Universidad de Málaga

Doctoral Thesis

Realistic Modeling of Smart Mobility Problems with Environmental Concerns

Author:
Christian Cintrano

Supervisor:
Dr. Francisco Chicano
Prof. Enrique Alba

A thesis submitted in fulfillment of the requirements for the degree of Doctorado en Tecnologías Informáticas

in the

Departamento de Lenguajes y Ciencias de la Computación
E.T.S.I. Informática

May, 2022

AUTOR: Christian Cintrano López

https://orcid.org/0000-0003-2346-2198

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

Declaration of Authorship

I, Christian CINTRANO , declare that this thesis titled, “Realistic Modeling of Smart Mobility Problems with Environmental Concerns” and the work presented in it are my own. I confirm that:

Under the supervision of Enrique ALBA and Francisco CHICANO.

The thesis submitted is an original work that does not infringe on intellectual property rights or industrial property rights, or others, by the current legal system (Royal Legislative Decree 1/1996, of April 12, approving the revised text of the Intellectual Property Law, regularizing, clarifying and harmonizing the current legal provisions on the subject), amended by Law 2/2019, of March 1.

I also assume, before the University of Malaga and before any other instance, the responsibility that may arise in case of plagiarism of content in the thesis submitted, according to the current legal system.

Date: May, 2022

Signed: Christian Cintrano	Signed: Enrique Alba
PhD Student	Supervisor

Signed: Francisco Chicano

Thesis Director

UNIVERSIDAD DE MÁLAGA

Abstract

E.T.S.I. Informática
Departamento de Lenguajes y Ciencias de la Computación

Doctorado en Tecnologías Informáticas

Realistic Modeling of Smart Mobility Problems with Environmental Concerns

by Christian CINTRANO

Traffic in cities has become a significant issue today. Not only because of the logistical problems involved but also because of the associated gas emissions. Taking into account the impact that mobility in cities has on the environment is vital for a sustainable future. Objective number 11 "Sustainable Cities and Communities", of the Sustainable Development Goals, promote Smart Cities from the perspective of sustainability.

In this thesis, we want to dive into the problem of environmental improvement and ecological transport. However, many works in the literature do not take into account these objectives or use synthetic scenarios to test their contributions. We rely on many existing open data sources, which we include in our models to obtain realistic results. In addition, we focus our study on the actual city of Malaga, Spain. Following this line, we have also offered all the results, codes, and instances as open research data.

We have used different optimization algorithms to work with the various problems raised in this thesis. We have focused mainly, but not exclusively, on metaheuristic algorithms, which have proven their incredible effectiveness in many real-world problems.

Several ecological mobility problems have been worked on within this thesis. This research has been reflected in different scientific publications that support this thesis: 3 articles in JCR journals (all Q1) and 14 congresses articles. Several problems covering various aspects of current mobility in cities have been highlighted. The selected problems are optimal traffic light planning, traffic analysis considering freight vehicles, intelligent placement of infrastructure needed for green transport (bicycles and electric vehicles), and the problem of vehicle routing considering various aspects such as travel time and emissions.

This thesis takes the first steps in environmental improvement and mobility optimization. However, we will not stop here, and we will continue working on sustainable initiatives to improve cities, help citizens' daily lives, and enhance the environment.

UNIVERSIDAD DE MÁLAGA

Resumen

E.T.S.I. Informática
Departamento de Lenguajes y Ciencias de la Computación

Doctorado en Tecnologías Informáticas

Realistic Modeling of Smart Mobility Problems with Environmental Concerns

by Christian CINTRANO

El tráfico en las ciudades se ha convertido en un problema de vital importancia. No solo por los problemas logísticos, sino también por las emisiones de gases asociadas. Considerar el impacto que la movilidad tiene sobre el medio ambiente es vital para un futuro sostenible. El objetivo número 11 "Ciudades y Comunidades Sostenibles", de los Objetivos de Desarrollo Sostenible, promueve las Ciudades Inteligentes desde la perspectiva de la sostenibilidad.

En esta tesis, queremos sumergirnos en el problema de la mejora medioambiental y el transporte ecológico. Sin embargo, muchos trabajos en la literatura no tienen en cuenta estos objetivos o utilizan escenarios sintéticos para probar sus aportaciones. Nosotros nos basamos en datos abiertos, que incluimos en nuestros modelos para obtener resultados realistas. Además, centramos nuestro estudio en la ciudad real de Málaga, España. Siguiendo esta línea, también hemos ofrecido todos los resultados, códigos e instancias como datos abiertos de investigación.

Hemos empleado diferentes algoritmos de optimización para trabajar con los problemas presentados en esta tesis. Nos hemos centrado principalmente, aunque no exclusivamente, en los algoritmos metaheurísticos, que han demostrado su eficacia en muchos otros problemas del mundo real.

En esta tesis se han trabajado con varios problemas de movilidad ecológica. Esta investigación se ha plasmado en diferentes contribuciones que avalan esta tesis: 3 artículos en revistas JCR (todos Q1) y 14 artículos en congresos. Hemos destacado varios problemas de la movilidad actual en las ciudades. Los problemas seleccionados son la planificación óptima de los semáforos, el análisis del tráfico considerando los vehículos de carga, la colocación inteligente de las infraestructuras necesarias para el transporte ecológico (bicicletas y vehículos eléctricos) y el problema del encaminamiento de vehículos considerando diversos aspectos como el tiempo de viaje y las emisiones.

Esta tesis da los primeros pasos en la mejora medioambiental y la optimización de la movilidad. Sin embargo, no nos detendremos aquí, y seguiremos trabajando en iniciativas sostenibles para mejorar las ciudades, ayudar a la vida cotidiana de los ciudadanos y mejorar el medio ambiente.

Acknowledgements

Many people have contributed to me during this thesis, both professionally and personally. I will try to mention them all and thank them all, and I am sorry if I have left anyone out.

First of all, I would like to thank my supervisors, Francisco Chicano and Enrique Alba. Thank you very much for all your advice and for not losing faith in me.

Of course, I also want to thank all my colleagues in the NEO group. I cannot list them all because the list would be longer than the thesis. They have given me a lot, both as a researcher and people. I would especially highlight Jamal Toutouh, Daniel H. Stolfi, Javier Ferrer, Gabriel Luque, Zakaria Dahi, José Ángel Morell, Andrés Camero, Yesnier Bravo, and Javier Arellano

I would also like to thank Thomas Stützle and the whole IRIDIA group in Brussels, who welcomed me with open arms and from whom I learned a lot.

For the rest of the acknowledgments I will write in Spanish.

Muchas gracias a mis grandes amigos David, Francisco y Guillermo por todo su apoyo y compañía en estos años. Sé que siempre puedo contar con vosotros. También dar las gracias a todos mis compañeros del aikido y el arco (entre otros), con los que comparto muchísimas aficiones y que me acogieron como uno más desde el primer momento.

Por supuesto, también quiero agradecer a mis padres María y Enrique, y a mi hermana Elisa. No dejaron de apoyarme en todo momento. Solo espero, que estén orgullosos del todo lo que he conseguido.

Y por último, quiero agradecer especialmente a las personas que más habría querido que leyeran esto, mis abuelos: Francisco, Antonia, Miguel y Ana María. De todas las personas del mundo, ellos eran los que más han creído en mí. Son los que más me comprendían y entendían. Y simplemente decir que esta tesis va por ellos para que estén donde estén, se sientan orgullosos de su nieto.

Financial support Christian Cintrano was partially supported by FPI grant (BES-2015-074805) from Spanish MINECO, FEDER project TIN2014-57341-R (`http://moveon.lcc.uma.es/`). Also, this PhD thesis was partially funded by the University of Malaga, Andalucía Tech, Consejería de Economía y Conocimiento de la Junta de Andaluía, the Spanish MINECO, and FEDER projects: TIN2014-57341-R, TIN2016-81766-REDT, TIN2017-88213-R, UMA18-FEDER-JA-003, and 2020-116727RB-I00.

Contents

List of Figures

List of Tables

A mi hermana y padres por aguantar todas mis locuras.
Y a mis abuelos por todo su apoyo y ejemplo.

Chapter 1

Introduction

1.1 Motivation

The Sustainable Development Goals are an initiative within the United Nations 2030 Agenda that seeks to abolish social inequalities, combat climate change, promote industrial and technological growth and, in general, a better future for all people. In particular, concern for the environment is vital within these objectives. Many moments in our daily lives can contribute to the improvement of the environment, such as recycling, using public transport, turning off the tap when we are not using it, etc. In particular, mobility in cities has a significant impact on gas emissions. In this line, it seems natural to make cities more sustainable by improving their mobility.

Cities are people's major ecosystems. The vast majority of the world's population lives in cities, and the experts predict that even more people will move from the countryside to the cities in the coming years. This state of growth has made it very difficult for many cities to adapt to these significant changes. The staff of the municipalities carries out the management of the cities. City administrators use data collection through surveys, dissemination of events and performances through posters, etc., to define and apply different strategies to improve the quality of life for citizens. However, decision making through expert knowledge alone must give way to a more optimal and objective one by using new computational advances. The relevant computer advances in data collection, dashboards for decision making, the most use of social networks, etc., have yielded a society that does not contemplate life without technology. Cities have been adapting to these changes, becoming "smarter".

However, transforming a city into a smart city is not easy. The problems to be solved are more complex and involve many variables. Besides, the information available is not always reliable and error-free. This lack of accuracy makes the classic problems, which were already complex from a computational point of view, hinder their models from adapting them to the reality of the city. Many researchers present fast algorithmic solutions to solve very complex real problems. However, in a considerable number of them, either the problems are not so "difficult" or the tests performed are made with synthetic or *ad hoc* data.

In this thesis dissertation, we propose several models of real problems, trying to represent the corresponding city as best as possible. We base our models on the broad state of the art of some of these problems (or similar ones). We use all the real data available to create an image of the city more faithful to its situation and characteristics.

1.2 Objectives and Phases

The main objective of this PhD thesis is the realistic modeling of problems of Smart Mobility taking into account the environmental impact. For this purpose, we have used open data and we have analyzed different optimization techniques and Smart City problems. New algorithmic solutions to various urban mobility problems have been proposed. The main objective of the thesis is decomposed in the following goals:

- Analyze intelligent mobility problems that are of importance to citizens and the environment.
- Model these problems so that they capture most of the details of the real situation.
- Use existing open data sources to solve the problems realistically instead of synthetic laboratory data.
- Apply diverse optimization techniques to solve our models and analyze the results through their impact on society and the environment.

The methodology of work in this thesis has followed the guidelines of the Scientific Method. Initially, an observation phase is carried out, in which the state of the art of different Smart Mobility problems and the techniques used to solve them are analyzed. Then, we explore possible improvements or new problems and formulate working hypotheses. In short, we have focused on modeling the problems, adding as much realism as possible. Then, we design and execute several experiments to validate or refute the different hypotheses. In this PhD thesis, we decided to use mainly Open Data to work in real cities scenarios. Finally, we analyze the obtained results and obtain different conclusions.

In general, the working hypothesis of this PhD thesis is:

> Multiple statistical techniques combined with the numerous open data offered by municipalities and public entities can be used to model/solve problems, especially those related to climate change, in a way very close to reality, without the need for large expenditures in deployment and field testing.

Multiple works corroborate this hypothesis and support this thesis. In addition to producing scientific articles supporting the results of the thesis also focused on disseminating the results through various magazines, news, public events, etc. Finally, we contributed to the open science and free software initiatives. We made available in open access all the papers, data, source code, and instances used in the work of this thesis. We want to make all our research publicly available for reuse by any scientific or industrial community member.

1.3 Contributions

In the following, we will present the main contributions of this PhD thesis. They are not the only contributions, but in our opinion, they are those that we consider having the most significant impact from an academic, social and "green" point of view:

- Propose new algorithms for traffic light optimization to make traffic more efficient and environmentally friendly.
- Analysis of the optimal type of vehicles used to goods delivery in order to improve the whole traffic in the city and reduce carbon footprint. We also perform traffic simulations with on-field measurements.
- Use of demographic and geographic data for the location of ecological urban furniture according to the needs of citizens (bicycle stations and charging stations for electric vehicles).
- Use of multiobjective strategies for the application of robustness in the vehicle routing problem to reduce travel times and gas emissions.
- Study of exact and metaheuristic strategies for the multiobjective shortest path problem.
- Environmental concern (reduction of pollutant gases or use of zero-emission vehicles) in all the work carried out. We are aligned with the Sustainable Development Goals of the 2030 Agenda.

1.4 Organization

This thesis is structured in three parts and three different appendices. The first part presents the context and theoretical framework of the thesis: from the Sustainable Development Goals, the current environmental problem, the city as a scenario for new ecological initiatives, to Smart Mobility as a topic to improve the environment and citizens' lives. We also present the different Open Data sources and the use case scenario that worked within this thesis. And we give some notions about optimization and the algorithms used in this thesis.

The second part presents five problems in which we developed realistic modeling based on open data and advanced algorithms in order not only to give "good" solutions but also to offer solutions adapted to the specific characteristics of each city. These problems are the optimization of traffic lights schedule to make the traffic more fluid, the adaptation of the types of cargo transportation to reduce pollution and traffic, the optimal location of public bicycle stations for citizen neighborhoods and charging stations for electric vehicles, and the robust and ecological routing of vehicles through the city. The last part presents the general conclusions of the thesis, the contributions made, the limitations found, and how it is hoped to overcome them in future research work.

Finally, this thesis has three appendices in which the list of works that support the thesis, several details for the reproducibility of the experiments, and the descriptions of the main programming languages and libraries used are presented. In the following, we detail the different chapters that make up each of the parts of this PhD thesis.

- Part I Background
  - Chapter 2 presents the context of the thesis. We go from de Sustainable Development Goals to ecological mobility, going through the concept of Smart City (SC). Among the potential problems within the framework of SC, we highlight those focused on mobility. The problems of Smart Mobility are those addressed in this thesis. Also, this chapter exposes one of the most outstanding aspects of the thesis: the use of Open Data as the base source of data that allows the applications for SCs. We also describe the main real scenario used in this PhD thesis, the city of Malaga, Spain.
  - Chapter 3 provides an overview of the optimization algorithms, some general concepts and metrics used to compare them. We focus mainly on metaheuristic algorithms because they are the principal type of solvers used in the thesis.
- Part II Problems
  - Chapter 4 presents a hybridization of evolutionary algorithms and a racing strategy to address the problem of optimal traffic light scheduling. This optimization aims to improve traffic flow and reduce gas emissions.
  - Chapter 5 shows us how to obtain more environmentally friendly traffic by modifying the type of vehicle mainly dedicated to the transport of goods. We use multi-target algorithms, a popular traffic simulator, and open data from multiple sources to provide a solution that adapts to the reality of the city.
  - Chapter 6 describes the problem of where to place the stations of a bike-sharing system. We present different models based on demographics, user demand data, and geographic distances. We show how a more adapted to the citizen's system could have been implemented and also reveal how we can use the proposed models and algorithms to optimize the current system of the city of Malaga.
  - Chapter 7 presents the problem of locating electric vehicle charging stations. We used a multiobjective viewpoint, trying to bring the stations as close as possible to the different urban centers while reducing the associated installation cost.
  - Chapter 8 describes how we can combine robustness and multi-target optimization techniques to obtain the shortest and most ecological route to move around the city. In this chapter, we present our robust model and also study its applicability with two different pollutants: CO_2 and NO_x.

- Part III Conclusions and Future Work
  - Chapter 9 summarizes the main conclusions of the thesis. It highlights the main contributions and facts made to the various issues presented in Part II of this document.
  - Chapter 10 describes the main lines of research derived from this PhD thesis. We highlight the main points of interest for future work in the different topics addressed in the thesis.
- Appendices
  - Appendix A presents the publications made by the PhD candidate during the thesis.
  - Appendix B shows different aspects of the experiment as an aid to reproducibility.
  - Appendix C briefly describes the main features of the programming languages used in the thesis, as well as some libraries of special interest in the work carried out.

Part I

Background

Chapter 2

State-of-the-art

This chapter will set out the context of this doctoral thesis. We will use a top-down approach to go from the Sustainable Development Goals to a review of related work on each of the problems addressed in the thesis.

2.1 Sustainable Development Goals

The United Nations General Assembly in 2015 defined seventeen goals to be achieved by all countries to reach a more sustainable, equitable, and fair future for the entire planet (United Nations, 2015). These targets are the so-called Sustainable Development Goals (SDG). Figure 2.1 shows the 17 goals.

FIGURE 2.1: Sustainable Development Goals diagram.

One of the most remarkable aspects of these goals is that six are directly related to the environment (goals 6, 7, 11, 13, 14, and 15). The need for all people to fight climate change and improve life on the planet is supported by these goals. Specifically, the goal *"11 Sustainable cities and communities"* shows that it is not necessary to go out into nature to carry out sustainable actions. Cities themselves are epicenters for sustainable and environmental initiatives. It is vital to be part of these initiatives and has been the primary context for this doctoral work.

2.1.1 Environmental Concerns

The most commonly used mean of transportation in modern cities is the private vehicle, which is causing major environmental problems (Dirección General de Tráfico (DGT), 2014; Mahmod et al., 2013). The noise and polluting gases associated with road transportation are having a direct impact on people's health (Lebrusán and Toutouh, 2020b; Soni and Soni, 2016). Poor air quality contributes to respiratory and cardiovascular diseases as well as to lung cancer (Loomis et al., 2013).

According to the European Commission, air pollution is the principal health hazard for European citizens (European Environment Agency, 2018). Besides, according to the European Environment Agency, exposure to high noise levels generates a health risk, causing some 12,100 premature deaths per year on the continent (European Environment Agency, 2020).

Local authorities are responsible for taking care of this matter (Haines et al., 2009) despite the importance of negative or zero cost options when formulating their climate policy (Kousky and Schneider, 2003). Some cities implement fixed speed policies where all the vehicles must observe a reduced maximum speed in the city's streets (Bel et al., 2015), while others have more green buildings, focus on pedestrian and bicycle infrastructure, or have implemented more programs to divert waste from methane-generating landfills (Millard-Ball, 2012). Some cities have implemented policies to limit the circulation of certain types of vehicles. In São Paulo, Brazil, the managers of the city have limited the use of heavy vehicles, obtaining significant reductions in pollution levels (Pérez-Martínez, Andrade, and Miranda, 2017).

It also has an economic impact, shortening lives, increasing medical costs, and has an impact on the climate, since some air pollutants act as greenhouse gases (Guerreiro, Leeuw, and Foltescu, 2013).

2.2 Smart Cities

Since the great migration from the countryside to the city, the urban centers have been the scene of the growth of our society. Not only because they allow a more significant number of jobs, but also because of the many benefits they offer: health, safety, leisure, etc. Multiple studies claim that in just 30 years, more than 66% of the world's population will be living in cities (Economic and Affairs, 2019).

Classically, the different services offered were conceived and managed by people, generally belonging to the various public bodies of the city. However, the tremendous technological advances in recent times have enabled communication and automation of different tasks, which have never been seen before. These technical advantages include advances in wireless telecommunications, low-cost sensors, actuators, and automatic and massive data collection from the city, to name a few.

These technological advances have led to a change from a traditional society to one in which life is inconceivable without technology. Of course, this change has not only taken place on a small scale, at the individual level. It has also taken place at the level of society and, of course, at the city level.

In this context, cities are modernizing and becoming so-called Smart City (SC). Improving all city processes from a holistic perspective is the key aspect that characterizes SC. All this improvement is thanks to technology, advances in communications, and artificial intelligence. Doing a thesis in computer science, framed within smart cities, is justified given its current and future impact on society.

Given the very applied nature of the research conducted, it is relevant to highlight the different stakeholders and beneficiaries of the proposed advances. Cities are large ecosystems in which various stakeholders interact. We can classify the stakeholders into citizens, companies, and city managers.

Citizens are the backbone of cities. Every change in the city makes the day-to-day life of citizens better or worse. Their importance is often overlooked. This research focuses on the citizens in the same modeling phase. This approach allows us to obtain mobility improvements that have a direct (and positive) impact on their lives.

Companies are essential when it comes to implementing initiatives and improvements in cities. Improving the lives of citizens is not only achieved by improving processes, but any progress must go hand in hand with an increase in the economy of the area. There are many types of public and private companies. Those that have relationships with public entities are the most interesting to us. Such companies have a significant impact on citizens, and by having such a close link with municipalities, they tend to share more and better data with the public. Such open data makes it possible to obtain, for example, information from electrical substations, giving this realistic data a great added value to the research carried out.

Municipalities cannot be left out if we talk about city improvements. City managers are the real experts of the city. Their knowledge is of great benefit to research. In turn, researchers can provide municipalities with tailored and objective tools that offer new information to help them in their work, a feedback process. Municipalities provide open data that can be used by applications that improve city processes. In turn, these IT systems generate new ones to offer to the municipalities, which they then make available to the public. This infinite cycle of improvement makes cities grow (from an improvement point of view), boosts their economy, and improves the quality of life of their citizens.

TABLE 2.1: Smart City parts.

Part	Description
Smart Governance	Municipalities are an essential part of the SCs. In addition to using technology for citizen management, such management must be transparent. Government data should be in the public domain, and it is up to the municipalities to provide the necessary mechanisms for its consultation (Carrasco and Sobrepere, 2015)
Smart Mobility	Traffic problems are one of the most worrying factors for modern cities. This thesis focuses on this kind of problem. Section 2.3 will describe it in more detail.
Smart Environment	Groups together the so-called "green" or ecological solutions. We have already highlighted the importance of environmental awareness for the planet and, of course, for cities. Together with Smart Mobility, it is one of the main focuses of this thesis.
Smart Living	Deals with people's day-to-day issues such as health, safety, culture, etc. Aspects such as social inclusion of all population levels and access to all social, health, and living resources are the basis of this section.
Smart Economy	All the economy-related aspects in a city. Solutions in entrepreneurship and innovation, productivity improvement, market, etc. Factors such as circular economy or new trends in payment methods and cryptocurrencies are also included in this part.
Smart People	It is the most abstract of all, trying to improve society by promoting education and creativity. It focuses more on the group than on the individual.

Even with this vision, research often focuses on specific problems rather than the more significant problems of the cities. For this reason, a classification of the different types of issues that can occur in SCs. Given this new concept's youthfulness, there is no official definition of these areas; even among researchers, there are problems in which the limit is not very well defined. One of the most common views used is the one proposed by Cohen (2012). This researcher suggested six aspects that made up the SCs: Smart Governance, Smart Mobility, Smart Environment, Smart Living, Smart Economy, Smart People. Table 2.1 describes each of them in more detail. This thesis will focus on intelligent mobility problems, given their intimate relationship with different environmental issues and their impact on citizens' daily lives.

2.3 Importance of Smart Mobility

Mobility has a significant impact on life in cities. Losing hours and hours in traffic jams can generate many health problems for citizens. In addition, the number of combustion cars has led cities such as Madrid and London to close entire areas to traffic. These measures, although necessary, generate significant problems of delays, logistics, etc.

All this makes it necessary to optimize the different aspects of urban mobility. Making mobility more sustainable and suitable for citizens is a pillar of this research. Smart Mobility is an area with a multitude of problems. Among the different topics, we can highlight the following four since they range from the closest to the citizen to the city managers and the policies to be applied:

- Optimizing vehicle trips according to different criteria.
- Analyze the impact of different types of vehicles on traffic and the environment.
- Installation of urban infrastructure for the eco-friendly transition (bicycles and electric vehicle charging points).
- Improving traffic flow by optimizing the traffic light network.

Below we will explain each of these problems in more detail.

Optimizing vehicle trips For years, vehicle routing has been formulated as the problem of finding the shortest path in a graph (the edges are the roads and the nodes are the intersections). Some previous works on vehicle shortest paths have solved the single-objective shortest path problem to optimize aspects such as distance (Cao et al., 2015), costs (Jamalluddin et al., 2014), congestion of the traffic (Chen and Nie, 2013), etc. Many variables are interesting when improving road trips. Some of them are the congestion on the roads (Jiang et al., 2013) and the emissions of polluting gases (Abdul-Hak et al., 2013). The latter is especially relevant nowadays. Among all types of pollutant gases, CO_2 is a good starting point for reducing pollution and improving air quality in cities. However, existing algorithms do not usually take into account traffic or vehicle aspects (Tatomir, Rothkrantz, and Suson, 2009), and they only shorten the travel time to get a reduction in carbon emissions. Instead of improving just one metric, minimizing travel time and gas emissions simultaneously are essential. Thus, we need to solve a Biobjective Shortest Path (BSP), which was defined by Hansen (1980).

While the single-objective version of the shortest path problem can be solved in $O(|E| + |N| \log |N|)$ time, where N is the set of nodes and E is the set of edges, time using Dijkstra's algorithm with Fibonacci heaps (Thomas H. Cormen and Stein, 2009), the BSP is NP-hard (Serafini, 1987). In BSP, the goal is to obtain a set of efficient solutions, that is, solutions that can be improved in one of the objectives only if they are worsened in the other. Most of the proposed solutions for BSP are maintained in the academic field (Erb, Kobitzsch, and Sanders, 2014; Ghoseiri and Nadjari, 2010), and usually, they are not implemented in real-world applications. Commercial systems offer several routes according to different objectives but only tend to focus on one at a time or by fixing some of them.

Another essential aspect of vehicle routing is the robustness of the solutions. A route must be as stable as possible so that the user can plan the trip as accurately as possible. Regarding the shortest path problem, there are several approaches to consider the robustness. Pascoal and Resende (2014) use a minmax regret strategy to find a solution that minimizes the cost in the worst-case scenario. Other authors as Cheng, Lisser, and Letournel (2013) add a random delay. They optimize the travel times in the optimization problem. This model of robustness allows adding some imprecision in the weight of the edge. The authors also describe how to use the knowledge of the probability distribution of this random variable. The authors solved this problem by applying a strategy similar to the minmax regret, using the maximum probability distribution of the delay. The main limitation of using minmax is that it only returns a single robust solution. Hasuike (2013) modeled robustness using confidence intervals. Although they tackled the single-objective problem, they transformed it into a multiobjective problem by moving from exact values to confidence intervals. Even though different variants of the BSP problem have been studied (Chen and Nie, 2013; Mickael Randour, Jean-François Raskin, 2015; Zhang et al., 2015), the RBSP problem has been less studied in the scientific literature.

Cargo Delivery Vehicles Proportions There is a noticeable increment in the number of trips that citizens have to take nowadays and their duration (TNS Opinion & Social, 2013), significantly because urban infrastructures are not scaling correctly for the massive amount of vehicles in their streets. Even though these journeys are usually to commute or take children to school, which often happens at the same time of day, several traffic jams are the consequence of private vehicles sharing streets with services for distribution of goods, deliveries, etc. Furthermore, the need for cargo space makes those services use small trucks to perform their commercial activities (Visser, Nemoto, and Browne, 2014), which represents an increment not only in the street space usage but also in the pollutant emissions.

Gan et al. (2018) carried out a pollution analysis where they modeled the gas emissions of delivery trucks in urban logistics. The authors analyze the effect on greenhouse gas emissions of different trips and conclude that traveled distance and vehicles' weight have a capital impact on the pollution levels emitted. But, the distance between origin and destination does not have as much effect on parcel distribution. Also, replacing trucks with other less polluting vehicles may be a solution for reducing greenhouse gas emissions.

Mahmod et al. (2013) proposed banning Heavy Duty Vehicles by 1.5 Light Duty Vehicles, among other strategies to keep the same cargo capacity while reducing gas emissions. Also, the total demand increased by 1.035 as more vehicles were needed. The authors reported a total reduction of 25.8% for CO_2, 50% for NO_x and 30.9% for PM_{10} emissions.

Cheng, Madanat, and Horvath (2016) proposed a transit system with hierarchical structure. The authors consider the interactions between the trunk and the feeder systems to provide quantitative basis for designing and operating integrated urban transit system to reduce emissions. They take into account public transport, such as subway and bus, to analyze the entire city evaluating cost-effectiveness aspects.

Allocation of Urban Infrastructure Mobility depends on the layout of the city. Not only the topology of the road network is essential. Other factors influence traffic performance in the city. Especially in green mobility, the need for particular infrastructure is particularly noticeable. For example, if we use a public bicycle or scooter system, we must have areas for their storage/collection. In bicycle-sharing systems, there are multiple problems to address, such as: predicting the filling of stations (Singhvi et al., 2015), the location-allocation of bicycles (Chen, Liu, and Liu, 2018; Chira et al., 2014) in the stations, routes for users or the transfer of bicycles (Hu and Liu, 2014), etc. These are exciting problems based on the presence of an already installed system (including the associated infrastructure). We think it is essential to apply intelligent techniques from the very moment of planning to the final location of the different elements of the system. There are also complete solutions that take into account multiple aspects of this kind of systems (Lin, Yang, and Chang, 2013). However, they are very complex solutions that require large amounts of information that are not always available.

If we focus on the optimal location of the stations, we find approaches such as those proposed by Chen et al. (2015). The authors use real data and machine learning techniques to find the best places to locate the stations in this paper. Liu et al. (2015) employed a similar approach to the above in New York City, which has a large network of bicycle stations. Furthermore, Park and Sohn (2017) presents a comparison between two models for the station location problem. The authors compare the p-median and the MCLP. However, they do not give details about the algorithm that is used. Several metaheuristic algorithms, such as Genetic Algorithm, Tabu Search, Iterated Local Search, Simulated Annealing, or Variable Neighborhood Search; have been successfully used for the p-median problem.

Chen and Sun (2015) try to find the best locations to ensure the availability to collect/deposit the bicycles, taking into account the demand (at peak times) and the possible routes between stations made by users. Kloimüllner and Raidl (2017) propose a discretization of the locations in plots. They use benchmark instances to test their solution. These works involve a demand for the use of the bicycle-sharing system. However, they do not consider potential users (citizens) who could use the system if it were closer to them.

On the other hand, new trends in mobility, such as the electric car, also require specific facilities. The optimal location of electric vehicle charging stations has been a relevant problem since the emergence of a renewed interest in electric transportation infrastructures in the early 21st century.

Frade et al. (2011) applied a maximal covering model to maximize the demand within a maximum desirable distance, assuming that coverage decays beyond that threshold distance. A MIP model was proposed, including a penalty term to prevent the installation of unnecessary supply points. The model was evaluated on four scenarios in Lisbon, Portugal, installing from 180 supply points in 29 charging stations up to 324 supply points in 43 charging stations in a higher-demand scenario. Accurate covered demand results were computed, providing an acceptable level of service.

Wagner, Götzinger, and Neumann (2013) proposed a business intelligence model for Electric Vehicle Charging Stations Location problem to maximize demand coverage, based on potential trip destinations of vehicle owners, defined using urban data analysis (Massobrio and Nesmachnow, 2020). An iterative method was proposed to find optimal locations using penalties to define ranks for points of interest and a Mixed-Integer Linear Programming model solved in CPLEX. The proposed model achieved promising results on two case studies from Amsterdam and Brussels. Chen and Nie (2013) proposed a Mixed-Integer Linear Programming formulation for locating charging stations minimizing the total walking distance according to parking patterns estimated using real urban data. The model was evaluated on a case study on 218 zones of Seattle, USA. Results achieved good accessibility: locating 20 charging station the walking distance was 1.1 km (average) and 3 km (maximum), whereas almost 80% of the demand was fulfilled.

Cavadas, Almeida Correia, and Gouveia (2015) proposed a MIP model for Electric Vehicle Charging Stations Location problem to maximize the satisfied demand subject to a maximum budget constraint, considering the activity patterns of travellers. A multi-period formulation was introduced to model time intervals within a day. The model was evaluated in a small real scenario in Coimbra, Portugal, with just nine stations and four charging points each installed on 129 candidate locations. Accurate solutions improved the realistic configuration of EV charging stations installed in the city. Brandstätter, Kahr, and Leitner (2017) proposed an ILP model for Electric Vehicle Charging Stations Location problem to maximize economic benefits in a car-sharing system, considering stochastic demands. The model was validated on medium-size synthetic scenarios and real-world instances from Vienna (up to 693 potential locations). For Vienna, the exact approach was only able to solve instances for eight central districts of the city, whereas a heuristic method was applied for larger problem instances. Solutions confirmed the economic viability of implementing an electric car-sharing system.

Çalık and Fortz (2017) proposed a Mixed-Integer Linear Programming formulation for Electric Vehicle Charging Stations Location problem to maximize the profit of a public one-way electric cars system. The model and two relaxations were studied for 63 instances in New York, USA, with 85 potential locations for installing non-identical charging stations. The impact of cost changes on the number of stations was studied. Bian et al. (2019) proposed an approach based on Geographic Information System (GIS) for Electric Vehicle Charging Stations Location to maximize the profit. GIS was applied to determine the probability for users to charge their EV in different areas, using relevant traffic information. The model was evaluated in a small case study in Västerås, Sweden, with 268 square zones. Two scenarios were studied, adding three and ten new charging stations to 40 already installed in the city. When adding three stations, the best option was selecting fast chargers in commercial areas, whereas slow chargers installed in residential areas were better when including ten stations.

Lin et al. (2018) proposed a Mixed-Integer Linear Programming model based on GIS to optimally select the location and the size Electric Vehicle Charging Stations in urban scenarios. The Mixed-Integer Linear Programming model maximizes the economic profits of installing new charging stations, computed according to the charging demand based on the traffic flow data, charging profiles, and city land-use classification. The authors generated an aggregated charging demand profile of the EVs based on the real-world travel data in the National Household Travel Survey and charging behaviors to compute the charging demand. These daily charging behaviors are represented by 24 hourly charging demands for each charging type of location. GIS is employed to calculate the charging demand in different locations by taking into account traffic flow and land-use classifications (e.g., residential with the region, residential with apartment, working, etc.). This study assumes that a charging station will only serve a specific area's demands. An acceptable walking distance from the charging station (parking lot) to the user's destination is defined. The researchers consider the costs of a new station (which could include fast and slow chargers). Thus, the station's costs consist of an aggregation of the economic costs of the equipment, installation, rent, maintenance and operation, and electricity consumption, which depend on the number and the type of installed chargers. The optimization problem objective (the economic profits of deploying the new stations) is computed by subtracting the costs of locating the new charging station from the revenues of charging EVs. The proposed approach was evaluated over an area of 67 Km^2 of Västerås, Sweden. Västerås had a population of 119,372 people, there were 44,192 personal cars, and the city had 324 plug-in EVs charging stations. The authors defined 532 tentative charging stations. The experimental analyses evaluated only the proposed method over four scenarios: installing three, five, ten, and 15 new charging stations. The results show that the proposed approach provided charging station locations with competitive profits.

In addition to those mentioned above, other types of furniture can be optimized in the context of Smart City, including roadside infrastructure for vehicular networks (Massobrio et al., 2017), bus stops (Fabbiani et al., 2018), and waste bins (Rossit, Toutouh, and Nesmachnow, 2020), among other relevant problems.

Traffic Lights Schedule In a metropolitan area, traffic jams, and congestion are among the most significant sources of greenhouse gas emissions and, therefore, contribute to global climate change. Urban traffic planning is a way to improve mobility efficiency and safety, thus producing a positive impact on the traffic flow. As in many real-world cities, the real-time control of traffic-lights (Cao et al., 2017) is not feasible because of various reasons (legal, technical, etc.), and we must instead find a highly-reliable global schedule of traffic lights that works well in the dynamic and uncertain traffic system (Bravo et al., 2016; Ferrer et al., 2016; García-Nieto, Alba, and Olivera, 2012; García-Nieto, Olivera, and Alba, 2013; Sánchez, Galán, and Rubio, 2008; Sánchez-Medina, Galán-Moreno, and Rubio-Royo, 2010; Stolfi and Alba, 2014; Stolfi and Alba, 2015a; Teklu, Sumalee, and Watling, 2007; Teo, Kow, and Chin, 2010; Péres et al., 2018a). When optimizing the light cycle programs of traffic signals within a city to improve traffic flow and reduce pollution, the fitness of a candidate traffic-light program is evaluated by simulating vehicle routes and velocities over a given traffic network (García-Nieto, Olivera, and Alba, 2013; García-Nieto, Alba, and Olivera, 2012; Sánchez-Medina, Galán-Moreno, and Rubio-Royo, 2010; Teo, Kow, and Chin, 2010; Li et al., 2016). The simulation of traffic flows in a specific city requires collecting network data (topology of the area and information about traffic lights), which is usually precise and static, and traffic data (the number of vehicles, their journeys, and velocities) is estimated from highly dynamic real-world data. Given the inherent uncertainty of this estimation, it is possible to generate distinct traffic scenarios that are all consistent with the real-world system (Stolfi and Alba, 2015a). One way to take into account this uncertainty is to compute an aggregated fitness value by simulating the same candidate solution multiple times by using different traffic scenarios (Ferrer et al., 2016). An alternative (or additional) way is to make each simulation itself stochastic by introducing random changes to the traffic scenario during simulation (Sánchez, Galán, and Rubio, 2008). The fitness of a reliable traffic-light program should not present a high variance when evaluated on the dynamic traffic of the actual city.

Despite the inherent uncertainty of simulated traffic scenarios, the literature on traffic-light optimization often relies on deterministic simulation of a single traffic scenario (Sánchez, Galán, and Rubio, 2008; Li et al., 2016; García-Nieto, Olivera, and Alba, 2013; García-Nieto, Alba, and Olivera, 2012; Péres et al., 2018a). When multiple scenarios are considered, they are used for evaluating the flexibility of the optimization algorithm by optimizing each scenario separately (Sánchez, Galán, and Rubio, 2008). Ferrer et al. (2016) validated the reliability of candidate solutions in multiple scenarios after the optimization phase. However, each solution is still optimized concerning a single scenario.

On the other hand, the scenarios used are usually quite small as the presented by García-Nieto, Olivera, and Alba (2013) (40 intersections, 184 traffic-lights, 500 vehicles) and other used to solved real-time control strategies (16–30 traffic-lights, 800 vehicles) (Cao et al., 2017).

2.4 Real-World Data

It is common sense that: it does not matter how sophisticated a model is, how many thousands of variables it uses, or it applies the newest resolution methods; if, when applying it, we use a synthetic scenario that little or nothing resembles reality. When we work on a solution for a Smart City, we could think that two cities of similar size and a geographical location can implement the same solution, which is a wrong assumption. Cities are like living organisms with their particularities. Although there are common features, models must consider this uniqueness when working with them. This section will present the main sources of real data on cities and describe the selected region to test our scientific contributions.

2.4.1 Open Data

The concept of Open Data (OD) (Kitchin, 2014) is a philosophy and practice that seeks to make certain types of data freely available to everyone, without restrictions of copyright, patents or other control mechanisms. Making data available without restrictions to anyone globally is the primary goal of OD. Data must be free of limitations on its exploitation, patents, access for a fee, copyright restrictions, etc., to get OD. There is a multitude of OD sources. The rise of technologies and paradigms such as the semantic web (Berners-Lee, Hendler, and Lassila, 2001), the increase of the use of sensors (Sinaeepourfard et al., 2016) and cloud computing (Hayes, 2008), among others, have favored the generation and use of OD.

OD is essential in the development of Smart City (Barns, 2018). OD allows the evaluation of different aspects of the city and the quality of life of its inhabitants (Lebrusán and Toutouh, 2020a; Lebrusán and Toutouh, 2020b; Lebrusán and Toutouh, 2021). Besides, it can be used to design applications that assist both citizens and managers to improve dwellers life (Estrada et al., 2018; Camero et al., 2018). Below we will highlight some of the entities that make this golden age of data sources possible.

Municipality Traditionally, governments have generated statistics about their cities and citizens. In the case of Spain, among the information generated are, for example, censuses, high-level economic data, data on unemployment levels, etc. These data were no more than summaries (averages, variances, etc.) that did not analyze the information in question intelligently. Lower-level governments, such as regional and municipal governments, have opted for OD to remedy this misinformation. Thanks to OD, citizens and city managers can benefit from new systems that make use of them. Many municipalities have made a solid commitment to this idea and offer a multitude of public OD on their web pages (Alcaide Muñoz, Rodríguez Bolívar, and Villamayor Arellano, 2019; Meijer and Potjer, 2018). For example the city of Malaga, Spain, has 2,338 datasets[1] (accessed on January 11th, 2022).

Companies Numerous companies generate OD that citizens can use. One of the alternatives to publishing such data is agreements with public entities. The data generated by the companies are transferred to the OD platforms of the municipalities, which make them available to the public. Some examples are energy consumption data provided by electricity companies (such

[1] Malaga OD website: https://datosabiertos.malaga.eu/

as Endesa), parking lots and vehicle-sharing stations (cars, electric motorcycles, electric scooters, and public bicycles), statistics on the car fleet (in Spain administered by the *Dirección General de Tráfico*), and so on.

Academic It has always been important to share the results and the data in research. In addition to research data, the scientific university encourages the generation of OD through challenges. For example, TSPLIB (Reinelt, 1991) is a dataset generated by one of these challenges. TSPLIB is composed of different sets of coordinate points distributed differently in space.

Reproducing the same research results is paramount to comparing our solutions. Reproducibility is sharing the data and the whole environment necessary to be able to replicate the research done (López-Ibáñez, Branke, and Paquete, 2021). In this thesis, we have tried to provide as much data as possible to reproduce the results obtained. Appendix B lists the different websites where the research data can be obtained.

2.4.2 Real Scenario - Malaga, Spain

For this thesis, it is imperative to achieve a real contribution that can improve the lives of citizens. So, it was necessary to work with an actual city in the research works described in this thesis. The selected geographical location was the region of Malaga in Spain. We will now describe the main characteristics of this Spanish region and the motivations that guided its choice. This territory represents a medium-size road map and a European region. Its geographical and road map was obtained from the Open Street Map website[2], and processed by our parser, which adapts the representation of a region to a graph as we need in our formulation.

Characteristics

We have decided to use this city as the target of our study for several reasons:

- It is a medium-sized European city.
- It has a wide variety of OD (see municipality OD website: `http://datosabiertos.malaga.eu/`), i.e., population census, filling status of bicycle stations, cartographic maps, etc.
- The City Hall has a lot of Smart City initiatives. In addition, during this thesis, there have been collaboration agreements between the research group and the City Council to develop several Smart City initiatives.

For this thesis, we can highlight the following data on Malaga:

- Extension of around 400 km^2 (city) and 7,300 km^2 (province),
- 577,405 citizens (in 2021),

[2]Open Street Map website of Malaga city: `https://www.openstreetmap.org/#map=13/36.7248/-4.4253`

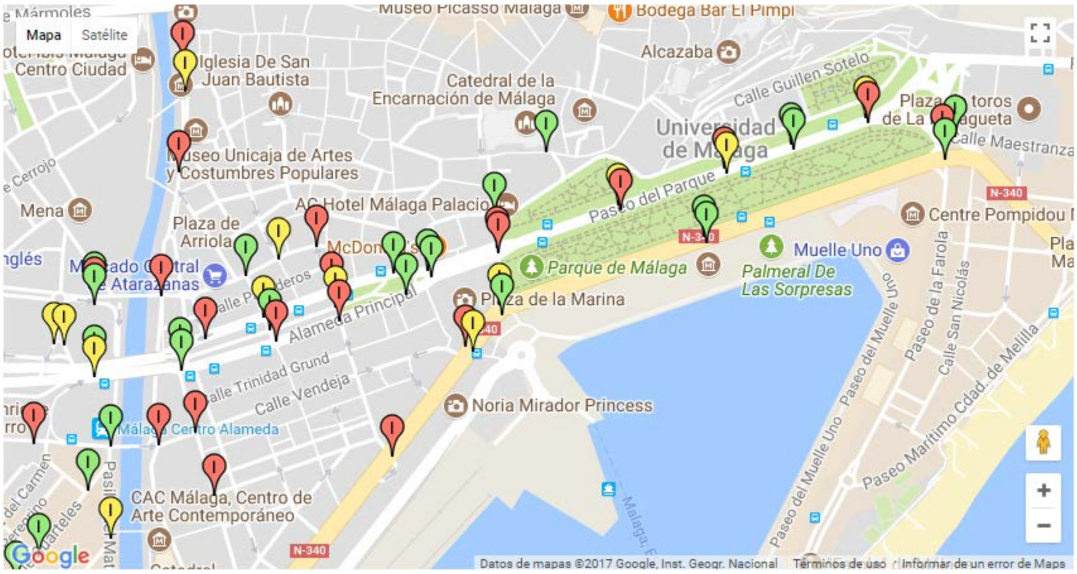


FIGURE 2.2: Locations of traffic lights in the downtown case of study. The colors show large (red), medium (yellow), and small (green) differences between the two different solutions.

