University Institute of Lisbon

Department of Information Science and Technology

A Graph-Based Approach for Sustainable Walking Tour Recommendations: the Case of Lisbon Overcrowding

Ana Rita Henrique Peixoto

Dissertation in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science and Business Management

Co-supervisor
Fernando Brito e Abreu, Associate Professor
ISCTE-IUL
Co-supervisor
Ana Maria de Almeida, Assistant Professor
ISCTE-IUL

December, 2019

[This page has been intentionally left blank]

to my mother

[This page has been intentionally left blank]

Abstract

Motivation: Mass tourism brought problems of carrying capacity in city management. More and more tourists flock to the most famous zones, thereby causing overcrowding situations, while other sustainable points of interest (POIs) are under-visited.

Goal: Allow local tourism managing authorities to assemble a database of georeferenced sustainable POIs. Then, combine the latter with local crowding data and implement a walking tour recommender system.

Proposal: A web platform to experts adds, in an intuitive way by using a map,POIs with sustainable data. Creating a new database of Lisbon (case of study) sustainable POIs. Implement a tour generator graph-algorithm that receives: user preferences, tour constraints, sustainable POIs and crowd data. Providing a customize tour, that obeys the domain constraints, suggests sustainable POIs and avoids the more crowded areas. Solving a multicriteria shortest path problem.

Conclusion: Evidence is provided on the feasibility of computing walking tour recommendations, meeting multiple and complex constraints, namely by promoting sustainability and mitigating crowding, using a graph search algorithm.

Keywords: Smart and Sustainable Tourism, Graph Algorithms, Tour Recommendation System.

[This page has been intentionally left blank]

Resumo

Motivação: O turismo em massa trouxe problemas de controlo da capacidade de carga na gestão das cidades. Os turistas, em número crescente, aglomeram-se nas zonas mais famosas, causando aí situações de sobrelotação, enquanto outros pontos de interesse (POIs, em Inglês) sustentáveis são sub-visitados.

Objetivo: Permitir que as autoridades gestoras do turismo local montem uma base de dados de POIs sustentáveis, georreferenciados. Em seguida, combinar estes últimos com os dados de aglomeração local e implementar um sistema de recomendação de passeios turísticos pedestres.

Proposta: Uma plataforma web para que o especialistas adicionem, de forma intuitiva através de um mapa, os pontos de interesse sustentáveis. Implementar um algoritmo de grafos, que gera caminhos e que recebe: as preferências do utilizador, as restrições do domínio do caminho, pontos de interesse sustentáveis e dados de congestionamento. Fornecendo assim, um caminho personalizado que obdece às restrições, sugere pontos de interesse sustentáveis e evita as áreas mais movimentadas. Deste modo, resolve o problema do caminho mais curto com multicriterias.

Conclusão: São fornecidas evidências sobre a viabilidade de computar recomendações de passeios turísticos a pé, atendendo a restrições múltiplas e complexas, nomeadamente promovendo a sustentabilidade e mitigando a superlotação, usando um algoritmo de pesquisa em grafos.

Palavras-chave: Turismo inteligente e sustentável, Algoritmos de grafos, Sistema de Recomendação de Caminhos.

[This page has been intentionally left blank]

Acknowledgements

I thank to Professor Fernando Brito e Abreu, to the deposited trust, since he first invited me to join the team of the "Sustainable Tourism Crowding" research project.
I thank to Professor Ana Maria de Almeida, for believing in my work, even when I doubted. For always being present to help me, and for having accepted to co-supervise my thesis work.
I thank the following professors for the provided knowledge in some specific areas of this work: Professor Américo Rio, Professor Jorge Louçã, and Professor Marta Pascoal.
I thank to my coworkers, the "dream team", that fought by my side on this research project: João Virtudes and Duarte Almeida.
I thank my boyfriend Rúben Silva, that also belonged to the "dream team", for the patience and for the massive constant support.
I would like to thank my family for all the love.
I thank my father Carlos Peixoto for all the life expertise provided.
And to my little sister Francisca, for showing me that I was capable of anything, because I have already done "a lot of hard stuff".
And finally, I thank my friends for being present in all moments.

Contents

List of Figures

List of Tables

Abbreviations

API - Application Programming Interface
BSP - Biobjective (Bicriteria) Shortest Path
HAGO - Historical narrative caching App for Going Out
HTTP - Hypertext Transfer Protocol
JSON - Javascript Object Notation
MOP - Mixed Orienteering Problem
MP - Multi-Period
MPP - Multiobjective (Multicriteria) Path Problem
MSP - Multiobjective (Multicriteria) Shortest Path
OP - Orienteering Problem
PHP - Hypertext Preprocessor
POI - Point of Interest
QoE - Quality of Experience
REST - Representational State Transfer
RS - Recommender System
SDG - Sustainable Development Goals
SP - Shortest Path
STC - Sustainable Tourism Crowding
TD - Time-Dependent
TRS - Tour Recommender System
TSP - Travelling Salesman Problem
TTDP - Tourist Trip Design Problem
TW - Time-Windows

[This page has been intentionally left blank]

Chapter 1

Introduction

1.1 Background and Motivation

Tourism is apparently a success story for Portugal, as proven by the fifteen World Travel Awards[1] won in the category of *World Winners 2018*. Among those, is the important *World's Leading Destination 2018* award, for Portugal as a whole, and the *World's Leading City Destination 2018* award, for Lisbon in particular. Tourism creates jobs and its continuous growth has turned it in a crucial part of the Portuguese economy [Statistics Portugal2018]. As such, it has been fulfilling the United Nations Sustainable Development Goal (SDG) number eight – *Decent work and economic growth*.

However, tourism is also creating serious sustainability and social consequences, specially on overcrowded locations, such as historical neighborhoods. Overcrowding happens when the number of visitors at a given location increases to levels that cannot be physically supported, or resources are not enough for the tourists and locals. Overcrowding overloads infrastructures (e.g. causing security and urban cleanliness problems), damages nature and threatens culture and heritage. It degrades tourist's experience (e.g. long queues, reduced authenticity) and local residents' quality of life (e.g. loss of privacy, increased noise). According to [Peeters et al.2018], that addresses the complex phenomenon of overtourism[2] in the EU, avoiding it requires custom-made policies in cooperation between destinations' stakeholders and policymakers.

