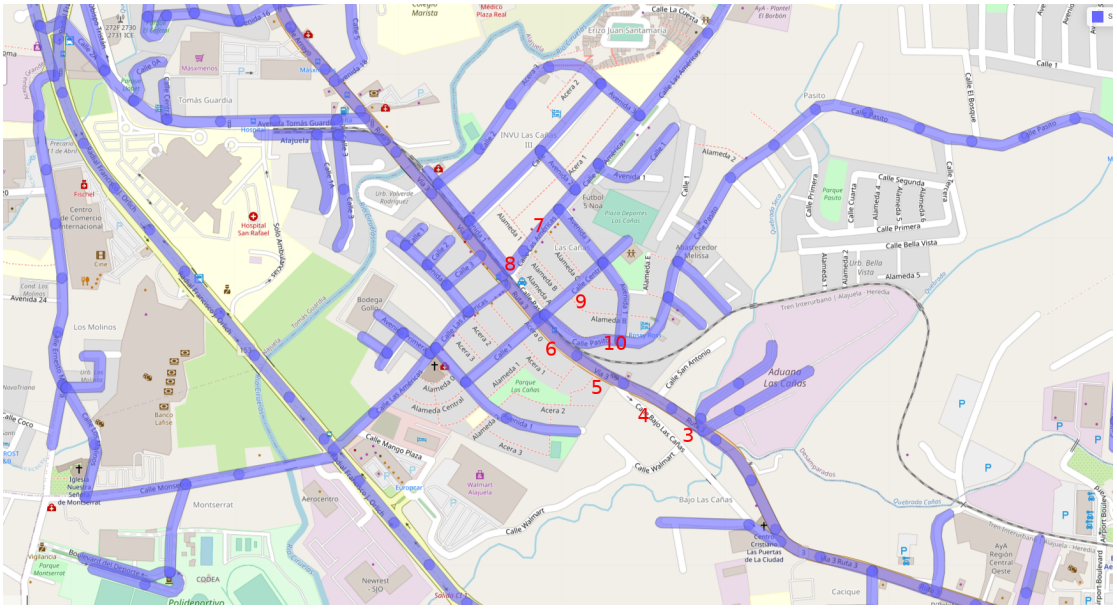


FIGURE A.3: 100-meter segment map of sector C

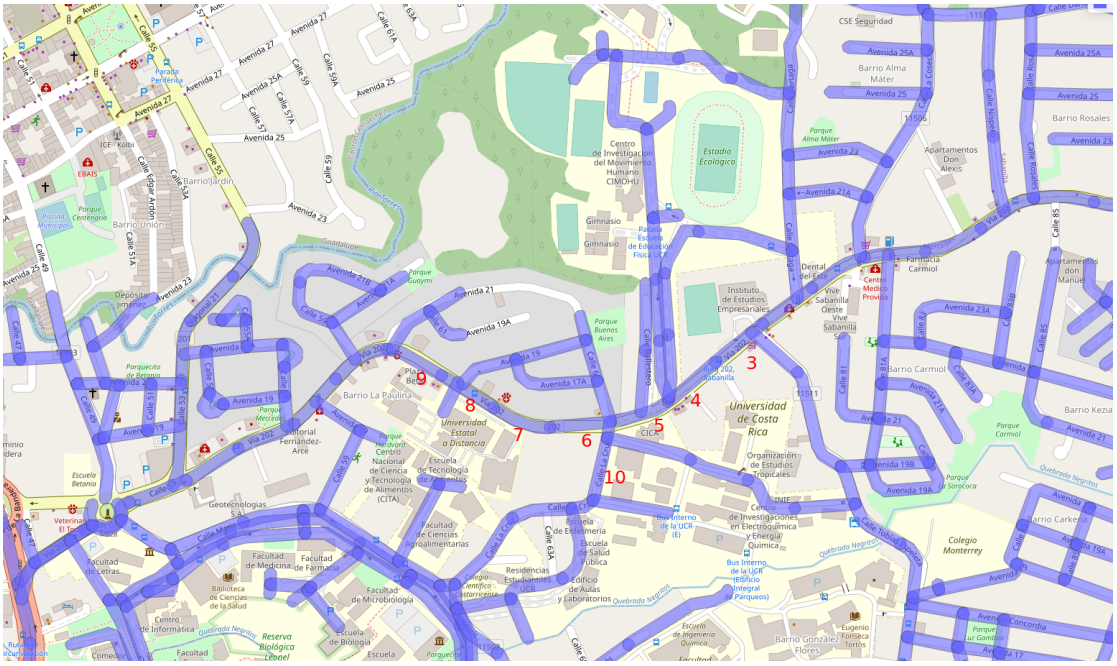


FIGURE A.4: 100-meter segment map of sector D

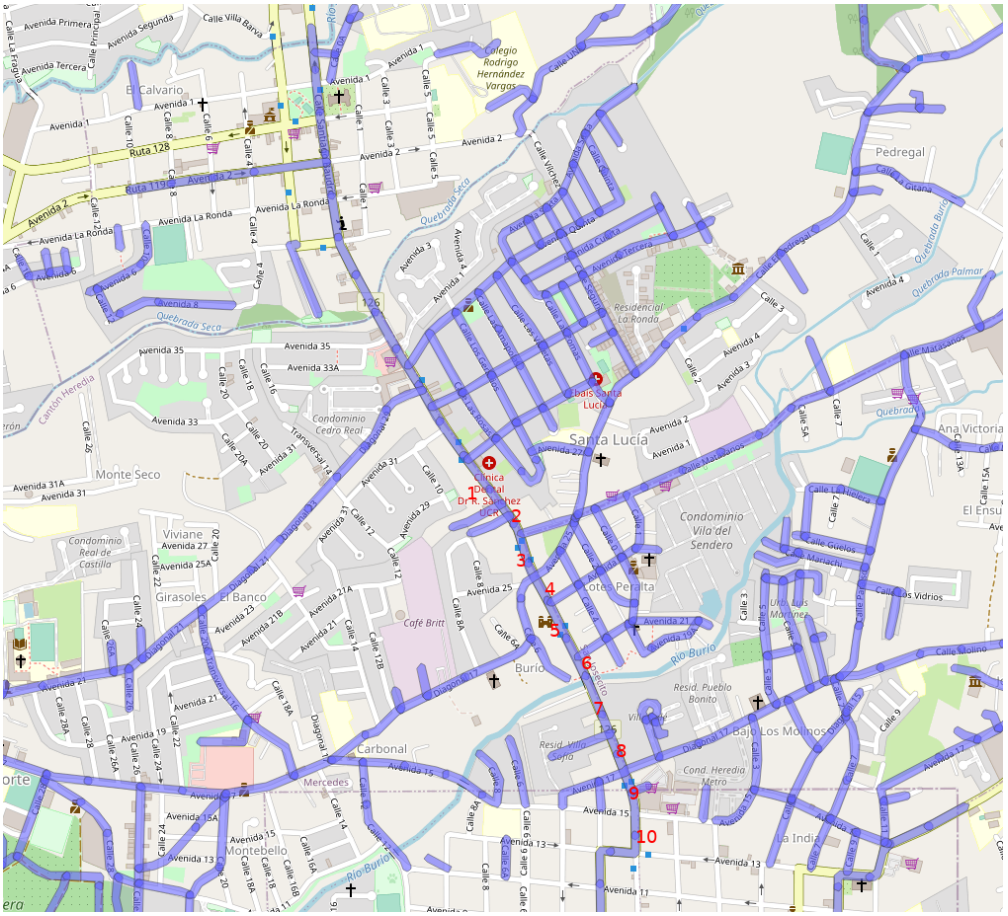


FIGURE A.5: 100-meter segment map of sector E

Bibliography

[1] S. Amaran, N. V. Sahinidis, B. Sharda, and S. J. Bury. Simulation optimization: a review of algorithms and applications. *Annals of Operations Research*, 240(1):351–380, 2016.

[2] S. Amini and C. Prehofer. Big Data Analytics Architecture for Real-Time Traffic Control. *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 710–715, 2017.

[3] C. ANTONIOU, J. BARCELÒ, M. BRACKSTONE, H. CELIKOGLU, B. CIUFFO, V. PUNZO, P. SYKES, T. TOLEDO, P. VORTISCH, P. WAGNER, V. PUNZO, and M. BRACKSTONE. Traffic Simulation: Case for guidelines. Technical report, Luxembourg, 2014.

[4] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri. Smarter traffic prediction using big data, in-memory computing, deep learning and gpus. *Sensors (Switzerland)*, 19(9):1–34, 2019.