- 363 neighborhoods,
- more than 33,550 road segments,
- 23 public bicycles stations, and
- 14 electrical substations.

Maps

We have used four different scenarios based on the Malaga region. Each one has different sizes. The scenarios are downtown, small, medium, and large. Following, we describe in more detail each one.

Downtown scenario corresponds with the main downtown street. This map was used previously in Stolfi and Alba (2015a), which encompasses an area of about 3 km^2 with 58 intersections controlled by 275 traffic lights (Figure 2.2). This little scenario allow us to test really well our approach as the new traffic lights plans.

Small scenario corresponds to the eastern area of the city of Malaga. We used this map for traffic measurements since it is a more confined space, allowing us to take measurements in the field. Figure 2.3 shows this area. In addition, we can see the locations of different sensors for measuring the number of vehicles (red dots) and where we made the manual measure of the traffic. This region includes zones where traffic jams are common. The geographical area studied encompasses an area of about 32 km^2.

FIGURE 2.3: Eastern area of Malaga. The markers are traffic measure points (red numbers) and the traffic sampling points (blue letters) to get the vehicles flows in the area.

Medium scenario corresponds to the whole city. Malaga is a coastal city with more than 569,000 inhabitants in an extension of about 400 km^2. This scenario allows us to analyze a realistic medium-sized city. Among other aspects, at this level, we have a multitude of OD available from the city council. Among the datasets we have used at this level are the city census, the location of the different neighborhoods and buildings, and the location of the urban furniture, specifically, the public bicycle stations. Figures 2.4 and 2.5 show two examples of geographical OD over the city's street map. We report the location of its neighborhood centers, and the location of public bicycle stations and the electric districts with their electrical substations.

Large scenario corresponds to the entire province of Malaga, Spain. This region of more than 7,300 km^2 includes many cities and towns. It is widespread for the citizens to live in one city and work in another. This is because of the spatial limitation of the number of homes and the excellent communication between the coastal cities. This situation makes the car the main means of transport for commuting to work. Under this assumption, we use the road map (shown in Figure 2.6) to find the fastest and most ecological way of commuting using citizens' vehicles. This road map has 10,601 intersections and 118,388 road segments (part of the road between two junctions).

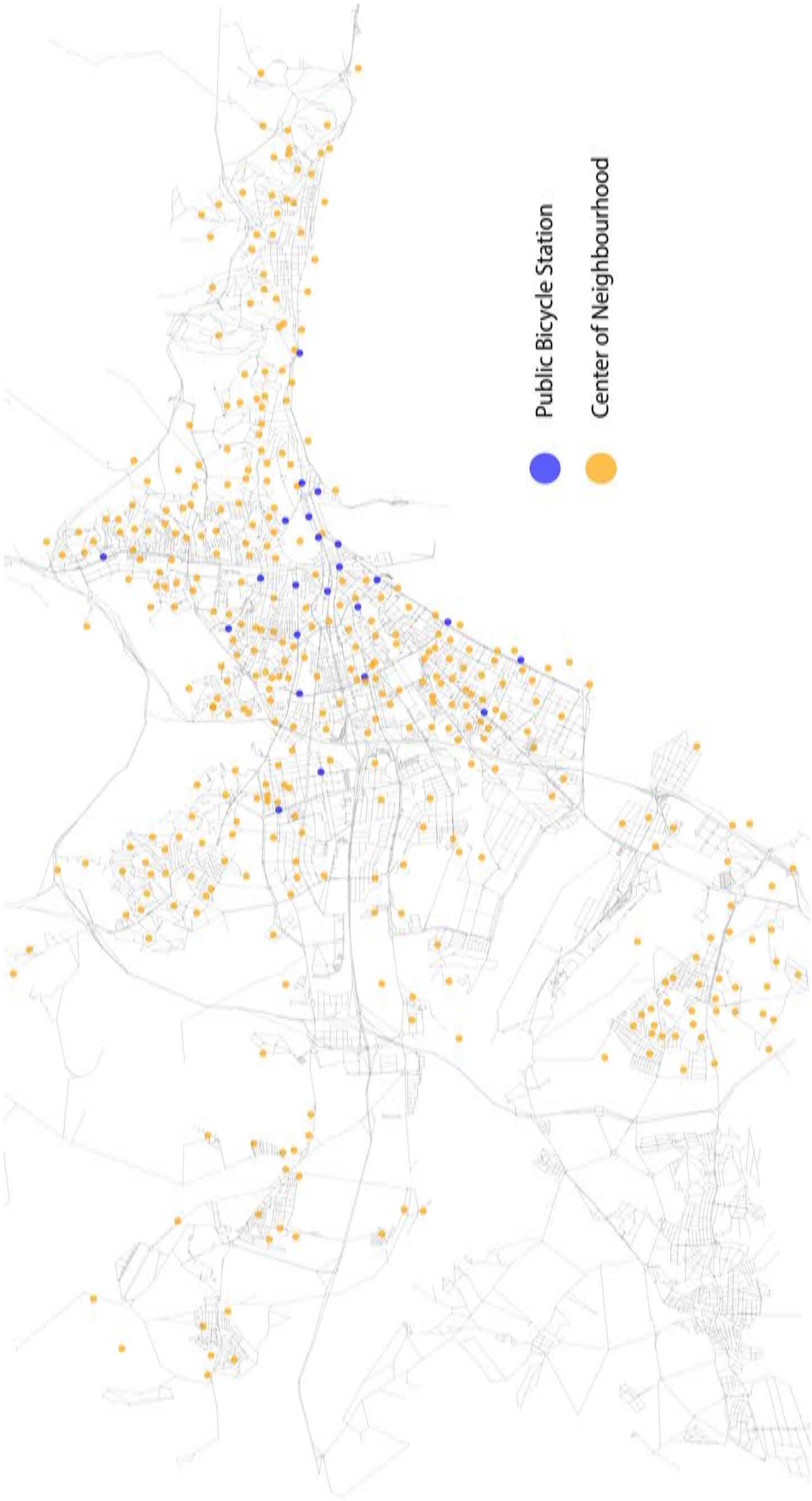


FIGURE 2.4: The center of each neighborhood (orange points) and current bicycle stations (blue points) in Malaga city.

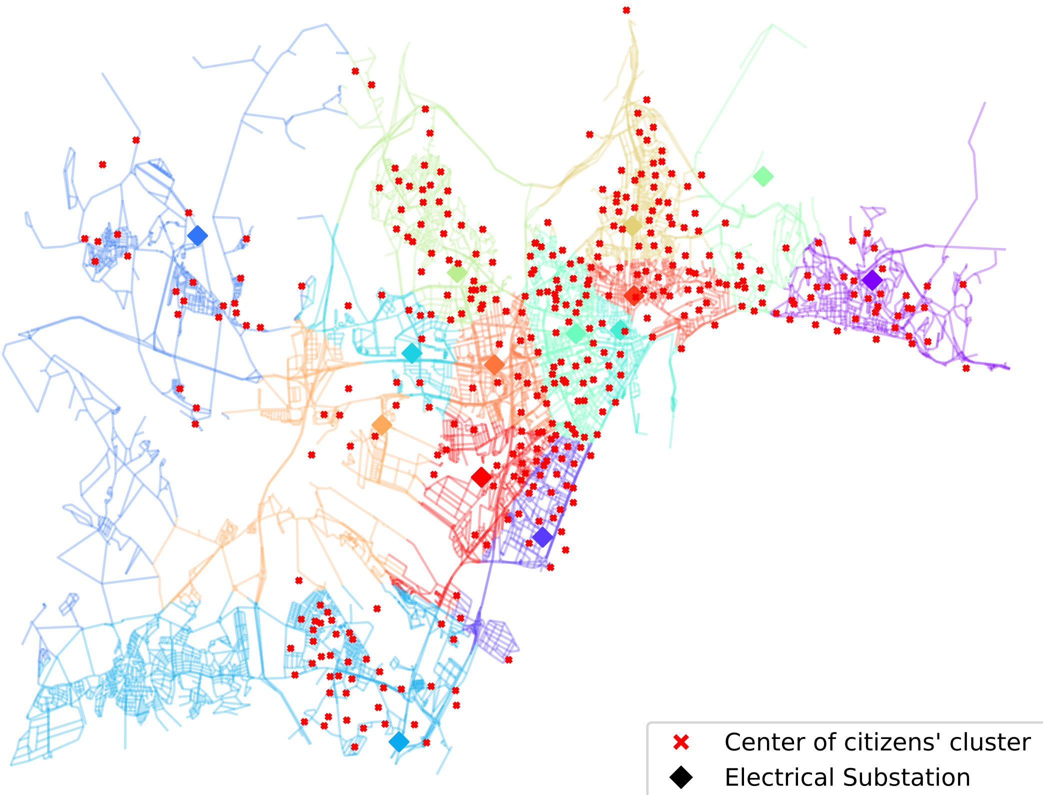


FIGURE 2.5: Citizen’ clusters, electrical substations, and road map of Malaga. Each edge represent a street segment and their color which substation it is associated with.

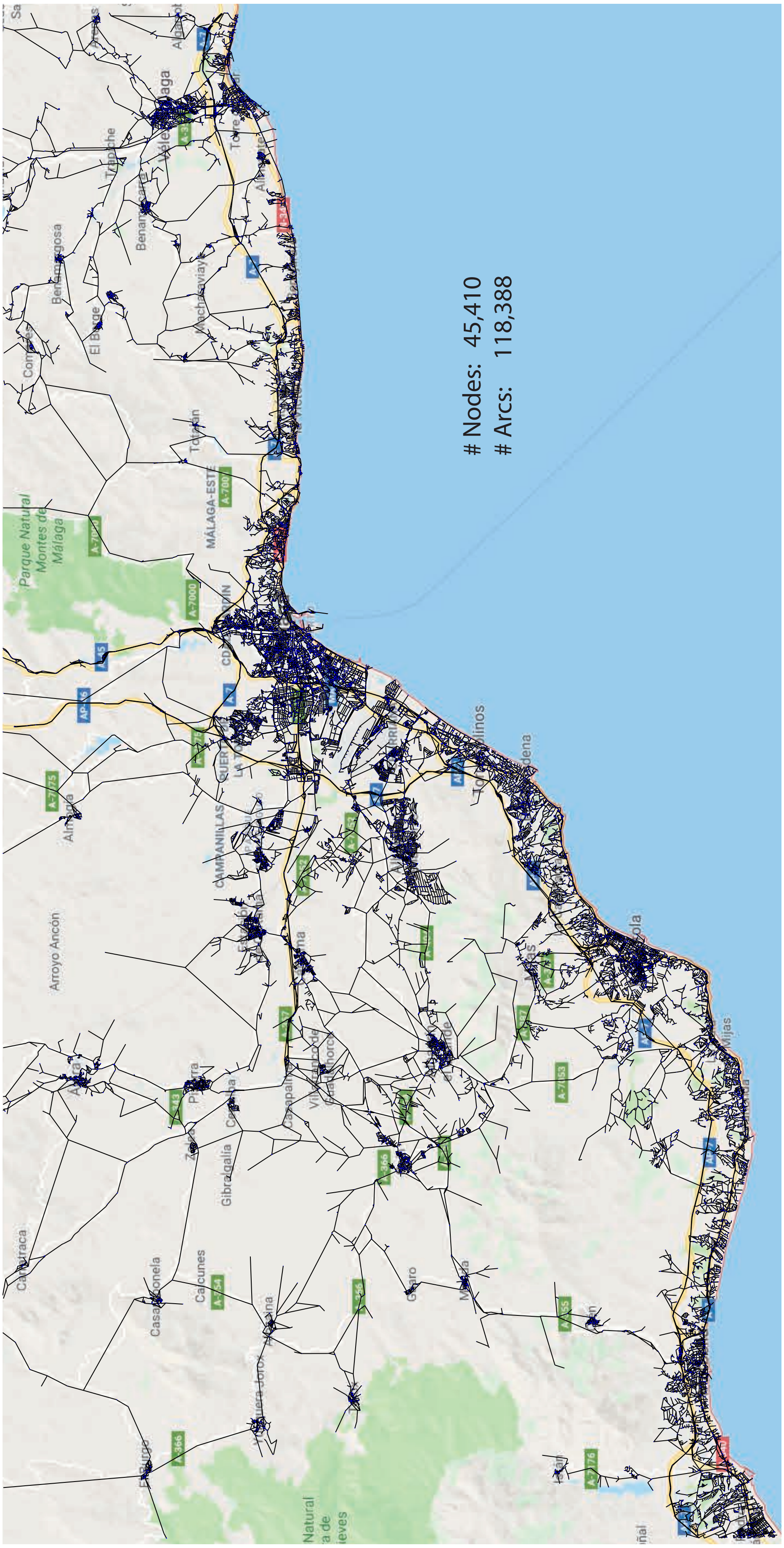


FIGURE 2.6: Graph of the road map over the map of the whole Malaga province.

Chapter 3

Optimization

3.1 Optimization Problems

An optimization problem consists of selecting the best option among all possible options according to some criterion. For example, if we are going to work in our car, the optimization problem could be choosing the route that generates the least gas emissions. Each of these routes would be a solution x to our problem. A solution is composed of a series of values (for example, the different streets through which it passes) called decision variables. Mathematically, a solution x can be seen as a vector $x = \{x_1, x_2, ..., x_n\}$, where $x_1, x_2, ..., x_n$ are the decision variables. And, the set of all possible solutions X is called the solution space or search space.

We have mentioned that we must choose the best option according to a criterion (for example, the least polluting). So, it is necessary to measure the quality of each solution. In optimization, we use an evaluation function $f : X \rightarrow \mathbb{R}$ that returns a numerical value for each solution x (Talbi, 2009). The evaluation function is also called the fitness function or objective function. The evaluation function f allows us to create a total order relation between the solutions. The result of an evaluation function is often referred to as the fitness value. There may be different functions to evaluate a solution depending on the problem. For example, we can evaluate routes according to their contamination, travel time, distance, etc. We can have two cases: (i) preferring solutions with higher fitness values than the others, or (ii) preferring lower values. In the first case, we will say that we have a *maximization* problem, and in the second case, a *minimization* problem. Traditionally, the evaluation function was only called fitness function and its resulting value fitness value in maximization problems. Nowadays, the scientific community has adopted fitness function as a *de facto* synonym of the objective function, and in this thesis, we will take both as synonyms.

At this point, we already have the necessary basis to define optimization problem and global optimum.

Definition 1 (Optimization problem) *An optimization problem consists in finding a solution $x \in X$ that optimizes (minimizes or maximizes) an objective function $f : X \rightarrow \mathbb{R}^d$, where X is the search space.*

Definition 2 (Global optimum) *A solution $x^* \in X$ is a global optimum if it has a better objective function (we suppose without loss of generality a minimization problem) than all solution in the search space, that is $\forall x \in X, f(x^*) \leq f(x)$.*

So, the goal of an optimization problem is to try to find the global optimum (or to get as close as possible to it).

If the image of f is $\mathbb{R}$ ($d = 1$), we have a single-objective optimization problem. However, if $d > 1$ then we say that the problem is multiobjective. In the latter case, we will use boldface for the objective function $\mathbf{f}$ to highlight that it is a vector function.

In a multiobjective problem, each objective is independent, equally important, and opposed to the other ones. In this kind of problems, we can have several optimal solutions. *Pareto dominance* is a useful concept to define this set of optimal solutions. Formally, we define it as follows.

Definition 3 (Pareto dominance) *Given a vector function $\mathbf{f} : X \rightarrow \mathbb{R}^d$, we say that solution $x \in X$* dominates *solution $y \in X$, denoted with $x \prec_{\mathbf{f}} y$, if and only if $f_i(x) \leq f_i(y)$ for all $1 \leq i \leq d$ and there exists $j \in \{1, 2, \ldots, d\}$ such that $f_j(x) < f_j(y)$. When the vector function is clear from the context, we will use $\prec$ instead of $\prec_{\mathbf{f}}$.*

Pareto dominance defines a partial order relationship in the solution space.

Definition 4 (Pareto Optimal Set and Pareto Front) *Given a vector function $\mathbf{f} : X \rightarrow \mathbb{R}^d$, the* Pareto Optimal Set *is the set of solutions $Y \subseteq X$ that are not dominated by any other in X:*

$$Y = \{x \in X |\ \nexists y \in X,\ y \prec x\} \tag{3.1}$$

The Pareto Front *is the image by $\mathbf{f}$ of the Pareto Optimal Set: $PF = \mathbf{f}(Y)$.*

Definition 5 (Set of Non-dominated Solutions) *We say that a set $Y \subseteq X$ is a* set of non-dominated solutions, *given a vector function $\mathbf{f} : X \rightarrow \mathbb{R}^d$, if there is not a pair of solutions in the set in which one solution dominates another. There is no pair of solutions $x, y \in Y$ where $y \prec x$.*

If a solution x is dominated, it means that there is at least one solution y that it is better than x in at least one objective. For that reason, algorithms should try to find only non-dominated solutions.

3.2 Evaluation Metrics

In our experiments, we will not always obtain global optima. Instead, we will have solutions whose quality we will have to compare. In addition to the basic statistical tools (means, medians, frequencies, etc.), some tests and metrics allow us to perform a deeper comparative study. In the following, we will highlight two types of metrics used in this thesis: statistical tests and quality metrics for multiobjective problems.

3.2.1 Statistical Tests

Given the stochastic nature of the algorithms used in this thesis, we have performed different hypothesis tests to validate our conclusions about the data. We assume a confidence level of 95% (p-value < 0.05) for all the tests in this thesis. The following is a brief description of the most relevant ones:

Wilcoxon rank-sum test (Sheskin, 2011) is a non-parametric test that compare if two group of samples are the same. It is also known as Mann-Whitney test. With more than two groups of data we need to use a correction, usually Holms's or Bonferroni.

Kruskal-Wallis test is a non-parametric test. It is used to know if there is any significant difference between the average of two groups of samples. Kruskal-Wallis is a binary test, so performing a p-value correction for more than two series is necessary to check if the observed differences are statistically significant. There are multiple corrections as Holms's or Bonferroni

$\widehat{A}_{12}$ non-parametric effect size measure statistic was proposed by Vargha and Delaney (2000). It is a single factor ANOVA post hoc test for pairwise comparisons. Given two random samples of a performance measure M obtained by running two stochastic algorithms 1 and 2, $\widehat{A}_{12}$ measures the probability that a run of algorithm 1 yields higher M values than a run of algorithm 2. If the two algorithms are equivalent, then $\widehat{A}_{12} = 0.5$. If $\widehat{A}_{12} = 0.3$ then one would obtain higher values for M with algorithm A, 30% of the time.

3.2.2 Multiobjective Quality Metrics

In multiobjective optimization, the non-dominated sets must be compared to find which set of solutions is the best one. There are many multiobjective optimization metrics proposed over the years to assess the quality of Pareto sets (Riquelme, Von Lücken, and Baran, 2015). These metrics usually work by comparing Pareto fronts. Some of the most popular ones are:

- Hypervolume (HV) and Relative Hypervolume (RHV)
- Generational Distance (GD) and Generational Distance Plus (GD$^+$)
- Inverse Generational Distance (IGD) and Inverse Generational Distance Plus (IGD$^+$)
- ϵ-indicator

Each of them is explained in more detail below.

HV is the most popular quality indicators to evaluate multiobjective algorithms. The HV value of Sol is the volume of the area that is dominated by objective vectors in Sol and bounded by the reference point q as it is shown in Equation (3.2), where the function *volume* is the Lebesgue measure (Zitzler and Thiele, 1998b). A large HV value indicates that Sol approximates the Pareto front well in terms of both convergence and diversity. The RHV metric is computed as the relative value of HV to the maximal hypervolume of the pareto front.

$$HV(Sol) = volume\left(\bigcup_{s\in Sol}[s_1,q_1]\times ... \times [s_n,q_n]\right) \tag{3.2}$$

GD measures the average distance from the solutions computed by the multiobjective algorithm to their closest solution in the pareto front (Van Veldhuizen, 1999). Let us assume the points found by the multiobjective algorithm are the objective vector set $Sol = \{s_1, s_2, ..., s_{|Sol|}\}$ and the reference points set (*reference front*) is $P = \{p_1, p_2, ..., p_{|P|}\}$. The reference front is obtained from all the solutions obtained by all the algorithms in all the executions. Then, GD is computed according to Equation (3.3), where d_i represents the distance from s_i to its nearest reference point in P. In turn, GD$^+$ (Ishibuchi et al., 2015) is evaluated according to Equation (3.4), where for minimization problems the modified distance between s_i and the nearest point $p_i \in P$ is computed as $d_i^+ = \max\{s_i - p_i, 0\}$.

$$GD(Sol) = \frac{1}{|Sol|}\cdot\left(\sum_{i=1}^{|Sol|} d_i^n\right)^{\frac{1}{n}} \tag{3.3}$$

$$GD^+(Sol) = \frac{1}{|Sol|}\cdot\left(\sum_{i=1}^{|Sol|} {d_i^+}^2\right)^{\frac{1}{2}} \tag{3.4}$$

IGD performance indicator inverts the GD and measures the distance from any point $p \in P$ to the closest point in Sol (Coello and Sierra, 2004). Equation (3.5) presents the IGD computation, where $\widehat{d_i}$ represents the distance from p_i to the closest reference solution in Sol. Besides, IGD$^+$ (Ishibuchi et al., 2015) performance metric is weakly Pareto compliant wheres the original IGD is not Pareto compliant. The IGD$^+$ metric es computed as it is shown in Equation (3.6), where for minimization problems the distance between p_i and the nearest reference point in Sol is computed as $d_i^+ = \max\{s_i - p_i, 0\}$.

$$IGD(Sol) = \frac{1}{|P|}\cdot\left(\sum_{i=1}^{|P|} \widehat{d_i}^n\right)^{\frac{1}{n}} \tag{3.5}$$

$$IGD^+(Sol) = \frac{1}{|P|}\cdot\left(\sum_{i=1}^{|P|} \widehat{d_i}^{+2}\right)^{\frac{1}{2}} \tag{3.6}$$

ϵ-indicator (Knowles, Thiele, and Zitzler, 2006) calculates the smallest distance needed to move a solution to dominate the entire reference front, as shown in Equation (3.7).

$$\epsilon\text{-indicator}(Sol) = \inf\{\epsilon \in \mathbb{R} | \forall p \in P, \exists y \in Sol, \forall 1 \le i \le n : x_i < \epsilon y_i\} \tag{3.7}$$

3.3 Algorithm Classification

An optimization algorithm is an algorithm that generates solutions to an optimization problem. Each candidate solution generated by an algorithm can be evaluated using the function f to get the best one. There are many optimization algorithms: specific or general, constructive or iterative, inspired by nature or following a mathematical analysis, etc. In the following, we will present the different optimization algorithms used in this PhD thesis. We proposed a simple classification of them, as shown in Figure 3.1. We classify the algorithms based on their accuracy, the number of objectives they work with, and the strategy employed. We will now analyze each of them.

There is a multitude of algorithms for a variety of problems. In particular, those that, given their formulation, ensure the best possible solution for a given problem are called exact algorithms. This thesis uses this type of algorithm to solve a specific kind of problem: the shortest path problem. The importance of the shortest path lies in part because it is the core of different GPS based vehicle navigators, such as Google Maps (*Google Maps* 2022) or Waze (Galeso, 2016).

It is often impossible to have an exact algorithm to solve our problem, or its computational requirements are too high. Metaheuristics are a type of algorithm that offer promising results for these problems. Their underlying idea is to "guide" the search processes intelligently to obtain the best possible solution. Indeed, we would not find the optimal solution in all cases. However, metaheuristics generate sufficiently good solutions to improve many real-world processes.

When we have a solution (not necessarily the best one) to a problem, it is located somewhere in the solution space and therefore in the objective space. That solution will change its location in those spaces if we modify some of their decision variables. These movements improve or worsen the fitness value. Trajectory-based algorithms are based on making changes to one single solution so that it "moves" in the decision space. In this way, the trajectory-based algorithms iteratively improve the quality of their solution in order to find better solutions, in other words, the solution follows a descending (in a minimization problem) or ascending trajectory. Under this spatial distribution of the solutions in the search space, we can define the neighborhood concept. A neighborhood of a solution is a set of solutions close to each other, i.e., whose decision variables are similar. Also, local search operators can be created, allowing moving a solution within its neighborhood to optimize the solution.

On the other hand, instead of improving a single solution, we can have a whole set of solutions. Population-based algorithms use a set of solutions that are continuously optimized. There are many algorithms based on this idea, but the most popular are Evolutionary Algorithm (EA), which are inspired by the evolution of species. EA calls population to the set of solutions and individual to the solution. In them, individuals are mixed and modified to improve generation after generation.

The different algorithms used in this thesis will be described as follows.

- Optimization Algorithms
  - Exact
    - Single-objective
      - Dijkstra
      - A⋆
    - Approximate (Multiobjective)
      - Pulse
  - Metaheuristic
    - Single-objective
      - Trayectory Based
        - Iterated Local Search
        - Simulated Annealing
        - Variable Neighborhood Search
      - Population Based
        - Genetic Algorithm
        - Differential Evolution
        - Particle Swarm Optimization
    - Multiobjective
      - NSGA-II
      - MOEA/D
      - SPEA2

FIGURE 3.1: Classification of the main algorithms used in this thesis.

Dijkstra shortest path algorithm is a well-known exact algorithm to find the shortest path between two nodes in a weighted graph proposed by Dijkstra (1959). All the weights in the graph have to be positive. Its pseudocode is provided in Algorithm 1. The underlying idea of the algorithm is to go through, starting from the origin node, the different adjacent nodes. Each node is assigned the minimum sum of the weights necessary to reach that node. If the node already has a value, the smaller of the two is set. Then the adjacent node with the lowest weight is chosen, and the process is repeated until the destination node is reached. Despite being a classic algorithm, it is still used today as a baseline in the design of new optimization strategies for the shortest path problem (Pascoal and Resende, 2015; Sedeño-Noda and Raith, 2015). This algorithm is commonly used in different vehicle navigator applications, such as Google Maps.

Algorithm 1 Dijkstra shortest path algorithm

```
Input a graph G(N, A), start node s, and end node e
for all n ∈ N do
    dist[n] ← ∞
    prev[n] ← null
end for
dist[s] ← 0
Q ← N
u ← null
while Q ≠ ∅ and u ≠ e do
    u ← v ∈ Q with minimum dist[v]
    Q ← Q \ u
    if u ≠ e then
        for all neighbor v of u do
            alt ← dist[u] + c_uv
            if alt < dist[v] then
                dist[v] ← alt
                prev[v] ← u
            else
                prev[v] ← u
            end if
        end for
    end if
end while
S ← []
u ← e
do
    S ← u · S
    u ← prev[u]
while u  is not null
return S
```

A* is an algorithm based on Dijkstra's algorithm. We can see the pseudocode in Algorithm 2. This algorithm allows, like Dijkstra, to find the shortest path in a graph. In this thesis, we use it to optimize road trips. The main difference concerning Dijkstra's algorithm is that A*uses a heuristic function to estimate the cost of reaching the destination node from the current one. The correct choice of the heuristic function can improve the performance of the algorithm. The heuristic function used in the following works is the Euclidean distance between the geographical coordinates of the nodes.

Algorithm 2 A* algorithm

Input a graph $G(N, A)$, start node s, and end node e

$closed \leftarrow \emptyset$
$open \leftarrow \{s\}$
$cameFrom \leftarrow$ empty map
$gScore \leftarrow$ map with default value of ∞
$gScore[s] \leftarrow 0$
$fScore \leftarrow$ map with default value of ∞
$fScore[s] \leftarrow h(s, e)$
while $open \neq \emptyset$ **do**
    $u \leftarrow n \in open$ with the lowest $fScore[n]$
    if $u = e$ **then**
        $path \leftarrow [u]$
        while $u \in cameFrom.Keys$ **do**
            $u \leftarrow cameFrom[u]$
            $path \leftarrow u \cdot path$
        end while
        return $path$
    end if
    $open \leftarrow open \setminus \{u\}$
    $closed \leftarrow closed \cup \{u\}$
    for all neighbors v of u **do**
        if $v \notin closed$ **then**
            if $v \notin open$ **then**
                $open \leftarrow open \cup \{v\}$
            end if
            if $gScore[u] + c_{uv} < gScore[v]$ **then**
                $cameFrom[v] \leftarrow u$
                $gScore[v] \leftarrow gScore[u] + c_{uv}$
                $fScore[v] \leftarrow gScore[v] + h(v, e)$
            end if
        end if
    end for
end while
return failure

Pulse algorithm, especially its version for more than two objectives, was proposed by Duque, Lozano, and Medaglia (2015). It is an exact method to solve the multiobjective shortest path problem. Pulse is based on a brunch and cut. This algorithm uses a strong pruning strategy to discard, as soon as possible, branches that are dominated by the solutions found so far. The set of solutions is updated when the algorithm obtains new non-dominated solutions. This allows us to get different sub-optimal Pareto sets (approximations of the optimal Pareto set) during the execution of the algorithm. The main pseudocode is presented in Algorithm 3.

Algorithm 3 Multiobjective Pulse algorithm

```
Input a graph G(N, A), start node s, and end node e
P ← {}
c⃗(P) ← 0
initialization(G)
Pulse(s, c⃗(P), P)
return X_E
function PULSE(v_i, c⃗(P), P)
    if isAcyclic(s, P) then
        if ¬checkNadirPoint(v_i, c⃗(P)) then
            if ¬checkEfficientSet(v_i, c⃗(P)) then
                if ¬checkLabels(v_i, c⃗(P)) then
                    store(c⃗(P))
                    P′ ← P ∪ {v_i}
                    for v_j ∈ outgoingNeighbors(v_i) do
                        for k = 1 to size(c⃗) do
                            c_k(P′) ← c_k(P) + c_{ij,k}
                        end for
                        Pulse(v_j, c⃗(P′), P′)
                    end for
                end if
            end if
        end if
    end if
end function
```

Iterated Local Search (ILS) was introduced by Lourenço, Martin, and Stützle (2003). The pseudocode is presented in Algorithm 4. ILS performs a shake on the current (best) solution x and applies a local search to get the nearest local optimal x'. If x' improves on the best solution found so far x, it becomes the new best solution $x \leftarrow x'$; otherwise, it is discarded. The underlying idea of this algorithm is to find the local optima quickly and move between them until the global optimum is reached. They need processes, local searches, that allow them to explore the search space near the current solution intelligently.

Algorithm 4 Iterated Local Search

$x \leftarrow \text{generation}()$
$x \leftarrow \text{localsearch}(x)$
$t \leftarrow 1$ ▷ generation counter
while non stopping condition **do**
    $x' \leftarrow \text{shake}(x, npert)$
    $x' \leftarrow \text{localSearch}(x')$
    $x \leftarrow \text{acceptance}(x, x')$
    $t \leftarrow t + 1$
end while
return x

Simulated Annealing (SA) was proposed by Kirkpatrick, Gelatt, and Vecchi (1983). The annealing of metals inspires this algorithm. By controlling the heating and cooling, specific properties are given to the metals. SA moves the solution to neighborhoods with worse fitness in the early stages (high temperature) based on this idea. At this stage, exploring new regions of the search space is favored. The probability of choosing a worse solution decreases as the temperature decreases, and therefore the nearby neighborhood is exploited more. Its pseudocode is provided in Algorithm 5. The cooling strategy estimates the temperature t in the current iteration of the algorithm. The potential function uses this temperature to accept the new solution generated by the shake function from the previous one, even if it does not improve to the best solution found so far. The potential function allows controlling the exploration of the algorithm during its execution. This thesis has made a slight adaptation of a classical SA. The function *next* indicates the distance of the neighborhood that we will explore in this iteration (number of decision variables that change). After that, *shake* performs the movement from the current solution x to another one x' in the neighborhood and repeats the algorithm.

Algorithm 5 Simulated Annealing

$x \leftarrow \text{generation}()$
$x \leftarrow \text{localsearch}(x)$
$t \leftarrow 1$ ▷ generation counter
while non stopping condition **do**
    $t \leftarrow \text{cooling}(t_0, t)$
    $k \leftarrow \text{next}(t, maxiter)$
    $x' \leftarrow \text{shake}(x, k)$
    $x \leftarrow \text{potential}(x, x', t_0, t)$
    $t \leftarrow t + 1$
end while
return x

Variable Neighborhood Search (VNS) was introduced by Mladenović and Hansen (1997). We have based ourselves on the version presented in Drezner et al. (2015). This algorithm is based on the concept of neighborhood defined before. Each possible solution has a neighborhood associated to it. The pseudocode is in Algorithm 6. The current solution x is modified according to these neighborhoods: we select the k-th neighbor using the function *next* and change the current solution to a new one x' in the neighborhood of x by the *shake* procedure. After we move to a new solution, a local search process improve it, we assume that the two local searches could be different between them. In this version, a number $maxAttempts$ of consecutive non-improvements is allowed before an iteration of the algorithm.

Algorithm 6 Variable Neighborhood Search

$x \leftarrow$ generation()
$x \leftarrow$ localSearch$_1(x)$
$restart \leftarrow true$
while $restart$ & non stopping condition **do**
    $restart \leftarrow false$
    $j \leftarrow 1$
    while $\neg restart$ & $j \leq maxAttempts$ **do**
        $t \leftarrow 1$
        while $\neg restart$ & $t \leq k_{max}$ **do**
            $k \leftarrow$ next(t, k_{max})
            $x' \leftarrow$ shake(x, k)
            $x' \leftarrow$ localSearch$_2(x')$
            $x \leftarrow$ acceptance(x, x')
            if $x = x'$ **then** $restart \leftarrow true$
            else $restart \leftarrow false$
            end if
            $t \leftarrow t + 1$
        end while
        $j \leftarrow j + 1$
    end while
end while
return x

Genetic Algorithm (GA) was originally presented by Whitley (1994). A basic pseudocode is shown in Algorithm 7. This kind of algorithms is inspired by natural evolution. In each iteration, the algorithm generates λ new solutions (new population). GA selects several solutions from the population set P according to a *selection* criteria. These solutions are called the parent solutions. In this thesis, the selection process used always return two parents. The parent solutions are mixed (*crossover*) between them to generate a new ones (called offspring). There are many crossover operators in the scientific literature, in the presented pseudocode the crossover operator only return one offspring solution. After that, the new solution is probabilistically disturbed by the *mutation*. At the end of an iteration, we have a new set Q of offspring. Two types of strategies are commonly used to update the population P with elements of Q

- (μ, λ): We select λ elements of Q and replace the same number of elements in P following some kind of criteria.
- $(\mu + \lambda)$: We merge the two sets $P \cup Q$ and select μ individuals following some kind of strategy as randomly, elitist, the worst ones, etc.

After that, they are the new population P for the next iteration. Finally, the algorithm returns the best solution found in the population of solutions.

Algorithm 7 Genetic Algorithm

$P \leftarrow \mathsf{generatePopulation}()$
$t \leftarrow 1$
while non stopping condition **do**
    $Q \leftarrow \emptyset$
    for $l \in \{1..\mathsf{size}(P)\}$ **do**
        $parent_1, parent_2 \leftarrow \mathsf{selection}(pop)$
        $x \leftarrow \mathsf{crossover}(parent_1, parent_2)$
        $x' \leftarrow \mathsf{mutation}(x)$
        $Q \leftarrow Q \cup \{x'\}$
    end for
    $P \leftarrow \mathsf{replacement}(P, Q)$
    $t \leftarrow t + 1$
end while
return P

Differential Evolution (DE) is an evolutionary algorithm that does not use a nature-inspired mutation of the individuals. Instead, this algorithm creates a new individual applying a mathematical formula of two vector parents. This algorithm is especially suitable for continuous optimization. DE was originally proposed by Storn and Price (1997). Algorithm 8 shows the pseudocode of a DE. The main feature of this algorithm is in its crossover operation. There are many of this kind of operators in the scientific literature (Price, 2013). Specifically, in this thesis, we use the "DE/best/1/bin" strategy (Price, Storn, and Lampinen, 2005) to create the new individuals.

Particle Swarm Optimization (PSO) was originally proposed by Kennedy and Eberhart (1995). Algorithm 9 shows a pseudocode of PSO. This swarm-intelligence metaheuristic tries to improve the whole population continuously. In each iteration, the algorithm moves the solutions (modify them), which it calls particles, into more promising areas of the solution space. Each solution uses information from its position, experience, and the rest of the particles (social knowledge) to modify its motion vector and approach better regions of the search space.

Algorithm 8 Differential Evolution

$t \leftarrow 1$
$\Theta_t \leftarrow \mathsf{generatePopulation}()$
$\mathsf{evaluate}(\Theta_t)$
while non stopping condition **do**
    $\Theta^{\text{new}} \leftarrow \emptyset$ ▷ auxiliary population
    for $i \in \{1..\mathsf{size}(P)\}$ **do**
        $\Theta^{\text{parents}} \leftarrow \mathsf{selection}(\Theta_t)$
        $\Theta^{\text{child}} \leftarrow \mathsf{DE\text{-}crossover}(\Theta^{\text{parents}})$
        $\mathsf{evaluate}(\Theta^{\text{child}})$
        $\Theta^{\text{new}} \leftarrow \Theta^{\text{new}} \cup \Theta^{\text{child}}$
    end for
    $\Theta_{t+1} \leftarrow \mathsf{Replace}(\Theta_t, \Theta^{\text{new}})$
    $t \leftarrow t + 1$
end while

Algorithm 9 Particle Swarm Optimization

$pop \leftarrow \mathsf{generatePopulation}()$
$p \leftarrow pop$ ▷ set of best known solutions
$best \leftarrow getBestSolution(pop)$
for $k \in \{1..pop\}$ **do**
    $v_k \leftarrow U(-|b_{upper} - b_{lower}|, |b_{upper} - b_{lower}|)$
end for
$i \leftarrow 1$
while $j \leq iter$ & non stopping condition **do**
    for $i \in \{1..\mathsf{size}(P)\}$ **do**
        $x \leftarrow pop_i$
        for $f \in \{1..\mathrm{x}\}$ **do**
            $r_p \leftarrow U(0, 1)$
            $r_g \leftarrow U(0, 1)$
            $v_{i,f} \leftarrow \omega v_{i,f} + \phi_p r_p (p_{i,f} - x_f) + \phi_g r_g (best_f - x_f)$
        end for
        $x \leftarrow x + v_i$
        if $\mathsf{evaluation}(x) < \mathsf{evaluation}(p_i)$ **then**
            $p_i \leftarrow x$
            if $\mathsf{evaluation}(p_i) < \mathsf{evaluation}(best)$ **then**
                $best \leftarrow p_i$
            end if
        end if
    end for
    $i \leftarrow i + 1$
end while
return $best$

Non-dominated Sorting Genetic Algorithm II (NSGA-II) was proposed by Deb et al. (2002a). The evolutionary search applied by NSGA-II uses a non-dominated elitist ordering to mitigate the complexity of the dominance check, a crowding technique to keep the diversity in the solutions, and a fitness assignment method that takes into account dominance ranks and crowding distance values. NSGA-II applies the (μ+λ) evolution model. Tournament selection is applied, with a tournament size of two individuals. The tournament criteria is based on dominance between solutions. If the two compared individuals are non-dominated, the selection is made based on crowding distance. Fitness assignment is performed considering Pareto dominance rank and crowding distance values.

Algorithm 10 Non-dominated Sorting Genetic Algorithm II

$offspring \leftarrow \emptyset$
$P \leftarrow$ generatePopulation()
while non stopping condition **do**
  evaluate(P) ▷ population evaluation
  $R \leftarrow P \cup offspring$
  $fronts \leftarrow$ non-dominated sorting(R)
  $Q \leftarrow \emptyset$
  $i \leftarrow 1$
  while $|Q| + |fronts(i)| \leq N$ **do**
    crowding distance($fronts(i)$)
    $Q \leftarrow Q \cup fronts(i)$
    $i \leftarrow i + 1$
  end while
  sortingByDistance($fronts(i)$)
  $Q \leftarrow Q \cup fronts(i)[1 : (N - |Q|)]$
  $selected \leftarrow$ selection(Q)
  $offspring \leftarrow$ evolutionaryOperators($selected$)
  $P \leftarrow Q$
end while
return computed Pareto front

Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) is an evolutionary algorithm proposed by Zhang and Li (2007). In this evolutionary algorithm, we have used the proposed version called MOEA/D-STM (Li et al., 2014). The pseudocode of this version is presented in Algorithm 11. The basic idea of MOEA/D is divide and conquer. The approach is to divide the multiobjective problem into multiple single-objective problems. MOEA/D uses a method of scaling the problem by several weight vectors $\vec{w}^{i_T}$ to divide the space. This strategy allows us to obtain different solutions in these regions simultaneously and thus, improve the approximate Pareto front in a coordinated way.

Algorithm 11 Multiobj. Evolutionary Algorithm Based on Decomposition

$P \leftarrow$ generatePopulation()
$weightVectors \leftarrow$ initializeWeightVectors()
$\vec{z}^{*} \leftarrow$ setIdealVector()
$\vec{z}^{nad} \leftarrow$ setNadirVector()
for $index \in \{1..maxGenerations\}$ **do**
    $B(i) \leftarrow \{i_1, \ldots, i_T\}$
        where $\vec{w}^{i_1}, \ldots, \vec{w}^{i_T}$ *are the* T *closest weight vectors to* $\vec{w}^{i}$
    $\pi^{i} \leftarrow 1$
end for
for $i \in \{1..maxGenerations\}$ **do**
    $Q \leftarrow \emptyset$
    for *each* $i \in I$ **do**
        if $U(0,1) < \delta$ **then** $E \leftarrow B(i)$
        else $E \leftarrow P$
        end if
        $newPopulation \leftarrow$ generateChildPopulation($poulation$)
        $parents \leftarrow$ selection($newPopulation$)
        $child \leftarrow$ crossing($parents$)
        $child \leftarrow$ mutate($child$)
        $Q \leftarrow Q \cup child$
        update($\vec{z}^{*}$)
        update($\vec{z}^{nad}$)
    end for
    $R \leftarrow P \cup Q$
    $P \leftarrow \mathrm{STM}(R, W, \vec{z}^{*}, \vec{z}^{nad})$
end for
return P

Strength Pareto Evolutionary Algorithm 2 (SPEA2) was an evolution of the SPEA algorithm Zitzler and Thiele (1998a) made by Zitzler, Laumanns, and Thiele (2001b). SPEA2 is distinct from other Multiobjective Evolutionary Algorithms because it applies the *strength* concept on the fitness computation, which is based on both Pareto dominance and diversity. Thus, the *strength* measures how many solutions dominate (and are dominated by) each candidate solution. In turn, a density estimation is also considered for fitness assignment. Furthermore, an elite population is defined to store the non-dominated individuals found during the search to apply elitism. SPEA2 working on a population P (size N) is shown in Algorithm 12. The *elitePop* parameter represents the elite population, which has *eliteSize* size. The most similar individuals are removed by a pruning method to assure that the size of the elite population is always *eliteSize* when the elite population is full.