[1] https://www.worldtravelawards.com/winners/2018/world

[2] overcrowding due to tourism

The work presented in this thesis is part of a larger research project called *Sustainable Tourism Crowding* (STC), aiming to mitigate overcrowding and promote tourism sustainability, by making good use of the available resources. Several tactics have been identified for mitigating and/or preventing overcrowding [Dichter and Gloria2017]. STC's tactic is to spread visitors geographically, avoiding congested Points of Interest (POI) and promoting more sustainable ones. The expected overall effect will be a decrease of overcrowding situations, along with an increase in tourist's quality of experience and location's sustainability, aligned with the United Nations SDG number eleven – *Sustainable cities and communities*. The STC project pilot will be deployed at the parish of *Santa Maria Maior*, a very touristic historical neighborhood in Lisbon, that is suffering from serious overcrowding problems. The global architecture of this project will be described in section 3.1 (Sustainable Tourism Crowding Project). However, other destinations are suffering with the overcrowding, the project solution presented could also be applied in those places.

1.2 Research Questions

The questions that the present dissertation aims to answer are:

1. Is it possible to create a database of sustainable POIs with knowledge from the parish council workers?
2. Is it possible to implement a tour generator graph-algorithm, that solves multicriteria shortest path with additional constraints (tour domain and tourist preferences)?

1.3 Research Objectives

The objectives of the present dissertation have the intention of answer to the previous questions.
To answer question 1, the objectives are:

- Creation of a POIs database, to store the sustainable data;

- Develop a web platform, to the parish council workers add the sustainable data;
- Assemble a REST API, to support the communications between the database and the web platform.

To answer question 2, the objectives are:

- Define the tour constraints and user preferences, in the multicriteria shortest path domain;
- Represent in a graph structure the Lisbon network (case of study);
- Fill the graph structure with nodes representing sustainable POIs;
- Develop a represent strategy for the crowding data on the graph;
- Implement the algorithm that combines all the requirements, providing a customized tour that obeys the problem constraints.

1.4 Contributions

We contribute, with the present work, in four areas:

1. Literature review, with a systematization of the capacity control strategies: it is of our understanding that there was not in the literature a list that compares the capacity control strategies for tourist locations and its advantages, (more details on Section 2.4 and Section 2.6);
2. In the Tourist Trip Design Problem (TTDP), with a new statement problem, MTDOPTW (more details on Section 3.3.1);
3. A new constraint added on the domain of Tour Recommender Systems (TRS), the sustainability, suggesting sustainable POIs and avoid people crowding (more details on Section 3.2);
4. In the STC research project, with the implementation of the main functionalities of the POINT (Chapter 4) and ROUTE (Chapter 5) microservices.

1.5 Methodology

The design science research methodology [Vaishnavi and Kuechler2004] has been followed in the work underlying this dissertation. Accordingly, the research process steps and corresponding outputs are:

1. Awareness of Problem: As noted, the evolution of tourism brought a massive amount of tourists in specific locals, which causes capacity control problems in cities' management. There is the need to disperse tourists, know and learn their patterns, and promote under-visited but sustainable points of interest. Those are not gathered in a data set to suggest to the tourists. Even, if we knew where the crowded areas are, there is the need to inform tourists and disperse them over other areas.

2. Suggestion: A database of sustainable points of interest. And a tours recommender system which provides tours suggesting those sustainable points of interest and avoid crowd areas.

3. Development: First a web platform to sustainable experts add the points of interest with the sustainable data. And second a tour generator algorithm.

4. Evaluation: The criteria for the web platform, sets on the fulfillment of the requirements: an easy way of adding POIs and its data. For the algorithm, if the tour suggests sustainable POIs and if adjusts avoiding the crowded areas.

5. Conclusion: The accomplishment of the web platform and is inherent POIs database. And a tour recommendation system, that uses successfully a graph-algorithm as a tour generator. Those tours match tourist preferences and promote sustainability through the visitation of sustainable POIs and avoidance of crowded areas.

1.6 Dissertation Outline

Chapter 2 (Literature Review), makes an overview of sustainability concepts, on the work on gathering crowding data and describes the capacity control strategies, with the focus on tour recommendation systems. Chapter 3 (Framework, Decisions and Theory) describes the STC project, clarifying the framework of the work proposed in this dissertation and consolidates the main theoretical concepts applied. Chapter 4 (Web Platform) describes the creation of the web platform for sustainable POIs collection. Chapter 5 (Walking Tour Recommendation) explains the tour generator algorithm implementation and presents the results of its validation. Finally, Chapter 6 enumerates the conclusions and Chapter 7 proposes future work.

[This page has been intentionally left blank]

Chapter 2

Literature Review

The proposed solution focuses on distributing tourists across POIs to decrease people congestion, increase the quality of experience and the location's sustainability. The related literature is organised according to the main themes involved in this dissertation development to achieve the thesis goals. As such, in Section 2.1, the protocol used to select the studies to be described in the remaining sections is explained. Section 2.2 consolidates the existing works about data collection for crowding detection with the aim of decreasing congestion. To promote the more sustainable POIs, we need to classify them accordingly. In Section 2.3 the sustainable classifications are explained. The need of distributing tourists has generated various strategies to control a place's capacity that as described in 2.4. Since the main strategy to be used in this dissertation is that of Tour Recommendations, Section 2.5 describes the most relevant applications of this strategy. Lastly, Section 2.6 presents a discussion of the state-of-art with focus on the tour recommendation systems, which is the main goal of the research presented in this dissertation.

2.1 Literature Review Protocol

This current protocol is divided in two Subsections: 2.1.1 Choice Criteria, where the criteria applied in the selection and inclusion of the studies for elaborating the literature review are described , and 2.1.2 Search Strategy, that presents the search methods and identifies the search strategies.

2.1.1 Choice Criteria

The works that have been selected are either empirical data experiments or theoretical studies, whose participants came from the following areas: Tourism Management, Mathematics, Computing or similar.