[5] R. Balakrishna, C. Antoniou, M. Ben-Akiva, H. N. Koutsopoulos, and Y. Wen. Calibration of microscopic traffic simulation models: Methods and application. *Transportation Research Record*, (1999):198–207, 2007.

[6] J. Barcelo. *Fundamentals of Traffic Simulation (International Series in Operations Research & Management Science)*. Springer US, 2010.

[7] C. N. Bowman and J. A. Miller. Modeling traffic flow using simulation and big data analytics. *Proceedings - Winter Simulation Conference*, pages 1206–1217, 2017.

[8] Y. Carson and A. Maria. Simulation optimization: Methods and applications. *Winter Simulation Conference Proceedings*, pages 118–126, 1997.

[9] M. W. Carter, C. C. Price, and G. Rabadi. *Operations Research. A Practical Introduction*. CRC Press, second edi edition, 2019.

[10] Y. Celik and A. T. Karadeniz. Urban Traffic Optimization with Real Time Intelligence Intersection Traffic Light System. *International Journal of Intelligent Systems and Applications in Engineering*, 3(6):214–219, 2018.

[11] Cluster CNCA. Guía de usuario. http://kabre.cenat.ac.cr/guia/.

[12] Colegio Federado de Ingenieros y Arquitectos de Costa Rica, CFIA. Congestionamiento del flujo vehicular en la gran Área metropolitana de san josÉ: recopilación, análisis y posicionamiento, 2005.

[13] M. Cubero-Corella, E. Durán-Monge, W. Díaz, E. Meneses, and S. Gómez-Campos. Modelling Road Saturation Dynamics on a Complex Transportation Network Based on GPS Navigation Software Data. *Communications in Computer and Information Science*, 1087 CCIS(February):136–149, 2020.

[14] W. Daamen, C. Buisson, and S. P. Hoogendoorn. *Traffic Simulation and Data: Validation Methods and Applications*. CRC Press, 2014.

[15] E. Felemban and A. A. Sheikh. A Review on Mobile and Sensor Networks Innovations in Intelligent Transportation Systems. *Journal of Transportation Technologies*, 04(03):196–204, 2014.

[16] C. Flitsch, K.-H. Kastner, K. Bósa, and M. Neubauer. Calibrating Traffic Simulation Models in SUMO Based upon Diverse Historical Real-Time Traffic Data – Lessons Learned in ITS Upper Austria. *SUMO 2018- Simulating Autonomous and Intermodal Transport Systems Engineering*, 2:25–6, 2018.

[17] A. S. Foundation. Apache parquet. `https://parquet.apache.org/documentation/latest/`, Last access: 30 April 2021.

[18] S. Gómez Campos and M. Cubero. Congestión vial en los cantones de Costa Rica. 2019.

[19] Instituto Nacional de Estadistica y Censos. X censo nacional de población y vi de vivienda 2011, resultados generales, censo 2011, 2011.

[20] N. Kheterpal, K. Parvate, C. Wu, A. Kreidieh, E. Vinitsky, and A. Bayen. Flow: Deep Reinforcement Learning for Control in SUMO. *SUMO 2018- Simulating Autonomous and Intermodal Transport Systems*, 2:134–115, 2018.

[21] R. Khokale and A. Ghate. Data Mining for Traffic Prediction and Analysis using Big Data. *International Journal of Engineering Trends and Technology*, 48(3):152–156, 2017.

[22] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent Development and Applications of {SUMO - Simulation of Urban MObility}. *International Journal On Advances in Systems and Measurements*, 5(3):128–138, 2012.

[23] D. Krajzewicz, G. Hertkorn, P. Wagner, and C. Rössel. An example of microscopic car models validation using the open source traffic simulation SUMO. *Proceedings of Simulation in Industry 14th European Simulation Symposium*, pages 318–322, 2002.

[24] D. Krajzewicz, G. Hertkorn, P. Wagner, and C. Rössel. SUMO (Simulation of Urban MObility) An open-source traffic simulation. *... Symposium on Simulation ...*, pages 63–68, 2002.

[25] C. Lassner. Pymp: Easy, openmp style multiprocessing for python on unix. `https://github.com/classner/pymp`, 2020.