Algorithm 12 Strength Pareto Evolutionary Algorithm 2

$t \leftarrow 0$
$elitePop \leftarrow \emptyset$
$\mathsf{generatePopulation}(P(0))$
while non stopping condition **do**
    $\mathsf{evaluate}(P(t))$
    $R \leftarrow P(t) \cup elitePop$
    for $s_i \in R$ **do**
        $si_{raw} \leftarrow \mathsf{computeRawEvaluation}(s_i, R)$
        $si_{density} \leftarrow \mathsf{computeDensity}(s_i, R)$
        $si_{fitness} \leftarrow si_{raw} + si_{density}$
    end for
    $elitePop \leftarrow \mathsf{nonDominated}(R)$
    if $\mathsf{size}(elitePoP) > eliteSize$ **then**
        $elitePop \leftarrow \mathsf{removeMostSimilar}(elitePop)$
    end if
    $selected \leftarrow \mathsf{selection}(R)$
    $offspring \leftarrow \mathsf{evolutionaryOperators}(selected)$
    $t \leftarrow t + 1$
end while
return computed Pareto front

Part II

Problems

Chapter 4

Hybridization for Traffic Lights Simulation Optimization

4.1 Motivation

In many real-world optimization problems, the evaluation of candidate solutions requires the simulation of a process under various scenarios that represent uncertainty about the real world. Good solutions should achieve good objective function values and show robustness, i.e., low variance across scenarios. To assess the robustness of solutions, it is often required to simulate each solution several times using different data, starting conditions or random numbers. For example, when planning the traffic light schedules within a city, it is desirable to find a schedule that works well under many different traffic conditions (Bravo et al., 2016; Ferrer et al., 2016; García-Nieto, Alba, and Olivera, 2012; García-Nieto, Olivera, and Alba, 2013; Sánchez, Galán, and Rubio, 2008; Sánchez-Medina, Galán-Moreno, and Rubio-Royo, 2010; Stolfi and Alba, 2014; Stolfi and Alba, 2015a; Teklu, Sumalee, and Watling, 2007; Teo, Kow, and Chin, 2010; Péres et al., 2018b). However, legal and technical limitations may make real-time traffic light control infeasible. A common approach simulates each candidate solution under several traffic scenarios generated from actual traffic data. However, there is a trade-off between the number of traffic scenarios used for evaluating each solution and the maximum number of candidate solutions evaluated.

Scientific work as Ferrer, López-Ibáñez, and Alba (2019) has shown that IRACE (López-Ibáñez et al., 2016) can find high-quality and low-variance traffic light schedules by dynamically adjusting the number of simulations performed per solution. The elitist iterated racing algorithm implemented by IRACE has been traditionally used to configure parameters in machine learning and optimization algorithms. Each configuration must be evaluated on several training instances of an optimization problem, and the algorithm themselves are often stochastic. The algorithm implemented in IRACE uses a learning mechanism inspired by reinforcement learning to sample new solutions from the best ones previously found. Although this approach tends to work well for configuring a mix of categorical and numerical parameters with dependencies and constraints, other operators may perform better when the problem consists only of numerical decision variables.

In this chapter, we introduce a hybridization of operators from Evolutionary Algorithm (EA) and the elitist iterated racing of IRACE. We evaluate its performance on the traffic light optimization problem and compare it with previous results from the literature. The idea of previous approaches to hybridizing EAs and racing were performing independent races to carry out the evaluation step within an EA (Heidrich-Meisner and Igel, 2009), whereas our proposal replaces the sampling mechanism in IRACE, which is not simply a sequence of independent races, with evolutionary operators.

Besides, to add value to our experimentation, we use an instance based on real data from the city of Malaga, Spain. We also utilize a traffic simulator, SUMO (Behrisch et al., 2011; Krajzewicz et al., 2012a), to evaluate each of the traffic light schedules generated by the algorithms. With this, we not only seek to analyze which algorithm is better but also to solve a real problem of a real city.

In summary, the main contributions of this chapter are:

- We propose new hybrid algorithms that combine racing strategies with evolutionary operators to obtain powerful and robust algorithms.
- We optimize the traffic light plan of the small map of Malaga using detailed micro-simulations.
- We offer an in-depth analysis of our hybrid algorithms and compare them with well-known EAs such as a Genetic Algorithm (GA) and a Differential Evolution (DE).
- We study which algorithm presents the most significant improvement to the city according to different traffic quality measures and emission reduction.

The rest of this chapter is organized as follows: Section 4.2 presents a description of the Traffic Light Scheduling Problem. Section 4.3 describes the hybridization between IRACE and EAs. Section 4.4 outlines the main aspects of our experimentation. We discuss the results obtained in Section 4.5. Finally, Section 4.6 presents some conclusions.

4.2 Problem Description

Traffic flow in large Smart City has become one of the most severe problems large cities face. This problem is further aggravated in some cases due to the high amount of traffic jams, traffic accidents, or even injured people or deaths. Therefore traffic must be regulated with some elements such as traffic lights. The larger the metropolitan area, the higher the number of traffic lights needed to control the traffic flow. Optimal management of traffic might be beneficial to minimize journey times and reduce fuel consumption and harmful emissions.

Traffic lights are coordinated in phases: green, yellow and red. In this way, when some traffic lights of the same intersection are in green, some others must be in red. Besides, the different pre-defined phases for an intersection are sequences repeated over time, we call Traffic Light Program (TLP) to each of those sequences.

The large number of program combinations that appear in traffic light schedules of large cities require automatic tools to generate optimal TLP, which motivates the Traffic Light Scheduling (TLS) problem (Sánchez, Galán, and Rubio, 2008; García-Nieto, Olivera, and Alba, 2013; Sánchez-Medina, Galán-Moreno, and Rubio-Royo, 2010). The main objective of this problem is to find optimized TLP for all the traffic lights located in the intersections of an urban area to reduce journey time, emissions, and fuel consumption.

Let us define the TLS problem as follows. Let $P = (I_1, \ldots, I_n)$ be a candidate TLP, where each I_i corresponds to a different intersection defined as a set of predefined valid phases $I_i = (\varphi_{i1}, \ldots, \varphi_{im_i})$, where $m_i = |I_i|$ and each $\varphi_{ij} \in \mathbb{N}^+$ represents the duration (in seconds) of phase j in intersection I_i, that is, the duration of each valid phase of light colors. Each sequence of light colors is fixed for each traffic light by the Traffic Control Center. A example of colors could be "*Red-Yellow-Green-Red-Green-Yellow*". The objective is to find a TLP P' that minimizes a fitness function $f\colon \Gamma \to \mathbb{R}$ such that:

$$P' = \arg\min_{P \in \Gamma} \{f(P)\} \tag{4.1}$$

where Γ is the space of all possible TLPs.

In order to define the fitness function, we need to explain some concepts used in the definition. The evaluation of a solution is performed using a traffic simulator that provides information regarding the flow of vehicles. Vehicles travel from a starting position to a destination position, then the travel time (t_v) of a vehicle v is the number of simulation steps (1 second per simulation step) in which its speed was above 0.1 m/s, while its waiting time (w_v) is the number of simulation steps in which its speed was below 0.1 m/s.

Long phase duration may lead to a collapse of the intersection. TLPs should prioritize those phases with more green lights on the directions with a high number of vehicles circulating. So, we should maximize the following ratio measure:

$$GR(P) = \sum_{i=1}^{n} \sum_{j=1}^{|I_i|} \varphi_{ij} \cdot \frac{G_{ij}}{R_{ij}} \tag{4.2}$$

where G_{ij} is the number of traffic lights in green, and R_{ij} is the number of traffic lights in red in phase j of intersection i and φ_{ij} is the duration of the phase. The minimum value of R_{ij} is 1 because if in a cross of vehicles there is a traffic light in green, must be another in red (to avoid crass accidents).

Finally, we define the following fitness function to be minimized:

$$f(P) = \frac{V^{\text{rem}}(P) \cdot t^{\text{sim}} + \sum_{v=1}^{V(P)} (t_v(P) + w_v(P))}{V(P)^2 + GR(P)} \tag{4.3}$$

where the presence of vehicles with incomplete journeys $V^{\text{rem}}(P)$ penalizes the fitness of a solution P proportionally to the simulation time t^{sim}. The number of vehicles that arrive at their destinations is squared ($V(P)^2$) to prioritize this criterion over the rest. García-Nieto, Alba, and Olivera (2012) and García-Nieto, Olivera, and Alba (2013) used successfully this fitness function.

4.3 Hybridization of IRACE and EAs

There are many definitions of hybrid algorithms, yet the general idea is to combine components or concepts from different techniques to exploit desirable characteristics of those components to tackle problems with particular features (Blum and Raidl, 2016). In this chapter, we combine the elitist iterated racing strategy from IRACE with evolutionary operators to obtain an algorithm that performs well on numerical optimization problems where the fitness of each solution is uncertain and must be evaluated using multiple simulations. The elitist iterated racing strategy of IRACE decides how many simulations should be performed per solution, how solutions are compared, and which solutions should be discarded at each iteration. The evolutionary operators are responsible for generating new solutions from the surviving population of solutions. Next, we will briefly explain the base algorithm, IRACE, and the different characteristics of the hybrid algorithm.

4.3.1 IRACE

IRACE (López-Ibáñez et al., 2016) has been previously used as a solver in real problems as the TLS problem (Ferrer, López-Ibáñez, and Alba, 2019). A description of the main functionality of IRACE and its pseudocode can be seen in Appendix C.2.2. Specifically, the pairwise paired Student's t-test is used as the elimination test because this statistic test performed well previously (Ferrer, López-Ibáñez, and Alba, 2019).

4.3.2 Hybrid Algorithms

Now, let us analyze our hybrid algorithms. The IRACE function $\mathsf{Sample}(\mathcal{M})$ generates a new set of candidate solutions to the problem. In our hybrid algorithms, we replace that function with operators taken from two EAs: GA and DE. We call IRACE+GA and IRACE+DE, respectively, to these new hybrid algorithms. These EAs have already demonstrated their effectiveness in solving the TLS problem (Ferrer, López-Ibáñez, and Alba, 2019), so we consider them to hybridize with IRACE. In this way, the racing step remains intact but the sampling of new solutions is carried out by these EAs. In IRACE, the $\mathsf{Sample}(\mathcal{M})$ procedure is equivalent to the mating selection and variation steps of an EA, i.e., selecting parents, generating new individuals from them (crossover), making some modification to the new solutions (mutation), and returning the new set of solutions. IRACE works with both numerical and categorical parameters. TLS problem only has numerical parameters, so, we do not have to deal with categorical parameters.

The set of elite solutions Θ^{elite} contains the best solutions found by IRACE after the race performed at each iteration. In our hybrid algorithm, the parents used by the evolutionary operators are selected from Θ^{elite}. However, the size of Θ^{elite} may vary each iteration and may be insufficient for the number of parents required by the evolutionary operators. We handle this situation by generating additional parents by random uniform sampling as in the initial iteration. This mechanism also introduces more diversity to the set of parent solutions. Because we use several evolutionary operators, the number of selected parents differs from one algorithm to another. IRACE+GA needs two parents for the operator execution, while IRACE+DE needs four. The restriction in the number of parents is given by the operators used by each algorithm, because each operator requires a different number of solutions to generate a new one.

Two variants of the proposed hybrid algorithm are implemented with the following operators:

- IRACE+GA: uniform crossover (Syswerda, 1989) and integer polynomial mutation (Deb and Agrawal, 1999)
- IRACE+DE: `DE/best/1/bin` strategy (Price, Storn, and Lampinen, 2005).

The evaluation of the new solutions returned after the $\mathsf{Sample}(\mathcal{M})$ phase is computed by performing several simulations, as carried out by IRACE. After the evaluation phase, we merge this set of new solutions Θ^{new} with set of elite solutions Θ^{elite} to execute the racing. This returns a new set of elite solutions, which will be used in the next iteration of the hybrid algorithms.

4.4 Experimental Setup

We describe here the experimental protocol followed. First, we describe the real-world case study of TLS problem that is the main motivation of our research. After that, we provide details about the experiments carried out. We will analyze these experiments in the next section.

4.4.1 Real-World Case Study

We consider a realistic scenario derived from the downtown traffic network of Malaga. Our network model was created from real data about traffic rules, traffic element locations, road directions, streets, junctions, etc. Also, we needed the routes and vehicles circulating and their speeds to create the testing and training traffic scenarios. This information was collected from sensorized points in particular streets, measuring traffic density at various time intervals. We have applied the Flow Generator Algorithm (Stolfi and Alba, 2015a), using sensed data extracted, to generate 60 different traffic scenarios with an average of 4,827 vehicles (or different vehicle routes) per scenario. We split the generated traffic scenarios into two equal sets of 30 scenarios each to evaluate the reliability of a candidate solution. One (training) set is exclusively used for optimization, that is, for identifying optimal TLS problem solutions. The other (testing) set of scenarios compares the solutions found during optimization.

4.4.2 Case Study Constraints

Real-world instances of the TLS problem often present additional constraints. In our case, we consider the constraints recommended by the City Council of Malaga. Phases with any yellow signals are called *fixed phases* because they have a predetermined duration, and Y will denote the set of such phases. These fixed phases correspond to pedestrian crosses, which last for a fixed time of 4 $\times$ *number of lanes* seconds. Non-fixed phases have a minimum duration of $\varphi_{\min} = 15$ seconds. Moreover, the total program time (Tp_i) within each intersection I_i, which is computed as the sum of its phase durations:

$$Tp_i = \sum_{j=1}^{|I_i|} \varphi_{ij} \tag{4.4}$$

Tp_i is constrained within $[Tp_{\min}, Tp_{\max}]$. For the City Council of Malaga, $Tp_{\min} = 60$ and $Tp_{\max} = 120$ seconds.

By default, the first programs of all intersections start simultaneously. However, we also optimize an offset time at each intersection (To_i) that represents a shift in seconds of the starting time of the program at the start of the simulation. If the offset value of an intersection is negative, then the program start time is shifted back that number of seconds, and the program starts on a phase before the first one; whereas if the offset is positive, the program begins as if that number of seconds has already passed, i.e., skipping those seconds from the duration of the first phase and, maybe, of later phases. Offset times enable the emergence of a series of coordinated traffic lights that produce a continuous traffic flow over several intersections in one main direction. Offset values are constrained within the time interval $To_i \in [To_{\min}, To_{\max}] = [-30, 30]$.

4.4.3 Repair Procedure

The TLS problem is subject to some constraints that we explained in Section 4.4.2. We propose a repair procedure used by all the algorithms before the simulation to ensure valid candidate solutions. The value of each phase duration φ_{ij} is already constrained within a range that is larger than the minimum phase duration $\varphi_{\min}$. However, we need to guarantee that the total program time Tp_i is within $[Tp_{\min}, Tp_{\max}]$. We can distinguish two cases.

In the first case, if the total program time for intersection I_i is smaller than $Tp_{\min}$, then we replace each non-fixed phase (those that do not contain a yellow signal, i.e., $\varphi_{ij} \notin Y$) with

$$\varphi_{ij} = \left\lceil \varphi_{ij} \cdot \frac{Tp_{\min} - Tp_i^Y}{Tp_i - Tp_i^Y} \right\rceil \tag{4.5}$$

where $Tp_i^Y = \sum_{\varphi_{ij} \in I_i \cap Y} \varphi_{ij}$ is the sum of the fixed phase durations within intersection I_i.

In the second case, if the total program time is larger than $Tp_{\max}$, then we replace each non-fixed phase ($\varphi_{ij} \notin Y$) with

$$\varphi_{ij} = \varphi_{\min} + \left\lfloor (\varphi_{ij} - \varphi_{\min}) \cdot \frac{Tp_{\max} - Tp_i^Y - \varphi_{\min} \cdot |I_i \setminus Y|}{Tp_i - Tp_i^Y - \varphi_{\min} \cdot |I_i \setminus Y|} \right\rfloor \tag{4.6}$$

where $|I_i \setminus Y|$ is the number of non-fixed phases within intersection I_i and Tp_i^Y is the total duration of the fixed phases within intersection I_i.

4.4.4 SUMO Simulator

The quality of a solution (traffic light program) is evaluated through the Simulator of Urban Mobility (SUMO) (Behrisch et al., 2011; Krajzewicz et al., 2012a). A description of SUMO is provided in Appendix C.2.5.

Since we already introduce variability by means of the different traffic scenarios, we fix the random seed used by SUMO to zero in all simulations. This means that, given a traffic scenario and a candidate solution, the simulation is deterministic. In all experiments, we stop each run of an algorithm, either a variant of IRACE or otherwise, after executing 30 000 calls to the SUMO simulator. Given that each solution is simulated on a number of different scenarios, the number of solutions evaluated per run is often much lower.

4.4.5 Algorithms

In our experiments we compare IRACE with the two hybrid variants described above, namely, IRACE+GA and IRACE+DE. In addition, to assess the contribution of the elitist racing mechanism, we also evaluate the classical GA and DE algorithms. Here, we describe the implementation details of these algorithms.

Following the conclusions from another state-of-the-art work on the TLS problem (Ferrer, López-Ibáñez, and Alba, 2019), we use default settings for IRACE and the hybrids, except the following. The population size is fixed to 10 individuals (also for GA and DE), the minimum number of traffic scenarios simulated per candidate solution is set to two ($T^{\text{first}} = 2$) and we enable the *deterministic* option that tells IRACE that the only source of uncertainty are the different scenarios and not the simulations themselves. The evolutionary algorithms use a fixed number of simulations per candidate solution. Each solution is simulated on five different training scenarios and its fitness is computed as the mean fitness over the five simulations. In (Ferrer, López-Ibáñez, and Alba, 2019), the authors already compared IRACE with differential evolution, genetic algorithm, particle swarm optimization, and a random search; and showed that IRACE obtained the best results with GA and DE being a close second, therefore, we focus here in the comparison of IRACE, GA, DE, and the hybrids.

Our GA implementation uses a ranking method for parent selection and elitist replacement for the next population, that is, the two best individuals of the current population are included in the next one. The operators used are uniform crossover and integer polynomial mutation with 1.0 of probability of crossover and 0.1 of probability of mutation. These parameter settings were found by additional experiments carried out in previous studies (Bravo et al., 2016) to produce a search behavior that is more exploitative rather than explorative, which is more appropriate for the TLS problem. Our DE implementation uses a "best/1/bin" strategy with difference factor $F = 0.5$ and probability of crossover 0.5. These are the default parameter values in jMetal (Nebro, Durillo, and Vergne, 2015). Finally, IRACE+GA and IRACE+DE use the same parameter settings as the GA and DE, respectively.

4.4.6 Experimental Details

As mentioned above, we generated 60 traffic scenarios from real sensor data and we split these scenarios into two sets of size 30. One set (training set) is used when running the algorithms to find TLS problem solutions, while the other set (testing set) is used for evaluating the fitness and reliability of these solutions and comparing the various strategies analyzed in this research. During optimization, the traffic is simulated up to a predefined time horizon (1 hour plus 10 minutes of warm-up, in our case) in order to simulate the peak period in our real-world case study. For the constraints of the TLS problem, we apply the same repair method as in Ferrer, López-Ibáñez, and Alba (2019).

The algorithms presented are non-deterministic algorithms, so we performed 30 independent runs for a fair comparison between them. After the executions, we applied the non-parametric Kruskal-Wallis test with a confidence level of 95% (p-value < 0.05) with Holms's p-value correction to check if the observed differences are statistically significant. In the cases where Kruskal-Wallis test rejects the null hypothesis, we run a single factor ANOVA post hoc test for pairwise comparisons. To properly interpret the results of statistical tests, it is always advisable to report effect size measures. For that purpose, we have also used the non-parametric effect size measure $\widehat{A}_{12}$ statistic proposed by Vargha and Delaney (2000).

4.5 Results

To give an in-depth view of the performance of our hybrid algorithms against the standard ones, we will analyze their performance in several sets of scenarios (training and testing). With this, we want to present the competitiveness of our proposal and give a solution to the TLS problem.

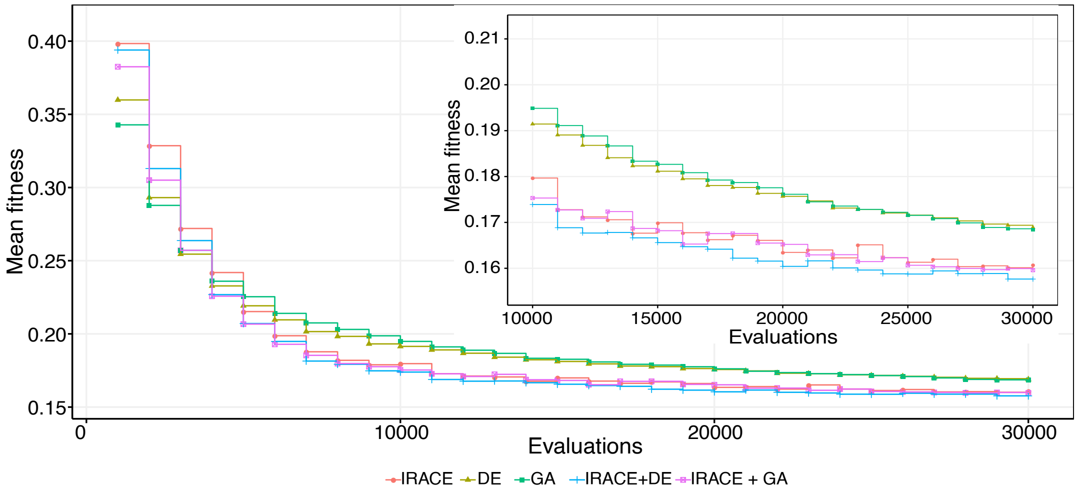


FIGURE 4.1: Mean fitness of the best solutions found so far within each run, as estimated by each algorithm at each moment of its execution on traffic scenarios from the training set. Results in the range [10,000, 30,000] are magnified.

4.5.1 Training Set

During the training phase, each algorithm performs a maximum of 30,000 simulations. Figure 4.1 shows the best fitness obtained, over the number of simulations, averaged over 30 runs of each algorithm. We can see that, in general, up to 10,000 simulations, all the algorithms improve significantly the quality of their solutions, but after this number of simulations, the improvement slows down. Although the GA and DE obtain the best results up to 5,000 simulations, they are quickly overtaken by IRACE and its hybrid variants. Figure 4.1 also shows in more detail the differences, starting from 10,000 simulations, between IRACE, IRACE+DE and IRACE+GA. We can notice that IRACE+DE consistently obtains the lowest mean fitness, while IRACE and IRACE+GA show a similar result. The plot also shows that the fitness reported by the IRACE hybrids sometimes increases due to the racing procedure performing additional simulations to refine the estimation of the fitness.

We have performed an $\widehat{A}_{12}$ test at the end of the training execution (30,000 simulations) to check if IRACE+DE is indeed better than the other algorithms. Table 4.1 shows the results of the $\widehat{A}_{12}$ test among the different algorithms, where each value gives the probability of the algorithm in the column returning a better solution than the one in the row. The test indicates that GA is better than the rest of the algorithms. However, we also look at the other statistics shown in Table 4.2. Although GA has better median than IRACE+DE's only by 10^{-4}, the standard deviation of IRACE+DE is 3.6 times less than GA. Thus, we conclude that IRACE+DE is more robust than GA. EAs obtain lower mean and median, while IRACE reports solutions with smaller variability. These results support our approach of hybridizing IRACE with EAs to obtain good quality robust solutions. Particularly, IRACE+DE looks like a good option if we want to apply these features.

TABLE 4.1: Results of the $\widehat{A}_{12}$ test for the evaluation of the last solutions found over training scenarios. Probability that the algorithm (column) is better than another algorithm (row). We highlight in bold the values when the algorithm in the column is better than the algorithm in the row.

	IRACE	IRACE+DE	IRACE+GA	GA	DE
IRACE	—	**0.6711**	**0.6100**	**0.6067**	0.4933
IRACE+DE	0.3289	—	0.3956	**0.5122**	0.3556
IRACE+GA	0.3900	**0.6044**	—	**0.5478**	0.4267
GA	0.3933	0.4878	0.4522	—	0.3811
DE	**0.5067**	**0.6444**	**0.5733**	**0.6189**	—

TABLE 4.2: Statistics of each algorithm from the best solutions obtained in the 30,000 simulation of the training. We mark in bold the lower value of each metric.

Algorithm	Mean	Median	SD
IRACE+DE	**0.1585**	0.1563	0.0101
IRACE+GA	0.1597	0.1590	0.0076
IRACE	0.1621	0.1615	**0.0064**
GA	0.1684	**0.1562**	0.0364
DE	0.1689	0.1596	0.0215

4.5.2 Testing Set

The above reported statistics were obtained after evaluating the final solutions on the same scenarios used during optimization, but the training scenarios will never arise exactly in the real world. We evaluate again the solutions on the 30 testing scenarios to properly assess their quality in unseen scenarios. Figure 4.2 shows the boxplots of each independent execution in each algorithm. EAs have the highest variability, while the other three algorithms have more robust boxplots.

To better compare the different algorithms, we summarize the mean, median, and standard deviation (see Table 4.3) between the independent runs.

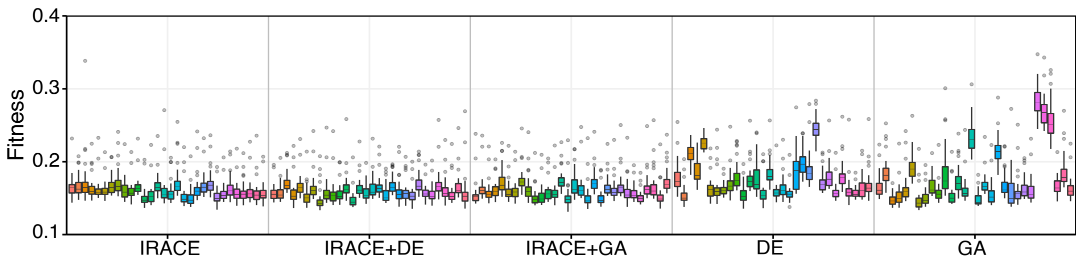


FIGURE 4.2: Fitness of the solutions obtained by the 5 algorithms. Each boxplot shows the distribution of fitness values of one solution on the 30 traffic scenarios in the test set.

TABLE 4.3: Statistics of each algorithm from the best solutions obtained in the testing phase. We mark in bold the lower value of each metric.

Algorithm	Mean	Median	SD
IRACE+DE	**0.1607**	**0.1571**	**0.0175**
IRACE+GA	0.1620	0.1577	0.0184
IRACE	0.1623	0.1581	0.0184
GA	0.1793	0.1630	0.0407
DE	0.1769	0.1676	0.0275

TABLE 4.4: Wilcoxon Test p-value of the testing set with Holm correction.

	IRACE	DE	IRACE+DE	GA
DE	$< 2e{-}16$	—	—	—
IRACE+DE	**0.0237**	$< 2e{-}16$	—	—
GA	$3.5e{-}11$	**0.0011**	$< 2e{-}16$	—
IRACE+GA	0.3284	$< 2e{-}16$	0.2122	$5.3e{-}13$

IRACE+DE gets the best results in each of these metrics, followed by IRACE and IRACE+GA, which are very similar, and the last ones, the EAs. This is an excellent result for IRACE+DE because, as it was proved in the training results, remarks the competitiveness of the algorithm also in the testing phase.

Anyway, the algorithms using IRACE obtain very similar results. This makes us wonder if there are significant differences between them. To study this, we perform a Wilcoxon rank-sum test between the algorithms to check if there are significant differences. Table 4.4 shows the p-values reported by the test. As we expected, IRACE+DE has significant differences compared to IRACE and EAs. This result support our working hypothesis: including EAs (specifically a DE) into IRACE can improve the performance. IRACE+GA and IRACE do not offer significant differences between them, which is not a bad result either, since at least the hybrid algorithm reaches a similar performance to IRACE. Lastly, EAs have significant differences with the others algorithms.

Finally, we perform an $\widehat{A}_{12}$ test to see if our hybrid algorithms (especially IRACE+DE) effectively beat the other competitors. Table 4.5 shows the results for the $\widehat{A}_{12}$ test. We observe that IRACE+DE is better than standard IRACE 53.62% of the time, and 66.17% better than evolutionary ones. While IRACE+GA is 51.33% of the time better than IRACE and 64.34% better than the evolutionary ones. These differences are in favour of our approach. After this experimentation, we can conclude that hybridizing IRACE with evolutionary algorithms is a viable and competitive option. With this idea, we join the best of both types of algorithms obtaining a powerful and robust algorithm, which allows us to find better solutions for TLS problem than the commonly used algorithms.

TABLE 4.5: Results of the $\widehat{A}_{12}$ test for testing. Probability that the algorithm (column) is better than another algorithm (row). We highlight in bold the values when the algorithm in the column is better than the algorithm in the row.

	IRACE	IRACE+DE	IRACE+GA	GA	DE
IRACE	—	**0.5362**	**0.5133**	0.4066	0.3198
IRACE+DE	0.4638	—	0.4780	0.3820	0.2946
IRACE+GA	0.4867	**0.5220**	—	0.3985	0.3147
GA	**0.5934**	**0.6180**	**0.6015**	—	0.4506
DE	**0.6802**	**0.7054**	**0.6853**	**0.5494**	—

4.5.3 Impact in Real World

The previous analysis has focused on the fitness function, an approximation which encompasses some knowledge of traffic flow to guide the search, however, it is quite complex to extract useful information for the domain's expert. Therefore in this section, we study the main traffic and environmental indicators which give the domain's expert more information about the solution.

In a real-world problem, it is desirable to analyze the impact that a representative solution of the different algorithms would have in a real environment. We choose one solution from each algorithm, as a typical traffic light plan as follows: (i) we calculate the mean of the fitness obtained in the 30 scenarios of the testing set by each of the 30 solutions of each algorithm, (ii) we order upwards these mean fitness for each algorithm, (iii) we select, as the representative, the solution whose fitness value is at the 16th position, that is, immediately following the median solution. We cannot select the median because there are an even number of solutions (30).

We simulate again each of the representative solutions in the test scenarios but allowing all the vehicles to reach their destination. This means that the fitness values are not penalized, hence, they are smaller than those reported in the previous boxplots. With these new simulations, we obtain 34 different traffic and environmental measures of the 30 testing scenarios. Figure 4.3 shows some of the most important measures for each algorithm. IRACE+DE obtains the best average values in *MeanTravelTime* and *MeanWaitingTime*, while IRACE+GA has the lowest *MaxTravelTime* and *MaxWaitingTime*. In practice, if we implement the IRACE+DE solution, citizens would complete the journeys in less time (329.60s) and with less waiting time at intersections (88.84s). If IRACE+GA solution were implemented, the *MeanTravelTime* is higher than in IRACE+DE solution, but in the worst case (*MaxTravelTime* and *MaxWaitingTime*), IRACE+GA obtains the minimum values. On the complete opposite side, we have GA and DE, with the worst results of the comparison.

Regarding the environmental impact (fuel consumption and CO_2 emissions), IRACE+DE gives the most eco-friendly solutions. Nowadays, pollution is a serious issue in many cities, so offering solutions that reduce emissions and fuel is of vital importance in today's cities.

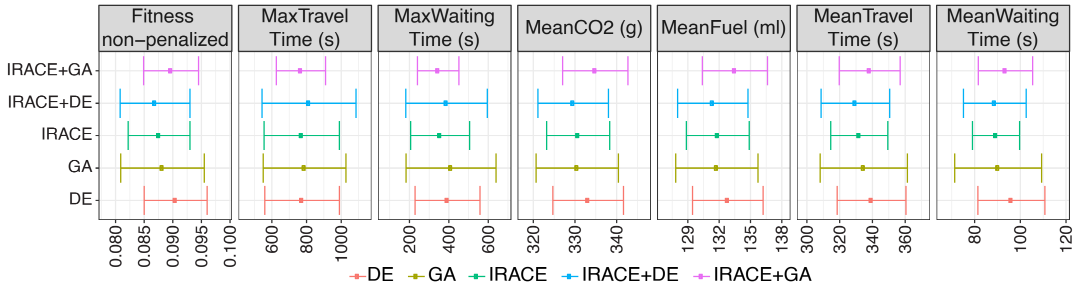


FIGURE 4.3: Traffic measures per vehicle. Mean values (and standard deviation) over 30 test traffic scenarios of the median solutions for the five algorithms.

With all these results, we can confirm that better TLPs result in less CO_2 emissions, less fuel consumption, and less journey time for the citizen. Our hybrid proposals, especially IRACE+DE, offer competitive solutions from a scientific point of view. Still, it would also positively impact the city at multiple levels, both environmental and the citizens' quality of life.

4.6 Conclusions

In this research line, we have proposed new hybrid algorithms combining IRACE with two evolutionary algorithms: GA and DE. These new hybrid algorithms are ideally suited for black-box numerical optimization problems under uncertainty, by using evolutionary operators designed for numerical optimization to generate better solutions, while handling uncertainty by means of the elitist racing strategy in IRACE. We have used these hybrid algorithms (IRACE+DE and IRACE+GA), IRACE, a GA, and a DE, to solve the TLS problem using the real instance of Malaga, Spain, and the SUMO traffic simulator to evaluate the solutions. The results obtained in the experiments confirm the competitiveness of the hybridization strategy. Both hybrid algorithms offer better results than GA (60% of the time) and DE (70% of the time) on realistic traffic scenarios. Particularly, IRACE+DE returns the best results during the testing, being also competitive during the training. Besides, we have seen the impact that the solutions would have on the city. Our hybridization strategies obtain the best results in travel times, fuel consumption, CO_2 emissions, etc. These results reinforce our algorithmic proposal and show the efficiency that IRACE+DE and IRACE+GA obtain when solving a real-world problem.

Chapter 5

Analysis of the Effect of Delivery Vehicles on Traffic

5.1 Motivation

There is a noticeable increment in the number and duration of trips that citizens have to take nowadays (TNS Opinion & Social, 2013), especially because urban infrastructures are not scaling properly. Several traffic jams are consequence of private vehicles sharing streets with services for distribution of goods, deliveries, etc. Furthermore, the need of cargo space makes those services to use small trucks to perform their commercial activities (Visser, Nemoto, and Browne, 2014), which represent an increment not only in the street space usage but also in the pollutant emissions.

Our proposal consists of studying different configurations of road traffic in a realistic scenario of Malaga, to better know how greenhouse gas emissions, Travel Time (TT), and fuel consumption change. Common sense would suggest that reducing the number of Heavy Duty Vehicles (HDV) in the city's streets and increment the Light Duty Vehicles (LDV) should be the right thing to do. We wished to check if it is so, when the cargo capacity is a constraint to be kept, as well as potential traffic jams are taken into account.

We wished to go further in our study and analyze not only the optimal vehicle proportion (instead of banning some types) but also how this proportion affects Travel Time, gas emissions, and fuel consumption. We have not considered hierarchical traffic or public transport, and driven distances are also out of our analysis. We are using a case study based on public open data, a model closer to reality than the previous studies.

After this study, city authorities would be capable of deciding the best strategy to apply when the pollutant levels are high or if they want to foster fuel saving or shorter TT.

The rest of this chapter is organized as follows. The problem description and our proposal are discussed in sections 5.2 and 5.3, respectively. Section 5.4 presents the characteristics of the real scenarios analyzed. Section 5.5 focuses on the numerical study and the discussion of the results. And, finally, in Section 5.6 the conclusions are outlined.

5.2 Problem Description

In this chapter, we present a new strategy to reduce TT, gas emissions, and fuel consumption in the city by using a multiobjective evolutionary strategy. We start from the real number of vehicles measured in the city, their proportions and routes, and calculate new vehicle proportions according to an evolutionary algorithm, to improve metrics, without losing the observed cargo capacity.

Formally, let $\vec{v} = (v_s, v_{mv}, v_{fsv}, v_t, v_m)$ be a vector containing the number of vehicles in the actual city (sedans, minivans, full-size vans, trucks and motorbikes) obtained from the proportions sampled during one hour. We assumed that only 20% of sedans ($v'_s = 0.2 \cdot v_s$) and motorbikes ($v'_m = 0.2 \cdot v_m$) are used for delivering goods. We also assumed that all HDV are used as cargo duty vehicles. So, we defined $\vec{v}' = (v'_s, v_{mv}, v_{fsv}, v_t, v'_m)$ as the vector with the number of cargo vehicles in the city.

According to the number of cargo vehicles $\vec{v}'$ and its average cargo capacity $\vec{t} = (t'_s, t_{mv}, t_{fsv}, t_t, t'_m)$, we can calculate the cargo capacity available in the real city during our study time as $T = \vec{v}' \cdot \vec{t}$.

Our objective is to obtain a Pareto set (Ehrgott, 2005) of N vectors $\vec{v}'^{\star}_j = (v'^{\star}_{s,j}, v^{\star}_{mv,j}, v^{\star}_{fsv,j}, v^{\star}_{t,j}, v'^{\star}_{m,j}), j \in N$ which contains different optimal solutions, minimizing TT, emissions, and fuel consumption in the city subject to:

$$T^{\star}_j = \sum_i t_{ij} \cdot v'^{\star}_{ij} \quad \geq \quad T, \quad \forall j \in N \tag{5.1}$$

The set of vectors $\vec{v}'^{\star}_j$ represents the number of vehicles delivering goods, while $\vec{v}^{\star}_j$ represents the total number of vehicles in the city.

A set of solutions would represent different key indicators for new policies (restrictions, tax reductions, etc.) to be applied to the road traffic in the city to foster shorter TT, less gas emissions, and saving fuel.

5.3 Solving the Problem

We are looking for the best proportion of vehicle types in the city so as to optimize different aspects of the entire road traffic in the city. Modifying the number and type of vehicles used in transportation of goods while keeping the total demanded cargo will allow us to change the whole traffic characteristics to improve the quality of life of drivers and citizens. We use a realistic scenario of Malaga featuring different vehicle proportions (Table 5.1) to be optimized. Using each new configuration we calculate the amount of cargo $T^{\star}$ for this new scenario (subject to the restriction in Equation (5.1)).

Figure 5.1 shows the architecture of our proposal. The algorithm calculates the optimal proportions of vehicles evaluating its individuals by using

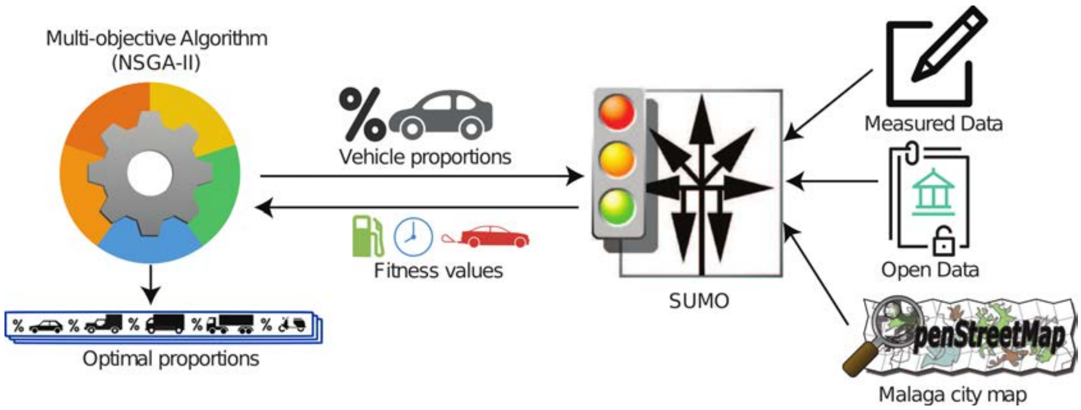


FIGURE 5.1: System architecture of the proposed vehicle proportion optimizer system.

the SUMO traffic simulator (Krajzewicz et al., 2012b). This evaluation comprises a realistic city model made of data measured *in situ*, open data published by the municipality, and the small version of the Malaga city map. The different parts of the architecture are explained as follows.

5.3.1 Solution Encoding

We have encoded each solution $\vec{x} = (x_s, x_{mv}, x_{fsv}, x_t, x_m)$ as a vector of real numbers in which each component is the proportion of sedans, minivans, full-size vans, trucks, and motorcycles intended to transport goods. However, a gas oil engine does not have the same fuel consumption as a gasoline engine, nor the same amount of gas emissions. Because of these differences, and to get closer to reality, it is interesting to take into account the proportions of engine types existing in the car fleet when we are going to evaluate a solution. Before evaluating a solution $\vec{x}$ we transform it into more detailed proportions $\vec{x}^\star$ according to their different characteristics of fuel and gas emissions. These proportions will be used by our evaluation function. We have two possibilities for fuel: gas oil and gasoline, and we have different gas emission classes according to the European emission standard (Eur. Com., 2018). Below we describe the steps to obtain $\vec{x}^\star$:

Step 1 Calculate the amount of cargo vehicles $N_c = T/(\sum_i^{|\vec{x}|} x_i \cdot t_i)$ that we need to supply the tonnage demand T (constraint).

Step 2 Compute the total amount of vehicles $N^\star = N_p + N_c$, being $N_p = v_s + v_m$ the number of vehicles intended for private use (constraint).

Step 3 Get the correction factor ϕ in the number of vehicles $\phi = N^\star/N$, being N the total amount of vehicles in the base solution (field sampling).

Step 4 For each proportion, we correct its value for the new total of vehicles: $x_i' = x_i \cdot N_c/N^\star$, obtaining the new vector $\vec{x}'$.

Step 5 Calculate the new proportion of private use vehicles (sedans and motorcycles) and add it to the solution $x_j' = x_j' + v_j/N^\star, j \in \{s, m\}$.

Step 6 According to the proportions of each vehicle classes in Table 5.1 we calculate the extended solution $\vec{x}^{\star} = \{x' \cdot f \cdot e | \forall x' \in \vec{x}', f \in Fuels, e \in Emissions\}$ where *Fuels* is the proportion of gasoline and gas oil of each vehicle, and *Emissions* the proportion of engine according to the data published in DGT, 2017.

Step 7 Return the extended solution $\vec{x}^{\star}$ and the factor ϕ.

5.3.2 Evaluation Function

We evaluate the quality of each solution, making use of the SUMO traffic microsimulator (Krajzewicz et al., 2012b). Each solution to be evaluated consists of an extended solution $\vec{x}^{\star}$ which are transformed from an algorithm's solution $\vec{x}$ as we have previously described. Additionally, we use an increment factor ϕ to increase (or decrease) the total number of vehicles N (measured number of vehicles) to be consistent with the global vehicle proportions. Note that the realistic scenario to be optimized has a factor $\phi = 1.00$.

Also, since Malaga has a large number of vehicles, solutions which notably increase this number could end in several traffic jams with many vehicles enqueued, waiting to enter the city after the analysis ends. To prevent that, we have calculated the proportion of vehicles entering the city in our realistic scenario (84%) and slightly penalized with the term k those configurations under this threshold. Concretely, we have set a threshold $\theta = 0.8$ to ensure that n vehicles enter the simulation, allowing up to 20% of vehicles to wait in the queue when the analysis time ends.