The inclusion of studies was not restricted to any specific type of intervention, although a preference for survey studies and real case scenarios has been used as a final filter. Nevertheless, regular papers, reports and books are the main types of publications that were accepted.

The kind of outcome of interest from the literature depends on the subject that is being searched but the main objectives to be fulfilled are: how to detect overcrowding, how to control location's capacity, how to measure tourism sustainability and, lastly, how are structured the touristic tour recommendations.

The publication date will be taken into consideration for the acceptance of any work. For real case scenarios, only publications from 2014 or above were included. However, an exception will be considered for algorithmic literature, which is not that much affected by ageing. Therefore, for graph-based algorithms literature we put no limits on the publication date.

The studies should be written preferably in English, but some Portuguese studies are used if the outcome is related with touristic influences in Portugal, since the real case scenario of this dissertation is Lisbon, which is the capital of Portugal, and so Portuguese scenarios are of paramount importance.

Any studies that does not matching the criteria explained above were excluded.

2.1.2 Search Strategy

The search strategies were applied for searching electronic databases and included information displayed in relevant journals in the search subject as a principal reference. However, electronic databases was not a precondition and scholar search engines were also used, namely *Google Schoolar*[1].

For the search, journals that publish papers in tourism innovation and computing were taken in consideration, as was also the respective impact factor, according to

[1]https://scholar.google.com/

the *Scimago Journal & Country Ranking* [2], together with the authors' H-index and the number of citations. Notwithstanding, the impact factor was not an exclusion criteria as long as the paper is scientifically sound.

The search keywords were the following:

1. (Crowding OR Overcrowding) AND (Detection OR Manage OR Tourism)
2. Sustainability AND (Classification OR Measure OR Tourism)
3. "Sustainable Development" AND (Tourism OR "Indicator System")
4. Tourism AND "Indicator System"
5. (Tourism OR Tour OR Travel) AND "Recommendation System"

2.2 Data for Crowding Detection

The overcrowding phenomenon happens when the number of persons at a given location increases to levels that cannot be physically supported or resources are not enough, [Dichter and Gloria2017]. If a local's capacity is for one hundred (1000) persons and there are nine hundred (900), the crowding value is of 90%. In the case of overcrowding, the value of crowding to be considered is higher than 100%. A question arises: How is it possible to monitor the number of individuals in a given area? How can we collect crowding data?

Every type of data has various sources and formats and can be acquired with structure, and thus, organised, or without it - unstructured. Various sources of crowding data have been studied, but the ones that are considered most important are: online social networks, mobile operators, image and sound capturing [Domínguez et al.2017], [Ranneries et al.2016], [Park et al.2016], [Stahlschmidt et al.2016], [Peng et al.2015], [Farrés2015] and [Andersson et al.2010]. It is noteworthy that most of the data sources for crowding are based on existing networks, like social networks, surveillance networks, etc. Nevertheless, the following studies aim to predict the crowding what can be deceiving, especially if there is an unexpected irregularity, such as a non-periodic event

[2] https://www.scimagojr.com/

Data from online social networks are used because of the ease of access and its large amount. The information is mainly posted through smart mobile devices applications, which provide GPS data linked with it. An analysis of geo-tagged photos of *Instagram* to find places that are under-visited by tourists but enjoyed by local residents and detect crowd anomalies in a city can be found in [Domínguez et al.2017]. The authors' main objective was to obtain a pattern for the behavior of the residents. Another study collected information from Twitter and Instagram to generate metrics for the distribution of people using GPS information and time [Ranneries et al.2016]. The biggest challenge consists in creating metrics for crowding using big data sets. Mobile operators are also a fruitful source of data due to the huge usage of their networks. The authors of [Park et al.2016] used historical data from the Spanish communications operator *Telefonica* to predict people's movement patterns.

Most of the big cities are somehow threatened by crowding impacts due to their dimensions. Since today most cities (municipalities) have a security camera's network assembled, the usage of that structure to collect information and analyze it to detect people's movement, is increasingly referred in the literature [Stahlschmidt et al.2016] and [Peng et al.2015]. In [Stahlschmidt et al.2016], the main objective is to avoid crowd "disasters", gaining knowledge of people's behaviour in a crowding situation. The authors attempt to find two types of patterns: for groups and for individuals. Comparing those, it is possible to find individuals that are acting differently (stand out) and, perhaps, in a conflicting way. Another example in the literature, that uses cameras from a city as a source of data for crowding, is [Peng et al.2015]. The author's goal is to track the pedestrians using multiple cameras that capture the same (crowded) local. The use of images involving citizens always raises privacy concerns, that have to be taken into consideration when performing the analysis. Furthermore, accurate image analysis requires a high level of computer processing.

Lastly, sound captures also enables the gathering of crowding data. Usually, this approach requires the creation of a new network of sound sensors distributed in the city. The technical report [Farrés2015] studies Barcelona's sound. An increase of the noise levels of the city lead to the assemblage of a distributed sound sensor network. The sounds thus captured can be directly correlated to crowding values of the city. Sounds emanating from individuals can be used to measure the people crowding, car sounds for traffic crowding, etc. In fact, after the information is gathered and processed, reports and heat-maps can be created to help with the city management.

Again, the capture of citizens sound and record of people's conversations raises privacy concerns. And likely to the image capture case, also demands high levels of computer processing. Nevertheless, some studies aim to take advantage of both sound and image, combining their data to obtain more precise results [Andersson et al.2010].

2.3 Sustainability Classification

Sustainability is a broad concept. Merriem-Webster[3] defines it as *of, relating to, or being a method of harvesting or using a resource so that the resource is not depleted or permanently damaged*. In other words, sustainability means that the existing resources must be used wisely, so no negative impact should occur. In a review of sustainability definitions [Glavič and Lukman2007], the author concludes that sustainability is a principle that lays in three dimensions: Environmental, Social and Economic. To guarantee the safe use of the available resources, the governance should develop sustainable policies, to mitigate environmental, social and economic problems. Ideally, a sustainable policy would center within the three dimensions (see figure 2.1), although in practice we find it more often between two dimensions. For example, a policy that sets between the social and environment dimension is recycling.