[26] S. B. Li, G. M. Wang, T. Wang, and H. L. Ren. Research on the Method of Traffic Organization and Optimization Based on Dynamic Traffic Flow Model. *Discrete Dynamics in Nature and Society*, 2017, 2017.

[27] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, P. Wagner, and E. Wiebner. Microscopic Traffic Simulation using SUMO. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018-Novem:2575–2582, 2018.

[28] Y. Lv, Y. Duan, and W. Kang. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *Ieee Transactions On Intelligent Transportation Systems*, pages 1–9, 2017.

[29] MINISTERIO DE VIVIENDA Y ASENTAMIENTOS HUMANOS. Plan de desarrollo urbano para la gran Área metropolitana (gam), 2013.

[30] D. Ni. *TRAFFIC FLOW THEORY TRAFFIC FLOW*. Elsevier, MA, USA, 2016.

[31] A. Nkaro. Traffic Data Collection and Analysis, 2004.

[32] OECD. Big Data and Transport Understanding and assessing options Corporate Partnership Board Report Corporate Partnership Board CPB. Technical report, International Transport Forum, 2015.

[33] K. Ozbay, S. Mudigonda, E. Morgul, and H. Yang. Big Data and the Calibration and Validation of Traffic Simulation Models. *Transportation Research Board 2015*, 2015.

[34] P. S. Pacheco. *Introduction to Parallel Programming*. Elsevier Inc, 2011.

[35] P. M. Pardalos, D. Z. Du, and R. L. Graham. *Handbook of Combinatorial Optimization*, volume 1-5. 2013.

[36] C. D. Paternina Arboleda, J. R. Montoya Torres, and A. Fábregas Ariza. Simulation-optimization using a reinforcement learning approach. *Proceedings - Winter Simulation Conference*, pages 1376–1383, 2008.

[37] E. Pebesma and R. S. Bivand. *Classes and Methods for Spatial Data: the sp Package*, 2015.

[38] Programa Estado de la Nación. Capítulo 7 : Patrones de la movilidad en tiempos de pandemia: una aproximación con técnicas del "big data" [Informe Estado de la Nación 2020]. pages 231–254, 2020.

[39] Programa Estado de la Nación. *Informe 2018, Estado de la Nación en desarrollo humano sostenible*, chapter 6. Transporte y movilidad: retos en favor del desarrollo urbano. Estado de la Nación, 2018.

[40] R. Rebba, S. Huang, Y. Liu, and S. Mahadevan. Statistical validation of simulation models. *International Journal of Materials and Product Technology*, 25(1-3):164–181, 2006.

[41] F. Rossi. Shared memory parallel programming in r. `https://apiacoa.org/teaching/big-data/R-smp.en.html`, Last update: 29 January 2021.

[42] R. Samarajiva, S. Lokanathan, K. Madhawa, G. Kreindler, and D. Maldeniya. Big data to improve urban planning. *Economic and Political Weekly*, 50(22):42–48, 2015.

[43] A. Sarok and R. Fujimoto. Smart city real-time data-driven transportation simulation. *Journal of Chemical Information and Modeling*, 53(9):1689–1699, 2019.

[44] Q. Shi and M. Abdel-Aty. Big Data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. *Transportation Research Part C: Emerging Technologies*, 58:380–394, 2015.

[45] SUMO. Sumo: From 30.000 feet. `https://sumo.dlr.de/docs/sumo.html#gui_only`.

[46] M. Treiber and A. Kesting. *Traffic flow dynamics: Data, models and simulation*. 2013.

[47] M. Wallig, M. Corporation, S. Weston, and D. Tenenbaum. *Foreach Parallel Adaptor for the 'parallel' Package*, 2020. R package version 1.0.16.

[48] L. F. Wang and L. Y. Shi. Simulation optimization: a review on theory and applications. *Zidonghua Xuebao/Acta Automatica Sinica*, 39(11):1957–1968, 2013.

[49] S. Weston. Using the foreach package. `https://cran.r-project.org/web/packages/foreach/vignettes/foreach.html`.

[50] J. L. Zambrano, C. T. Calafate, D. Soler, J. C. Cano, and P. Manzoni. Using Real Traffic Data for ITS Simulation: Procedure and Validation. *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, pages 161–170, 2016.

[51] G. Zeng. Application of Big Data in Intelligent Traffic System. *IOSR Journal of Computer Engineering*, 17(1):2278–661, 2015.

[52] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen. Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1624–1639, 2011.