We show in Equation (5.2) the fitness function used in the optimization process and in Equation (5.3) the penalty term.

$$\vec{\mathrm{f}}(\vec{x}) = \left(k + \frac{1}{n}\right) \cdot \sum_{i=1}^{n} (travel\ time(x_i), emissions(x_i), fuel(x_i)), \quad (5.2)$$

$$k = \begin{cases} 0 & \text{if } \frac{n}{N} \geq \theta, \\ \frac{100}{n} \cdot (\theta - \frac{n}{N}) & \text{otherwise.} \end{cases} \quad (5.3)$$

5.3.3 Algorithm

In order to solve our optimization problem, we use a well-known multi-objective metaheuristic algorithm: the Non-dominated Sorting Genetic Algorithm II (NSGA-II) proposed by Deb et al. (2002a). Our goal is not to research in the algorithm itself, but to make a first model of this real problem and see if we can create an intelligent advisor for city managers. As described above, each individual is represented as a vector of real numbers. This allows us to use simple and fast operators as the following ones:

- Crossover: Simulated binary crossover with probability 0.9.
- Mutation: Polynomial mutation with probability 0.25.

- Selection: Same selection applied in Deb et al. (2002a).
- Replacement: Elitist without including repeated individuals.

To avoid unfeasible solutions, before evaluating each solution, we normalize it and then calculate $\vec{x}^{\star}$ and ϕ. We also performed 200 evaluations of the algorithm with a population of 48 individuals (the initial population was randomly generated). These values were selected after a preliminary study in which we tested two population sizes: 24 and 48 and two maximum numbers of evaluations: 100 and 200. Then, we selected the best configuration to maximize the diversity of the calculated Pareto set, taking into account the limitation of time (the mobility scenarios require long computation times).

5.4 Case Study

We used in the study the small map of Malaga. The city map was imported from OpenStreetMap into the SUMO traffic microsimulator. This allows us to work with a real scenario, e.g., streets, traffic lights, left turns, and roundabouts. From our observations, we have defined five types of vehicle for representing the road traffic in the streets of Malaga. The average characteristics of vehicles according to the manufacturers are shown in Table 5.1. The vehicle distribution was obtained by counting and classifying the type of vehicles at four different locations in the city. We observed that 68.9% of vehicles are sedans, 6.4% are minivans, 7.0% are full-size vans, 2.9% are trucks, and 14.9% are motorcycles. Some types are divided into gasoline and gas oil variants (Table 5.1) as stated by the data for Andalusia (DGT, 2017) (the region of Spain where Malaga belongs) published by the Spanish authorities (*Dirección General de Tráfico*), and into their equivalent emission classes in SUMO that is the HBEFA3 (Hausberger et al., 2009).

Using the data published by the local council of Malaga (Ayto. de Malaga, 2017) corresponding to 23 measurement points (red numbers in previously presented Figure 2.3) and the Flow Generator Algorithm (Stolfi and Alba, 2015b), we have obtained the average traffic per hour for working days in the third quarter of 2015. The Flow Generator Algorithm assigns vehicles to the traffic flows generated by the program DUARUTER included in the SUMO software package, and adjusts its number and routes in the simulation map to the values measured by the 23 real sensors in the city. After this process, we ended up with a scenario, consisting of 10,438 vehicles (7,193 sedans, 616 minivans, 768 full-size vans, 297 trucks, and 1,564 motorbikes), to be optimized with the aim of finding the best policies for reducing TT, gas emissions and fuel consumption.

Each vehicle type has an assigned capacity t obtained from standard commercial models of vehicles as shown in Table 5.1. By multiplying the number of vehicles n by their capacity t, we obtained a total cargo of 4,101 tons, which is the lower bound T of the problem (Equation (5.1)). Each solution generated by our algorithm must be able to provide the total of vehicles needed to delivery this tonnes of goods, so that the city is not losing cargo capacity.

TABLE 5.1: Characteristics of the vehicles in our case study, the observed proportions, the cargo availability, and the individual and total cargo capacity.

Type	Accel. (m/s^2)	Decel. (m/s^2)	Length (m)	Max.Speed (m/s)	Rate (%)
Sedan	0.720	12.341	4.500	25.25	68.9
Minivan	0.720	12.341	4.500	25.25	6.4
Full-size van	0.720	12.341	4.878	25.25	7.0
Truck	0.263	3.838	7.820	16.67	2.9
Motorcycle	0.460	8.147	2.200	16.67	14.9
Type	**Gasoline (%)**	**Gas Oil (%)**	**Cargo (%)**	**Capacity t (ton)**	**Total T (ton)**
Sedan	44.86	55.15	20	0.20	288
Minivan	10.88	89.12	100	1,00	616
Full-size van	10.88	89.12	100	2.00	1,536
Truck	0.00	100.00	100	5.50	1,633
Motorcycle	100.00	0.00	20	0.09	28

5.5 Results

After presenting our problem we are going to study the different solutions obtained by the algorithm and compare our improvements on the city with other strategies carried out by traffic managers for reducing pollution.

5.5.1 Pollutants Correlations

Since SUMO provides different types of pollutants (CO_2, CO, HC, etc.), we wished to select the ones with the lowest correlations to TT and fuel consumption. After carrying out the simulation and measuring these emissions we calculated the Pearson correlation coefficients (see Table 5.2) where hydrocarbons (HC) presented the lowest correlation with TT (0.23) and fuel consumption (0.40). Hence, we have chosen HC as a gas emission measure in our experiments.

5.5.2 Solution Analysis

Having defined the three objectives to be minimized, we analyzed the data obtained from 30 independent runs. Table 5.3 shows different quality indicators (Riquelme, Von Lücken, and Baran, 2015) of each of the Pareto sets returned by our algorithm: Hypervolume (HV), ϵ-indicator, Generational Distance (GD), and Inverse Generational Distance (IGD). HV values of each instance were similar to each other, so we could infer that the different executions obtained a similar global quality. For the other indicators, the instance with the largest HV (run 27) was used as the reference front. The rest of them

TABLE 5.2: Pearson correlation coefficients.

	TT	Dist.	CO_2	CO	HC	NO_x	PM_x	Fuel
TT	1.00	0.70	0.51	0.38	0.23	0.25	0.25	0.53
Distance	0.70	1.00	0.39	0.19	0.14	0.15	0.18	0.41
CO_2	0.51	0.39	1.00	0.29	0.38	0.81	0.51	1.00
CO	0.38	0.19	0.29	1.00	0.67	-0.13	-0.05	0.35
HC	0.23	0.14	0.38	0.67	1.00	0.25	0.30	0.40
NO_x	0.25	0.15	0.81	-0.13	0.25	1.00	0.80	0.77
PM_x	0.25	0.18	0.51	-0.05	0.30	0.80	1.00	0.48
Fuel	0.53	0.41	1.00	0.35	0.40	0.77	0.48	1.00

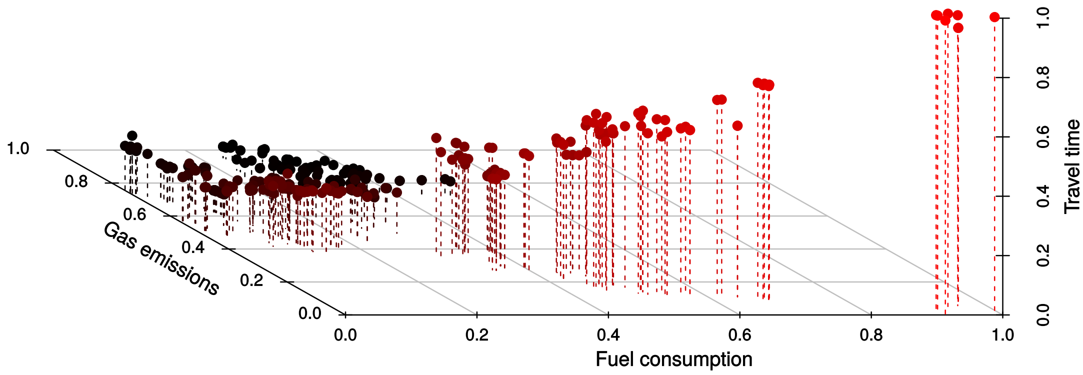


FIGURE 5.2: 25%-attainment surface. Fitness values are scaled to the [0,1] range.

(ϵ-indicator, GD, and IGD) had small values compared to the scale of fitness values. In short, the average HV is 95,736, the standard deviation is 6,550, their average ϵ-indicator is 12.6 and the deviation is 6.3. In general, all these quality indicators showed that the different instances obtain Pareto sets with little variation between them. This similarity between quality indicators is a good advantage for the algorithm as it is stable in the search for solutions.

We studied the different solutions using the attainment surfaces (Fonseca and Fleming, 1996) technique. We selected the 25%-attainment surface to ensure the quality of the selected solutions (Figure 5.2). By analyzing the individuals in this Pareto front, we were able to spot the differences in the number of vehicles in each class (Figure 5.3).

The most used cargo vehicles were vans: 1,419 minivans (median) and 1,332 full-size vans. Their speed and cargo capacity thus are an ideal choice for delivery goods throughout the city. However, HDV, like trucks, and also motorcycles were not good options, as the former are too slow and pollutant and the latter have too little cargo capacity. Sedans had a greater presence than motorcycles and trucks, but were still overshadowed by the utility of vans for delivering goods.

TABLE 5.3: Quality indicators of each algorithm's run. The best result in each indicator is marked in bold.

Run	HV	ϵ-indicator	GD	IGD
1	88,918.1	16.3	13.2	9.2
2	93,400.2	14.0	8.1	7.0
3	89,931.2	24.2	2.8	13.8
4	95,003.0	12.5	8.0	6.8
5	91,301.1	14.1	6.9	7.9
6	86,087.9	21.2	2.6	12.0
7	101,238.9	5.3	4.3	3.5
8	96,478.4	11.1	8.4	4.9
9	97,950.9	7.6	5.5	5.2
10	92,891.3	21.8	2.7	14.2
11	102,349.8	12.5	2.8	8.2
12	100,904.0	6.2	4.2	3.5
13	93,092.9	15.0	10.8	9.9
14	95,836.9	9.9	3.6	7.6
15	93,791.1	21.4	3.3	13.0
16	93,736.7	11.0	8.3	6.0
17	93,629.9	11.2	7.1	7.2
18	101,677.1	5.9	3.8	4.0
19	106,831.4	4.1	3.2	**2.7**
20	108,745.5	**3.8**	2.9	2.9
21	94,511.5	13.6	5.3	6.7
22	92,807.9	21.5	3.0	12.0
23	90,134.2	16.7	7.3	8.6
24	106,488.1	5.5	6.9	3.7
25	95,875.0	9.3	5.0	6.1
26	93,543.0	10.5	7.2	6.9
27	**108,879.1**	0.0	0.0	0.0
28	92,451.3	12.3	7.3	7.4
29	80,521.4	16.3	5.8	9.5
30	93,063.1	22.8	**1.8**	11.0

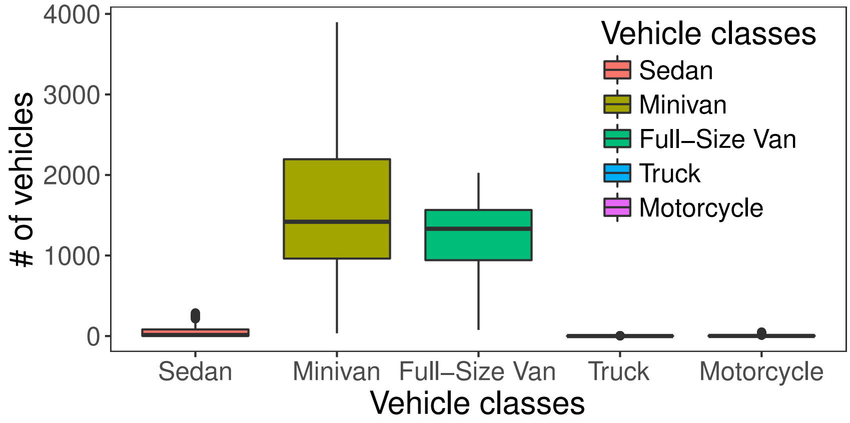


FIGURE 5.3: Number of cargo vehicles in the 25%-attainment surface analysis.

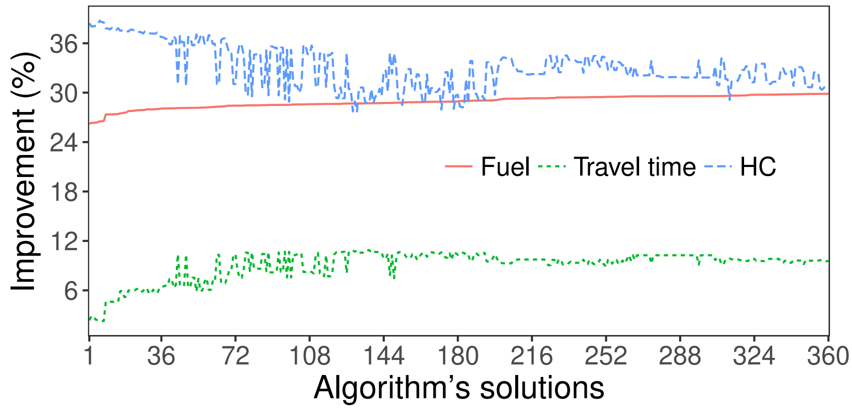


FIGURE 5.4: Improvements observed in the 25%-attainment surface analysis.

5.5.3 Solution Improvement

Next, after discussing the solutions obtained by the algorithm, we will analyze the fitness values of each one of them. Figure 5.4 shows the percentage improvement of each solution in the approximated Pareto front with respect to the current situation (scenario) in the city of Malaga. Although TT was improved by 10%, fuel and all the emissions (not only HC) were considerably better than the ones measured in Malaga. This is good news for the city's environment, as not only there were fewer emissions of polluting gases, but also fewer fossil fuels were consumed.

5.5.4 Comparing with other Strategies

This new way of improvement in the transport of goods is different from previous strategies applied by the city managers for reducing the carbon footprint and getting more fluid traffic on the roads (Mahmod et al., 2013). We compared our solution with other methods used in the cities in Table 5.4, which shows the fitness value of each objective (TT, gas emissions, and fuel)

TABLE 5.4: Fitness values in the different strategies.

Strategies	TT	Gas emissions	Fuel
NSGA-II Min	**633.77**	**434.15**	**406.20**
NSGA-II Max	**698.66**	**523.08**	**434.06**
Malaga	711.12	708.68	579.19
Limit 30 km/h	2,048.74	1,412.43	1,089.75
Limit 70 km/h	709.47	621.21	499.71

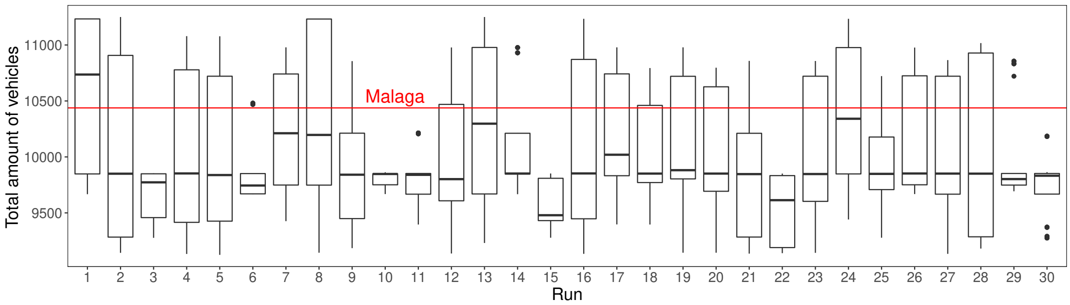


FIGURE 5.5: Total number of vehicles in each run.

obtained by our algorithm (minimum and maximum values found among all the Pareto sets).

Even in the worst case, our proposal achieved better fitness (434.06, 698.66, 523.08) than the rest of the strategies (30% improvement on average). These results show that employing intelligent techniques can help city managers to discover new ways of improving traffic flows and reducing air pollution.

We also compare the number of vehicles in each solution found with those currently moving in Malaga. Figure 5.5 shows the number of vehicles of each algorithm's run. According to our case study, the red line marks how many vehicles (10,438) were in Malaga. We can see that the total number of vehicles is lower than in Malaga (74.09% of solutions are under the red line). But, this does not mean that more trucks need to be added instead of LDV, as we saw in the 25%-attainment surface. The original, realistic scenario is close to the maximum capacity of vehicles that fits into the analyzed area of the city without producing traffic jams. Consequently, our solutions do not only have fewer emissions while improving the efficiency of transport operators but also citizens and drivers enjoy less congestion on the roads.

5.5.5 City Improvements

The global improvements achieved on Malaga are shown in Tables 5.5 and 5.6. Being Malaga our realistic case study, we have improved all the metrics, except the distance (in which case the algorithm is in the second position), using the algorithm. However, we report only the best solutions for each objective. We can see that the lowest TT are obtained when there are 1,734 full-size vans as they are faster than trucks their emissions are lower. The less emitting solution is the one that uses 3,193 minivans for delivering cargo

TABLE 5.5: Improvements of metrics when using the analyzed strategies in Malaga and the number of vehicles of each type.

Strategy	TT (s)	CO (g)	CO_2 (kg)	HC (g)	NO_x (g)	PM_x (g)	Fuel (l)	Distance (m)
Malaga	696.3	19,439.8	1,270.0	626.7	5,493.1	191.0	513.8	5,552.0
Limit 30 km/h	912.7	20,686.9	1,198.8	629.2	5,102.4	178.0	485.5	**5,261.4**
Limit 70 km/h	709.5	18,926.5	1,234.9	621.2	5,257.4	183.3	499.7	5,527.5
NSGA-II TT	**633.8**	16,205.3	1,014.5	504.8	**3,624.3**	**156.6**	412.9	5,605.5
NSGA-II HC	694.5	**15,069.5**	1,055.0	**434.1**	4,007.5	162.1	425.6	5,532.9
NSGA-II Fuel	642.9	15,440.8	**1,000.5**	491.7	3,731.0	165.0	**406.2**	5,593.7

TABLE 5.6: Number of vehicles of each type when using the analyzed strategies in Malaga.

Strategy	Sedan	Minivan	Full-size van	Truck	Motor-cycles	Total
Malaga	6,095	521	649	254	1,307	8,826
Limit 30 km/h	5,655	484	600	237	1,224	8,200
Limit 70 km/h	6,004	512	643	246	1,302	8,707
NSGA-II TT	5,118	109	1,734	1	1,084	8,046
NSGA-II HC	5,034	3,193	76	1	1,011	9,315
NSGA-II Fuel	5,027	1,274	1,153	0	1,093	8,547

instead of the other vehicles. However, for some gases, the other solutions also work very well. Finally, citizens would save more fuel in a more equilibrate distribution of both van types. The algorithm has banned trucks from the city as they are large, slow, and pollutant. Additionally, we present the 30 and 70 km/h metrics and conclude that their improvements are marginal despite being commonly used by city councils when the emissions are high.

5.6 Conclusions

After studying how changing the proportions of HDV and LDV affects the metrics in the city, we have obtained results that show that the number of trucks should be kept at a minimum inside the city. However, due to the limited capacity of the city's streets, the number of LDV vehicles cannot be considerably increased as this makes traffic jams very likely to occur. The multi-objective algorithm was capable of identifying this restriction and obtained solutions where using minivans and full-size vans for delivering goods is advisable.

It does not mean that companies have to sell all their trucks, but HDVs should be used as freight transport by highways and then use vans for local delivery. All this information would be beneficial for city managers. These results would serve as goals for creating municipal strategies to promote the well-being of drivers, workers, and citizens.

Chapter 6

Optimizing the Location of Public Bicycle Stations

6.1 Motivation

In recent years there has been an increase in the options for citizens to move around the city. One of the reasons is the appearance of new transport models such as car-sharing applications, transportation network companies, electric cars, etc. In this chapter, we will focus on bicycle-sharing systems, which present multiple problems to be optimized: how many bicycles to place in each station, the routes to transport bike lots from one station to another, load balancing at the stations, selecting where to place the stations, etc. Most of them have been well studied in the scientific literature (Singhvi et al., 2015; Hu and Liu, 2014; Chira et al., 2014; Chen, Liu, and Liu, 2018). To support these systems, we will focus on the problem of allocating stations where users can pick up/deposit public bicycles. Also, this problem is less studied than the others but is equally important. These systems base their models only on topological aspects. We propose a change of perspective, focusing on how the population is distributed (customers of the system) and the use it makes of the system, instead of how the city is shaped.

There are multiple problems in the scientific literature for the localization of resources (Drezner and Hamacher, 2001): Quadratic assignment, p-center, p-mean, etc. One of the most popular location problems is the p-median problem (Daskin and Maass, 2015). It consists of finding the location of a set of facilities to minimize the distance from a set of clients to their nearest facility. This problem arises especially when deciding the location of the infrastructure of a city, a needed task in today's Smart City. We will formulate the problem of allocating bicycle stations as a p-median problem, and using Open Data; we will find locations that suit the citizens of Malaga.

The scientific contributions presented in this chapter are:

- We have used data from the public bicycle-sharing system provided by the municipality of Malaga, Spain
- We have considered real population data, geographic locations of stations, city maps, etc. We have also included information on the use that citizens make of the current system (bicycle collections and deposits).

- We have used five different algorithms to solve the problem: Iterated Local Search (ILS), Simulated Annealing (SA), Variable Neighborhood Search (VNS), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO).
- We have configured all these algorithms using IRACE (López-Ibáñez et al., 2016), in order to optimize the parameters for each of them. We select different operators (general and specific) for each algorithm.
- We have not just been looking for a ranking of algorithm performances, but also for finding out which models better approach the current needs of users. We have also compared our results using statistical tests.
- We have carried out an in-depth analysis of the evolution of fitness during the execution of the algorithms and how we can improve the current system of Malaga by adding more stations.

The sections are organized as follows: Section 6.2 presents the formulation of the p-median problem. Section 6.3 models the problem of location of bicycle stations and the various realistic data used in this study. Section 6.4 describes the parameters of the algorithms. Section 6.5 analyze the experimentation and conclude in Section 6.6.

6.2 Background

The p-median problem is one of the most-studied NP-hard discrete location problems (Dantrakul, Likasiri, and Pongvuthithum, 2014; Megiddot and Supowits, 1984). It can be formulated as follows: Given a set of customers N and a set of facilities F, the p-median problem seeks to allocate p facilities in F while minimizing the weighted sum of the distances between the customers and their closest facility. Formally, the problem is defined as:

$$\min \sum_{i=1}^{|N|} w_i \min_{j \in L} d_{ij}, \tag{6.1}$$

where $L \subseteq F$, with $|L| = p$ is the set of potential locations for the facilities, w_i is the weight of the customer i, and d_{ij} is the distance between customer i and facility j. If we have $w_i = 1$, $i = 1, \ldots, |N|$, that is, all weights are one, we have the unweighted version of the problem.

In our study, the customers are the citizens and the facilities are the stations, that can be placed on different street segments across the city. Any street segment between two intersections can be a station location. We selected this formulation of the problem for two reasons: (i) the model is easy to understand and implement, and (ii) the p-median problem is a classical location problem that has been well-studied in the scientific literature (Dantrakul, Likasiri, and Pongvuthithum, 2014). From the allocation of bicycle stations perspective, the p-median may serve as a good model to identify interesting places and relevant distributions of the stations across an urban area.

There are other formulations to the location of bicycle stations problem requiring different information, as the proposed by Chen et al. (2015), Liu et al. (2015), and Kloimüllner and Raidl (2017). The p-median problem requires little information (only the distance matrix and the weights). Although this may seem like a limitation, it is relatively simple to add additional information, either by pre-processing the weights or distances, (e.g., by considering the slopes of the streets) or by adding terms in the formulation itself (e.g., by adding capacity information related to the slots in each bicycle parking site).

6.3 Problem Definition

Our problem is to find the optimal positions for public bicycle stations so that citizens have to walk as little as possible to them. Instead of using synthetic benchmark instances, we took advantage of many Open Data offered by the city council and specialized websites. In this way, we worked with a realistic instance as the city of Malaga. Specifically, we work with the medium size map presented in Section 2.4.2.

In addition, the formulation of our problem as a p-median needs two sets of points: customers and facilities. The customers are the citizens. We have chosen the centers of the different neighborhoods ($|N| = 363$) of the city as the most natural way of grouping the population (we excluded six neighborhoods on the outskirts of Malaga). The center of each these neighborhood is the position assigned to each customer in our problem. The facilities are the bicycle stations. We selected all possible street segments as potential locations to the facilities. In total, there are $|F| = 33,550$ potential locations.

Figure 2.4 shows the layout of the city, the neighborhood centers (orange points), and the 23 public bicycle stations (blue points). We observe a more significant number of stations in the city's central area and only a few stations on the outskirts. This distribution does not have to be helpful for the citizens since many of the population and points of interest (universities, sports centers, schools, etc.) are on the outskirts. Because of that, a study carried out using intelligent techniques is beneficial not only when we need to install infrastructure for citizens but also in assessing the quality of existing solutions.

In our study, we have considered different values of distances and weights used in the p-median problem. In this way, we will be able to study how these data sets affect and with which data we obtain results that adapt better to the city. We have used two types of distance d between customers and stations: Straight-line Euclidean distance and the shortest path through the city streets (calculated using Dijkstra's algorithm); we refer to this last one as *real distance*. In addition, we have considered three types of weights $w_i, i \in N$, that a final application could consider modeling the demand on each customer:

- Uniform: $w_i = 1$, is the basic option for solving the standard p-median. It deals with the whole population fairly.

- Citizens: $w_i = c_i$, where c_i is the number of residents in the i-th neighborhood of the city. Neighborhoods with more people should have a closer station.

- Demand: $w_i = p_i$, where p_i is an estimate of demand in the i-th neighborhood of the city. We have modeled this estimate based on the data of the use of the city's current public bicycle system. The Open Data about the stations allowed us to calculate the activity of each station as $act_j = mean(o_j)/mean(s_j)$, where o_j are the number of slots occupied and s_j the total number of slots in the j-th station. o_j and s_j were obtained by sampling them each minute for a whole week (from 7th October 2018 to 13th October 2018) and we compute the value of act_j in each station. Finally, we calculated the demand as $p_i = c_i \cdot act_{near_i}$, where c_i is the number of inhabitants of the i-th neighborhood and act_{near_i} is the activity of the nearest station to the i-th neighborhood.

6.4 Algorithms

We have used three metaheuristic algorithms based on trajectory search: (i) Iterated Local Search (ILS) (Lourenço, Martin, and Stützle, 2003), (ii) Simulated Annealing (SA) (Kirkpatrick, Gelatt, and Vecchi, 1983), and (iii) Variable Neighborhood Search (VNS) (Mladenović and Hansen, 1997); and two population-based algorithms: (iv) Genetic Algorithm (GA) (Whitley, 1994) and (v) Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995).

With these algorithms, we want to review the effect that local searches and shaking processes, commonly used in trajectory algorithms, have on this type of problem. We also want to check if a population-based strategy returns better results than a trajectory-based one. Besides, each of these algorithms has multiple choices of parameters and operators. We have implemented several alternatives for each to give a broad view. We selected the best configuration to solve this problem among the possible parameter setting, as we will see in Section 6.5.1. Next, we give a brief description of each parameter and some abbreviations that we will use in the experimental section.

General parameters

- Number of iterations (iter). The total number of iterations of the algorithms can be calculated according to one of the following equations: $N \cdot p/5, 100 \cdot p, max(2 \cdot N, 100)$, and one million of iterations (`1M`). Where N is the number of customers, and p is the number of facilities to select. These options are in several scientific articles.

- Domain model. We use two different models to assign to each facility a set of d nearby facilities (this model is used in the shaking process): The facilities with the closest Euclidean distance (`NEAR`) and the closest by quadrants (`QUAD`). The latter model is calculated as follows: Each facility's geographic space is divided into four quadrants using their coordinates (latitude and longitude). Then iteratively, we select the nearest point in each quadrant (if there are no more points, skip the quadrant) until we have picked d facilities. The `QUAD` model provides more uniform coverage of the geographical space at the cost of allowing more remote facilities. Figure 6.1 shows an example of these two models.

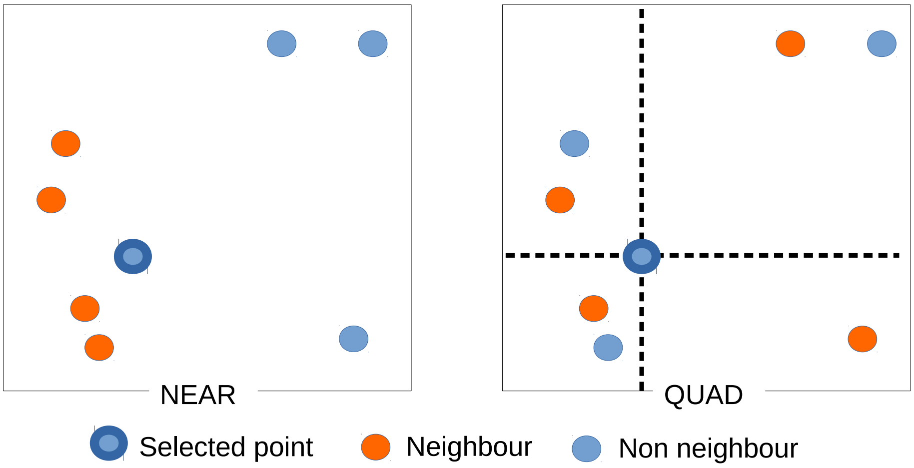


FIGURE 6.1: Graphic example of the two domain models, $d = 4$.

Specific parameters of trajectory-based algorithms (ILS, SA, and VNS)

- Local search. We can apply a local search procedure to the initial solution ($\mathtt{ls}_1$) and in each iteration ($\mathtt{ls}_2$). The local search algorithms are: Fast Interchange (FI) (Whitaker, 1983), Improved Alternate Approach (IALT) (Drezner et al., 2016) with its parameter L, and Improvement Algorithm (IMP) (Drezner et al., 2015) with its parameter p.
- Initial solution. We have four procedures to generate the initial solution: Random (RAND), the constructive algorithm proposed in Drezner et al. (2015) (START), and the best of 100 random solutions (100RAND).
- Shake. We use three different permutation algorithms to change k facilities of a solution: Either we exchange the facilities with randomly chosen other ones in the set of close facilities defined in the domain model (CLOSE), we select the new facilities at random from all (RAND), or we do not perform any change of facilities to the solution (NONE).

Specific parameter of population-based algorithms (GA and PSO)

- Population size μ: It is between 8 and 30.

GA parameters

- Number of offspring λ. It can take values between 1 and μ.
- Selection. To generate a new individual, the algorithm selects two solutions from the population. This selection can be random (RAND), selecting the two solutions with the best fitness value (BETTER), or the two with the worst fitness values (WORSE).
- Crossover. This operator combines two solutions (parents) to generate a new solution (offspring). It can be one of the following strategies: the merging process used in the GA described in Drezner et al. (2015) (MERGING), one-point crossover (1POINT), the proposed crossing operator in Colmenar, Martí, and Duarte (2018) (CUPCAP), or a copy of one of the parent solutions randomly selected (1RANDPARENT).

- Mutation. We use the same operations, `CLOSE` and `RAND`, as we do in the shaking process of the trajectory-based algorithms. We apply this operator with a probability in the interval $[0, 1]$.
- Replacement. We can perform the replacement in the population according to one of these strategies: (μ, λ) or $(\mu + \lambda)$.

ILS parameters

- Number of perturbations (npert). Between 1 and 20.

SA parameters

- Next solution. We use two strategies to select how many facilities have to be shaked: The same value as the iteration index i (`SEQ`) or the one used in the distribution based variable neighborhood search proposed in Drezner et al. (2015) (`DVNS`). The latter defines a probability distribution to avoid following the same pattern (sequential case) every time.
- Initial temperature (t_0). It is between 1 and 100.
- Cooling strategy. The strategy can be lineal (`LIN`), exponential (`EXP`), or the iteration index as the new temperature (`NONE`).

PSO parameters

- Inertia parameter ω, in the interval $[0, 1]$.
- Cognitive parameter ϕ_p, in the interval $[0, 1]$.
- Social parameter ϕ_g, in the interval $[0, 1]$.

VNS parameters

- Next solution. The same as in the SA.
- Exploration size (k_{max}). Number of neighbors that will be explored from the same solution between $[1, 50]$.
- Acceptance strategy. We have three options: Elitist, walk, or elitist but allowing to choose worse solutions according to a certain probability (acceptProb) between $[0, 1]$.
- Number of consecutive non-optimal solutions allowed (maxAttempts). Between 1 and 100

6.5 Experimental Results

In this section, after calculating the best parameter settings of each algorithm, we will study: which algorithm works better to solve the problem, which model is better adapted to the reality of the city, and how our system can assist to the municipalities to improve the expansion of their bicycle-sharing systems. To carry out these studies, we took into account different data sets to the distances and weights (as we explained before in Section 6.3) and we used the same number of stations $p = 23$ that are currently placed in Malaga. The following are the results for each of the studies listed above.

6.5.1 Parameter Settings

Before starting with the results, we study the best parameter configuration for each algorithm. Each algorithm has multiple variants in terms of operators and parameter settings. Since manual multi-factorial complete experimental design is very time-consuming, we use an automatic algorithm configuration stage through the use of the iterated racing procedure implemented by the IRACE package (López-Ibáñez et al., 2016). This produces a statistically best parameterized algorithm for our experiments since IRACE does this job for us. We used a budget (number of configurations to explore) of 5,000. As training instances, we chose instances of the TSPLIB library between 1,000 and 10,000 points. These are instances typically used in this type of problems (Crainic et al., 2004; Irawan, Salhi, and Scaparra, 2014; Rabie, El-Khodary, and Tharwat, 2014). The locations of the cities are the positions of the customers and the possible facilities. We used values of $p \in \{10, 20, 30, 40, 50\}$ for each training instance. For the configuration process, we included several components and parameters to overview the algorithms. These parameters were described earlier in Section 6.4. Table 6.1 shows the final parameters selected by IRACE for each algorithm.

6.5.2 Algorithm Comparison

First, we will analyze the fitness values obtained by our optimization algorithms in each scenario (type of distances and weights used). Figure 6.2 shows for each of our realistic scenarios the fitness values in the 30 independent executions. In general, GA gets better fitness values than the other ones. This highlights the power and versatility of these algorithms, a feature that has helped them gain popularity in part. The second one is the VNS, which corroborates the popularity that this algorithm has when solving this type of localization problem. GA shows a 5-16% improvement over the second-best algorithm. The following positions correspond to ILS, SA, and finally, PSO, in all cases. It is interesting to note that while one population-based algorithm obtains the best results, GA, another one obtains the worst results, PSO. The operators used in the GA may be better suited to solve this problem than those in the PSO. On the other hand, if we analyze the three trajectory-based algorithms between them, we see the importance of using a powerful local search to improve the performance of the algorithms.

To contrast these results, we will analyze whether the differences between the algorithms are statistically significant. We have performed a Wilcoxon Sum Rank Test, with Bonferroni correction, between each pair of data sets. There are differences between the different combinations of algorithms and weights, p-value < 0.01 in all cases. Despite this, and given the results, we can conclude that GA is, in general, the best option for solving the problem of locating bicycle stations.

TABLE 6.1: Best parameter configuration found by IRACE.

Parameter	GA	ILS	PSO	SA	VNS
iter	$N{\cdot}p/5$	1M	$max(2N, 100)$	$max(2N, 100)$	1M
$\mathtt{ls}_1$	—	$\mathtt{IALT}_{L=9}$	—	$\mathtt{IALT}_{L=16}$	$\mathtt{IMP}_{p=2}$
domain m.	QUAD	NEAR	—	QUAD	NEAR
d	20	29	—	34	49
generation	RAND	100RAND	RAND	100RAND	100RAND
shake	—	CLOSE	—	RAND	CLOSE
npert	—	6	—	—	—
next	—	—	—	SEQ	SEQ
t_0	—	—	—	4.45	—
cooling	—	—	—	$\mathtt{EXP}_{opt=0.39}$	—
$\mathtt{ls}_2$	—	—	—	—	$\mathtt{IALT}_{L=16}$
k_{max}	—	—	—	—	5
maxAttempts	—	—	—	—	57
accept	—	—	—	—	ELITIST
μ	14	—	29	—	—
λ	10	—	—	—	—
selection	BETTERS	—	—	—	—
crossover	CUPCAP	—	—	—	—
mutation	RAND	—	—	—	—
mut. prob.	0.91	—	—	—	—
replacement	(μ, λ)	—	—	—	—
ω	—	—	0.0001	—	—
ϕ_p	—	—	0.53	—	—
ϕ_g	—	—	0.69	—	—

6.5.3 Improvement over the Current System

After analyzing the algorithms among themselves, we will compare them against the real solution in Malaga. We compute the objective function for the actual location of the 23 stations in Malaga (evaluated according to each scenario). Figure 6.3 shows the empirical cumulative distribution of the percentage of improvement in the objective function value in each scenario and algorithm to Malaga. In general, unweighted and Euclidean distances get the best improvement rates over the actual city, around 60% in distance reduction. If we do not consider the city's most and least populated areas, the stations will spread across the city. We reach more areas near the borders with this behavior, reducing the distance to each district. However, the Malaga system is considered the most densely populated area of the city, where it collects the highest number of bicycle stations. When we use information about the population, we obtain substantial improvements of about 50% over the base scenario. An interesting aspect is that the type of distances or weights does not particularly affect the overall behavior of the algorithms. In other words, there does not seem to be a unique key parameter to improve or give realism to the system, but a combination of them.

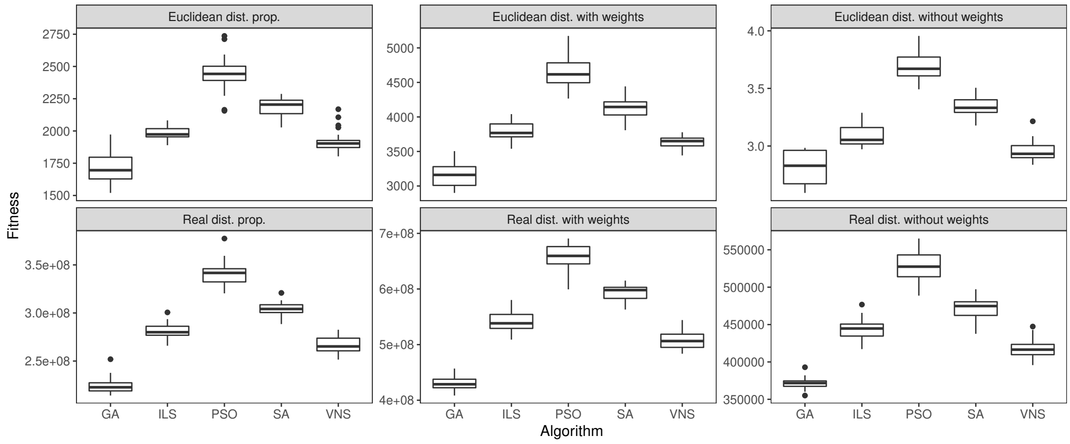


FIGURE 6.2: Fitness values in each scenario and algorithm.

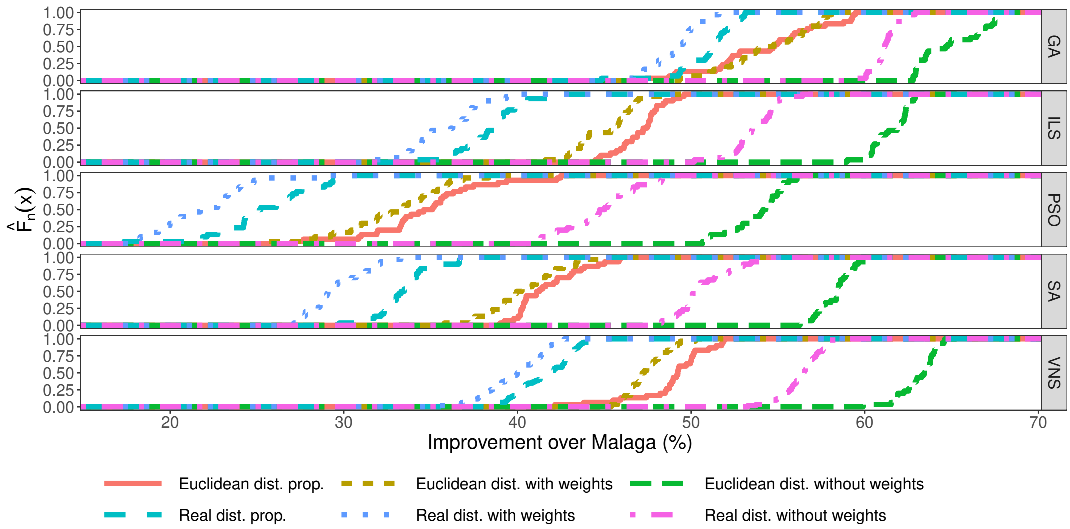
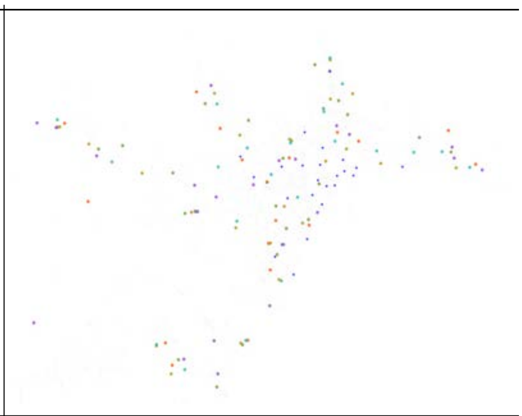


FIGURE 6.3: Empirical cumulative distribution of the percentage of improvement of our solutions in each algorithm and scenario, compared to the current location of bicycle stations in Malaga. We evaluate the current solution of Malaga in each scenario using the specific type of distance and weight.

6.5.4 Geographical Spread

After discussing the quality of the results, we will study their spread around the city. Table 6.2 shows the geographical positions of the stations in each instance of the problem. We offer the solution with the lowest fitness value for each of the combinations of distances and weights. As expected, the weighted versions allocate more stations in the city's central area, the more populated ones. But, they still offer more uniform coverage of the main neighborhoods in the city than Malaga's system so that each citizen has a nearby station to use the service. There are usually no more than one or two stations in each neighborhood furthest from the city center in the weighted versions. The quality of the service to the citizens will be improved if the city council expends a small amount on infrastructure located in the outskirts.

TABLE 6.2: Geographical distributions of the best solutions found by each algorithm in each of the six scenarios.

	Euclidean distance	Real distance
$w_i = 1$		
$w_i = c_i$		
$w_i = p_i$		
	● Current stations ● ILS ● SA ● VNS	

On the other hand, the city's current solution uses many stations to cover the coastline. However, our algorithms have shown that other avenues in the city are more suitable for citizens.