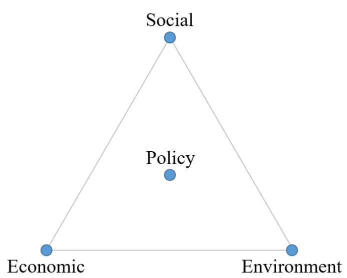


FIGURE 2.1: Sustainability Dimensions

The definition of sustainability policy is *...a plan of what to do in particular situations that has been agreed officially by a group of people, a business organization, a government or a political party, about environmental, economic and social issues,*

[3]https://www.merriam-webster.com/dictionary/sustainability

[Glavič and Lukman2007]. Planning policies for the sustainable use of resources leads to the definition of a new concept: sustainable development.

Sustainable development is a planned process, with a fixed timeline, that incorporates the vision of a sustainable future. In the literature, its definition is *the development that meets the needs of the present without compromising the ability of future generations to meet their own needs*, [Burton1985]. In view of the current environmental and humans concerns, the importance of a global sustainable development lead into the formalisation by the United Nations of the Sustainable Development Goals[4] *to promote prosperity while protecting the planet*, Figure 2.2 are all the goals. There are seventeen goals and the present work fits directly in two of them: Goal 8 - Decent work and economic growth, and Goal 11 - Sustainable cities and communities. Recommending sustainable POIs, that tourists are agnostic about, like local stores rather than franchisees, promotes local economic growth and local employment. To do so, while dispersing tourists, reduces the crowds and helps to increase both the quality of experience and the authenticity of the destination. Social issues, like the discomfort of residents regarding agglomerates of tourists can be solved.

FIGURE 2.2: Sustainable Development Goals Illustration

With this sustainable awareness, comes the need to be able to “measure” sustainability. This is a complex task because sustainability is organised in interrelationships between the three principals dimensions. In literature, the most used and reliable way of measuring sustainability is by using indicators [Hanai and Espíndola2012]. These can measure tourism sustainability and inherently tourism sustainable development. Indicators represent an attribute of a system, compiling information into something measurable and tangible, while helping in the management of the

[4] https://www.un.org/sustainabledevelopment/sustainable-development-goals/

system and serving as support for decision. In [Hanai and Espíndola2012], the authors list indicators and their functions. Based on that list, it is possible to summarize the indicators functionalities into three large categories: *Objective-oriented*, *alert information*, and *state monitoring*.

Various institutions have also formulated a list of indicators to be applied in several countries, to help measure and manage a sustainable development. Those lists are called *Indicator Systems*. The concept of sustainability is interpreted in various ways, thus generating different indicator systems. To determinate a set of indicators able of measuring sustainable development is, in fact, a hard task, as can be ascertained by reading the *International Institute for Sustainability Development* report [Bossel1999], where the process for building a Indicator System is described. Two examples of Indicator Systems that Portugal can apply for measuring sustainable development are:

- Indicator System for Sustainable Development (*Sistema de Indicadores para Desenvolvimento Sustentavel* - SIDS ([Gomes et al.2008])):
  This system has one hundred and thirty-two indicators designed for Portugal, of which: seventy-two are environmental, twenty-nine are economic, twenty-two are social and nine are institutional indicators. Every economic sector is considered, including tourism. An example of a tourism indicator is an economic one called "Accommodation Capacity", that measures the number of beds in Portugal.

- European Tourism Indicator System - ETIS ([European Commission2016]):
  This system focus only on tourism indicators that European countries can use, so those indicators are more segmented and explicit. It presents twenty-seven basic indicators and forty optional ones, grouped into four sections: (1) Destination Management, (2) Economic Value, (3) Social and Cultural Impact, and (4) Impact on the Environment. In [Tudorache et al.2017], the author explains the implementation of ETIS in *Brasov County*, Romania, and the difficulties that were felt.

In conclusion, while the indicators help monitoring and improve touristic destinations, some definitions are ambiguous and a complete and large amount of data is needed to feed the monitoring system.

Recently, some studies propose an evolution of tourism indicators. The authors [Blancas et al.2018] propose a vectorial composite indicator called the *Differential Dynamic Index* (DDI). It consists in two components: one static (as every indicator explained above) and one dynamic. The static serves to position the sustainability destination relatively to others. The dynamic indicator measures the sustainability evolution of the destination itself.

2.4 Capacity Control Strategies

Although there are many capacity control strategies, under the context of this dissertation, only the ones we consider more useful and relevant were surveyed and organised into three categories: 1) *Forbidden access*, 2) *Limit and schedule capacity*, and 3) *Tour recommendations.*

Forbidden access is a prohibition of access to a determinate location. A recent example for this category can be found in Sintra[5], Portugal, where the main street was transformed into a pedestrian street so that residents and tourists can walk widely and safe. However, approaches like this are very restrictive and tend to generate controversial opinions from residents. A less restrictive strategy is to limit and schedule the capacity, like what happens in the famous *Eiffel Tower*[6] in Paris, France. Because of its popularity, enormous wait lines were becoming a constant, resulting in a very congested area. In order to scatter tourists at that specific location, it is possible to schedule a visit online. In consequence, the ones that book will only come to the location around the scheduled hour, which tends to disperse the tourists over time. Unfortunately this only works for a specific POI. A more efficient strategy is to schedule tourists over space and time, can be performed with tour recommendations, that act on both dimensions.

To be effective, tour recommendations must follow tourist's preferences and, with that information, plan a visit to possibly multiple POIs over space and time. The problem lays in how to recommend which POIs and in which order they should be visited, to achive an interesting touristic tour. This is a challenging problem since it involves various domain constraints such as available time,

[5]https://www.publico.pt/2018/03/14/local/noticia/sintra-limita-circulacao-no-centro-historico-com-criacao-de-sentidos-unicos-1806664

[6]https://www.eiffeltickets.com/

budget, POI's visiting time, opening and closing hours, etc. In the extant literature, this problem is formulated as the *Tourist Trip Design Problem* (TTDP) [Vansteenwegen and Van Oudheusden2007], [Gavalas et al.2014], [Gunawan et al.2016]. Mainly, it consists in identifying POIs and, with them, plan a tour obeying fixed trip constraints.