6.5.5 Model Comparison

Next, we will compare the different scenarios with each other to discover which of them suits the needs of the citizens. We have calculated the average distance a person must travel from home to the nearest station (real distance)

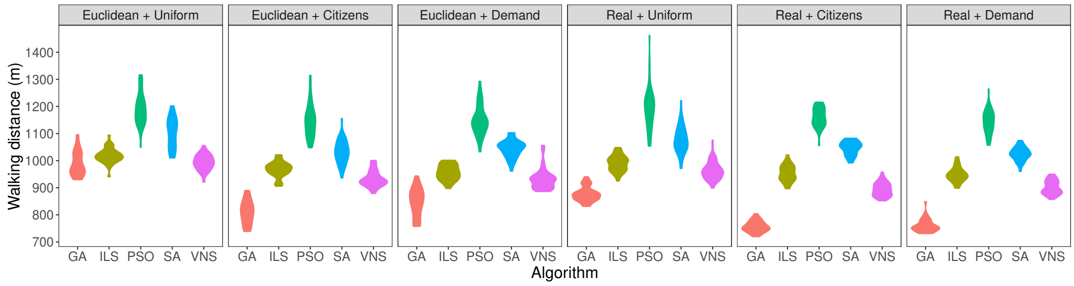


FIGURE 6.4: Average distance (m) walked by the citizens to reach its nearest station. Each algorithm is shown in each combination of distances and weighting model.

for each solution. Table 6.3 presents these distances in meters. GA obtains the shortest distances with the weights of the number of citizens. This result validates our way of weighing the citizens according to the use of the stations. However, there is not much difference between the weight models $w = c$ and $w = p$ (even $w = p$ is better in some cases).

We see in Figure 6.4 that the distributions for these weighting models are very similar. Interestingly, in algorithms that use local searches, in half the cases, two modes are observed instead of one. This means that there are (at least) two local minima in the search space. Algorithms based on exploitation rather than exploration have more difficulties getting out of the local minima solutions. On the other hand, GA presents very pronounced modes in the scenarios with real distances, which denotes a remarkable convergence of the algorithm. In general, all algorithms have a lower variability and get better results when using real distance instead of Euclidean. All algorithms start the optimization process with similar quality solutions, so the use of distances closer to reality makes the optimization process faster.

To verify these results, we analyze the obtained solutions by the algorithms during their executions. Figure 6.5 shows the mean distance walked by a citizen to the nearest station in the solutions obtained during the execution. We have represented the average value obtained by $\pm 25\%$ of the solutions (smooth area) in each iteration. We observe that the algorithms find a better solution faster when using real distances. The quality of the solutions is more stable than in the other cases when we use a uniform weighting model. Interestingly, GA has significant variability when using Euclidean distance, obtaining even worse solutions than trajectory-based algorithms. On the other hand, when we calculate the solutions using the real distances, the competitive advantage that GA has becomes manifest. Notably, this difference is emphasized when we consider information about the citizens.

6.5.6 Increase in the Number of Stations

Finally, as a logistical issue, we will study how the city councils could improve the current bicycle shared system of the city. We added stations to the current solution in Malaga using the GA algorithm, which is the best algorithm in our experimentation. We have increased the 23 actual stations to

TABLE 6.3: Distance traveled per inhabitant to the nearest station in the different scenarios, evaluated as real distance and citizen weighting model ($w_i = c_i$). The minimum values in each column are marked in bold.

Algorithm	Distance	Weigh model	Min	Max	Mean	Median
GA	Euclidean	Uniform	929.51	1,096.05	991.22	976.19
		Citizens	738.55	890.20	810.11	815.00
		Demand	757.17	943.67	846.50	854.50
	Real	Uniform	830.95	940.85	876.80	875.57
		Citizens	**719.16**	**804.55**	**756.48**	754.84
		Demand	730.77	848.97	760.73	**754.62**
ILS	Euclidean	Uniform	940.23	1,094.37	1,018.23	1,017.88
		Citizens	905.33	1,017.62	967.54	968.99
		Demand	896.95	1,001.83	954.95	956.22
	Real	Uniform	924.32	1,048.65	989.06	991.78
		Citizens	896.22	1,021.60	950.88	945.79
		Demand	898.08	1,007.24	947.37	945.20
PSO	Euclidean	Uniform	1,049.01	1,316.87	1,192.22	1,184.86
		Citizens	1,047.30	1,315.17	1,149.32	1,147.55
		Demand	1,032.34	1,293.27	1,150.80	1,142.29
	Real	Uniform	1,053.65	1,462.62	1,198.15	1200.42
		Citizens	1,055.30	1,216.14	1,160.79	1,161.21
		Demand	1,056.60	1,264.73	1,143.84	1,145.35
SA	Euclidean	Uniform	1,009.83	1,202.68	1,102.89	1,113.43
		Citizens	935.82	1,155.31	1,033.12	1,032.55
		Demand	961.25	1,103.43	1,043.84	1,049.32
	Real	Uniform	970.81	1,221.67	1,079.69	1,071.86
		Citizens	991.36	1,083.02	1,046.17	1,052.98
		Demand	959.92	1,074.92	1,024.44	1,023.37
VNS	Euclidean	Uniform	920.54	1,056.32	993.77	995.12
		Citizens	878.62	1,001.12	933.26	924.58
		Demand	885.85	1,056.88	933.54	931.91
	Real	Uniform	898.52	1075.87	964.38	958.81
		Citizens	851.56	958.11	894.28	891.61
		Demand	856.47	950.29	900.94	893.72

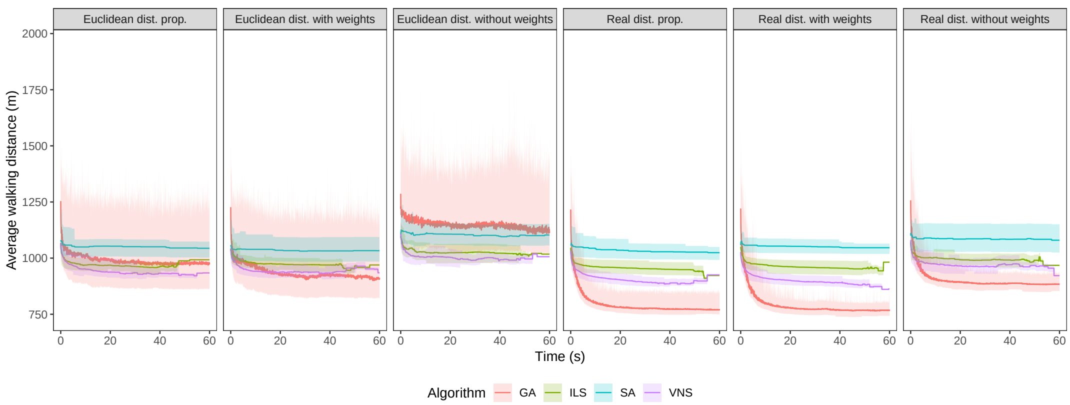


FIGURE 6.5: Smooth function over the average distance walked by a citizen to its nearest station in the solution obtained in each iteration of the four betters algorithms. We decided to exclude PSO in order to clarify the figure, as its lowest value among all scenarios was 1,018.24 m.

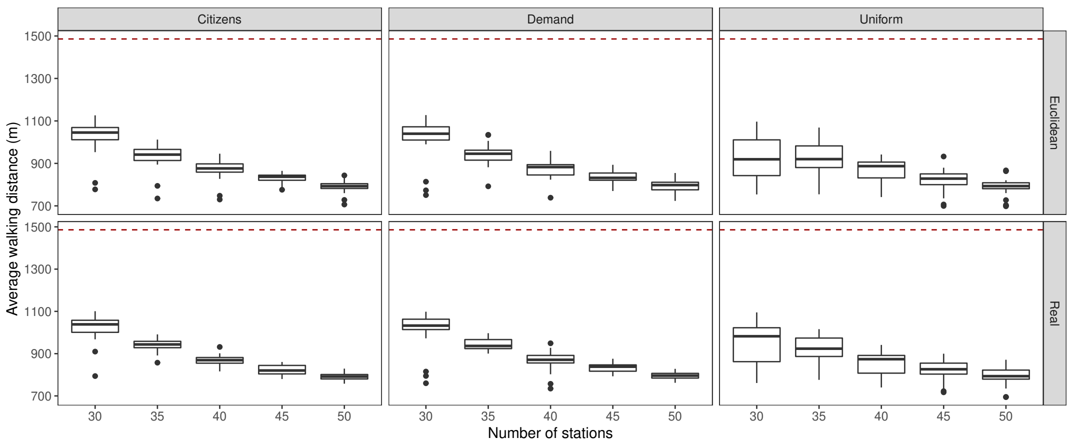


FIGURE 6.6: Average distance (m) walked by the citizens to reach its nearest station when we increase the number of stations. The dashed red line indicates the average distance in the city's current system, 1,485.97 m.

reach 30, 35, 40, 45, and 50 stations and, thus, be able to verify the improvement obtained. Figure 6.6 shows the mean distance values obtained in each of our instances. We get a decrease of 33% on average in the walking distance by adding seven stations (smartly located). This is an excellent improvement for the citizens with a reduced cost in infrastructure. Furthermore, if we increase the number of stations to 50, the distance covered is reduced to 53%, which means that a citizen has a station only 738 meters away. This expense in infrastructure allows public bicycles to be a daily aspect for the population with the direct benefit of reducing the use of private vehicles and endorsing a healthy lifestyle.

6.6 Conclusions

In this chapter, we remarked how solving the classic location problem *p*-median we can solve the problem of where to place bicycle stations. We analyzed the state-of-the-art and selected five algorithms commonly used to solve the *p*-median and others NP-hard problem: GA, ILS, PSO, SA, and VNS. Each of them was tunned using IRACE over a set of benchmark instances. In addition, because we are solving a problem for the cities, we used several realistic datasets from the city of Malaga. In conclusion, GA has been the algorithm that returned the best results, opposite to the state-of-the-art. We also obtain improvements of 50% of quality when we apply metaheuristic techniques compared to the assignment made by an expert of the city council. Also, we check that real distance allows us to find better locations for the bicycle stations and the number of inhabitants per neighborhood enables us to accommodate the results to the citizen's needs. Overall, our proposal can be used to improve the city's current system by adding stations, obtaining improvements between 33% and 53%.

Chapter 7

Multiobjective Electric Vehicle Charging Station Location

7.1 Motivation

In recent years, sustainability has become an important goal for the industry, academia, and society. Society has steadily moved towards ecological awareness, adapting their lifestyles to promote environmental initiatives, cleaner means of production, and environmentally friendly energy sources. One of the most critical changes in current activities concerns urban mobility, where an effective transition towards inclusive, efficient, and low-carbon means of transport is experienced (Coffman, Bernstein, and Wee, 2016). Electric mobility provides citizens with a cleaner and safer way of getting around without gas emissions.

Electric vehicles (i.e., electric cars, scooters, and motorbikes) are powered by an electric motor charged using energy from the electricity grid. They have shown rapid and sudden growth and expansion, as they are preferred mobility options for the younger generation. Consequently, electric vehicles have a relevant socio-economic impact (Kumar and Alok, 2020). In addition, electric vehicles have a higher overall efficiency than combustion engine vehicles. They save (on average) 40% of energy while contributing to reducing gas emissions when the electricity used to charge the vehicles is obtained from green sources (Hipogrosso and Nesmachnow, 2020).

When considering the deployment of electric vehicles in large cities, a relevant logistical problem arises, similar to other location problems related to public services in the context of Smart City (Fabbiani et al., 2018; Massobrio et al., 2017; Rossit, Toutouh, and Nesmachnow, 2020). One of the most relevant sub-problems is the effective and efficient location of charging points for electric vehicles (Falchetta and Noussan, 2021). The main objective of this problem is to provide a good quality of service to citizens while keeping costs reasonable for the city administration.

Different authors have tried to address this siting problem from different perspectives. One of the common ways in the literature is to address the location of charging stations as Integer Linear Programming or Mixed-Integer Linear Programming problems. The researchers usually used these methods to maximize the economic profits of installing new charging stations (Brandstätter, Kahr, and Leitner, 2017; Lin et al., 2018; Çalık and Fortz, 2017), minimizing the total walking distance according to parking patterns estimated using realistic urban data (Chen and Nie, 2013), or maximizing coverage to improve demand (Frade et al., 2011; Wagner, Götzinger, and Neumann, 2013). Some authors have relied on open data to improve the quality of service offered to citizens by taking into consideration the energetic constraints of the area (Risso et al., 2021). The cited articles work with a mono-objective view of the problem. However, in the real world, the location of charging stations have different objectives.

This research presents a new multiobjective variation of the problem of locating Electric Vehicle Charging Stations (EVCS) in a city known as the Multiobjective Electric Vehicle Charging Stations Location (MO-EVCS-L) problem. Two objectives are considered: maximizing the quality of service of the charging station network to the citizens and minimizing the deployment and installation cost of the new stations. Both objectives need data about different aspects of a city: locations of neighborhoods, streets, etc., energy data such as types of charging stations or energetic capability of electrical substations, and economic data such as installation costs. To obtain this data, different open data sources were used. This research uses a realistic case study defined in the city of Malaga, Spain.

The main contributions of this chapter are:

- Defining and formulating a new realistic multiobjective problem for locating electric vehicle charging station on a city scale, taking into account the quality of service, power restrictions, and deployment costs.
- Proposing two multiobjective metaheuristics to address the proposed problem.
- Devising specific evolutionary operators.
- Handling the problem on a realistic instance defined using real-world data from Open Data sources.

The rest of the chapter is organized as follows: Section 7.2 presents the problem addressed in this study. Section 7.3 describes the algorithmic contribution. The experimental setup and evaluation are reported in Sections 7.4 and 7.5. Finally, Section 7.6 presents the main conclusions of this research.

7.2 Multiobjective Electric Vehicle Charging Stations Location Problem

The problem considered in this chapter aims to select the best locations of EVCSs to maximize the quality of service provided to users and simultaneously take into account the infrastructure deployment costs. Different types of EVCS are considered. The kind of EVCSs determines the number of users that served per unit of time, the charging time, and the installation costs.

The quality of service is evaluated according to the users that can be served (each EVCS may attend the users that live within a defined *service distance*), the charging time, and the citizens that any charging station does not serve. The deployment cost has two main components: the infrastructure installation expenses for the charging equipment and construction of a new station and the cost of connecting the installed station to the power grid.

The two discussed objectives (quality of service and deployment costs) are in conflict because installing charging stations close to the residences of all tentative clients would require a significant investment, which in turn may not produce in adequate expected revenues for the institutions in charge of the management of the electric vehicle charging system. Thus, to assist the decision-makers, the main research outcome of the addressed problem is to provide solutions (i.e., EVCS locations) that adequately sample the different trade-offs between these problem objectives.

7.2.1 Mathematical Formulation

The mathematical formulation of the addressed optimization problem is defined considering the following elements:

- A set $S = \{s_1, \ldots, s_M\}$ of candidate road segments for installing EVCSs. Each road segment s_i can be the location of only one charging station.
- A set $C = \{c_1, \ldots, c_N\}$ of the locations of the tentative users. Nearby locations are grouped in clusters, as usual in the related literature. The number of clients to serve at each cluster c is u_c. The distance from the cluster c to the charging station $s \in S$ is $dc_{c,s}$. A cluster of clients c is served by the charging station located in s if the $dc_{c,s}$ distance is lower or equal to the Ds_s *service distance*, i.e., $dc_{c,s} \leq Ds_s$. $C_s \subseteq C$ represents the set of clusters of clients served by station installed in s, and $NC \subseteq C$ defines the set of clusters not served by any charging station.
- A set $E = \{e_1, \ldots, e_T\}$ of electrical substations that supply the power to the charging stations. Due to the power distribution restrictions, each electrical substation e can serve electricity only to a given subset of candidate road segments enclosed in a given city area $A_e \subset S$, named electrical substation influence area. The distance (in meters) from a charging station $s \in A_e$ to the electrical substation e that provide it is de_s. The maximum distance between substation e in E and its assigned charging station s is De. In turn, the maximum power allocated for EVCSs, i.e., the electricity distributed by substation e that can be used to feed the electric vehicle charging stations is limited by MP_e.
- A set $J = \{j_1, \ldots, j_H\}$ of EVCS types. Each type has its own charging time ct_j, equipment and building cost cc_j, connection to the grid cost cg_j, and required electric power from the electrical substations ep_j. In the segments where no charging station is located, we assume $cc_0 = cg_0 = 0$ and $ct_0 = \infty$.

After the definition of these several variables, we can present the mathematical modeling of the MO-EVCS-L problem:

$$\max \sum_{s\in S}\left(\sum_{c\in C_s}\frac{u_c}{ct_{x_s}}\right) - \sum_{nc\in NC} u_{nc} \tag{7.1}$$

$$\min \sum_{s\in S}(de_s \cdot cg_{x_s} + cc_{x_s}) \tag{7.2}$$

subject to

$$SC = \bigcup_{s\in S} C_s \qquad \forall\, s \in S:\ x_s \neq 0 \tag{7.3}$$

$$NC = C \setminus SC \tag{7.4}$$

$$\sum_{s\in S} y_{e,s} \cdot ep_{x_s} \leq MP_e \qquad \forall\, e \in E \tag{7.5}$$

where x_s is an integer variable, $x_s = j_i$ when a charging station of type $j_i \in J$ is installed in segment s, and $x_s = 0$ otherwise; and $y_{e,s}$ is a binary variable, $y_{e,s} = 1$ if the electrical substation e is feeding the charging station located in s and 0 otherwise.

The quality of service provided by the deployed infrastructure is defined in Equation (7.1) as the sum of the service provided by each charging station installed in $s \in S$ to the subset C_s of clusters within its *service distance* minus the number of clients in clusters not served by any charging station NC. NC, defined in Equation (7.4), is the complementary set of the set of all clusters served by all charging stations, see Equation (7.3). The service provided by the EVCS deployed in s is proportional to the number of citizens in the cluster u_c and inversely proportional to the time required to charge an electric vehicle ct_{x_s}. The quality of service is proposed to be maximized.

The installation cost of a EVCS considers the sum of the infrastructure cost cc_{x_s} and the cost of connecting the station to its electrical substation, defined in Equation (7.2). The budget required to connect the charging station to the electrical substation is proportional to the distance between them de_s and the cost of wiring cg_{x_s}. The cost is proposed to be minimized.

Regarding the problem constraints, The constraint in Equation (7.5) guarantees that the total power consumption of all charging stations that are fed by a given electrical substation is lower or equal than MPe.

7.3 Algorithms

We need to use a multiobjective algorithm to deal with a multiobjective problem. Two well-known Multiobjective Evolutionary Algorithm (MOEA)s are applied in this study: Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Strength Pareto Evolutionary Algorithm 2 (SPEA2). Both algorithms have been successfully applied to many problems in different application areas in Smart City (Rossit, Toutouh, and Nesmachnow, 2020; Nesmachnow, Rossit, and Toutouh, 2018; Massobrio et al., 2017; Péres et al., 2018a; Toutouh, Rossit, and Nesmachnow, 2020). The proposed NSGA-II and SPEA2 for the locating the EVCS include the main following features:

Solution encoding Solutions are encoded as a vector of integers s which $s_k \in [0, |J|], 1 \leq k \leq M$. Each position in the vector represent a possible location for the charging station and the corresponding integer value on index s_k represents one of the possible electric vehicle charging type, i.e., $j_i \in J$. The special value '0' is used to represent the situation where no charging station is installed in the segment s_k. Figure 7.1 presents an example of solution encoding for a sample scenario with eight tentative locations $\{1, \ldots, 8\}$ and two types of charging stations $\{1, 2\}$.

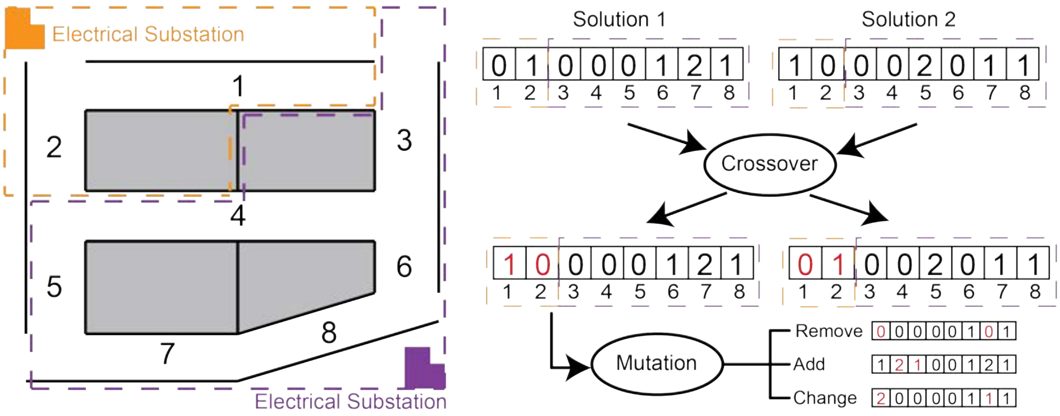


FIGURE 7.1: Example of solution encoding of an scenario with eight possible locations for EVCS and two types of stations, and the evolutionary operators.

Initialization The population is initialized by applying a random procedure that creates feasible solutions. The initialization process iterates over the areas of influence of each electrical power station A_e. For each A_e, it randomly selects a tentative location s_i and adds a randomly chosen EVCS j_k to it. If the power consumption restriction in Equation (7.5) is met, it selects another location in A_e to install a new EVCS. This process is repeated while power consumption restriction is fulfilled.

Evolutionary operators The recombination operator applied is a variation of the standard n-point crossover used over two selected individuals specifically devised to address MO-EVCS-L, named k electrical substation influence area crossover (k-A_eX). It randomly exchanges the deployment configuration of the electrical areas of influence of k power stations (A_e defined in Section 7.2). This operator works as follows: given two parents (individuals), k-A_eX randomly selects k A_e and exchanges between parents the assigned stations in the selected A_e to create two offspring. Figure 7.1 shows an example of two substations $A_1 = \{1, 2\}$ and $A_2 = \{3, 4, 5, 6, 7, 8\}$ which A_1 is crossed between the parent solutions and generate two new offspring. The mutation operator is based on randomly modifying a specific attribute of a randomly selected segment in the A_e of a given individual (i.e., x_s). There are three different potential changes shown in Figure 7.1: *i)* if there is a charging station (i.e., $x_s \neq 0$), the charging station can be removed; *ii)* if there is no any charging station (i.e., $x_s = 0$), a randomly chosen charging station is selected to be added to the represented segment(i.e., x_s is replaced by an integer value uniformly chosen in the range [0,Z-1]); and *iii)* the values of two different attributes x_s and $x_{s'}$ are exchanged with each other regardless of their values. The recombination and mutation operators are applied with probability p_C and p_M, respectively. Figure 7.1 represents both evolutionary operators.

Solution feasibility The restriction defined in Equation (7.5) may not be met after the application of evolutionary operators, i.e., the total power consumption of all charging stations in A_e could be higher than MP_e. Thus an operator is applied to randomly remove charging stations installed in A_e until the restriction is fulfilled.

7.4 Experimental Setup

This section summarizes the methodology applied for the experimental analysis of the proposed MOEAs to address MO-EVCS-L.

7.4.1 Problem Instance

The experimentation is performed over a realistic scenario defined on the medium size Malaga. Around 567,953 citizens spread over 363 neighborhoods live in this city. The road map is composed by 33,550 road segments. Each road can be selected for the placement of a EVCS (as we saw previously in Figure 2.5). The city's electric power is supplied by 14 electrical substations. These electrical substations limits the number of stations that we can install in a specific area. When we install a EVCS we can choose between different types of stations according to the different charging speed. Two different types of charging stations are considered to be installed: fast charging stations (type 1) and super-fast charging stations (type 2). Each one have different energy consumption requirements, installation (equipment/building and connection) costs, and also times for fully charging a standard electric vehicle. Table 7.1 summarizes the main characteristics of both electric vehicle charging station types.

TABLE 7.1: Main features of the considered charging stations.

Type (j)	ct_j	ep_j	cc_j	cg_j
fast (1)	120 minutes	7.4 kW	13,915 €	1.15 €
super-fast (2)	15 minutes	50.0 kW	39,930 €	1.35 €

TABLE 7.2: Relative hypervolumes in the preliminary experiments for each configuration of the algorithms.

(A) NSGA-II

		Mutation		
		1/14	4/14	7/14
Cross.	0.5	**0.950**	0.949	0.914
	0.7	0.915	0.935	0.909
	0.9	0.890	0.942	0.896

(B) SPEA2

		Mutation		
		1/14	4/14	7/14
Cross.	0.5	0.930	**0.975**	0.913
	0.7	0.897	0.956	0.896
	0.9	0.916	0.946	0.918

7.4.2 Parameter Settings and Execution Platform

A set of parametric setting experiments were performed to determine the best parameter values for the proposed MOEAs. The parameter setting analysis were made over the proposed scenario. Both MOEAs apply the same initialization, crossover, and mutation operators. The population size ($\#p$) and the maximum number of generations ($\#g$) were calibrated in preliminary experiments. The analysis confirmed that using $\#p = 20$ and $\#g = 500$ provided a good exploration pattern for both MOEAs. In SPEA2, the size of the elite population was set to 5 individuals, following rules-of-thumb from the related literature (Zitzler, Laumanns, and Thiele, 2001a).

For p_C and p_M, candidate values were $p_C \in \{0.5, 0.7, 0.9\}$ and $p_M \in \{1/14, 4/14, 7/14\}$ (we have 14 zones in one scenario). Each configuration was evaluated over 30 independent executions performed for the proposed MOEAs. The distribution of the relative hypervolume (see Figure 7.2) results obtained using each configuration were analyzed by applying the non-parametric Kruskal-Wallis statistical test to determine the configuration that allowed computing the best results. Thus, for NSGA-II, the most competitive results were achieved with $p_C = 0.5$ and $p_M = 1/14$, and for SPEA2, with $p_C = 0.5$ and $p_M = 4/14$.

7.4.3 Baseline Method

In order to test the effectiveness of our algorithms, an intelligent Random Search (RS) method to get a baseline of solutions is defined. RS generates feasible solutions using the same constructive method applied to generate the initial population in NSGA-II and SPEA2. The method keeps all non-dominated solutions generated during the process. RS iterates generating new solutions until a stop criteria is reached. In this case, it stops after reaching the maximum execution time of the two MOEAs analyzed here.

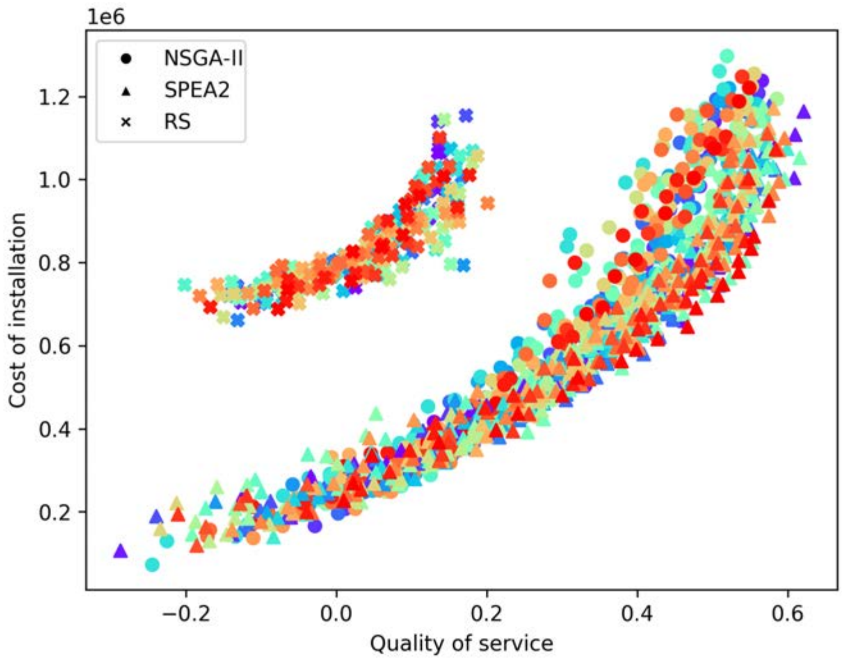


FIGURE 7.2: All computed solutions.

7.5 Experimental Evaluation

This section reports the experimental analysis of the proposed MOEAs to address the real-world case study of MO-EVCS-L.

7.5.1 Multiobjective Optimization Analysis

Figure 7.2 shows the non-dominated solutions computed by each independent execution of the evaluated algorithms. Marker colors represent the results of each independent execution. In turn, Figure 7.3 illustrates the three Pareto fronts computed by NSGA-II, SPEA2, and RS, i.e., all non-dominated solutions computed considering all the 30 independent executions performed by each method. The arrow points in the direction of the *best* solutions.

Figure 7.2 indicates that RS is the least competitive algorithm. The RS set of solutions represent charging covering shorter ranges of quality of service and deployment costs than the MOEAs. All solutions computed by NSGA-II and SPEA2 dominate the RS solutions, i.e., the RS solutions provide less quality of service while requiring higher deployment costs. In turn, NSGA-II and SPEA2 show robustness because the average dispersion of solutions for the same value of each problem objective was below 20% of each of the independent runs.

Results in Figure 7.3 show that the MOEAs can compute accurate solutions, properly sample the Pareto front of the problem, and demonstrate the practical applicability of the proposed approach. For deployment costs lower than 0.5×10^6, both methods present solutions with close trade-offs between quality of service and deployment costs. However, for higher installation

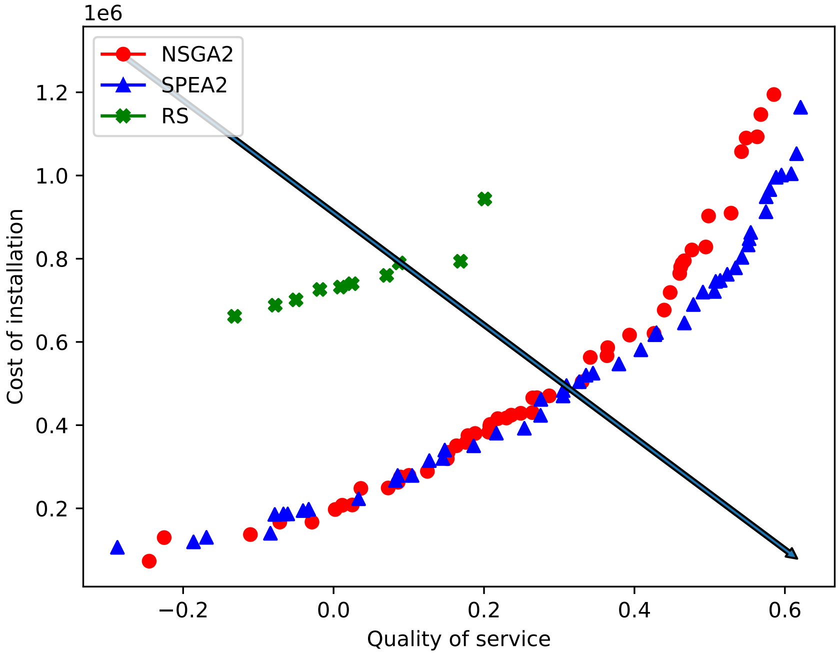


FIGURE 7.3: Pareto fronts of the multiobjective algorithms.

costs, SPEA2 seems to improve over the solutions computed by NSGA-II, i.e., SPEA2 solutions are able to provide a better quality of service at the same installation costs.

Regarding multiobjective optimization metrics, Table 7.3 reports relevant statistical values of the evaluated multiobjective metrics (i.e., minimum, maximum, mean, standard deviation, median and IQR) for the evaluated MOEAs. SPEA2 is the most competitive method among the evaluated ones addressing MO-EVCS-L. SPEA2 presents lower values for the distance-based metrics, i.e., GD, GD^+, IGD, and IGD^+ (see in bold in Table 7.3), which represents that SPEA2 computed solutions are closer to the reference front than the NSGA-II ones. The RHV results of SPEA2 are higher than the NSGA-II ones (see in bold in Table 7.3), which indicates that SPEA2 converged to more competitive solutions than NSGA-II.

Statistical analysis is performed to evaluate the statistical significance of these results. As the distribution of results follow a non-normal distribution, Kruskal-Wallis statistical test is applied. The results confirmed that SPEA2 outperforms NSGA-II with a confidence higher than 99%, i.e., p-values $\ll$ 0.001 for all evaluated metrics. These results imply better convergence towards the Pareto front of the problem and a better coverage of the Pareto space by solutions computed by SPEA2

Table 7.4 reports the values of the multiobjective metrics evaluated for the Pareto front obtained from the solutions provided by NSGA-II and SPEA2. Results in Table 7.4 show that the Pareto front computed by SPEA2 provides the best values for each metric.

TABLE 7.3: Statistics of multiobjective metrics for each run of the executions of each algorithm.

Algor.	Metric	Min	Mean±SD	Median	IQR	Max
NSGA-II	RHV	0.74	0.82±0.03	0.83	0.05	0.88
	GD	6,028.16	9,777.69±2,569.45	9,174.03	2,487.18	17,151.89
	GD^+	0.07	1,853.90±2,396.43	986.55	3,165.70	9,132.95
	IGD	18,213.75	29,024.71±8,099.82	27,602.76	7,603.97	57,134.82
	IGD^+	0.17	13,727.04±9,850.17	12,702.03	7,333.46	44,933.12
SPEA2	RHV	**0.83**	**0.89**±0.03	**0.89**	0.03	**0.95**
	GD	**3,738.85**	**7,460.10**±1,472.98	**7,211.45**	1,496.41	**10,550.35**
	GD^+	**0.02**	**66.54**±179.47	**0.06**	0.02	**712.32**
	IGD	**13,494.63**	**21,287.26**±5,191.87	**20,642.87**	6,307.04	**37,576.88**
	IGD^+	**595.11**	**8,327.67**±6,036.19	**7,301.21**	7,973.87	**26,004.91**

TABLE 7.4: MOEAs metrics for the Pareto front computed by NSGA-II and SPEA2.

Algorithm	RHV	GD	GD^+	IGD	IGD^+
NSGA-II	0.947	5,790.885	590.648	10,131.143	0.031
SPEA2	**0.992**	**1,175.294**	**0.009**	**2,776.893**	**595.026**

7.5.2 Computational Time Evaluation

This section discusses the execution time of the evaluated methods. Table 7.5 shows the relevant statistics of the execution time in seconds of NSGA-II and SPEA2 when addressing the proposed instance of MO-EVCS-L. RS is not included since its stop condition is set as running for the maximum running time of both MOEAs, i.e., 1515 seconds.

Results in Table 7.5 show that there are no significant differences between the execution time required by the evaluated MOEAs. The execution time is between 20 and 25 minutes, which entails a low computational cost because we are dealing with an NP-hard problem.

7.5.3 Comparative Analysis

A few samples of computed solutions are compared in terms of two metrics of quality of service: a) sum of service provided by each station, defined in

TABLE 7.5: NSGA-II and SPEA2 execution times (in seconds).

Algorithm	Minimum	Mean±SD	Median	IQR	Maximum
NSGA-II	1,214.37	1,271.17 ±61.70	1,242.04	61.86	1,515.38
SPEA2	1,214.25	1,272.85 ±49.20	1,260.53	47.11	1,515.96

TABLE 7.6: Quality of service metrics for the representative solution by cost of each algorithm.

Cost	Algorithm	QoS	du	# stations type=1	# stations type=2
50%	NSGA-II	1,204,666	317,563	21	9
	SPEA2	**1,413,229**	**298,201**	19	10
75%	NSGA-II	1,072,286	280,995	39	10
	SPEA2	**1,529,501**	**275,702**	31	13
90%	NSGA-II	1,502,587	**262,766**	43	13
	SPEA2	**1,619,564**	274,091	40	14

Equation (7.6), and b) sum of disconnected users, defined in Equation (7.7), which represents the number of citizens not served by any charging station.

$$QoS = \sum_{s \in S} \left(\sum_{c \in C_s} \frac{u_c}{ct_{x_s}} \right) \tag{7.6}$$

$$du = \sum_{nc \in NC} u_{nc} \tag{7.7}$$

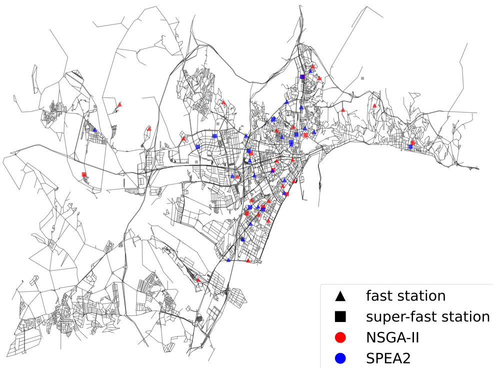
For a fair comparison, the solutions compared are the ones that require the same deployment cost. Three were selected according to a percentage over the maximum deployment cost computed: 50%, 75%, and 90%. Table 7.6 reports the results. Besides, it includes the number of stations of each type installed. Figure 7.4 illustrates the solution in each percentage of cost.

According to the results in Table 7.6, SPEA2 provides the best QoS values for the three evaluated solutions. For the costs of 50% and 75%, SPEA2 leaves fewer users disconnected. However, NSGA-II has fewer citizens that are not served by any charging station for the cost of 90%. Finally, it can be seen that SPEA2 deployments have more EVCS of type 2 than NSGA-II, and NSGA-II installs more EVCS of type 1 than SPEA2. Super-fast charging stations allow eight times more vehicles to be assigned than type 1 stations. If energy constraints permit, it is more beneficial for the public to install super-fast charging stations, even if that means fewer stations.

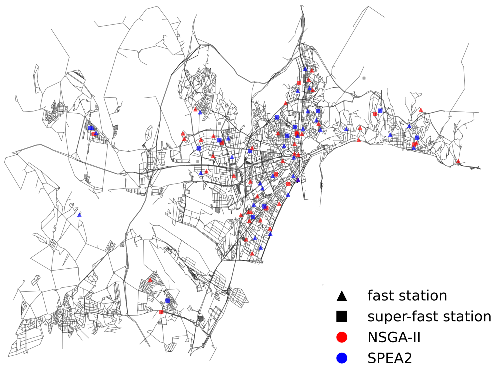
7.6 Conclusions

This chapter presents a multiobjective evolutionary approach to address the problem of locating electric vehicle charging stations in a city, a relevant challenge of the current sustainability and clean mobility concerns.

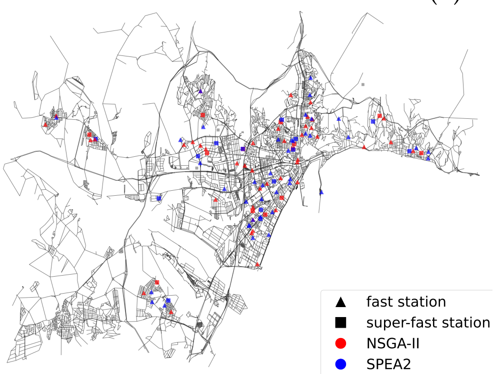
The proposed problem formulation as MO-EVCS-L is more realistic than previous approaches. On the one hand, it considers the two types of users: citizens served by the charging stations and those disconnected from the charging station network (i.e., not attended by any charging station). It is important considering the unserved users because this may make it difficult



(A) cost=50%



(B) cost=75%



(C) cost=90%

FIGURE 7.4: Geographical locations of the solutions computed by NSGA-II and SPEA2 with different percentages of cost.

for these citizens to purchase electric vehicles. On the other hand, it explicitly models real energy supply constraints and deployment costs.

Two variations of MOEAs (NSGA-II and SPEA2) that apply specific evolutionary operators have been proposed to solve MO-EVCS-L. The problem has been solved over a real city-scale scenario, the city of Malaga (Spain). The results obtained show that the SPEA2 is the most competitive approach. However, both MOEAs provide accurate location plans to assist in making decisions on the location of EVCSs taking into account the quality of service and the cost of installation.

Chapter 8

Robust Biobjective Shortest Path Problem

8.1 Motivation

Millions of citizens around the globe have to face every day the problem of deciding the route to follow in their cities to go from an origin (e.g., home) to a destination (e.g., work). Many software tools solve this problem by representing the city as a graph and assigning a cost to every edge (street segment) in the graph. Then, Dijkstra's shortest path algorithm (Dijkstra, 1959) is used to find the route minimizing the cost function. The time to traverse the route is commonly used as cost. Navigator tools like Tomtom, Google Maps, or Waze traditionally do that. Distance is also a common parameter that can be easily minimized using this approach.

However, reducing the travel time (or the distance) is not the only important goal in the real world. Many citizens also want to spend less money on fuel, drive through quiet streets, and, in general, fulfil several of these criteria simultaneously. The shortest path problem that simultaneously deals with two objectives is called a Biobjective Shortest Path (BSP) problem. Sometimes the data required to solve the problem is not reliable or accurate. For example, the time needed to traverse a street depends on the traffic load, the vehicles used, the speed limit, the state of the traffic lights, and even the driver's skills. Often, all of these parameters are not accurately known. For example, a citizen that goes to work usually prefers the fastest route. However, the main roads generally have traffic jams during rush hour, so that the time may be longer than another route with a higher distance but less traffic. Considering the uncertainty in the data is vital to propose realistic solutions. Given a solution, we would like to know how much its quality attributes can change due to the variation of the uncertain data we have about the city. This quality of the solver technique and problem definition is usually called *robustness* (Ben-Tal, El Ghaoui, and Nemirovski, 2009).

In this chapter, we include robustness in our new model of the BSP problem to take into account the uncertainty in the data. We consider two objectives: travel time and the gas emission (CO_2 and NO_x). We select these two goals guided by our present societies and modern needs. We need to waste as little time as possible on traveling and pollution problems in cities affect the health of their citizens, as we mentioned in previous chapters. Our contributions in this research line can be summarized as follows:

- We introduce a robustness model into the definition of the BSP itself. We model an optimization problem that searches for routes with the lowest average travel time and gas emitted, and also for the smallest variability of these two goals. This new formulation including robustness for this problem is not present in the literature, where usually one single robust solution is returned by the algorithms (Chassein and Goerigk, 2015; Hasuike, 2013; Pascoal and Resende, 2014).
- We use explicit multiobjective modeling of the problem along with a shift from fixed parameters to random variables.
- We apply different algorithms to solve the multiobjective problem in order to check their performance. Multiobjective algorithms and weighted sums in single-objective algorithms are the chosen strategies.
- We use a real map with real data in our study. We select the Malaga Province, Spain, as the base map to our experiments. In addition to being a large scenario full of real data, it also represents a case study in a traditional European region, with interesting implications in creating services for tourists and commuters.

We used five algorithms in this study:

- two multiobjective algorithms: NSGA-II and MOEA/D;
- and three exact algorithms: Pulse (Duque, Lozano, and Medaglia, 2015), Dijkstra's algorithm, and A*, the last two without taking into account the robustness.

We also use twenty-nine different instances with real data and present an in-depth analysis of the different results obtained. In addition, this research studies single-objective algorithms to solve multiobjective problems using weighted sums to get an approximation of the Pareto optimal set.

This chapter is organized as follows. Section 8.2 mathematically formulates the BSP problem. Section 8.3 describes our proposed model to the robust BSP problem. Section 8.4 presents the experimentation baseline. Section 8.5 discusses the main results obtained by experimentation. Section 8.6 describes different aspects of robustness related to the present work. Finally, Section 8.7 presents the conclusions of this research.

8.2 Background

Our model treats the robustness of the problem as a variability in the parameters that must be minimized to obtain more robust routes.

Let $G(N, A)$ be a directed graph, where N is the set of nodes, and A the set of edges between nodes, $A \subseteq N \times N$. We define a path p with n nodes as a sequence of nodes $p_i \in N$ with $1 \leq i \leq n$, where $(p_i, p_{i+1}) \in A$ for $1 \leq i \leq n-1$. That is, consecutive nodes in the sequence are adjacent in the graph. We define $\mathcal{P}_{s,e}$ as the set of all possible paths between a start node $s = p_1$ and an end node $e = p_n$. In our problem, the graph represents the road map. The edges are the street segments between every two intersections (nodes of the graph). This is a classic way of modeling a road map (Garaix et al., 2010), which allows us to formalize the real problem, while we keep its main features.

We also define a cost function $C : A \to \mathbb{R}^+$ in the graph G. This function assigns a non-negative number to each edge of the graph. In order to simplify the formulation, we write $C((i,j)) = c_{ij}$ as the cost of the edge (i,j). These costs represent the values of the objectives to be minimized, e.g., the values of travel time and emission in each of the road segments. The definition of the cost function can be extended to a whole path as follows:

$$z(p) = \sum_{(i,j)\in p} c_{ij}, \tag{8.1}$$

where we write $(i,j) \in p$ when nodes i and j are consecutive in path p.

In this chapter, we present a model of robustness for the BSP problem. The BSP problem is an NP-hard multiobjective optimization problem (Serafini, 1987) that searches for the paths between two points in a graph $G(N,A)$ minimizing two objectives simultaneously in the Pareto sense. As we have already explained, this graph problem is analogous to our routing problem. The cost function is defined as $C : A \to \mathbb{R}^+ \times \mathbb{R}^+$, which associates two weights to each edge in the graph. As in the single-objective case, we simplify the formulation by writing $C((i,j)) = c_{ij,k}$ to represent the cost of edge (i,j) in the objective k. We formulate the BSP problem as follows.

Definition 6 (Biobjective Shortest Path Problem) *Given a graph $G(N,A)$ and two nodes in the graph s and e, the biobjective shortest path problem consists in finding the Pareto optimal set of the biobjective problem defined over the objective function:*

$$\mathbf{z}(p) = \sum_{(i,j)\in p} (c_{ij,1}, c_{ij,2}) \tag{8.2}$$

subject to $p \in \mathcal{P}_{s,e}$, that is, only paths from nodes s and e are considered.

Once we have introduced the necessary background for our proposal we can present our robust model for the BSP problem.

8.3 Proposed Robust Model

We call *robustness of a problem* to the characteristic of the problem to deal with inaccuracies in the parameters that define an instance of the problem. The street length, travel times, or the map itself in a routing problem are examples of parameters of a problem. We say that a problem formulation is robust if it considers the inaccuracies in the parameters. Robustness in problem formulations is a crucial characteristic to consider when developing real-world applications. Our way of introducing robustness in the formulation differs from most other models because we consider robustness an additional objective to optimize.

We can consider problem's parameters as another argument of the objective function. The new objective function has the form $f^P : X \times P \to \mathbb{R}^d$, where P is the space of parameters. P is a tuple $P = (P_1,\ P_2\ , \ldots,\ P_m)$, where P_i with $1 \leq i \leq m$ is the domain for the i-th parameter of the problem.

In real-world applications for the cities, we do not have the same control over the environment of the problem as in the lab. The real data usually have inaccuracies, missing information, etc. There are many strategies to deal with these inaccuracies (see Section 8.6). We assume that the parameters are random variables, and we work with two statistics of them: their mean and variance. This is different from previous studies because it considers both the general character of the road and its variation. Given the i-th parameter of the problem with domain $P_i \subseteq \mathbb{R}$, we consider that its value is defined by a random variable with mean μ_i and variance σ_i^2. We say that the parameter is *inaccurate* if $\sigma_i^2 > 0$.

In a Shortest Path problem the cost function $C : A \to \mathbb{R}^+$ is a parameter of the problem that can have uncertainties. Thus, the cost associated to edge (i, j) is modelled with a random variable with mean μ_{ij} and variance σ_{ij}^2. We will assume that random variables associated with different edges are independent, what allows us to sum means and variances to get the mean and variance of the sum of costs. Figure 8.1 shows an example of this model of robustness. Given a graph $G(N, E)$, with $N = \{A, B, \ldots, F\}$, the shortest path between nodes A and F is calculated. In the top network, the shortest route (using the cost of each edge) has a cost of 10. In the bottom network, the variance associated with each edge is introduced (the robust version of the problem). Two routes are relevant in this case. The first is (A, B, D, F) with $\mu = 10$ and $\sigma^2 = 6$. The second one is (A, B, E, F) with $\mu = 12$ and $\sigma^2 = 4$. The first one has a lower mean cost but higher variance, while the second has a lower variance and higher mean cost. We say that the second solution is more robust than the first one since the variation of its cost due to the uncertainties in the parameters of the problem is lower than in the case of the first solution.

In the case of the BSP, we consider two parameters for each edge. These parameters are Travel Time (TT) and gas emission (CO_2 and NO_x). Each of them are characterized by a mean $\mu_{ij,k}$ and a variance $\sigma_{ij,k}^2$, with k being TT for travel time and GAS for gas emission. Our robust formulation for the Robust Biobjective Shortest Path (RBSP) problem has a total of four objectives that are defined as follows:

$$\mathbf{z}^R(p) = \sum_{(i,j)\in p} (\mu_{ij,TT},\ \mu_{ij,GAS},\ \sigma_{ij,TT}^2,\ \sigma_{ij,GAS}^2), \tag{8.3}$$

where $p \in \mathcal{P}_{s,e}$ for a given s and e defined by the instance. In this way, we shift from a basic biobjective problem to another one with four objectives.

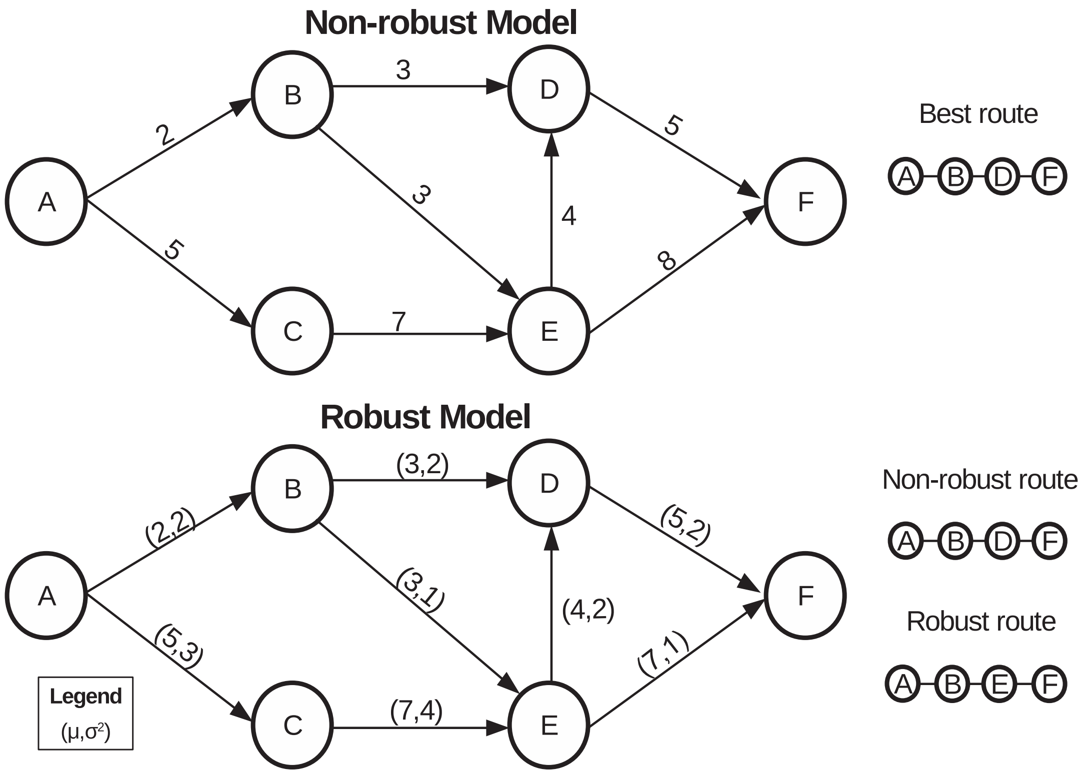


FIGURE 8.1: Standard non-robust model (top) and our proposed robust model (bottom). We also show the best route for the first one and two routes for the second.

This formulation allows us to obtain solutions with different degrees of robustness (higher or lower variance) in both parameters (TT or gas emission). Having one set of solutions is interesting for many types of applications. According to the circumstances, a user may need solutions that focus on some or all of the parameters. For example, a police officer may prefer more stable routes (low variability) in general. Still, it may need to find a faster route at specific times even though it is risky. In short, providing more options to users means that they can choose the most suitable one in each situation.

Figure 8.2 shows some sample Pareto fronts of RBSP and compares it with that of BSP. The solutions and Pareto front on the left considers that $\sigma^2 = 0$ for all the edges and the two parameters. The solutions and Pareto front on the right assume that $\sigma^2 > 0$. The latter could make us find solutions with a low average cost, but with high variability (not very robust). However, if we expand the Pareto front to four dimensions, this will expand our set of non-dominated solutions with those that have low variability in our objectives.

8.4 Experimentation Baseline

In this section, we describe the inputs, obtained from public open data websites, and algorithms used in the experimentation. With this section, we want to remark the applied intention of our work and the usability of our proposal in real-world applications.

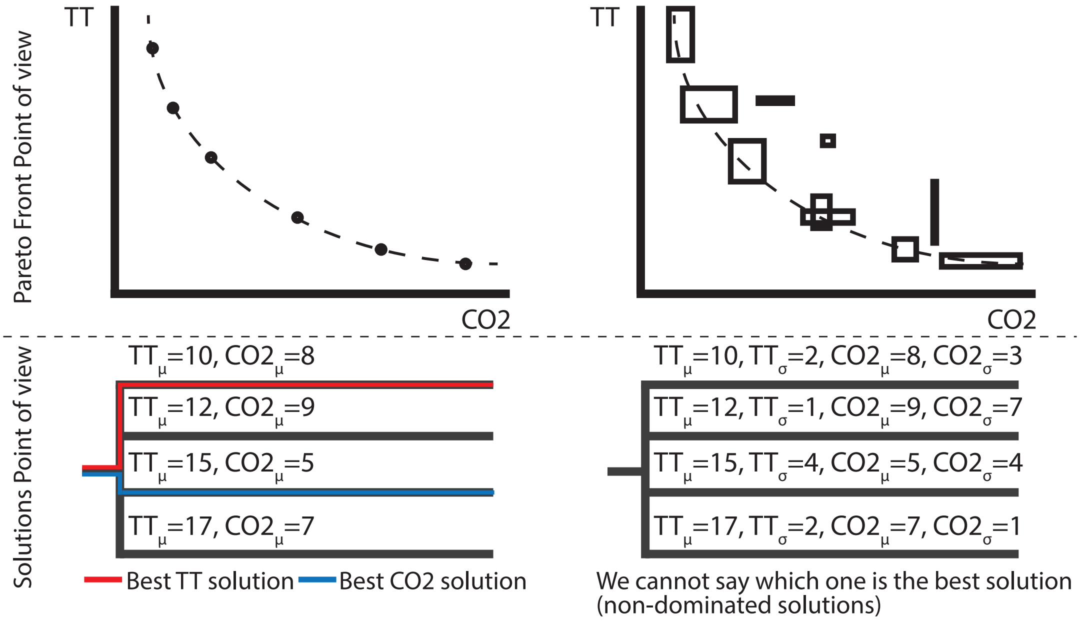


FIGURE 8.2: BSP (left) and RBSP (right) problem models. They are shown from the point of view of the Pareto fronts, as well as the selection of a edge when building the solution (a path).

8.4.1 Real Map

We used the large map of Malaga presented previously in Section 2.4.2. This road map has 10,601 nodes and 118,388 edges. We selected random points (according to a uniform distribution) to serve as the origin and destination nodes for the instances of the experimentation. We generated 29 instances.

8.4.2 Realistic Inaccurate Parameters

As mentioned in Section 8.3, the two uncertain parameters considered in this research are travel time and gas emission. The mean of each parameter is computed for each edge of the graph as follows:

- Travel Time (TT). For each road segment the mean travel time was estimated from the speed and distance obtained from Open Street Maps during the parse process of the source map.
- Gas emission (CO_2 and NO_x). We estimated the gas emitted by the vehicles using the HBEFA model (Hausberger et al., 2009) and the estimated speed obtained from Open Street Maps. This model computes the pollutant emission rate using the speed of the vehicle as follows:

$$h(v) = \max\left(0, c_0 + c_1 v\frac{dv}{dt} + c_2 v\left(\frac{dv}{dt}\right)^2 + c_3 v + c_4 v^2 + c_5 v^3\right) \quad (8.4)$$

where the coefficients c_i with $0 \leq i \leq 5$ depend on the car and the type of gas emission. These coefficients are calculated based on averaging measurements of hundreds of types of vehicles, models, engines, etc.

The mean of the inaccurate parameters considers the real speed limits on roads and a commercial sedan as the vehicle for our study. We selected these specific values as an example of a possible driver of the city. Factors like traffic jams, traffic lights, works, etc., can also affect each parameter. The variances in both parameters were randomly generated. However, our model is prepared to consider the inaccuracies due to all these factors too. According to these real data, the fitness function used by our solvers is $\mathbf{z}^R(p)$ (see Equation (8.3)). This function has four objectives: the means and variances of the travel time and gas emission.

8.4.3 Optimization Algorithms

We choose five algorithms to solve the problem with different search strategies. The selected algorithms are Dijkstra algorithm (Dijkstra, 1959), A* (Hart, Nilsson, and Raphael, 1968), Pulse (Duque, Lozano, and Medaglia, 2015), Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002b), and Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Zhang and Li, 2007). All of them return paths between two single nodes s and e. These nodes and the graph are the inputs for the solvers.

In A*, the heuristic function used in our study is the Euclidean distance between the geographical position of the nodes.

Dijkstra and A* are single-objective. We chose them because they are typically used in applications and state-of-the-art comparisons (Ardakani and Tavana, 2015; Pascoal and Resende, 2015; Sheng and Gao, 2016). But, we are solving a multiobjective problem. So, we transform the problem from multiobjective to single-objective using a weighed sum of the objectives. This method allows us to find supported solutions from the Pareto front. The supported solutions are those belonging to the convex hull of the front. The transformed problem is as follows:

$$\min_{p \in \mathcal{P}_{s,e}} w(p) = \sum_{(i,j)\in p} (w_0\mu_{ij,1} + w_1\mu_{ij,2} + w_2\sigma^2_{ij,1} + w_3\sigma^2_{ij,2}) \tag{8.5}$$

with w_i, $0 \leq i \leq 3$ the weights associated to each objective.

We considered different weight values: $\{0.0001, 0.25, 0.50, 0.75, 1\}$ for each weight. Weight 0.0001 is used instead of 0 to ensure that the solution found is efficient (and not only weakly efficient). We execute the single-objective algorithm (Dijkstra or A*) for each combination of weight values. This means a total of 625 runs of the single-objective algorithms per instance (all combinations of weight values for the four weights: 5^4). With these weights evenly distributed in the interval (0, 1] we try to explore uniformly the objective space with these single-objective algorithms. After the runs, we collect the obtained solutions to find a subset of the Pareto optimal set. We denote with WDijkstra and WA* the Dijkstra's algorithm and A* solving the RBSP problem using this approach, respectively.

NSGA-II and MOEA/D have some common configuration parameters. In order to do a fair comparison among them we have used the same parameters in both algorithms. These are:

- Selection operator: random selection.
- Crossover operator: one-point crossover with probability 0.9.
- Mutation operator: custom mutation with probability 0.1. This mutation selects a random middle point k in a (s-e)-path and a random vertex i and compute the two paths executing the A* algorithm: (k-i)-path and (i-e)-path. After computing the shortest path between the pairs (s-k), (k-i), and (i-e), it joins them to form the new solution path (s-k-i-e)-path.
- Replacement operator: elitist tournament.
- Neighbourhood of 5 individuals (only MOEA/D).

The stopping condition in NSGA-II and MOEA/D is to reach 10,000 fitness evaluations. The population size is 10 individuals.

8.5 Experimental Results

In this section we will present the main results of our study. First, we will analyze the Pareto fronts obtained, followed by the robustness of the different solutions found for each instance. Finally, the algorithms WDijkstra and WA* will be specifically analyzed apart, focusing on the relationship between the weights and the fitness values for the different solutions found.

8.5.1 Pareto Optimal Set Analysis

In this section we analyze the quality of the approximated Pareto optimal set found by the algorithms in the different instances for two different types of polluting gases: CO_2 (29 instances) and NO_x (28 instances). NSGA-II and MOEA/D got solutions which do not reach the same quality as the exact algorithms in all the experiments. In fact, the quality of their solutions with a similar execution time was much lower than the rest of the proposals. They did not get any solution from the Pareto front in any of the instances. That is why they are not explained in detail in this section. This behavior makes these metaheuristic algorithms a bad option for solving this problem.

Travel times and CO_2

Table 8.1 reports the execution time per solution and size of the sets of non-dominated solutions obtained by the deterministic algorithms when they analyze the travel times and CO_2. Pulse returns the entire front, so the reported time t is the total running time of the algorithm T divided by the number n of solutions in the Pareto front: $t = T/n$. An interesting result is that WDijkstra and WA* obtained the same results for each instance, but WA* is much faster, as expected. The Pulse algorithm returned the whole Pareto front at the cost of a higher run time.

Figure 8.3 shows the time per solution, in milliseconds, for each algorithm and instance. The running times follow ascending lines. However, a higher

TABLE 8.1: Runtime to compute one solution for WDijkstra, WA* and Pulse and number of different solutions found in each of the TT-CO_2 instances. We mark in bold the shortest time and largest number of solutions found for each instance.

Inst.	Start	End	Runtime per solution (ms)			# of solutions		
			WD.	WA*	Pulse	WD.	WA*	Pulse
1	32310	27542	1,801	**55**	53,398	4	4	**48**
2	12175	40684	1,743	**732**	72,523	4	4	**11**
3	40684	12175	1,375	**756**	77,136	3	3	**9**
4	27542	32310	1,936	**45**	77,957	5	5	**42**
5	13360	10386	1,677	**543**	113,866	6	6	**23**
6	30159	8481	2,032	**324**	132,024	13	13	**157**
7	23385	11669	1,731	**225**	159,944	10	10	**49**
8	10789	11798	1,361	**343**	178,261	4	4	**70**
9	11669	23385	1,771	**178**	180,364	6	6	**20**
10	28960	28582	1,534	**284**	182,434	7	7	**86**
11	39559	20777	2,000	**63**	188,075	7	7	**63**
12	10386	13360	1,208	**138**	211,592	6	6	**16**
13	11798	10789	2,205	**643**	257,288	5	5	**57**
14	42100	21155	1,261	**291**	261,992	6	6	**161**
15	15181	30200	2,060	**531**	279,094	9	9	**77**
16	14141	33551	1,377	**371**	302,760	5	5	**223**
17	33551	14141	1,559	**294**	308,817	5	5	**136**
18	8481	30159	1,630	**192**	343,680	6	6	**82**
19	19215	39301	1,405	**120**	406,114	5	5	**10**
20	39301	19215	1,062	**87**	437,103	4	4	**27**
21	20777	39559	1,771	**260**	470,971	8	8	**17**
22	21155	42100	2,056	**862**	476,278	4	4	**80**
23	45081	34504	1,139	**215**	778,485	12	12	**263**
24	30200	15181	1,184	**292**	854,915	15	15	**168**
25	34504	45081	1,474	**471**	872,470	7	7	**88**
26	28582	28960	1,554	**183**	1,200,626	8	8	**28**
27	25319	35869	1,525	**591**	2,286,564	10	10	**27**
28	20842	9585	2,130	**747**	2,312,745	5	5	**206**
29	35869	25319	1,263	**210**	4,591,774	11	11	**32**
Total			46,824	**10,046**	18,069,250	200	200	**2276**

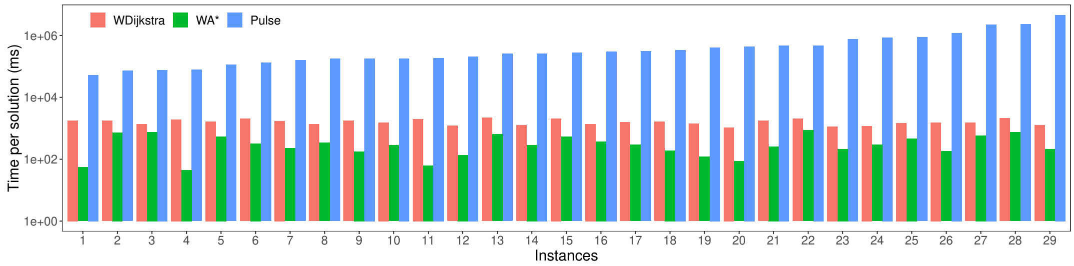


FIGURE 8.3: Execution time (ms) per solution of each deterministic algorithm in each instance of the problem with CO_2 as pollutant.

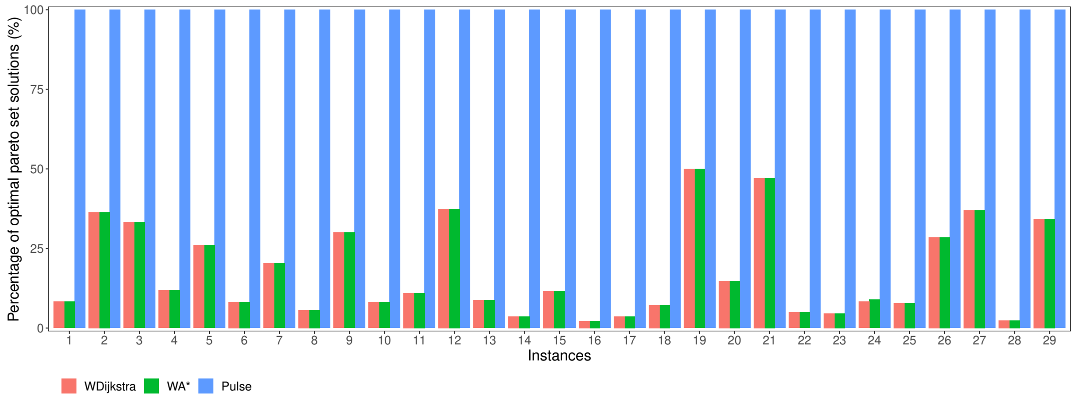


FIGURE 8.4: Percentage of the optimal Pareto set found for each deterministic algorithm in each instance of the problem with CO_2 as pollutant.

computation time does not imply a more extensive Pareto set. WDijkstra had stable behavior in their running time while WA* was less stable.

The sizes of the non-dominated sets are shown in Figure 8.4. The percentage of solutions found in the optimal front by WDijkstra and WA* is 17.77% on average. This percentage suggests that the number of supported solutions in the front is low. Despite the 625 weight combinations computed in WDijkstra and WA*, the obtained non-dominated sets have between three and twelve solutions only. This is because each point in the target space has a polytope associated with it in the weight space and several weight combinations are in the same polytope. In Section 8.5.3 we will study the combinations of weights in more detail.

Now, we will analyze the quality of the approximated Pareto fronts by some quality indicators (Riquelme, Von Lücken, and Baran, 2015). We select as metrics the Hypervolume (HV), ϵ-indicator, and Inverse Generational Distance (IGD) because they are commonly used in the multiobjective literature. Their values of the indicators are in Table 8.2. Figure 8.5 shows the HV percentage in each instance. We can see that, in some cases, the HV of the deterministic algorithms are very close to each other (especially in instance 29). NSGA-II has better HV than MOEA/D, even getting close to the deterministic algorithms. In general, both metaheuristics offer worse HVs, except in instance 28, in which the few solutions obtained by WDijkstra and WA*

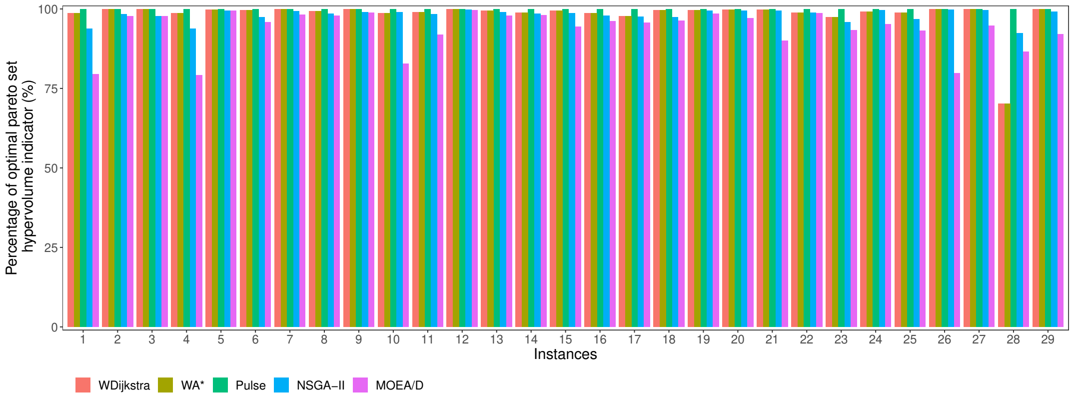


FIGURE 8.5: Percentage of the HV indicator of the approximated optimal Pareto set for the execution of each algorithm in every instance of the problem with CO_2 as pollutant.

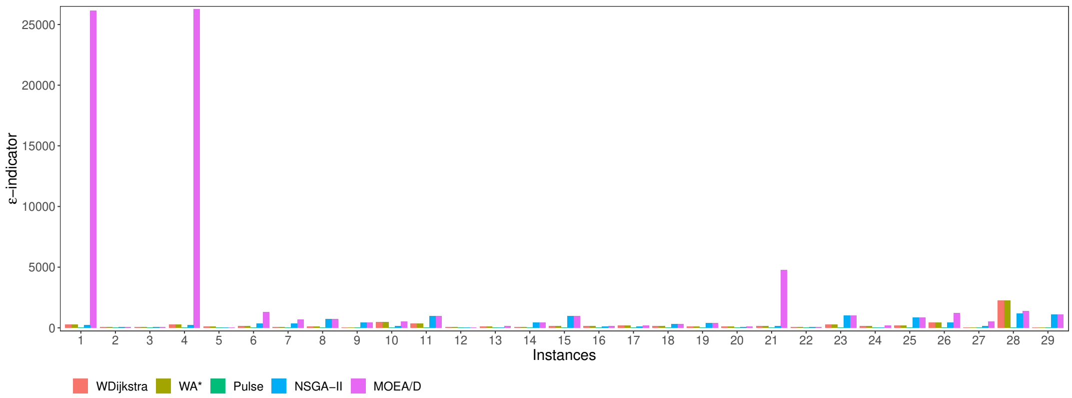


FIGURE 8.6: ϵ-indicator for the execution of each algorithm in every instance of the problem with CO_2 as pollutant.

are not enough to obtain a high HV. This is a good result for these solvers, because only with the supported solutions we can cover very well the objective space in the general case. The results for the number of non-dominated solutions and the HV suggest that many of the solutions are non-supported and their contribution to the HV is low.

If we see the ϵ-indicator (see Figure 8.6) of each algorithm (where lower values are better), we will find that in some instances the distance of the approximated Pareto front found by WDijkstra and WA* is very close (11 units) to the optimal front, and in general it is not so far from the Pareto front found by the Pulse algorithm. MOEA/D obtained the worst measures for this indicator, especially in three instances. This is an advantage of using single-objective algorithms and weighted sum to get an approximation of the Pareto front in the RBSP problem since a very fast approximated front is obtained with acceptable quality, even discarding the non-supported solutions.

TABLE 8.2: Quality indicators of the approximated Pareto fronts in each instance for the TT-CO_2 case. The highest values for the HV and the lowest values for ϵ-indicator and IGD are marked in bold.

Ins.	HV					ϵ-indicator					IGD				
	WD.	WA*	Pulse	NSGA-II	MOEA/D	WD.	WA*	Pulse	NSGA-II	MOEA/D	WD.	WA*	Pulse	NSGA-II	MOEA/D
1	6.84E+18	6.84E+18	**6.94E+18**	6.51E+18	5.52E+18	282	282	**0**	218	26135	14695	14695	**0**	1867	14663
2	1.09E+18	1.09E+18	**1.09E+18**	1.07E+18	1.07E+18	41	41	**0**	46	64	3388	3388	**0**	7492	2030
3	1.11E+18	1.11E+18	**1.11E+18**	1.09E+18	1.09E+18	59	59	**0**	64	64	1496	1496	**0**	11158	3920
4	6.82E+18	6.82E+18	**6.91E+18**	6.49E+18	5.48E+18	271	271	**0**	216	26244	13838	13838	**0**	9453	17030
5	1.12E+18	1.12E+18	**1.12E+18**	1.12E+18	1.12E+18	117	117	**0**	20	20	4726	4726	**0**	1890	1534
6	2.33E+18	2.33E+18	**2.34E+18**	2.28E+18	2.25E+18	137	137	**0**	341	1292	6927	6927	**0**	1904	2367
7	2.26E+18	2.26E+18	**2.26E+18**	2.24E+18	2.22E+18	42	42	**0**	346	665	5654	5654	**0**	3543	8213
8	1.73E+18	1.73E+18	**1.74E+18**	1.71E+18	1.70E+18	112	112	**0**	724	724	17597	17597	**0**	2757	1506
9	2.33E+18	2.33E+18	**2.33E+18**	2.31E+18	2.30E+18	28	28	**0**	422	422	2692	2692	**0**	7775	3284
10	2.12E+18	2.12E+18	**2.15E+18**	2.13E+18	1.78E+18	489	489	**0**	145	509	8673	8673	**0**	1500	4729
11	3.46E+18	3.46E+18	**3.50E+18**	3.44E+18	3.21E+18	334	334	**0**	951	951	11747	11747	**0**	1393	13320
12	1.08E+18	1.08E+18	**1.08E+18**	1.08E+18	1.08E+18	68	68	**0**	8	15	2280	2280	**0**	3087	2521
13	1.73E+18	1.73E+18	**1.74E+18**	1.73E+18	1.71E+18	103	103	**0**	37	145	12218	12218	**0**	2561	2733
14	7.72E+16	7.72E+16	**7.81E+16**	7.70E+16	7.66E+16	69	69	**0**	420	420	7980	7980	**0**	6081	17281
15	2.71E+17	2.71E+17	**2.73E+17**	2.70E+17	2.58E+17	140	140	**0**	961	961	8282	8282	**0**	30696	44340
16	2.63E+18	2.63E+18	**2.67E+18**	2.61E+18	2.56E+18	157	157	**0**	104	164	26941	26941	**0**	47710	53018
17	2.74E+18	2.74E+18	**2.80E+18**	2.73E+18	2.68E+18	183	183	**0**	118	168	27166	27166	**0**	15004	91262
18	2.35E+18	2.35E+18	**2.36E+18**	2.30E+18	2.27E+18	135	135	**0**	313	313	7628	7628	**0**	1680	2951
19	4.04E+18	4.04E+18	**4.05E+18**	4.03E+18	4.00E+18	94	94	**0**	398	398	5441	5441	**0**	1939	2555
20	4.08E+18	4.08E+18	**4.09E+18**	4.07E+18	3.97E+18	92	92	**0**	61	102	12275	12275	**0**	11908	3205
21	3.56E+18	3.56E+18	**3.57E+18**	3.55E+18	3.21E+18	144	144	**0**	144	4758	1457	1457	**0**	7211	10852
22	8.58E+16	8.58E+16	**8.68E+16**	8.58E+16	8.56E+16	70	70	**0**	62	66	7611	7611	**0**	4015	12351
23	4.92E+17	4.92E+17	**5.05E+17**	4.85E+17	4.72E+17	251	251	**0**	999	999	8523	8523	**0**	6371	41705
24	2.77E+17	2.77E+17	**2.79E+17**	2.78E+17	2.66E+17	144	144	**0**	33	167	6226	6226	**0**	13961	43379
25	7.33E+17	7.33E+17	**7.41E+17**	7.18E+17	6.91E+17	196	196	**0**	842	842	17948	17948	**0**	12639	33830
26	2.11E+18	2.11E+18	**2.11E+18**	2.10E+18	1.68E+18	444	444	**0**	444	1229	644	644	**0**	638	2086
27	7.84E+17	7.84E+17	**7.84E+17**	7.80E+17	7.42E+17	34	34	**0**	160	530	2079	2079	**0**	1448	7484
28	1.48E+17	1.48E+17	**2.11E+17**	1.95E+17	1.82E+17	2249	2249	**0**	1159	1403	22209	22209	**0**	1569	6347
29	7.02E+17	7.02E+17	**7.02E+17**	6.96E+17	6.46E+17	11	11	**0**	1109	1109	3802	3802	**0**	2238	4437

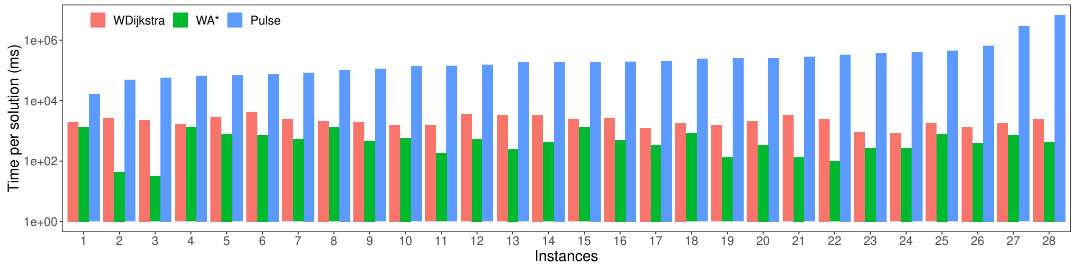


FIGURE 8.7: Execution time per solution (ms) of each deterministic algorithm in each instance of the problem with NO_x as pollutant.

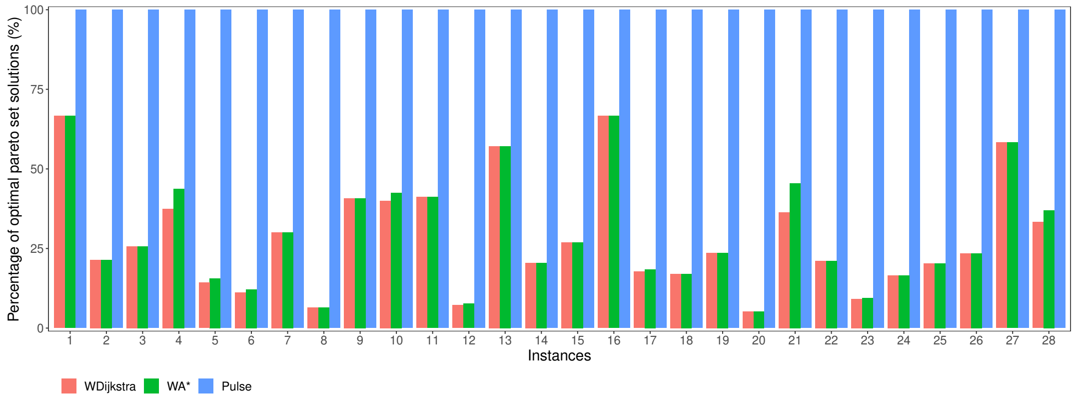


FIGURE 8.8: Percentage of the optimal Pareto set found for each deterministic algorithm in each instance of the problem with NO_x as pollutant.

Travel times and NO_x

In this section, we repeat the previous analysis replacing the CO_2 by NO_x. Table 8.3 reports the running time per solution and the number of them found by WDijkstra, WA*, and Pulse. While the execution times are similar to those of the version with CO_2 (compare Figures 8.3 and 8.7) the number of solutions dound differs. The number of solutions obtained by the strategy of weighted sums is greater than in the case of CO_2. In general, 86% more solutions have been found in fronts that have 14% fewer solutions, so the approximations are generally better than with CO_2 (see Figure 8.8). It should also be noted that in nine cases WA* found one solution more than WDijkstra.

We can state that the problem has more supported solutions when NO_x is considered, compared to the case of CO_2. If the number of solutions and the runtime are important in an application, this result suggests that NO_x should be considered in the optimization.

Regarding the quality indicators of the different instances, shown in Table 8.4, the results in the three metrics outperform those obtained in the case of CO_2. In the case of the HV (see Figure 8.9), we again observe that the three deterministic algorithms obtain a similar value and the two metaheuristic algorithms have the lowest performance. Figure 8.10 shows the results of the ϵ-indicator, which improve the ones returned for the CO_2 experiments.

TABLE 8.3: Runtime to compute one solution for each deterministic algorithm in each instance and number of different solutions found in the TT-NO_x case. We mark the shortest time and largest Pareto front size for every instance.

Ins.	Start	End	Runtime per solution (ms)			# of solutions		
			WDijkstra	WA*	Pulse	WDijkstra	WA*	Pulse
1	32310	27542	2,702	**44**	48,945	9	9	**42**
2	12175	27543	2,007	**1,310**	16,081	8	8	**12**
3	40684	27544	1,323	**389**	659,375	8	8	**34**
4	27542	27545	2,351	**32**	57,746	9	9	**35**
5	13360	27546	1,703	**1,301**	66,766	6	7	**16**
6	30159	27547	3,467	**415**	188,996	36	36	**176**
7	23385	27548	2,404	**522**	85,042	22	22	**73**
8	10789	27549	2,963	**772**	68,870	11	12	**77**
9	11669	27550	2,011	**469**	112,745	11	11	**27**
10	28960	27551	2,093	**1,353**	101,731	4	4	**62**
11	39559	27552	2,520	**101**	332,607	11	11	**52**
12	10386	27553	1,508	**184**	142,696	7	7	**17**
13	11798	27554	4,309	**710**	75,567	12	13	**107**
14	42100	27555	1,227	**338**	203,377	27	28	**152**
15	15181	27556	1,550	**597**	135,527	16	17	**40**
16	14141	27557	3,516	**528**	154,292	15	16	**208**
17	33551	27558	1,862	**824**	242,193	14	14	**82**
18	8481	27559	1,498	**131**	250,644	13	13	**55**
19	19215	27560	3,340	**247**	185,996	4	4	**7**
20	39301	27561	3,350	**133**	283,093	4	5	**11**
21	20777	27562	2,608	**506**	196,655	8	8	**12**
22	21155	27563	2,488	**1,296**	190,553	14	14	**52**
23	45081	27564	909	**265**	376,058	25	26	**273**
24	30200	27565	819	**265**	397,895	24	24	**145**
25	34504	27566	1,825	**813**	447,447	11	11	**54**
26	28582	27567	2,089	**335**	257,131	5	5	**96**
27	25319	27568	1,812	**739**	2,901,262	7	7	**12**
29	35869	27570	2,408	**422**	6,811,269	9	10	**27**
Total			65,128	**5,639**	14,990,559	372	381	**1,956**

TABLE 8.4: Quality indicators of the Pareto fronts in each instance of the TT-NO_x case. The highest values for the HV and the lowest values for ϵ-indicator and IGD are marked.

Ins.	HV					ϵ-indicator					IGD				
	WD.	WA*	Pulse	NSGA-II	MOEA/D	WD.	WA*	Pulse	NSGA-II	MOEA/D	WD.	WA*	Pulse	NSGA-II	MOEA/D
1	1.49E+13	1.49E+13	**1.49E+13**	1.45E+13	1.36E+13	7	7	**0**	12	32	14	14	**0**	17	33
2	3.06E+12	3.06E+12	**3.06E+12**	2.92E+12	2.92E+12	0	0	**0**	11	11	2	2	**0**	14	19
3	3.07E+12	3.07E+12	**3.07E+12**	2.93E+12	2.92E+12	3	3	**0**	9	17	5	5	**0**	15	23
4	1.49E+13	1.49E+13	**1.49E+13**	1.48E+13	1.34E+13	9	9	**0**	9	33	11	11	**0**	106	131
5	3.07E+12	3.07E+12	**3.08E+12**	2.97E+12	2.92E+12	6	6	**0**	10	11	13	13	**0**	15	27
6	5.43E+12	5.43E+12	**5.44E+12**	5.23E+12	4.80E+12	12	12	**0**	11	33	9	9	**0**	15	34
7	5.69E+12	5.69E+12	**5.69E+12**	5.41E+12	5.27E+12	3	3	**0**	12	18	3	3	**0**	212	176
8	4.22E+12	4.22E+12	**4.23E+12**	4.06E+12	3.45E+12	22	22	**0**	13	47	28	27	**0**	16	56
9	5.90E+12	5.90E+12	**5.90E+12**	5.82E+12	5.68E+12	3	3	**0**	4	9	3	3	**0**	9	19
10	4.97E+12	4.97E+12	**5.01E+12**	4.72E+12	4.46E+12	18	18	**0**	18	33	139	139	**0**	19	41
11	8.05E+12	8.05E+12	**8.07E+12**	7.41E+12	6.94E+12	25	25	**0**	26	43	51	51	**0**	14	32
12	3.06E+12	3.06E+12	**3.06E+12**	3.05E+12	2.95E+12	4	4	**0**	6	7	5	5	**0**	7	15
13	4.20E+12	4.20E+12	**4.21E+12**	4.04E+12	3.44E+12	27	27	**0**	12	46	29	29	**0**	12	49
14	5.71E+11	5.71E+11	**5.72E+11**	5.50E+11	4.26E+11	9	9	**0**	19	30	9	9	**0**	22	37
15	1.11E+12	1.11E+12	**1.11E+12**	1.01E+12	1.05E+12	6	6	**0**	21	30	5	5	**0**	27	30
16	6.16E+12	6.16E+12	**6.18E+12**	5.97E+12	5.94E+12	37	37	**0**	24	24	50	37	**0**	29	28
17	6.38E+12	6.38E+12	**6.42E+12**	6.41E+12	6.01E+12	40	40	**0**	25	29	52	52	**0**	31	97
18	5.43E+12	5.43E+12	**5.43E+12**	5.06E+12	4.91E+12	12	12	**0**	27	27	9	9	**0**	40	29
19	8.58E+12	8.58E+12	**8.60E+12**	8.59E+12	8.57E+12	18	18	**0**	3	13	15	15	**0**	15	11
20	8.69E+12	8.69E+12	**8.69E+12**	8.61E+12	8.61E+12	9	9	**0**	3	3	12	8	**0**	41	95
21	8.17E+12	8.17E+12	**8.17E+12**	7.92E+12	7.72E+12	0	0	**0**	9	16	5	5	**0**	38	30
22	5.64E+11	5.64E+11	**5.65E+11**	5.42E+11	4.15E+11	9	9	**0**	19	30	9	9	**0**	12	17
23	1.71E+12	1.71E+12	**1.72E+12**	1.40E+12	1.25E+12	11	11	**0**	33	51	11	11	**0**	51	51
24	1.13E+12	1.13E+12	**1.13E+12**	1.12E+12	1.06E+12	9	9	**0**	9	34	12	12	**0**	23	72
25	2.19E+12	2.19E+12	**2.19E+12**	1.89E+12	1.74E+12	12	12	**0**	27	42	14	14	**0**	67	55
26	4.87E+12	4.87E+12	**4.94E+12**	4.78E+12	4.62E+12	19	19	**0**	16	21	263	263	**0**	64	85
27	2.28E+12	2.28E+12	**2.28E+12**	2.24E+12	2.24E+12	1	1	**0**	5	5	2	2	**0**	26	78
29	2.15E+12	2.15E+12	**2.15E+12**	2.13E+12	2.13E+12	7	7	**0**	7	7	7	6	**0**	4	5

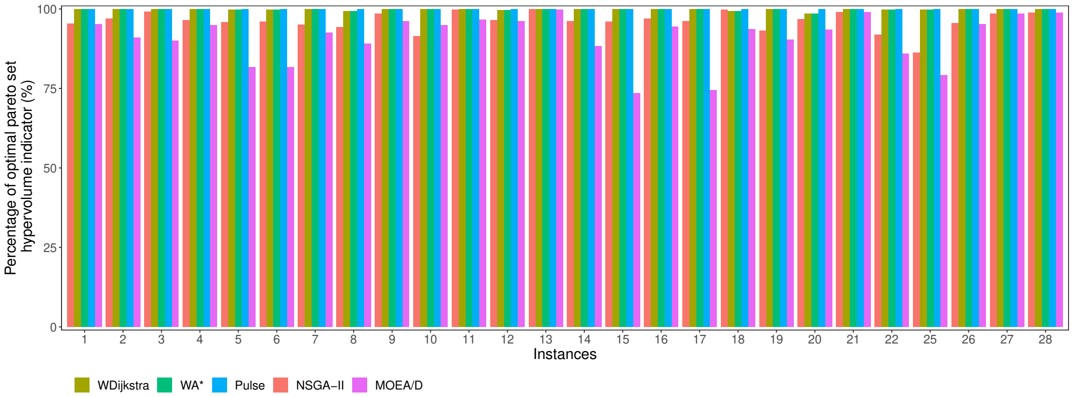


FIGURE 8.9: Percentage of the HV indicator of the approximated optimal Pareto set for the execution of each algorithm in every instance of the problem with NO_x as pollutant.