2.5 Touristic Tour Recommendations

With tourism and technology evolution, and tourist's need to schedule their trips to unfamiliar places, smart tour recommendations for tourists naturally arise because, while planning a tour, the tourist faces difficulties in organising an ordered set of points of interest. As previously referred, this problem is formulated as the Tourist Trip Design Problem (TTDP).

Nowadays, tourists use smart technologies as tools and applications to help and provide personalised tours, called *Tour (or Travel) Recommender Systems* (TRS). A Recommender System (RS) is a software tool that provides suggestions to a specific user [Ricci et al.2015]. The origin of RS lays on the fact that users usually rely on recommendations provided by others. For example, recommendation letters in recruiting selection, or a simple recommendation of a book by a critic. In the 1990's, with the Internet, RS emerged based on collaborative filtering, to address more targeted information to each user. Nowadays, RS are largely used in e-commercial sites, with machine learning algorithms suggesting items that the user may buy. In platforms of entertainment, like *Netflix*, suggestion of movies or series that meet the user's preferences are common place [Koren et al.2009]. In the field of tourism, RS are used to recommend a set of unordered POIs to a Tourist. In [Logesh et al.2018], [Zhou et al.2015] and [Domínguez et al.2017] were implemented POIs recommender systems were implemented using collaborative clustering. But it is also used in recommendation systems to solve the TTDP and provide tours of the user's interest. This approach is called a Tour Recommendation System (TRS), where user preferences are gathered to combine them with tour constraints and POIs using a tour generating algorithm to provide a customized tour, [Ricci2002]. This global framework of a TRS can be visualized in Figure 2.3.

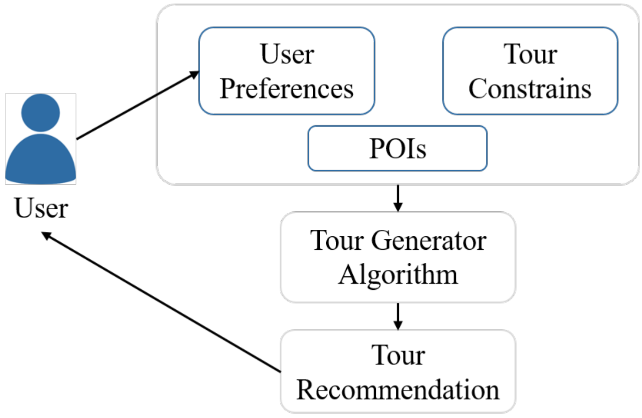


FIGURE 2.3: Tour Recommender System Global Framework

The survey [Borràs et al.2014] is a compilation of various systems and provides a guideline for the implementation of smart tour recommendations. The authors conclude that multi-channel applications, that is, that can be accessed using the web and in a mobile device, are useful for the tourist. The rationale behind, is that it allows to plan the trip still at the tourist's home (or hotel) and use the mobile application as a guide in the destination and during the walk. As another advantage, the authors conclude that tourists find reviews on POIs from other users useful and reliable. Furthermore, they also concluded that images (photos) play an important role and, are appealing for the tourist. The creation of a profile made by the tourist to suggest places of his/her interest, in a first approach, is highly advised by the authors, since it enables to learn from the tourist's actions and adapt the profile preferences. For example, using the time spent at each POI or if POIs different from the ones suggested were selected. This technique has been key to the success of many tour recommenders. However, to fully profit from this approach, it is crucial to categorize the POIs.

Recently, several TRSs have been developed, both for the web and for mobile applications. The difference between them, besides the algorithmic approach, is how they build the set of POIs and which tour constraints are included. For the selection of the set of POIs, an elementary distinction must be made: POIs are based on a) an initial subset or b) opinion mining. In (a), it is required for the user to provide his/her preferences, by indicating a subset of POIs of interest [Kotiloglu et al.2017]. From there, the categories of POIs that the tourist may enjoy can be inferred. This strategy does not provide new experiences to the tourist neither does it take in account the popularity of the POIs. The second one, (b) opinion mining, requires background work on the POIs at the destination, involving a high level of computer processing and text mining approaches, and finaly, the tour construction from the extracted knowledge. In this approach, the main data sources are social media or trip reviews platforms and crucial information to classify the POIs based on their features and opinions must be extracted [Logesh et al.2018]. In [Zhou et al.2015], the author found and clustered tourist destinations using *Flickr*, due to the popularity of the geo-tagged photos. Another study used a tour planning platform, *Minube*, where user opinions could be found and only suggested the POIs that were highly valued, [Cenamor et al.2017]. In [Domínguez et al.2017], the author used social media, namely *Instagram*, to analyze the geo-tagged photos with the aim of find under-visited places by the tourists, but enjoyed by the residents. Some studies, like [Lim et al.2015], combined both ways to get a customised set of POIs that included the most popular ones. In the latter case, the authors chose geo-tagged photos from *Flickr* to suggest popular POIs, and then asked the tourist for to select a subset of those POIs, that he/she considers visiting. This system also learns from the POIs that the user did visit and from the ones that he/she has discarded.

Tour constraints aim to solve destination domain problems that affect the tourist and the tour. In the extant literature, according to each study's objectives, constraints chosen between what the authors deem as relevant. We have divided the types of constraints into three categories: i) *budget*, ii) physical effort and iii) *time*.

It is important to understand that there are other specific constraints that do not fit in any of these categories like nourishment, in [Cenamor et al.2017]. The authors incorporated in the tour a “feed plan” the user was asked to fill in the number of times expected to have a meal and the tour suggests places to go when the user is more likely to be hungry.

Budget is a simple constraint but is well valued by the tourist. The user should state the available budget and the tour suggests POIs where the sum of prices does not overcome that amount, [Lim et al.2015],[Kotiloglu et al.2017]. The *physical effort* tries to customise the tour in terms of the adequate physical effort for each user. It is based on the length of the tour and in a preference provided by the users, [Gavalas et al.2017]. *Time* is a feature that has major importance from the users view of quality of experience and is stated in the literature as being a particularisation of the TTDP, [Vansteenwegen and Van Oudheusden2007], [Gunawan et al.2016].