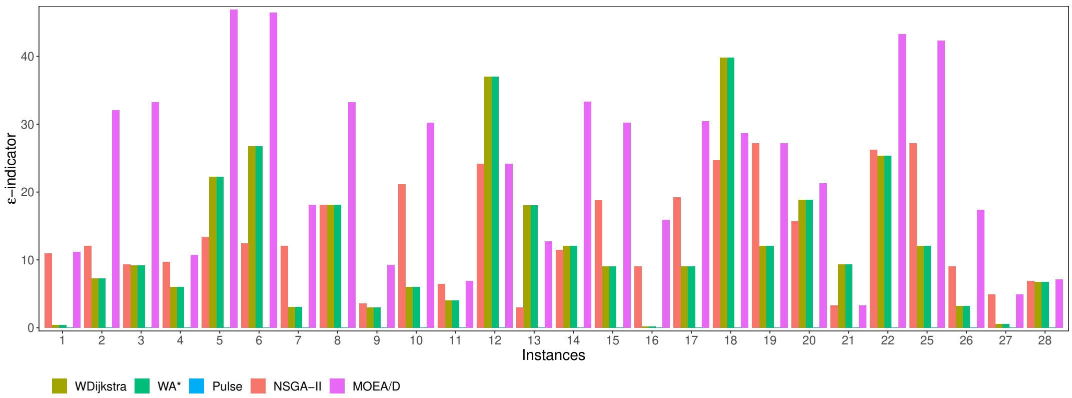


FIGURE 8.10: ϵ-indicator for the execution of each algorithm in every instance of the problem with NO_x as pollutant.

There are no differences between WDijkstra and WA* in these two quality indicators. However, in the IGD, there are differences in some cases. Although WA* improved the number of solutions in nine scenarios, the metric is only improved in three of them. This result suggests that despite good approximations being obtained, there may still be regions in which there are supported solutions that have not been found. Metaheuristics continue to get the worst results.

8.5.2 Robustness Analysis

In this section, we analyze the degree of robustness of the solutions. Our proposal differs from others because we offer a set of solutions (not only one) with varying levels of quality and robustness in each objective (TT and gas emission). The final user could choose which solution to apply among all optimal solutions provided by us. As an example, Figure 8.11 shows the Pareto front of instance 13 optimized for TT and CO_2. The colors represent the different algorithms used in this study, and the size of the boxes represent

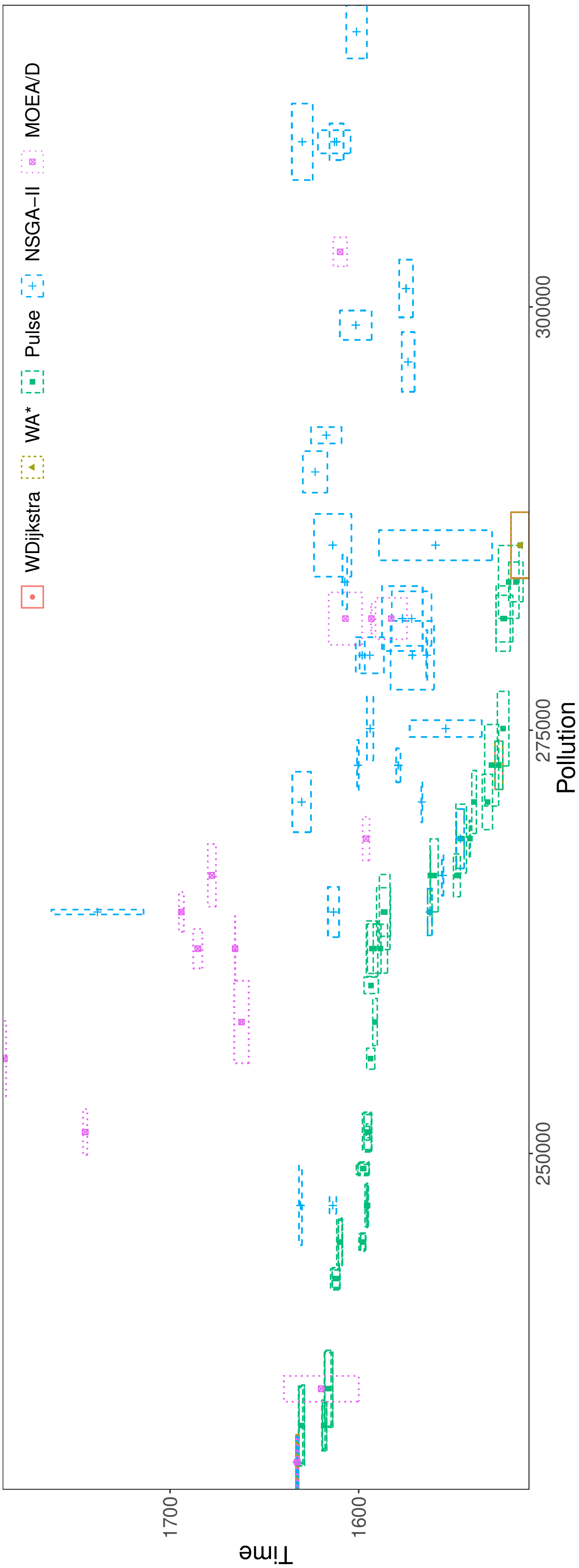


FIGURE 8.11: Approximated Pareto fronts of instance 13 for TT-CO_2 found by the each algorithm. The dimension of the squares is the standard deviation for the travel time and CO_2 emission, respectively.

the robustness of a solution (its standard deviation). Each solution is represented by a box which is centered in the mean CO_2 emission and travel time, and its width and height are the standard deviation for CO_2 emission and travel time, respectively. Thus, a very robust solution has a small box, while a large box indicates that the solution quality has a large variability in their objectives (TT or CO_2).

We can observe that a large portion of the solutions (obtained by Pulse) is in the concave region of the front. WDijkstra and WA* algorithms cannot find these solutions. It can also be seen how the solutions that in a non-robust version would be dominated have a smaller variance (smaller box) in the objectives. On the other hand, NSGA-II and MOEA/D do not reach the level of quality of the deterministic algorithms, obtaining solutions very far from the Pareto front.

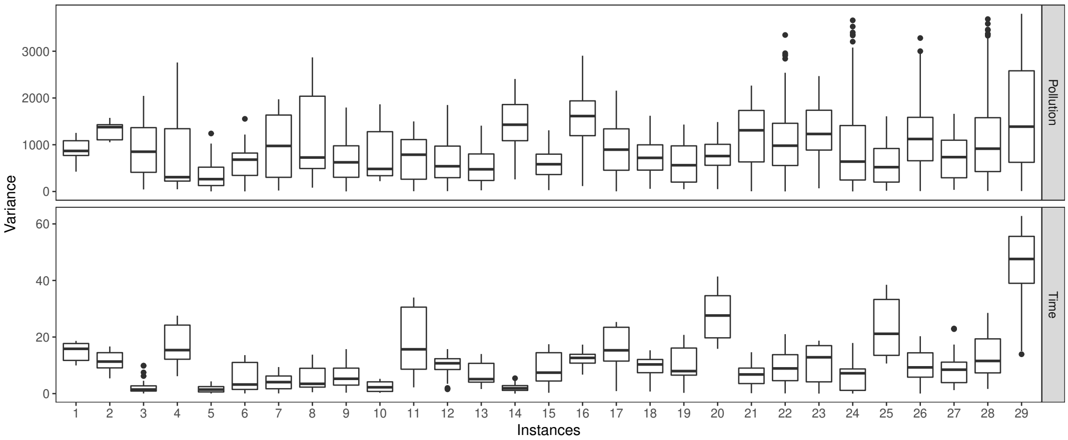
The variation in the objectives between the most and least robust solutions is slight in most cases. But this does not mean that robustness does not have to be considered because the box area is not the same size. Also, there is a subset whose size is smaller in each front, i.e., a more robust solution.

Now, we analyze in more depth the Pareto fronts obtained by Pulse. In most cases, the variation in the objectives between the most and least robust solutions is small. But this does not mean that robustness does not have to be taken into account because the box area is not the same size. Also, there is a subset whose size is smaller in each front, which means a more robust solution. Figure 8.12 shows the variance of the pollution gases, CO_2 and NO_x, and TT among the different solutions in the Pareto front. We can see that the TT has a smaller variation than the CO_2, being especially small in some of the instances, which slightly increases the user's travel times greatly reduces the levels of CO_2 emitted. This makes us very optimistic with our model, as it is interesting to explore solutions with low variation in objectives. However, as in instance 29, the variance changes a lot between the solutions, so there are cases in which TT is greatly affected by the chosen route. In the case of NO_x, it is observed that the variances are lower than in the case of CO_2. More robust solutions are obtained if NO_x is used as a pollutant gas. But, in general, more extreme solutions are observed compared to the CO_2 case.

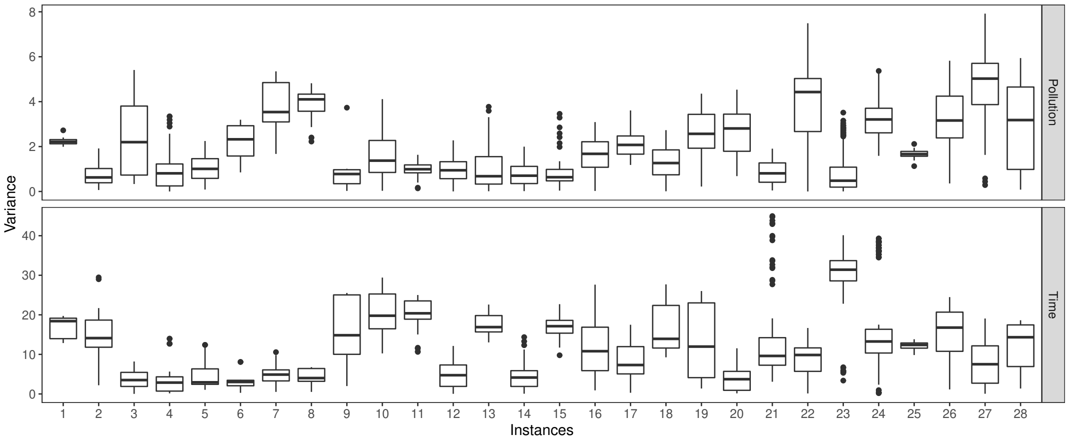
8.5.3 Weights Analysis

The difference between the number of total weight combinations (625) and different efficient solutions (a few tens) found by the algorithms WDijkstra and WA* opens the possibility of studying which weights return the same solutions. Discovering these relationships between weights and fitness can help us to reduce the computation times of this strategy and allows us to discard weight combinations very quickly.

Formally, a supported non-dominated solution in the Pareto front is associated to a polytope in the weights space, e.g., a set of weights values that return the same solution in the optimization process. We only need a single weight combination in the interior of each polytope to get the whole set of supported solutions associated with that polytope.



(A) Travel times and CO_2 case.



(B) Travel times and NO_x case.

FIGURE 8.12: Variances of each objective in relation to the mean for each instance in the Pulse algorithm.

We will analyze one of the instances as an example. Figure 8.13 shows the different weights combinations in a single instance. In the plot, the colors indicate a specific solution obtained with the weight combination. Several boxes with the same color form a set of combinations of weights that generate the same solution. There is a pattern in the colored grids. When $w_3 > 0.0001$, the number of solutions found is considerably limited. This weight has a large influence on the exploration of the region where most of the supported solutions are located. Using two values for w_3 could make us reduce the computational time. Weights w_0, w_1 and w_2 do not have a direct influence in the supported solutions explored (as w_3 has): varying these three weights many polytopes defining different supported solutions are reached.

These results confirm our idea of reducing execution times by analyzing the weights. In our experimentation this would suppose a reduction of 98.47 %. We think that these patterns in the relationship between the weights and the solutions can be calculated from the characteristics of the problem (map and start and end points), but we defer this issue to future work.

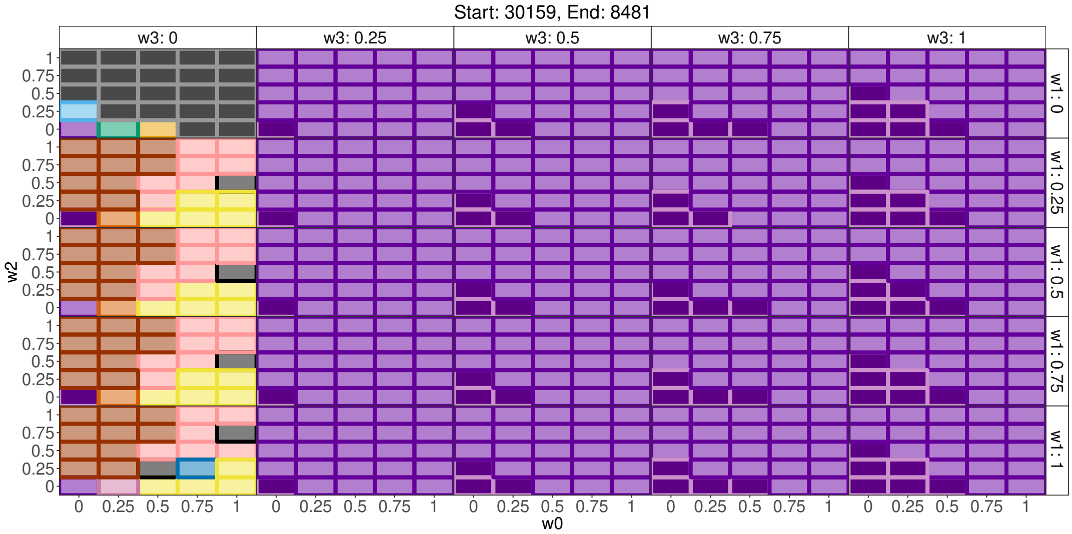


FIGURE 8.13: Weight combinations to obtain each supported solution (different colour) in the instance number 6 optimized for TT and CO_2.

8.6 Related Work

We review in this section some works related to the main topic of this research. Starting with the concept of robustness, Jin and Branke (2005) defines it as the desirable characteristic of taking into account the uncertainties in different sources of information (parameters, data, etc.). One way of looking at robustness is considering the inaccuracies in the parameters of the problem that come from multiple sources, such as approximate data or incorrect sensor measures. Usually, the inaccuracies can be defined in different ways:

- The parameter p can have values from a finite and defined set of values $p \in S = \{s_1, s_2, \ldots, s_n\}$. Each element in the set is called a *scenario* (Kuhn et al., 2016).
- The parameters can take values from an infinite set, e.g., real numbers, $p \in [a, b]$. Because the techniques for solving them are often similar, this group also includes probabilistic functions and confidence intervals (Ehrgott, Ide, and Schöbel, 2014; Gabrel, Murat, and Thiele, 2014; Gong, Qin, and Sun, 2011), among others.

The type of values in the parameters directly implies the chosen model of robustness and the techniques to solve the robust problem. Our proposal is included in the second definition of inaccuracies in the parameters.

There are several surveys that analyze distinct aspects of robustness. An overview of the whole paradigm of robustness is described in the survey of Gabrel, Murat, and Thiele (2014). This article defines and lists most of the scientific works done in robust optimization (in its general meaning) in the last years. Ide and Schöbel (2016) present a classification of different types of robust problems is presented. The robust solutions are classified according to the number of scenarios in which the solution is optimal. In this way, one solution is more robust than another when it is the optimal solution in a larger number of scenarios. But, it is not always possible to have well-defined scenarios. Moreover, not all scenarios are the same or have the same relevance. In this taxonomy, the minmax strategy is adapted for each kind of robustness. The minmax strategy (Kjeldsen, 2001) tries to minimize the following expression:

$$\min_{x \in X} \max_{s \in S} f_s(x) \tag{8.6}$$

where S is the set of scenarios, X the solution space, and $f_s(x)$ the objective function in the scenario s. The solution x is an upper bound for all the scenarios. Because of this, it is very common to use the minmax strategy when solving the shortest path problem with scenarios (Pascoal and Resende, 2015). However, minmax fairly considers usual and unusual scenarios. In many cases, users may prefer to lose a bit of robustness in exchange of a significant improvement in the quality of the solution.

If the robustness cannot be modeled in a discrete way, a range of possibilities opens up: fuzzy logic (Keshavarz and Khorram, 2009), random delay variables (Cheng, Lisser, and Letournel, 2013), confidence intervals (Gong, Sun, and Ji, 2013; Gong, Sun, and Miao, 2018; Hasuike, 2013; Sun et al., 2018), probabilistic distributions (Chassein and Goerigk, 2015), etc. The minmax strategy can also be applied when the data can change in a defined interval $[a, b]$. This limitation in the decision variables can be used by a minmax strategy to minimize the worst case (upper bound of each interval) scenario, as done in (Hazir, Erel, and Günalay, 2011; Raith et al., 2018).

Besides saying whether a solution is robust or not, a certain degree or level of robustness could be given to a single solution. According to the classification that is presented in (Kuhn et al., 2016), we can assign a level of robustness to a solution, i.e., each robust definition of a problem have an associated level of robustness for its solutions. A common way to measure the robustness is by a confidence ratio (Mirjalili, Lewis, and Mostaghim, 2015), a.k.a. radius of robustness (Goberna et al., 2015). It defines a region in the parameters space where the parameters can move for a given solution without changing the objective values more than a predefined amount. The radius defines the degree of robustness: (i) if the radius is small, the solver will give us a solution specific to these parameters; (ii) if it is large, the solver gave us a more robust solution. In the first case (less robust solution), the solution will have better fitness value than in the second case.

Some authors as Ehrgott, Ide, and Schöbel (2014) or Kuhn et al. (2016) have done works in robust optimization from a theoretical point of view. Ehrgott, Ide, and Schöbel (2014) classify robust solutions according to criteria of dominance between solutions. To solve the problem, they propose three approaches based on weighted sum scalarization, ϵ-constraint scalarization, and objective-wise worst case. Kuhn et al. (2016) present a classification of the different degrees of robustness of a single solution according to the number of scenarios in which it is efficient. However, they only take into account the uncertainty in one of the objectives, while we consider the two objectives with the same uncertainty criteria. They also present an algorithm based on the two-phase method to obtain the Pareto set of efficient solutions from the set of robust solutions that are not dominated by any other solution in the whole set of solutions.

In general, the state-of-the-art in the field of robustness is treated in very different ways. These approaches have in common that they offer a single robust solution. In contrast, our proposal gives a set of solutions that is calculated by computing the variance in all the objectives.

8.7 Conclusions

In this chapter, we presented a new robust model to deal with the uncertainty in the parameters of a problem in smart cities: the biobjective shortest path problem. We have proposed a new model for the robust biobjective shortest path problem based on multiobjective optimization. We optimized simultaneously the mean and the variance of the costs of the routes. We have selected parameters of interest to citizens for the costs of the edges: travel times and gas emission (CO_2 and NO_x). To test our proposal, we used real data and different algorithms: Dijkstra, A*, Pulse, NSGA-II, and MOEA/D.

We measure the degree of robustness of each solution by its variance in the objective space. We check that if we consider the robustness, we can obtain more robust solutions at the expense of a minimal penalty in the quality of the solutions.

We check that not only the mean values of the objectives are opposed but also the variances. This result is significant when we want to implement the robustness model in a final application. It is interesting to choose a path with little variability and make an order of priority in optimizing the objectives.

Part III

Conclusions and Future Work

Chapter 9

Conclusions

In this thesis, different problems related to Smart Mobility have been addressed. Realistic data has been prioritized instead of working with synthetic data and unrealistic maps. In particular, many Open Data sources and a real scenario have been used for the experimentation carried out. In this line, instead of comparing ourselves with academic benchmarks, we have used the city of Malaga in the south of Spain. This city is simultaneously the scenario and a competitor for our solutions. Overall, we have achieved remarkable realism in our modeling and problems, offering solutions that can be implemented in the city.

On the other hand, the importance of placing sustainability in the focus of our research has become apparent. Reducing emissions has been a constant in most of the problems addressed. It has been possible to improve traffic, generate fewer emissions, and have more fluid traffic, thus reducing travel times and fuel consumption.

In addition to working with fuel-powered vehicles, we have dealt with problems of means of transportation that have gained popularity in recent years. Bicycles have become the number one vehicle in sustainability, and bicycle-sharing services have become very popular. On the other hand, electric vehicles are becoming more and more accessible in terms of price and performance for citizens. These forms of green mobility require specific infrastructures. In contrast to other works that deal with the placement of bicycle or vehicle charging stations from an industrial perspective, we offer a more social vision. We bring these stations closer to potential customers using different metaheuristics. We even simultaneously optimize the installation costs and the quality of service of electric car charging stations.

Another critical aspect of realism has been to include the robustness feature in several of our problems. We cannot make mistakes to obtain adapted and valuable solutions for citizens, so we offer stable routes for road traffic or improve the traffic lights so that a good traffic flow is obtained whatever the number of vehicles.

Finally, we can conclude that Open Data allows us to provide adaptable solutions to the needs of cities. Considering environmental and social aspects in our research is possible thanks to Open Data. Although there is still much work to be done in this line, this thesis serves as a precedent for future work that delves into sustainability in Smart Mobility and other areas of society.

Chapter 10

Future work

To conclude this thesis, different lines of future work will be presented. Each of the problems will be analyzed individually. In addition, some global ideas will be given to further developing this line of research.

In the problem of optimizing traffic lights programs, we present a hybridization between IRACE (López-Ibáñez et al., 2016) and evolutionary algorithms. A natural step in this line of research is to consider other algorithms and operators that have proven to be effective in numerical optimization problems for hybridization with IRACE. Our best results are computed by IRACE+DE, which encompasses the canonical version of DE. Thus, we test other well-known variants such as JADE (Zhang and Sanderson, 2009). Although preliminary experiments hybridizing IRACE with JADE, did not improve the results over the IRACE+DE proposed in this thesis, we plan to perform a deeper analysis of IRACE+JADE to extract any insights about the behavior of the new hybrid algorithms. Also, we do not want to limit the developed hybrid algorithms to this problem. We plan to test our hybrid algorithms on other black-box numerical optimization problems under uncertainty to validate our results further and check the advantages and disadvantages of this solver.

When dealing with problems with a large number of vehicles, such as the optimal proportions of goods vehicles, there are multiple objectives to consider. Noise is one traffic phenomenon that affects the population, including their health. It would be interesting to include, in addition to the economic (travel times and fuel consumption) and environmental objectives, noise as a fourth objective to be reduced. This will imply a four-objectives analysis which will also lead us to test other (Many-objective) solvers to improve the running time of the algorithm and its results.

This thesis presents two problems of infrastructure location for ecological and current urban transport: bicycle stations and charging stations for electric vehicles. For bicycle stations, we want to extend the model to include even more real-world information such as the morphology of the city (hills and slopes), types of roads, number of slots per station and the importance of specific points of interest, as hospitals, schools, shopping centers, etc.

Another future work would be to expand our model with other parts of a commercial system, such as transporting bicycles from one station to another to balance the load or studies of taxes and economic impact on citizens. Including socio-economic information would be interesting to bring this type of transport closer to the population sectors that may make greater use, such as neighborhoods with a more significant number of young people.

Besides, we would like to add even more operators to each algorithm. Designing operators that consider the geographical distribution of the stations could be a step forward in this line of research. And perform a more exhaustive analysis of the impact of the operators, specifically the local searches, on the global performance.

On the other hand, in the case of charging stations for electric vehicles, one of the main lines for future work is related to evaluating exact approaches such Integer Linear Programming variations to address this multiobjective optimization problem. We have already worked on using Integer Linear Programming to solve a single-objective version of this problem (Risso et al., 2022). Despite the increase in the number of variables to consider, we believe that with some debugging and filtering of the data, it may be possible to use exact solvers such as CPLEX (Cplex, 2009) to solve the multiobjective version. In this line, we also want to study the point at which the exact results give us solutions in a time suitable for use in a commercial application. From that point on, metaheuristics can still give us good results, so we plan to test different algorithms and operators and combine both strategies.

Also, it will be interesting to increase the realism of the model (as in the bicycle station problem) by considering general citizen's mobility behavior, the location of points of interest or the vehicle fleet.

Exciting future work is to associate charging stations with different electrical substations. These new decision variables would directly impact the city's power grid. The free association between charging stations and electrical substations would allow us to better balance electricity consumption, for example, by allowing a more significant number of charging stations in the same area without the power grid suffering.

The last problem presented is the routing of vehicles. It would be interesting as future research to include more realism in our model, including traffic light information. In this line, it would be interesting to analyze how the certainty of the traffic light plans would affect our robustness model. Our model is also open to adding new objectives, like the monetary cost of the routes (including fees of toll stations), other pollutants, noise, etc. or the use by electric vehicles.

In a different line of work, we want to analyze how to select the weights for solving the multiobjective problem using single-objective algorithms and a weighted sum of objectives. We think it is possible to use the graph to select the most appropriate weight combinations.

All these problems have been tested in Malaga. Using an actual city has allowed us to test our contributions with a medium-sized European city. It would be interesting to experiment with other cities with different sizes, small and large ones, and layouts (e.g., New York with its square blocks).

Finally, the ultimate goal is to integrate all the services developed. Being able to design new neighborhoods, improve disadvantaged areas or make the city more adaptable are some of the goals we can achieve by integrating all services. In addition, we cannot forget environmental needs. Without generating polluting gases and using renewable energy sources, our future must make our cities 100% sustainable. Therefore, after this thesis, we will not stop working in this line, generating research and solutions that contribute to improving the environment and people's lives.

Part IV

Appendix

Appendix A

List of Publications Supporting the Doctoral Thesis

This appendix presents the set of papers that have been published as a result of the research developed throughout this doctoral thesis. These publications endorse this doctoral thesis's interest, validity, and contributions in the literature since these works have been published in prestigious forums and, therefore, have been subjected to review processes.

Indexed Journals

Cintrano, C, F Chicano, and E Alba (2019). "Facing robustness as a multi-objective problem: A bi-objective shortest path problem in smart regions". In: *Information Sciences* 503, pp. 255–273.

— (2020). "Using metaheuristics for the location of bicycle stations". In: *Expert Systems with Applications* 161, p. 113684.

Cintrano, Christian, Javier Ferrer, Manuel López-Ibáñez, and Enrique Alba (to appear). "Hybridization of Evolutionary Operators with Elitist Iterated Racing for the Simulation Optimization of Traffic Lights Programs". In: *Evolutionary Computation*.

International Scientific Congresses and Workshops

Cintrano, Christian, Francisco Chicano, and Enrique Alba (2017). "Robust Bi-objective Shortest Path Problem in Real Road Networks". In: *International Conference on Smart Cities*. Springer, Cham, pp. 128–136.

Cintrano, Christian, Javier Ferrer, and Enrique Alba (2020). "Intelligent System for the Reduction of Injuries in Archery". In: *International Conference on Optimization and Learning*. Springer, pp. 128–137.

Cintrano, Christian, Javier Ferrer, Manuel López-Ibáñez, and Enrique Alba (2021a). "Hybridization of Racing Methods with Evolutionary Operators for Simulation Optimization of Traffic Lights Programs". In: *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*. Springer, pp. 17–33.

Cintrano, Christian, Daniel H Stolfi, Jamal Toutouh, Francisco Chicano, and Enrique Alba (2016). "CTPATH: a real world system to enable green transportation by optimizing environmentaly friendly routing paths". In: *International Conference on Smart Cities*. Springer International Publishing, pp. 63–75.

Cintrano, Christian and Jamal Toutouh (2022). "Multiobjective electric vehicle charging station locations in a city scale area: Malaga study case". In: *International Conference on the Applications of Evolutionary Computation (Part of EvoStar 2022)*. Springer.

Cintrano, Christian, Jamal Toutouh, and Sergio Nesmachnow (2021). "User-centric multiobjective location of electric vehicle charging stations in a city-scale area". In: *2021 Ivannikov Ispras Open Conference (ISPRAS)*, pp. 89–95.

Risso, Claudio, Christian Cintrano, Jamal Toutouh, and Sergio Nesmachnow (2022). "Exact Approach for Electric Vehicle Charging Infrastructure Location: A Real Case Study in Málaga, Spain". In: *Smart Cities*. Ed. by Sergio Nesmachnow and Luis Hernández Callejo. Cham: Springer International Publishing, pp. 42–57. ISBN: 978-3-030-96753-6.

Stolfi, Daniel H, Christian Cintrano, Francisco Chicano, and Enrique Alba (2018b). "Natural evolution tells us how to best make goods delivery: use vans". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, pp. 308–309.

Toutouh, Jamal, Irene Lebrusán, and Christian Cintrano (2022). "Using Open Data to Analyze Public Bus Service from an Age Perspective: Melilla Case". In: *Smart Cities*. Ed. by Sergio Nesmachnow and Luis Hernández Callejo. Cham: Springer International Publishing, pp. 223–239. ISBN: 978-3-030-96753-6.

Spanish Conferences

Cintrano, Christian and Enrique Alba (2016). "Genetic Algorithms Running into Portable Devices: A First Approach". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 383–393.

Cintrano, Christian, Francisco Chicano, Thomas Stützle, and Enrique Alba (2018). "Studying Solutions of the p-Median Problem for the Location of Public Bike Stations". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 198–208.

Cintrano, Christian, José Ángel Morell, Enrique Alba, Jamal Toutouh, and Enrique Alba (2021b). "Estudio de la calidad de servicio de autobuses empleando datos abiertos: caso de Melilla". In: *Conference of the Spanish Association for Artificial Intelligence*, pp. 968–973.

Cintrano, Christian, Jamal Toutouh, and Enrique Alba (2021b). "Citizen centric optimal electric vehicle charging stations locations in a full city: case of Malaga". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 247–257.

Stolfi, Daniel H, Christian Cintrano, Francisco Chicano, and Enrique Alba (2018a). "An Intelligent Advisor for City Traffic Policies". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 383–393.

Appendix B

Replicability, Reproducibility, and New Open Data

Research is a living entity. Science is continually being improved, challenged, hypotheses are accepted and refuted, new research lines appear, etc. A correct contribution to science must provide the necessary mechanisms to compare and even replicate research results. In this thesis, replicability is essential, so in this appendix, we will describe the different repositories, data sources, and computational details used.

B.1 Hardware platform

The computation platform used in the thesis is composed of a cluster of 144 cores, equipped with three Intel Xeon CPU (E5-2670 v3) at 2.30 GHz and 64 GB memory. The cluster is managed by HTCondor 8.2.7, which allows us to perform parallel independent runs to reduce the overall execution time.

B.2 Problems

B.2.1 Traffic Lights Programs

Programming language GA and DE are implemented in Java using jMetal 5.0 (Nebro, Durillo, and Vergne, 2015). IRACE and the hybrids are implemented in R. We used IRACE version 2.3 as the baseline (Available at `https://cran.r-project.org/package=irace`).

Source code `https://github.com/NEO-Research-Group/irace-ea`

Operating system Ubuntu 20.04 LTE.

B.2.2 Allocation of Bicycles Stations

Programming language C++ compiled with flags `-std=c++11` and `-O3`.

Source code `https://github.com/cintrano/p-median`

Operating system Ubuntu 20.04 LTE.

Data sources We used the TSPLIB library http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/ for the tunning of the algorithms.

Execution parameters We have carried out 30 runs of each algorithm to statistically compare the results obtained. The algorithms were run for 60 CPU seconds and reports the best solution found in each of the runs.

B.2.3 Electric Vehicle Charging Station Location

Programming language Python 3.8 using the library DEAP (De Rainville et al., 2014) to develop the metaheuristic algorithms.

Source code https://github.com/cintrano/EV-CSL

Operating system Ubuntu 20.04 LTE.

Execution parameters We perform 30 independent runs of each algorithm.

B.2.4 Biobjective Shortest Path Problem

Programming language Java 7.

Source code https://github.com/cintrano/Robust-shortest-path

Operating system Ubuntu 16.04 LTE.

Data sources The map and instances can be found in http://christiancintrano.com/research/RBSPproblem/RBSPproblem.html

Execution parameters We perform 29 independent executions of each metaheuristic algorithm.

Appendix C

Programming Languages and Libraries

One of the main positive points of this thesis is using different programming languages. This appendix will briefly review the different programming languages used during the thesis and the most relevant libraries and packages. All the source codes used in this doctoral thesis can be found in the following repository: https://github.com/cintrano.

C.1 Programming Languages

In this PhD thesis, different algorithms and applications have been developed. One of the remarkable aspects of the thesis is the use of multiple programming languages. The use of several programming languages allowed us to obtain high-performance, adaptable, and fast prototyping/development software parts. The programming languages chosen were the following:

- Java: https://www.java.com/es/
- C++: https://www.cplusplus.com/
- Python: https://www.python.org/
- R: https://www.r-project.org/

C.2 Libraries and Packages

The following describes the most relevant software libraries used in the different software artifacts developed in the thesis:

C.2.1 DEAP

DEAP is a framework developed in Python for the design and use of evolutionary algorithms. It is similar to jMetal. The documentation of the framework can be found in https://deap.readthedocs.io/en/master/.

C.2.2 Irace

The IRACE package developed by López-Ibáñez et al. (2016) is a well-known tool for automatic (hyper-)parameter configuration of optimization and machine learning algorithms. In the context of automatic parameter configuration, decision variables correspond to algorithmic parameters, candidate solutions correspond to potential configurations of an algorithm, and evaluating the fitness of a solution requires running the algorithm with a particular parameter configuration on multiple training data or problem instances. However, IRACE can be seen as an optimization method for mixed-integer black-box problem under uncertainty, and, hence, it may be used to tackle simulation-optimization problems.

Algorithm 13 briefly presents IRACE. Initially, a set of solutions are sampled uniformly at random. Then a race is performed to identify the best solutions among the initial set. Each solution is simulated multiple times in different scenarios within a race until there is enough evidence to eliminate it because it is performing worse than the best solution found so far. The race stops once a minimum number of solutions remains alive in the race, the budget assigned to the race is exhausted, or multiple elimination tests fail to eliminate any solution. The solutions that remain alive after the race are called elite. These elite solutions are used to update a sampling model, similar to reinforcement learning, which generates new solutions. New and elite solutions together form a new population that is raced again. In elitist racing, results from previous races are re-used in subsequent races, and elite solutions cannot be eliminated from the race until the contender has been evaluated in as many scenarios as the elite solution. This process is iterated until a maximum budget of simulations is exhausted. The main benefit of the racing strategy is that poor solutions are discarded quickly to avoid wasting simulations. In contrast, good solutions are simulated in many scenarios to estimate their fitness well. Moreover, the elimination test considers the mean value over multiple simulations and the variance and the number of simulations performed so far.

Algorithm 13 Pseudocode of IRACE

Input: Training and testing traffic scenarios.
Output: Best solution found.

$t \leftarrow 1$
$\Theta_t \leftarrow \mathsf{SampleUniformRandomPopulation}$
$\Theta^{\text{elite}} \leftarrow \mathsf{Race}(\Theta_t)$
while $evals < totalEvals$ **do**
    $t \leftarrow t + 1$
    $\mathcal{M} \leftarrow \mathsf{Update}(\Theta^{\text{elite}})$
    $\Theta^{\text{new}} \leftarrow \mathsf{Sample}(\mathcal{M})$
    $\Theta_t \leftarrow \Theta^{\text{new}} \cup \Theta^{\text{elite}}$
    $\Theta^{\text{elite}} \leftarrow \mathsf{Race}(\Theta_t)$
end while
Output: best solution from Θ^{elite}

C.2.3 JMetal

JMetal (Nebro, Durillo, and Vergne, 2015) is one of the most popular libraries for developing evolutionary algorithms. It is written in Java, although there are versions for Python and C++. One of its main advantages lies in the possibility of using algorithms (single and multiobjective), operators, and even problems on which to develop our contributions. The source code of the standard operators used here is publicly available at https://github.com/jMetal/jMetal.

C.2.4 Open Street Maps

Open Street Maps (OpenStreetMap contributors, 2017) is a geographic mapping platform very popular in both the academic and business world. It is based on open source and has a community of thousands of users who are responsible for maintaining and updating all geographic data. Open Street Map offers a multitude of data such as: type of road, length of the street, types of intersections, etc.

In this thesis, we have used Open Street Map as another source of Open Data. In particular, we have extracted the necessary information in each of the works carried out. The transformation of the data sets allowed us, among other benefits, to work in different programming languages. In addition, and as an added value of this thesis, it allowed us to create reusable datasets, thus contributing to the reproducibility of the experimentation performed.

C.2.5 SUMO

Simulator of Urban Mobility (SUMO) (Behrisch et al., 2011; Krajzewicz et al., 2012a) is a microscopic road traffic simulator that provides detailed information about vehicles like velocity, fuel consumption, emissions, journey time, waiting time, etc. It is developed by the German Aerospace Center[1]. The study of realistic scenarios according to real patterns of mobility of the target city is possible due to the fine-grained realistic micro-simulations provided by SUMO. In this thesis, the version of SUMO used was the 0.22.

[1] http://dlr.de/ts/sumo

Bibliography

Abdul-Hak, Mohamad, Youssef Bazzi, Oliver Cordes, Nizar Alholou, and Malok Alamir (Apr. 2013). “Dynamic Eco-Routing Methodology Using Petri Net”. In: *SAE Technical Papers*. Vol. 2. DOI: 10.4271/2013-01-0989.

Alcaide Muñoz, Laura, Manuel Pedro Rodríguez Bolívar, and Cinthia Lorena Villamayor Arellano (2019). “Open Government Initiatives in Spanish Local Governments: An Examination of the State of the Art”. In: *Governance Models for Creating Public Value in Open Data Initiatives*. Ed. by Manuel Pedro Rodríguez Bolívar, Kelvin Joseph Bwalya, and Christopher G. Reddick. Cham: Springer International Publishing, pp. 123–139. ISBN: 978-3-030-14446-3. DOI: 10.1007/978-3-030-14446-3_6. URL: https://doi.org/10.1007/978-3-030-14446-3_6.

Ardakani, M K and M Tavana (2015). “A decremental approach with the A* algorithm for speeding-up the optimization process in dynamic shortest path problems”. In: *Measurement: Journal of the International Measurement Confederation* 60, pp. 299–307.

Arostegui, Marvin A., Sukran N. Kadipasaoglu, and Basheer M. Khumawala (2006). “An empirical comparison of Tabu Search, Simulated Annealing, and Genetic Algorithms for facilities location problems”. In: *International Journal of Production Economics* 103.2, pp. 742 –754. ISSN: 0925-5273.

Avella, Pasquale, Maurizio Boccia, Saverio Salerno, and Igor Vasilyev (2012). “An aggregation heuristic for large scale p-median problem”. In: *Computers & Operations Research* 39.7, pp. 1625–1632. ISSN: 0305-0548.

Ayto. de Malaga (2017). *Area de Movilidad*. URL: http://movilidad.malaga.eu (visited on 01/01/2017).

Barns, Sarah (2018). “Smart cities and urban data platforms: Designing interfaces for smart governance”. In: *City, culture and society* 12, pp. 5–12.

Behrisch, Michael, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz (2011). “SUMO - Simulation of Urban MObility: An Overview”. In: *SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind. Barcelona, Spain, pp. 63–68.

Bel, Germà, Catalina Bolancé, Montserrat Guillén, and Jordi Rosell (2015). “The environmental effects of changing speed limits: A quantile regression approach”. In: *Transportation Research Part D: Transport and Environment* 36, pp. 76–85. ISSN: 1361-9209.

Ben-Tal, Aharon, Laurent El Ghaoui, and Arkadi Nemirovski (2009). *Robust optimization*. Princeton University Press.

Berners-Lee, Tim, James Hendler, and Ora Lassila (2001). “The semantic web”. In: *Scientific american* 284.5, pp. 34–43.

Bian, Caiyun, Hailong Li, Fredrik Wallin, Anders Avelin, Lu Lin, and Zhixin Yu (2019). "Finding the optimal location for public charging stations–a GIS-based MILP approach". In: *Energy Procedia* 158, pp. 6582–6588.

Blum, Christian and Günther R. Raidl (2016). *Hybrid Metaheuristics—Powerful Tools for Optimization*. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, Berlin, Germany: Springer.

Brandstätter, Georg, Michael Kahr, and Markus Leitner (2017). "Determining optimal locations for charging stations of electric car-sharing systems under stochastic demand". In: *Transportation Research Part B: Methodological* 104, pp. 17–35.

Bravo, Yesnier, Javier Ferrer, Gabriel J. Luque, and Enrique Alba (2016). "Smart Mobility by Optimizing the Traffic Lights: A New Tool for Traffic Control Centers". In: *Smart Cities (Smart-CT 2016)*. Ed. by Enrique Alba, Francisco Chicano, and Gabriel J. Luque. LNCS. Cham, Switzerland: Springer, pp. 147–156. DOI: 10.1007/978-3-319-39595-1_15.

Çalık, Hatice and Bernard Fortz (2017). "Location of stations in a one-way electric car sharing system". In: *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, pp. 134–139.

Camero, Andrés, Jamal Toutouh, Daniel H Stolfi, and Enrique Alba (2018). "Evolutionary deep learning for car park occupancy prediction in smart cities". In: *International Conference on Learning and Intelligent Optimization*. Springer, pp. 386–401.

Cao, Z., S. Jiang, J. Zhang, and H. Guo (2017). "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion". In: *IEEE Transactions on Intelligent Transportation Systems* 18.7, pp. 1958–1973.

Cao, Zhiguang, Hongliang Guo, Jie Zhang, Dusit Niyato, and Ulrich Fastenrath (2015). "Finding the Shortest Path in Stochastic Vehicle Routing: A Cardinality Minimization Approach". In: *IEEE Transactions on Intelligent Transportation Systems* PP.99, pp. 1–15. ISSN: 1524-9050. DOI: 10.1109/TITS.2015.2498160.

Carrasco, Carlos and Xavier Sobrepere (2015). "Open government data: An assessment of the Spanish municipal situation". In: *Social science computer review* 33.5, pp. 631–644.