Different approaches to incorporate time into the tour have been suggested, namely *Time Window* (TW), *Time-dependent* (TD) and *Multi-period* (MP). The first one, TW, aims to suggest only POIs that are open at the time when the user arrives there. The TD focus in providing tours that do not surpass the available time of the user for the tour (similarly to the budget constraint), and MP recommends a tour that can be carried on for more than one day.

The tour generator algorithms used in the TRSs are mainly search algorithms or derivations of it. For example in [Kotiloglu et al.2017], the author used *Iterated Tabu Search*. In [Cenamor et al.2017], *Enforced Hill-Climbing* and the *Weight Best-First Search* were combined. And in [Mrazovic et al.2017], *Variable Neighbourhood Search* was used.

2.6 TRS State-of-art Discussion

Based on the literature review, it is possible to conclude that to forbid or limit the access is a highly restrictive technique, leaving the tourist with virtually no choice. Furthermore, it may cause the opposite effect, as forbidding or limiting access to a given area will increase the pressure felt by tourism crowding in the surrounding areas. Hence, tour recommendations is the best strategy for capacity control that uses smart technology for providing a larger choice of action to the tourists.

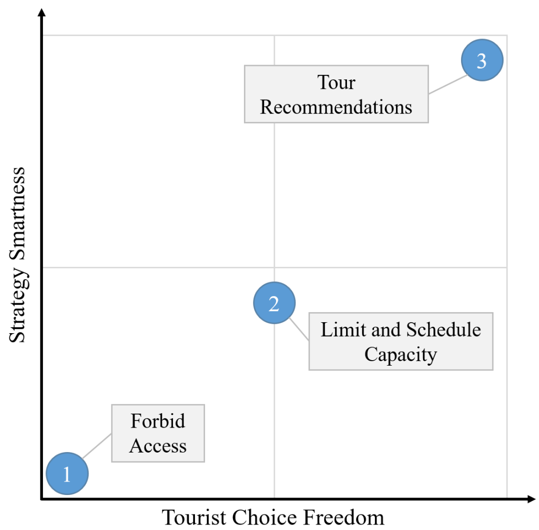


FIGURE 2.4: Analysis of Capacity Control Strategies

With the reservations' information, it is possible to manage the tourist in time and space and, as the tourist can book it in advance although the reserves are limited, it gives more freedom of choice to the tourist. Thus, as illustrated by Figure 2.4, the specific kind of strategy controlling capacity by limiting and scheduling visits, while being a smart use of technology by adding some flexibility and load management to the process, is still rather restrictive. Moreover, it helps solving the problem at a specific point of interest but does not solve the problem in other public spaces like streets or plazas. In a nutshell, it decreases the pressure felt at one POI but does not help managing the pressure of overall touristic activities in a city.

Tour recommendations aim to solve the latter problem by dispersing the pressure of touristic activities throughout the city or location. It also aims to adjust and fulfil the preferences of the tourist, giving him/her more freedom of choice, while allowing for a more holistic management of tourism for the managers of that location.

Based on the previous Literature Review, it is possible to conclude that there is space available for research within the area of Tour Recommendation Systems adding new domain constraints. Especially constraints that belong to the sustainable area of tourism, those are less searched and used, but nowadays with tourism's evolution and sustainability awareness is crucial to incorporate in tour recommendations. This awareness also appears with the need to support the Sustainable Development Goals, proposed by the United Nations. The present dissertation aims to achieve a Tour Recommendation System that combines sustainable POIs and avoid crowded areas, to mitigate the overcrowding phenomena and increase the local economy, providing smart sustainable tourist tours.

[This page has been intentionally left blank]

Chapter 3

Framework, Decisions and Theory

The present chapter describes where the main work within this dissertation fits into the overall project, where it is included and how the work done was structured. Section 3.1 describes the global project, *Sustainable Tourism Crowding Project*. Section 3.2 clarifies and illustrates the framework of the work described by the dissertation. And, in Section 3.3, the main theoretical concepts applied in the development of the thesis work are exposed.

3.1 Sustainable Tourism Crowding Project

The *Sustainable Tourism Crowding* (STC) research project aims to locally mitigate tourism overcrowding and promote tourism sustainability, using a smart platform. The latter intends to distribute tourists over less crowded POIs and promote the visitation of more sustainable ones.

STC's smart tourism platform is based on a distributed architecture, including the following components, as represented in Figure 3.1:

- **APP** – mobile app that records user's preferences in terms of types of POIs, the current start location (obtained through GPS) and the desired arrival hour and location; passes on that information to the ROUTE microservice that replies providing a recommended walking tour that is shown to the app user; it sends location permanence pings to the HEAT microservice;

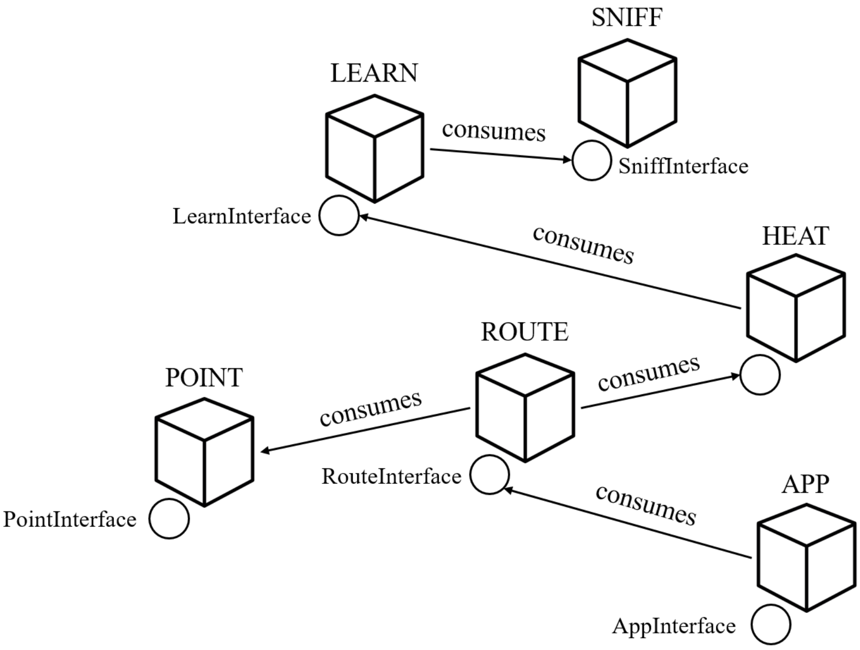


FIGURE 3.1: STC project microservices architecture

it records the walking tours effectively traveled by tourists (including timestamps) and sends both instances (recommended and traveled) to the LEARN microservice; allows publishing comments (also sent to the LEARN microservice) associated to visited POIs on effectively traveled tours; for enforcing tourist's engagement, this app uses a gamified approach upon the fruition of georeferenced historical contents, inspired in the geo-caching game and that is the reason why it is called *HAGO*[1]; its logo, with a traditional Lisbon's crow[2] is shown by Figure 3.2;

- **SNIFF** – edge computing device that detects the number of mobile devices active in its vicinity, considered to be a good surrogate for the number of people there; it sends that crowding data to the HEAT microservice;

- **<u>ROUTE</u>** – microservice that concurrently serves requests for tour recommendations from multiple APP instances; ROUTE has both a slave **POIs database** and a slave **current population cartogram database**, both in read-only mode, that are a replication of the master ones in the POINT and HEAT microservices, respectively; the recommendation algorithm operates upon a graph representing the target neighborhood, obtained from its map; besides being based on real-time data concerning the actual crowding state

[1]Historic narrative caching App for Going Out
[2]related to Lisbon's legend of S. Vincent's crows

detected in POIs' vicinity, this graph-based multicriteria algorithm takes into account the available time for the tour, visiting preferences expressed by the user and POIs' opening hours, promotes the visitation of more sustainable POIs and demotes the visitation of POIs whose vicinity is more crowded;

- **POINT** – microservice providing a web interface, that allows scrolling and zooming in a *OpenStreetMap*[3] and to edit POIs' data (e.g. description, opening hours, expected visit duration, photos and links to videos) upon it; its local **POIs database** is replicated by the ROUTE microservice;

- **HEAT** – microservice responsible for the population cartogram (heat-map) of the target neighborhood; it receives crowding data from multiple SNIFF devices, and location permanence pings from the APP instances and updates its local **current population cartogram database** accordingly; at regular points in time, it sends the current population cartogram snapshot to the LEARN microservice, requesting a forecast to reset the current state;

- **LEARN** – microservice that stores the population cartogram snapshots, received from the HEAT microservice, in the **historical population cartogram database** and uses that information to produce forecasts for a future period of time, to be passed on to the HEAT microservice; it provides a web interface that displays:

  1. a longitudinal visualization of the population cartogram, using a slider, upon a selected time period;
  2. a sentiment analysis on comments about visited POIs, to infer their attractiveness and assess how delays affect those comments;
  3. an analysis of the differences between recommended and traveled walking tours, to assess which factors affect the adherence to recommended ones.

Microservice HEAT is the provider of these information and its visualisation by using a heat-map format to quickly ascertain if there are and which are the current crowding areas. The data combines real-time sensing, coming from the SNIFF microservice, with predictions, coming from the microservice LEARN.

[3] https://www.openstreetmap.org

Figure 3.2: *HAGO mobile app logo*

The STC - LEARN goal is to study the historical data, also provided by SNIFF, and extract patterns to predict future crowding points. People normally go around carrying mobile devices (cellphones, smartwatches). Hence, real crowding data can be acquired by tracing mobile devices (via Wi-fi, Bluetooth, GMS, 3G and 4G connections), and estimating the quantity of devices present at a specific location, making it possible to know in real-time how densely populated that local is. This data will be collected once the necessary sensing network is deployed.

This dissertation's main work concerns the development of the POINT and ROUTE microservices. The tour recommendation structure, that is at the core of ROUTE, and its requirements are described in Section 3.2.

The main strategy for the optimisation problem that consists in deciding the best routing tour for the user is that of solving a multicriteria shortest path with additional constraints and is addressed in Chapter 5.

The STC - POINT (whose implementation is described in Chapter 4) consists in a web platform that allows the local government body (in this case, a Lisbon parish council workers) to define sustainability values for their POIs (more details can be found in Section 4.1).

3.2 Tour Recommendation System Framework

The main goal is the implementation of a TRS, able to suggest tours that try to distribute tourists in a uniform manner using the crowding data. As already explained, the TRS needs as inputs the preferences and visiting overall duration given by the tourist to the mobile app and should recommend tours that meet those requirements and, at the same time, prevent crowding and suggest sustainable POIs for the visit. To accomplish this task, we will need to use graph optimisation algorithms and, thus, model the network of the city, which in the present case study is Lisbon downtown, using a graph that will have to incorporate data on sustainable Lisbon POIs and crowding data. This requirements and framework are illustrated by the figure 3.3.