Cavadas, Joana, Gonçalo Homem de Almeida Correia, and Joao Gouveia (2015). "A MIP model for locating slow-charging stations for electric vehicles in urban areas accounting for driver tours". In: *Transportation Research Part E: Logistics and Transportation Review* 75, pp. 188–201.

Chassein, André B. and Marc Goerigk (Aug. 2015). "A new bound for the midpoint solution in minmax regret optimization with an application to the robust shortest path problem". In: *European Journal of Operational Research* 244.3, pp. 739–747. ISSN: 03772217.

Chen, Longbiao, Daqing Zhang, Gang Pan, Xiaojuan Ma, Dingqi Yang, Kostadin Kushlev, Wangsheng Zhang, and Shijian Li (2015). "Bike Sharing Station Placement Leveraging Heterogeneous Urban Open Data". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*. NY, USA: ACM Press, pp. 571–575. ISBN: 9781450335744.

Chen, Peng (Will) and Yu (Marco) Nie (June 2013). "Bicriterion Shortest Path Problem with a General Nonadditive Cost". In: *Procedia - Social and Behavioral Sciences* 80, pp. 553–575. ISSN: 18770428.

Chen, Qun, Mei Liu, and Xinyu Liu (2018). "Bike Fleet Allocation Models for Repositioning in Bike-Sharing Systems". In: *IEEE Intelligent Transportation Systems Magazine* 10.1, pp. 19–29. ISSN: 1939-1390.

Chen, Qun and Tingyuan Sun (Dec. 2015). "A model for the layout of bike stations in public bike-sharing systems". In: *Journal of Advanced Transportation* 49.8, pp. 884–900. ISSN: 01976729.

Cheng, Han, Samer Madanat, and Arpad Horvath (2016). "Planning hierarchical urban transit systems for reductions in greenhouse gas emissions". In: *Transportation Research Part D: Transport and Environment* 49, pp. 44 – 58. ISSN: 1361-9209. DOI: https://doi.org/10.1016/j.trd.2016.08.033. URL: http://www.sciencedirect.com/science/article/pii/S1361920916300803.

Cheng, Jianqiang, Abdel Lisser, and Marc Letournel (June 2013). "Distributionally robust stochastic shortest path problem". In: *Electronic Notes in Discrete Mathematics* 41, pp. 511–518. ISSN: 15710653.

Chira, Camelia, Javier Sedano, José R. Villar, Mónica Cámara, and Emilio Corchado (July 2014). "Urban bicycles renting systems: Modelling and optimization using nature-inspired search methods". In: *Neurocomputing* 135, pp. 98–106. ISSN: 0925-2312.

Cintrano, C, F Chicano, and E Alba (2019). "Facing robustness as a multi-objective problem: A bi-objective shortest path problem in smart regions". In: *Information Sciences* 503, pp. 255–273.

— (2020). "Using metaheuristics for the location of bicycle stations". In: *Expert Systems with Applications* 161, p. 113684.

Cintrano, Christian and Enrique Alba (2016). "Genetic Algorithms Running into Portable Devices: A First Approach". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 383–393.

Cintrano, Christian, Francisco Chicano, and Enrique Alba (2017). "Robust Bi-objective Shortest Path Problem in Real Road Networks". In: *International Conference on Smart Cities*. Springer, Cham, pp. 128–136.

Cintrano, Christian, Francisco Chicano, Thomas Stützle, and Enrique Alba (2018). "Studying Solutions of the p-Median Problem for the Location of Public Bike Stations". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 198–208.

Cintrano, Christian, Javier Ferrer, and Enrique Alba (2020). "Intelligent System for the Reduction of Injuries in Archery". In: *International Conference on Optimization and Learning*. Springer, pp. 128–137.

Cintrano, Christian, Javier Ferrer, Manuel López-Ibáñez, and Enrique Alba (2021a). "Hybridization of Racing Methods with Evolutionary Operators for Simulation Optimization of Traffic Lights Programs". In: *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*. Springer, pp. 17–33.

Cintrano, Christian, Javier Ferrer, Manuel López-Ibáñez, and Enrique Alba (to appear). "Hybridization of Evolutionary Operators with Elitist Iterated Racing for the Simulation Optimization of Traffic Lights Programs". In: *Evolutionary Computation*.

Cintrano, Christian, José Ángel Morell, Enrique Alba, Jamal Toutouh, and Enrique Alba (2021b). "Estudio de la calidad de servicio de autobuses empleando datos abiertos: caso de Melilla". In: *Conference of the Spanish Association for Artificial Intelligence*, pp. 968–973.

Cintrano, Christian, Daniel H Stolfi, Jamal Toutouh, Francisco Chicano, and Enrique Alba (2016). "CTPATH: a real world system to enable green transportation by optimizing environmentaly friendly routing paths". In: *International Conference on Smart Cities*. Springer International Publishing, pp. 63–75.

Cintrano, Christian and Jamal Toutouh (2022). "Multiobjective electric vehicle charging station locations in a city scale area: Malaga study case". In: *International Conference on the Applications of Evolutionary Computation (Part of EvoStar 2022)*. Springer.

Cintrano, Christian, Jamal Toutouh, and Enrique Alba (2021a). "Citizen centric optimal electric vehicle charging stations locations in a full city: case of Malaga". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 247–257.

— (2021b). "Citizen centric optimal electric vehicle charging stations locations in a full city: case of Malaga". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 247–257.

Cintrano, Christian, Jamal Toutouh, and Sergio Nesmachnow (2021). "User-centric multiobjective location of electric vehicle charging stations in a city-scale area". In: *2021 Ivannikov Ispras Open Conference (ISPRAS)*, pp. 89–95.

Coello, Carlos and Margarita Reyes Sierra (2004). "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm". In: *Mexican international conference on artificial intelligence*. Springer, pp. 688–697.

Coffman, Makena, Paul Bernstein, and Sherilyn Wee (2016). "Electric vehicles revisited: a review of factors that affect adoption". In: 37.1, pp. 79–93.

Cohen, Boyd (2012). "What exactly is a smart city". In: *Co. Exist* 19.

Colmenar, J.M., R. Martí, and A. Duarte (2018). "Multi-objective memetic optimization for the bi-objective obnoxious p-median problem". In: *Knowledge-Based Systems* 144, pp. 88 –101. ISSN: 0950-7051.

Cplex, IBM ILOG (2009). "V12. 1: User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53, p. 157.

Crainic, Teodor Gabriel, Michel Gendreau, Pierre Hansen, and Nenad Mladenović (2004). "Cooperative parallel variable neighborhood search for the p-median". In: *Journal of Heuristics* 10.3, pp. 293–314.

Dantrakul, Sittipong, Chulin Likasiri, and Radom Pongvuthithum (June 2014). "Applied p-median and p-center algorithms for facility location problems". In: *Expert Systems with Applications* 41.8, pp. 3596–3604. ISSN: 09574174.

Daskin, Mark S and Kayse Lee Maass (2015). "The p-median problem". In: *Location science*. Springer, pp. 21–45.

De Rainville, François-Michel, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, and Christian Gagné (Feb. 2014). "DEAP: Enabling Nimbler Evolutions". In: *SIGEVOlution* 6.2, 17–26.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (Apr. 2002a). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. ISSN: 1089-778X.

— (Apr. 2002b). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II". In: *Trans. Evol. Comp* 6.2, pp. 182–197. ISSN: 1089-778X.

Deb, Kalyanmoy and S. Agrawal (1999). "A Niched-Penalty Approach for Constraint Handling in Genetic Algorithms". In: *Artificial Neural Nets and Genetic Algorithms (ICANNGA-99)*. Ed. by Andrej Dobnikar, Nigel C. Steele, David W. Pearson, and Rudolf F. Albrecht. Springer Verlag, pp. 235–243. DOI: 10.1007/978-3-7091-6384-9.

DGT (2017). *Tablas estadisticas*. URL: http://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/parque-vehiculos/tablas-estadisticas/ (visited on 12/20/2017).

Dijkstra, Edsger W (1959). "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1.1, pp. 269–271.

Dirección General de Tráfico (DGT) (2014). *El impacto medioambiental del tráfico*. https://www.dgt.es/PEVI/documentos/catalogo_recursos/didacticos/did_adultas/impacto.pdf.

Drezner, Zvi, Jack Brimberg, Nenad Mladenović, and Said Salhi (Oct. 2015). "New heuristic algorithms for solving the planar p-median problem". In: *Computers & Operations Research* 62, pp. 296–304. ISSN: 0305-0548.

— (Nov. 2016). "New local searches for solving the multi-source Weber problem". In: *Annals of Operations Research* 246.1-2, pp. 181–203. ISSN: 0254-5330.

Drezner, Zvi and Horst W Hamacher (2001). *Facility location: applications and theory*. Springer Science & Business Media.

Duque, Daniel, Leonardo Lozano, and Andrés L. Medaglia (May 2015). "An exact method for the biobjective shortest path problem for large-scale road networks". In: *European Journal of Operational Research* 242.3, pp. 788–797. ISSN: 03772217.

Economic, United Nations Department of and Social Affairs (2019). *World Urbanization Prospects: The 2018 Revision*.

Ehrgott, Matthias (2005). *Multicriteria optimization*. Vol. 491. Springer Sci. & Bus. Media.

Ehrgott, Matthias, Jonas Ide, and Anita Schöbel (Nov. 2014). "Minmax robustness for multi-objective optimization problems". In: *European Journal of Operational Research* 239.1, pp. 17–31. ISSN: 03772217.

Erb, Stephan, Moritz Kobitzsch, and Peter Sanders (2014). "Parallel Bi-objective Shortest Paths Using Weight-Balanced B-trees with Bulk Updates". In: *Experimental Algorithms - 13th International Symposium, SEA 2014, Copenhagen, Denmark, June 29 – July 1, 2014. Proceedings*. Ed. by Joachim Gudmundsson and Jyrki Katajainen. Vol. 8504. Lecture Notes in

Computer Science. Cham: Springer International Publishing, pp. 111–122. ISBN: 978-3-319-07958-5. DOI: 10.1007/978-3-319-07959-2.

Estrada, Elsa, Rocio Maciel, Adriana Peña Pérez Negrón, Graciela Lara López, Víctor Larios, and Alberto Ochoa (2018). "Framework for the analysis of smart cities models". In: *International Conference on Software Process Improvement*. Springer, pp. 261–269.

Eur. Com. (2018). *Transport & Environment*. URL: https://web.archive.org/web/20121111033537/http://ec.europa.eu/environment/air/transport/road.htm (visited on 06/01/2018).

European Environment Agency (2018). "Noise pollution is a major environmental health concern in Europe". In: *Collection - Environment and health*. URL: https://www.eea.europa.eu/themes/human/noise/noise-2.

— (2020). *Health risks caused by environmental noise in Europe*. https://www.eea.europa.eu/publications/health-risks-caused-by-environmental.

Fabbiani, Enzo, Sergio Nesmachnow, Jamal Toutouh, Andrei Tchernykh, Arutyun Avetisyan, and Gleb Radchenko (2018). "Analysis of mobility patterns for public transportation and bus stops relocation". In: *Programming and Computer Software* 44.6, pp. 508–525.

Falchetta, Giacomo and Michel Noussan (2021). "Electric vehicle charging network in Europe: An accessibility and deployment trends analysis". In: *Transportation Research Part D: Transport and Environment* 94, p. 102813.

Ferrer, Javier, José García-Nieto, Enrique Alba, and Francisco Chicano (2016). "Intelligent Testing of Traffic Light Programs: Validation in Smart Mobility Scenarios". In: *Mathematical Problems in Engineering* 2016, pp. 1–19. DOI: 10.1155/2016/3871046.

Ferrer, Javier, Manuel López-Ibáñez, and Enrique Alba (2019). "Reliable simulation-optimization of traffic lights in a real-world city". In: *Applied Soft Computing* 78, pp. 697–711. DOI: 10.1016/j.asoc.2019.03.016.

Fonseca, Carlos M. and Peter J. Fleming (1996). "On the performance assessment and comparison of stochastic multiobjective optimizers". In: *Parallel Problem Solving from Nature — PPSN IV*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 584–593. ISBN: 978-3-540-70668-7.

Frade, Inês, Anabela Ribeiro, Gonçalo Gonçalves, and António Pais Antunes (2011). "Optimal location of charging stations for electric vehicles in a neighborhood in Lisbon, Portugal". In: *Transportation Research Record* 2252.1, pp. 91–98.

Gabrel, Virginie, Cécile Murat, and Aurélie Thiele (June 2014). "Recent advances in robust optimization: An overview". In: *European Journal of Operational Research* 235.3, pp. 471–483. ISSN: 03772217.

Galeso, Michael (2016). *Waze: An Easy Guide to the Best Features*. 1st. North Charleston, SC, USA: CreateSpace Independent Publishing Platform. ISBN: 1539660141.

Gan, Mi, Xiaobo Liu, Si Chen, Ying Yan, and Dandan Li (2018). "The identification of truck-related greenhouse gas emissions and critical impact factors in an urban logistics network". In: *Journal of Cleaner Production*. ISSN: 0959-6526.

Garaix, Thierry, Christian Artigues, Dominique Feillet, and Didier Josselin (July 2010). "Vehicle routing problems with alternative paths: An application to on-demand transportation". In: *European Journal of Operational Research* 204.1, pp. 62–75. ISSN: 03772217.

García-Nieto, José, Enrique Alba, and Ana Carolina Olivera (Mar. 2012). "Swarm intelligence for traffic light scheduling: Application to real urban areas". In: *Engineering Applications of Artificial Intelligence* 25.2, pp. 274–283.

García-Nieto, José, Ana Carolina Olivera, and Enrique Alba (Dec. 2013). "Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization". In: *IEEE Transactions on Evolutionary Computation* 17.6, pp. 823–839. DOI: 10.1109/TEVC.2013.2260755.

Ghoseiri, Keivan and Behnam Nadjari (Sept. 2010). "An ant colony optimization algorithm for the bi-objective shortest path problem". In: *Applied Soft Computing* 10.4, pp. 1237–1246. ISSN: 15684946. DOI: 10.1016/j.asoc.2009.09.014.

Goberna, M.A., V. Jeyakumar, G. Li, and J. Vicente-Pérez (2015). "Robust solutions to multi-objective linear programs with uncertain data". In: *European Journal of Operational Research* 242.3, pp. 730–743. ISSN: 03772217.

Gong, D., J. Sun, and Z. Miao (Feb. 2018). "A Set-Based Genetic Algorithm for Interval Many-Objective Optimization Problems". In: *IEEE Transactions on Evolutionary Computation* 22.1, pp. 47–60. ISSN: 1089-778X.

Gong, Dun wei, Na na Qin, and Xiao yan Sun (2011). "Evolutionary algorithms for optimization problems with uncertainties and hybrid indices". In: *Information Sciences* 181.19, pp. 4124 –4138. ISSN: 0020-0255.

Gong, Dunwei, Jing Sun, and Xinfang Ji (2013). "Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems". In: *Information Sciences* 233, pp. 141 –161. ISSN: 0020-0255.

Google Maps (Jan. 12, 2022). Google. URL: https://www.google.com/maps.

Guerreiro, Cristina, Frank de Leeuw, and Valentin Foltescu (2013). *Air quality in Europe - 2013 report*. Tech. rep. European Environment Agency.

Haines, Andy, Anthony J McMichael, Kirk R Smith, Ian Roberts, James Woodcock, Anil Markandya, Ben G Armstrong, Diarmid Campbell-Lendrum, Alan D Dangour, Michael Davies, Nigel Bruce, Cathryn Tonne, Mark Barrett, and Paul Wilkinson (2009). "Public health benefits of strategies to reduce greenhouse-gas emissions: overview and implications for policy makers". In: *The Lancet* 374.9707, pp. 2104–2114. ISSN: 0140-6736.

Hansen, Pierre (1980). "Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter, West Germany, August 20–24, 1979". In: ed. by Günter Fandel and Tomas Gal. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. Bicriterion Path Problems, pp. 109–127. ISBN: 978-3-642-48782-8.

Hart, Peter E, Nils J Nilsson, and Bertram Raphael (1968). "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107.

Hasuike, Takashi (Feb. 2013). "Robust shortest path problem based on a confidence interval in fuzzy bicriteria decision making". In: *Information Sciences* 221, pp. 520–533. ISSN: 00200255.

Hausberger, Stefan, Martin Rexeis, Michael Zallinger, and Raphael Luz (2009). *Emission Factors from the Model PHEM for the HBEFA Version 3*. Tech. rep. I, p. 679.

Hayes, Brian (2008). *Cloud computing*.

Hazir, Öncü, Erdal Erel, and Yavuz Günalay (Mar. 2011). "Robust optimization models for the discrete time/cost trade-off problem". In: *International Journal of Production Economics* 130.1, pp. 87–95. ISSN: 09255273.

Heidrich-Meisner, Verena and Christian Igel (2009). "Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search". In: *Proceedings of the 26th International Conference on Machine Learning, ICML 2009*. Ed. by Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman. New York, NY: ACM Press, pp. 401–408. DOI: 10.1145/1553374.1553426.

Hipogrosso, Silvina and Sergio Nesmachnow (2020). "Analysis of Sustainable Public Transportation and Mobility Recommendations for Montevideo and Parque Rodó Neighborhood". In: *Smart Cities* 3.2, pp. 479–510.

Hu, Shou-Ren and Chao-Tang Liu (Oct. 2014). "An optimal location model for a bicycle sharing program with truck dispatching consideration". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1775–1780. ISBN: 978-1-4799-6078-1.

Ide, Jonas and Anita Schöbel (Oct. 2016). "Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts". In: *OR Spectrum* 38.1, pp. 235–271. ISSN: 0171-6468.

Irawan, Chandra Ade, Said Salhi, and Maria Paola Scaparra (2014). "An adaptive multiphase approach for large unconditional and conditional p-median problems". In: *European Journal of Operational Research* 237.2, pp. 590 –605. ISSN: 0377-2217.

Ishibuchi, Hisao, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima (2015). "Modified distance calculation in generational distance and inverted generational distance". In: *International conference on evolutionary multi-criterion optimization*. Springer, pp. 110–125.

Jamalluddin, Mohd Helmi, Mohd Azrul Jaafar, Mohd Iskandar Amran, Mohd Sharizal Ainul, Aqmar Hamid, Zafirah Mohd Mansor, and Zulkifli Mohd Nopiah (June 2014). "Minimizing travel claims cost with minimal-spanning tree model". In: *Proceeding of the 3rd International Conference on Mathematical Sciences*. Vol. 1602. 1. AIP Publishing, pp. 541–545. DOI: 10.1063/1.4882538.

Jiang, Bin, Xiao XuChao Yang, Renfa Li, and Takao Terano (2013). "Highlights on Practical Applications of Agents and Multi-Agent Systems". In: *International Workshops of PAAMS 2013, Salamanca, Spain, May 22-24, 2013. Proceedings*. Ed. by Juan M. Corchado, Javier Bajo, Jaroslaw Kozlak, Pawel Pawlewski, Jose M. Molina, Vicente Julian, Ricardo Azambuja Silveira, Rainer Unland, and Sylvain Giroux. Vol. 365. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 36–48. ISBN: 978-3-642-38060-0. DOI: 10.1007/978-3-642-38061-7.

Jin, Y. and J. Branke (June 2005). "Evolutionary Optimization in Uncertain Environments - A Survey". English. In: *IEEE Transactions on Evolutionary Computation* 9.3, pp. 303–317. ISSN: 1089-778X.

Kennedy, James and Russell Eberhart (1995). "Particle swarm optimization". In: *Proceedings of ICNN'95-International Conference on Neural Networks*. Vol. 4. IEEE, pp. 1942–1948.

Keshavarz, Esmaile and Esmaile Khorram (Aug. 2009). "A fuzzy shortest path with the highest reliability". In: *Journal of Computational and Applied Mathematics* 230.1, pp. 204–212. ISSN: 03770427.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). "Optimization by Simulated Annealing". In: *Science* 220.4598, pp. 671–680. ISSN: 0036-8075.

Kitchin, Rob (2014). *The data revolution: Big data, open data, data infrastructures and their consequences*. Sage.

Kjeldsen, Tinne Hoff (Nov. 2001). "John von Neumann's Conception of the Minimax Theorem: A Journey Through Different Mathematical Contexts". In: *Archive for History of Exact Sciences* 56.1, pp. 39–68. ISSN: 1432-0657.

Kloimüllner, Christian and Günther R. Raidl (2017). "Hierarchical Clustering and Multilevel Refinement for the Bike-Sharing Station Planning Problem". In: *Learning and Intelligent Optimization*. Ed. by Roberto Battiti, Dmitri E. Kvasov, and Yaroslav D. Sergeyev. Cham: Springer International Publishing, pp. 150–165. ISBN: 978-3-319-69404-7.

Knowles, Joshua D, Lothar Thiele, and Eckart Zitzler (2006). "A tutorial on the performance assessment of stochastic multiobjective optimizers". In: *TIK-report* 214.

Kousky, Carolyn and Stephen H Schneider (2003). "Global climate policy: will cities lead the way?" In: *Climate Policy* 3.4, pp. 359–372. ISSN: 1469-3062.

Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch, and Laura Bieker (2012a). "Recent development and applications of SUMO - Simulation of Urban MObility". In: *International Journal On Advances in Systems and Measurements* 5.3-4, pp. 128–138.

— (2012b). "Recent Development and Applications of SUMO - Simulation of Urban MObility". In: *International Journal On Advances in Systems and Measurements* 5.3, pp. 128–138. ISSN: 1942-261x.

Kuhn, K., A. Raith, M. Schmidt, and A. Schöbel (July 2016). "Bi-objective robust optimisation". In: *European Journal of Operational Research* 252.2, pp. 418–431. ISSN: 03772217.

Kumar, Rajeev and Kumar Alok (2020). "Adoption of electric vehicle: A literature review and prospects for sustainability". In: *Journal of Cleaner Production* 253, p. 119911.

Lebrusán, Irene and Jamal Toutouh (2020a). "Car restriction policies for better urban health: a low emission zone in Madrid, Spain". In: *Air Quality, Atmosphere & Health* 14 (3), pp. 333–342.

— (2020b). "Using Smart City Tools to Evaluate the Effectiveness of a Low Emissions Zone in Spain: Madrid Central". In: *Smart Cities* 3.2, pp. 456–478.

— (2021). "Smart City Tools to Evaluate Age-Healthy Environments". In: *Smart Cities*. Ed. by Sergio Nesmachnow and Luis Hernández Callejo. Cham: Springer International Publishing, pp. 285–301. ISBN: 978-3-030-69136-3.

Li, K., Q. Zhang, S. Kwong, M. Li, and R. Wang (Dec. 2014). "Stable Matching-Based Selection in Evolutionary Multiobjective Optimization". In: *IEEE Transactions on Evolutionary Computation* 18.6, pp. 909–923. ISSN: 1089-778X.

Li, Zhiyi, Mohammad Shahidehpour, Shay Bahramirad, and Amin Khodaei (2016). "Optimizing Traffic Signal Settings in Smart Cities". In: *IEEE Transactions on Smart Grid* 3053.4, pp. 1–1. DOI: 10.1109/TSG.2016.2526032.

Lin, Haiyang, Caiyun Bian, Hailong Li, Qie Sun, and Ronald Wennersten (2018). "Optimal siting and sizing of public charging stations in urban area". In: *2018 Joint International Conference on Energy, Ecology and Environment (ICEEE 2018) and International Conference on Electric and Intelligent Vehicles (ICEIV 2018)*, p. 7.

Lin, Jenn-Rong, Ta-Hui Yang, and Yu-Chung Chang (2013). "A hub location inventory model for bicycle sharing system design: Formulation and solution". In: *Computers & Industrial Engineering* 65.1, pp. 77–86. ISSN: 0360-8352.

Liu, Junming, Qiao Li, Meng Qu, Weiwei Chen, Jingyuan Yang, Hui Xiong, Hao Zhong, and Yanjie Fu (Nov. 2015). "Station Site Optimization in Bike Sharing Systems". In: *2015 IEEE International Conference on Data Mining*. IEEE, pp. 883–888. ISBN: 978-1-4673-9504-5.

Loomis, Dana, Yann Grosse, Béatrice Lauby-Secretan, Fatiha El Ghissassi, Véronique Bouvard, Lamia Benbrahim-Tallaa, Neela Guha, Robert Baan, Heidi Mattock, and Kurt Straif (2013). "The carcinogenicity of outdoor air pollution". In: *The Lancet Oncology* 14.13, pp. 1262–1263.

López-Ibáñez, Manuel, Juergen Branke, and Luís Paquete (2021). "Reproducibility in evolutionary computation". In: *ACM Transactions on Evolutionary Learning and Optimization* 1.4, pp. 1–21.

López-Ibáñez, Manuel, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle (2016). "The irace package: Iterated racing for automatic algorithm configuration". In: *Operations Research Perspectives* 3, pp. 43–58. ISSN: 2214-7160.

Lourenço, Helena R, Olivier C Martin, and Thomas Stützle (2003). "Iterated local search". In: *Handbook of metaheuristics*. Springer, pp. 320–353.

Mahmod, M, B van Arem, R Pueboobpaphan, and R de Lange (Mar. 2013). "Reducing local traffic emissions at urban intersection using ITS countermeasures". In: *Intelligent Transport Systems, IET* 7.1, pp. 78–86. ISSN: 1751-956X.

Massobrio, Renzo and Sergio Nesmachnow (2020). “Urban Mobility Data Analysis for Public Transportation Systems: A Case Study in Montevideo, Uruguay”. In: *Applied Sciences* 10.16, p. 5400.

Massobrio, Renzo, Jamal Toutouh, Sergio Nesmachnow, and Enrique Alba (2017). “Infrastructure deployment in vehicular communication networks using a parallel multiobjective evolutionary algorithm”. In: *International Journal of Intelligent Systems* 32.8, pp. 801–829.

Megiddot, Nimrod and Kenneth J Supowits (1984). “On the Complexity of Some Common Geometric Location Problems”. In: *SIAM Journal on Computing* 13.1.

Meijer, Albert and Suzanne Potjer (2018). “Citizen-generated open data: An explorative analysis of 25 cases”. In: *Government Information Quarterly* 35.4, pp. 613–621. ISSN: 0740-624X. DOI: https://doi.org/10.1016/j.giq.2018.10.004.

Mickael Randour, Jean-François Raskin, Ocan Sankur (2015). “Variations on the Stochastic Shortest Path Problem”. In: *Verification, Model Checking, and Abstract Interpretation*. Ed. by Deepak D’Souza, Akash Lal, and Kim Guldstrand Larsen. Vol. 8931. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–18. ISBN: 978-3-662-46080-1.

Millard-Ball, Adam (2012). “Do city climate plans reduce emissions?” In: *Journal of Urban Economics* 71.3, pp. 289–311. ISSN: 0094-1190.

Mirjalili, Seyedali, Andrew Lewis, and Sanaz Mostaghim (Oct. 2015). “Confidence measure: A novel metric for robust meta-heuristic optimisation algorithms”. In: *Information Sciences* 317, pp. 114–142. ISSN: 00200255.

Mladenović, N. and P. Hansen (Nov. 1997). “Variable neighborhood search”. In: *Computers & Operations Research* 24.11, pp. 1097–1100. ISSN: 0305-0548.

Mladenović, Nenad, Jack Brimberg, Pierre Hansen, and José A. Moreno-Pérez (June 2007). “The p-median problem: A survey of metaheuristic approaches”. In: *European Journal of Operational Research* 179.3, pp. 927–939. ISSN: 0377-2217.

Nebro, Antonio J., Juan J. Durillo, and Matthieu Vergne (2015). “Redesigning the jMetal Multi-Objective Optimization Framework”. In: *GECCO (Companion)*. Ed. by Juan Luis Jiménez Laredo, Sara Silva, and Anna I. Esparcia-Alcázar. New York, NY: ACM Press, pp. 1093–1100.

Nesmachnow, Sergio, Diego Gabriel Rossit, and Jamal Toutouh (2018). “Comparison of Multiobjective Evolutionary Algorithms for Prioritized Urban Waste Collection in Montevideo, Uruguay”. In: *Electronic Notes in Discrete Mathematics* 69, pp. 93–100.

OpenStreetMap contributors (2017). *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org.

Park, Chung and So Young Sohn (Nov. 2017). “An optimization approach for the placement of bicycle-sharing stations to reduce short car trips: An application to the city of Seoul”. In: *Transportation Research Part A: Policy and Practice* 105, pp. 154–166. ISSN: 0965-8564.

Pascoal, Marta M.B. and Marisa Resende (Aug. 2014). “The minmax regret robust shortest path problem in a finite multi-scenario model”. In: *Applied Mathematics and Computation* 241, pp. 88–111. ISSN: 00963003.

Pascoal, Marta M.B. and Marisa Resende (Dec. 2015). "Dynamic preprocessing for the minmax regret robust shortest path problem with finite multi-scenarios". In: *Discrete Optimization*. ISSN: 15725286.

Péres, Matias, Germán Ruiz, Sergio Nesmachnow, and Ana C. Olivera (2018a). "Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay". In: *Applied Soft Computing* 70, pp. 472–485.

— (2018b). "Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay". In: *Applied Soft Computing* 70, pp. 472–485.

Pérez-Martínez, Pedro José, María de Fátima Andrade, and Regina Maura de Miranda (2017). "Heavy truck restrictions and air quality implications in São Paulo, Brazil". In: *J. Environ. Manage.* 202, pp. 55 –68. ISSN: 0301-4797.

Price, Kenneth, Rainer M. Storn, and Jouni A. Lampinen (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Springer, New York, NY. DOI: 10.1007/3-540-31306-0.

Price, Kenneth V (2013). "Differential evolution". In: *Handbook of optimization*. Springer, pp. 187–214.

Rabie, Hassan M, Ihab A El-Khodary, and Assem A Tharwat (2014). "Particle swarm optimization algorithm for the continuous p-median location problems". In: *2014 10th International Computer Engineering Conference (ICENCO)*. IEEE, pp. 81–86.

Raith, A., M. Schmidt, A. Schöbel, and L. Thom (2018). "Multi-objective minmax robust combinatorial optimization with cardinality-constrained uncertainty". In: *European Journal of Operational Research* 267.2, pp. 628–642.

Reese, Josh (2006). "Methods for Solving the p-Median Problem: An Annotated Bibliography". In:

Reinelt, Gerhard (1991). "TSPLIB—A traveling salesman problem library". In: *ORSA journal on computing* 3.4, pp. 376–384.

Riquelme, Nery, Christian Von Lücken, and Benjamin Baran (2015). "Performance metrics in multi-objective optimization". In: *Computing Conference (CLEI), 2015 Latin American*. IEEE, pp. 1–11.

Risso, C., C. Cintrano, J. Toutouh, and S. Nesmachnow (2021). "Exact approach for electric vehicle charging infrastructure location: a real case Study in Málaga, Spain". In: *Smart Cities*. Springer International Publishing.

Risso, Claudio, Christian Cintrano, Jamal Toutouh, and Sergio Nesmachnow (2022). "Exact Approach for Electric Vehicle Charging Infrastructure Location: A Real Case Study in Málaga, Spain". In: *Smart Cities*. Ed. by Sergio Nesmachnow and Luis Hernández Callejo. Cham: Springer International Publishing, pp. 42–57. ISBN: 978-3-030-96753-6.

Rossit, Diego Gabriel, Jamal Toutouh, and Sergio Nesmachnow (2020). "Exact and heuristic approaches for multi-objective garbage accumulation points location in real scenarios". In: *Waste Management* 105, pp. 467–481.

Sánchez, Javier, Manuel Galán, and Enrique Rubio (2008). "Applying a traffic lights evolutionary optimization technique to a real case: "Las Ramblas" area in Santa Cruz de Tenerife". In: *IEEE Transactions on Evolutionary Computation* 12.1, pp. 25–40.

Sánchez-Medina, J. J., M. J. Galán-Moreno, and E. Rubio-Royo (Mar. 2010). "Traffic Signal Optimization in "La Almozara" District in Saragossa Under Congestion Conditions, Using Genetic Algorithms, Traffic Microsimulation, and Cluster Computing". In: *IEEE Transactions on Intelligent Transportation Systems* 11.1, pp. 132–141. ISSN: 1524-9050. DOI: 10.1109/TITS.2009.2034383.

Sedeño-Noda, Antonio and Andrea Raith (May 2015). "A Dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem". In: *Computers & Operations Research* 57, pp. 83–94. ISSN: 03050548.

Serafini, Paolo (1987). "Some Considerations about Computational Complexity for Multi Objective Combinatorial Problems". In: *Recent Advances and Historical Development of Vector Optimization: Proceedings of an International Conference on Vector Optimization Held at the Technical University of Darmstadt, FRG, August 4–7, 1986*. Ed. by Johannes Jahn and Werner Krabs. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 222–232. ISBN: 978-3-642-46618-2.

Sheng, Y and Y Gao (2016). "Shortest path problem of uncertain random network". In: *Computers and Industrial Engineering* 99, pp. 97–105.

Sheskin, David J. (2011). *Handbook of Parametric and Nonparametric Statistical Procedures*. 5th. Chapman & Hall/CRC.

Sinaeepourfard, Amir, Jordi Garcia, Xavier Masip-Bruin, Eva Marín-Tordera, Jordi Cirera, Glòria Grau, and Francesc Casaus (2016). "Estimating Smart City sensors data generation". In: *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. IEEE, pp. 1–8.

Singhvi, Divya, Somya Singhvi, Peter I Frazier, Shane G Henderson, Eoin O 'mahony, David B Shmoys, and Dawn B Woodard (2015). "Predicting Bike Usage for New York City's Bike Sharing System". In:

Soni, Nikhil and Neetishree Soni (2016). "Benefits of pedestrianization and warrants to pedestrianize an area". In: *Land use policy* 57, pp. 139–150.

Stolfi, Daniel H. and Enrique Alba (2014). "Red Swarm: Reducing travel times in smart cities by using bio-inspired algorithms". In: *Applied Soft Computing* 24, pp. 181–195. DOI: 10.1016/j.asoc.2014.07.014.

— (2015a). "An Evolutionary Algorithm to Generate Real Urban Traffic Flows". In: *Advances in Artificial Intelligence, CAEPIA 2015*. Ed. by José M. Puerta, José A. Gámez, Bernabe Dorronsoro, Edurne Barrenechea, Alicia Troncoso, Bruno Baruque, and Mikel Galar. Vol. 9422. LNCS. Springer, pp. 332–343. DOI: 10.1007/978-3-319-24598-0_30.

Stolfi, Daniel H and Enrique Alba (2015b). "An Evolutionary Algorithm to Generate Real Urban Traffic Flows". In: *Advances in Artificial Intelligence*. Vol. 9422. Lecture Notes in Computer Science. Springer International Publishing, pp. 332–343. ISBN: 978-3-319-24597-3.

Stolfi, Daniel H, Christian Cintrano, Francisco Chicano, and Enrique Alba (2018a). "An Intelligent Advisor for City Traffic Policies". In: *Conference of the Spanish Association for Artificial Intelligence*. Springer, pp. 383–393.

Stolfi, Daniel H, Christian Cintrano, Francisco Chicano, and Enrique Alba (2018b). "Natural evolution tells us how to best make goods delivery: use vans". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, pp. 308–309.

Storn, Rainer and Kenneth Price (1997). "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces". In: *Journal of global optimization* 11.4, pp. 341–359.

Sun, Jing, Dunwei Gong, Xiaojun Zeng, and Na Geng (2018). "An ensemble framework for assessing solutions of interval programming problems". In: *Information Sciences* 436-437, pp. 146 –161. ISSN: 0020-0255.

Syswerda, Gilbert (1989). "Uniform Crossover in Genetic Algorithms". In: *Proc. of the Third Int. Conf. on Genetic Algorithms*. Ed. by J. David Schaffer. Morgan Kaufmann Publishers, San Mateo, CA, pp. 2–9.

Talbi, El-Ghazali (2009). *Metaheuristics: from design to implementation*. Vol. 74. John Wiley & Sons.

Tatomir, Bogdan, Leon J. M. Rothkrantz, and Adriana C. Suson (Dec. 2009). "Travel time prediction for dynamic routing using Ant Based Control". In: *Proceedings of the 2009 Winter Simulation Conference (WSC)*. IEEE, pp. 1069–1078. ISBN: 978-1-4244-5770-0. DOI: 10.1109/WSC.2009.5429648.

Teklu, Fitsum, Agachai Sumalee, and David Watling (Jan. 2007). "A Genetic Algorithm Approach for Optimizing Traffic Control Signals Considering Routing". In: *Computer-Aided Civil and Infrastructure Engineering* 22.1, pp. 31–43. DOI: 10.1111/j.1467-8667.2006.00468.x.

Teo, K. T. K., W. Y. Kow, and Y. K. Chin (2010). "Optimization of traffic flow within an urban traffic light intersection with genetic algorithm". In: *Proceedings - 2nd International Conference on Computational Intelligence, Modelling and Simulation, CIMSim 2010*. IEEE. IEEE Press, pp. 172–177.

Thomas H. Cormen Charles E. Leiserson, Ronald L. Rivest and Clifford Stein (2009). *Intoduction to Algorithms*. 3rd. MIT Press.

TNS Opinion & Social (2013). *Attitudes of Europeans towards urban mobility*. Tech. rep. June.

Toutouh, Jamal, Irene Lebrusán, and Christian Cintrano (2022). "Using Open Data to Analyze Public Bus Service from an Age Perspective: Melilla Case". In: *Smart Cities*. Ed. by Sergio Nesmachnow and Luis Hernández Callejo. Cham: Springer International Publishing, pp. 223–239. ISBN: 978-3-030-96753-6.

Toutouh, Jamal, Diego Rossit, and Sergio Nesmachnow (2020). "Soft computing methods for multiobjective location of garbage accumulation points in smart cities". In: *Annals of Mathematics and Artificial Intelligence* 88.1, pp. 105–131.

United Nations, Assembly General (2015). "Sustainable development goals". In: *SDGs Transform Our World* 2030.

Van Veldhuizen, David (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Institute of Technology.

Vargha, A. and H. D. Delaney (2000). "A critique and improvement of the CL common language effect size statistics of McGraw and Wong". In: *Journal of Educational and Behavioral Statistics* 25.2, pp. 101–132.

Visser, Johan, Toshinori Nemoto, and Michael Browne (2014). "Home Delivery and the Impacts on Urban Freight Transport: A Review". In: *Procedia. Soc. Behav. Sci.* 125, pp. 15–27. ISSN: 1877-0428.

Wagner, Sebastian, Markus Götzinger, and Dirk Neumann (2013). "Optimal location of charging stations in smart cities: A points of interest based approach". In:

Whitaker, R A (1983). "A Fast Algorithm For The Greedy Interchange For Large-Scale Clustering And Median Location Problems". In: *INFOR: Information Systems and Operational Research* 21.2, pp. 95–108.

Whitley, Darrell (1994). "A genetic algorithm tutorial". In: *Statistics and computing* 4.2, pp. 65–85.

Yu, Vincent F. and Shin-Yu Lin (2015). "A simulated annealing heuristic for the open location-routing problem". In: *Computers & Operations Research* 62, pp. 184 –196. ISSN: 0305-0548.

Zarandi, MH Fazel, S Davari, and SA Haddad Sisakht (2015). "An empirical comparison of simulated annealing and iterated local search for the hierarchical single allocation hub median location problem". In: *Scientia Iranica. Transaction E, Industrial Engineering* 22.3, p. 1203.

Zhang, Jingqiao and Arthur C. Sanderson (2009). "JADE: adaptive differential evolution with optional external archive". In: *IEEE Transactions on Evolutionary Computation* 13.5, pp. 945–958. DOI: 10.1109/TEVC.2009.2014613.

Zhang, Q. and H. Li (Dec. 2007). "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition". In: *IEEE Transactions on Evolutionary Computation* 11.6, pp. 712–731. ISSN: 1089-778X.

Zhang, Yan, Pei Liu, Lixing Yang, and Yuan Gao (Apr. 2015). "A bi-objective model for uncertain multi-modal shortest path problems". In: *Journal of Uncertainty Analysis and Applications* 3.1, p. 8. ISSN: 2195-5468.

Zitzler, E., M. Laumanns, and L. Thiele (2001a). In: *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*. Ed. by K. Giannakoglou, D. Tsahalis, J. Périaux, K. Papailiou, and T. Fogarty, pp. 95–100.

Zitzler, Eckart, Marco Laumanns, and Lothar Thiele (2001b). "SPEA2: Improving the strength Pareto evolutionary algorithm". In: *TIK-report* 103.

Zitzler, Eckart and Lothar Thiele (1998a). "An evolutionary algorithm for multiobjective optimization: The strength pareto approach". In: *TIK-report* 43.

— (1998b). "Multiobjective optimization using evolutionary algorithms—a comparative case study". In: *International conference on parallel problem solving from nature*. Springer, pp. 292–301.

Acronyms

BSP Biobjective Shortest Path. xvi, 13, 14, 101–106

DE Differential Evolution. 38, 48, 50, 53–60, 129

EA Evolutionary Algorithm. 33, 48, 50, 56, 57

EVCS Electric Vehicle Charging Stations. xvi, 90–94, 99, 100

GA Genetic Algorithm. 15, 37, 38, 48, 50, 53–60, 76–82, 84–86

GD Generational Distance. 29, 30, 66–68

GD$^+$ Generational Distance Plus. 29, 30

GIS Geographic Information System. 17

HDV Heavy Duty Vehicles. 15, 61, 62, 67, 71

HV Hypervolume. xvi, xvii, xx, 29, 30, 66–68, 109, 111–113, 115, 116

IGD Inverse Generational Distance. xx, 29, 30, 66–68, 109, 112, 113, 116

IGD$^+$ Inverse Generational Distance Plus. 29, 30

ILP Integer Linear Programming. 89, 130

ILS Iterated Local Search. 15, 33, 74, 76–80, 83, 85, 86

LDV Light Duty Vehicles. 15, 61, 70, 71

MILP Mixed-Integer Linear Programming. 89

MO-EVCS-L Multiobjective Electric Vehicle Charging Stations Location. 90, 91, 94, 96–100

MOEA Multiobjective Evolutionary Algorithm. xx, 41, 93–98, 100

MOEA/D Multiobjective Evolutionary Algorithm Based on Decomposition. 41, 102, 107–109, 111, 112, 116, 117, 122

NSGA-II Non-dominated Sorting Genetic Algorithm II. xvi, xx, 39, 41, 64, 70, 71, 93, 95–100, 102, 107–109, 112, 116, 117, 122

OD Open Data. 2, 4, 19, 20, 22, 127, 141

PSO Particle Swarm Optimization. xvi, 39, 76–81, 85, 86

RBSP Robust Biobjective Shortest Path. xvi, 14, 104–106, 108, 111

RHV Relative Hypervolume. 29, 30

RS Random Search. 95, 96, 98

SA Simulated Annealing. 15, 37, 74, 76–80, 83, 85, 86

SC Smart City. xix, 2, 4, 11, 12, 19

SDG Sustainable Development Goals. xv, 1, 3, 4, 9

SM Smart Mobility. 2–4, 12, 13, 127

SPEA2 Strength Pareto Evolutionary Algorithm 2. xvi, xx, 41, 43, 93, 95–100

TLP Traffic Light Program. 49, 59

TLS Traffic Light Scheduling. 48–55, 58, 59

TT Travel Time. xvii, xx, 61, 62, 65–67, 69–71, 104, 106, 110, 112, 114, 116–118, 120

VNS Variable Neighborhood Search. 15, 37, 74, 76–80, 83, 85, 86