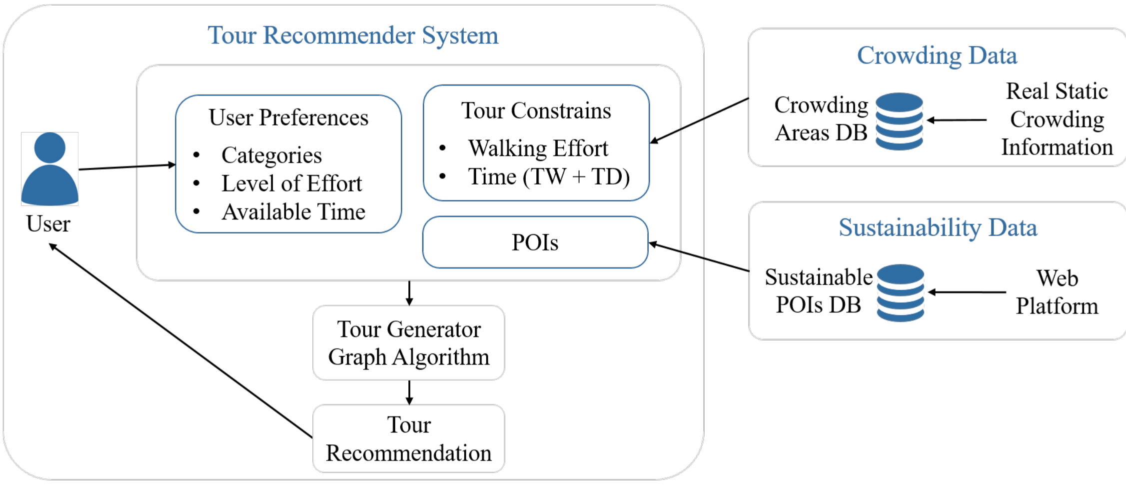


FIGURE 3.3: Proposed Solution Framework

User preferences

For the user to be able to specify his/her interests, the extant literature shows the importance of using categories to enable good suggestions of POIs that meet those interests, see Section 2.5. In this aspect, and besides having to define which are the categories that will be used for the tourist to choose from (more details at Section 4.2, Table 4.1), to provide walking tours to a user, details as age and physical conditions are needed. In order to maintain the tour comfort and increase the quality of experience (QoE) and avoid being intrusive, levels of effort are used to accommodate the user preferences, (more details Section 5.1.2, Table 5.1).

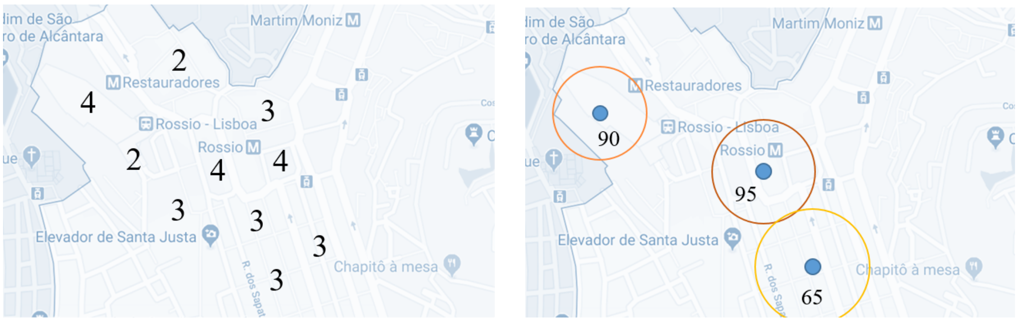
Of course that one of the most important points here is the available time of the user for the visit so that only tours that do not surpass that limited time are recommended.

Tour constraints

As already stated, to improve the QoE, walking (or physical) effort and the time satisfiability upper bound (TD) are defined by the user choices. Time windows (TW) are used so that only tours presenting POIs open for visit shall be suggested to the tourist.

Another tour constraint specific of this project domain is the crowding data, so that the tour generation algorithms avoid the more congested areas. In particular, data on crowding points at the location must be provided from the service HEAT. Unfortunately, it was not completed on useful time to provide the needed crowding data. As such, we consulted the relevant persons from the Lisbon parish council where our case study is being deployed and used the static data that was indicated by the professionals. The data is a human estimation of what they believe it is the real crowding on average.

With a range between 1 and 4, from less to more congested, they evaluated every area of the map. That evaluation is in the map on the appendices, figure 1. Using that information it was created epicenters with a radius of incidence, simulating a crowd percentage. This transformation is exemplified in figure 3.4. In extremes scenarios, the crowd value can exceed the 100 percentage, simulating a situation of local capacity exceed.



(a) Parish Council Data Excerpt (b) Epicenters Transformation

FIGURE 3.4: Crowding Data Transformation

TABLE 3.1: Crowding Data Transformation in Database

Id	Latitude	Longitude	Radius (meters)	Crowding
1	38,714301	-9,142206	125	90
2	38,713587	-9,138463	150	95
3	38,711546	-9,137297	200	65

In figure 3.4 (a) is represented an excerpt of the total map and in 3.4 (b) a possible crowding scenario using epicenters and estimation percentages. That data was inserted in a local database, and for the tour recommender tests was simulated various possible scenarios with less and more crowding always based on the data provided. In the table 3.1 is exemplified the type of data inserted into the crowding database relative to the illustrated transformation. Are stored the coordinates of the epicenters, latitude and longitude, the radius of incidence (in meters) and the percentage of crowding in that circular area. Third the data set of POIs, due to the nonexistence of a database with Lisbon POIs and its sustainable value was implemented a web platform to populate a new database. The web platform was created to provide an easy way for experts introduce sustainable information. The creation of that platform and its database is described in the Chapter 4. The user preferences, the tour constraints and the POIs were then used to feed the algorithm that provide tour recommendation. In this project, graph algorithms were used, due to the map be a road network easily structured as a graph. The implementation and all the information relative to the graph algorithm is stated in the Chapter 5.

3.3 Incorporated Theory

This section propose is to explain the theory incorporated in the formulation of the Tour Recommender System explained above. Since it intends to solve the tourist trip design problem, subsection 3.3.1 describes the problem and its requirements. As the tour generator algorithm of the TRS is a graph algorithm, subsection 3.3.2 describes the inherent theory. Both subsections apply its concepts to the proposed solution of this dissertation.

3.3.1 Tourist Trip Design Problem

The tourist trip design problem (TTDP) aims to plan a tour with multiple points of interest that match user preferences. The problem is to select the POIs and take into account the various domain constraints. Such as respecting the time available, the distance between the POIs, the visiting time of each POI and the opening and closing hours. Also adding constraints that suit the problems for a specific domain or project that the TTDP may be inserted.

Some extensions of the TTDP referred to the time problems are studied and formulated in the literature, [Gavalas et al.2014]:

- Time-Windows, consider for each POI an opening and closing hour. The selected tour must only suggest POIs that are open when the tourist arrives there. So it also may consider the initial time of the tour.

- Time-Dependent, consider the estimation of the time required to move between POIs and the expected time that the tourist will spend on each POI.

- Multi-Period, consider tours for more than one period, for example a tour for a weekend vacation that covers both Saturday and Sunday.

There are two classic variants of the TTDP with a single tour: Travel Salesman Problem (TSP) and Orienteering Problem (OP). In the TSP, [Ilavarasi and Joseph2014], is intended to minimize the distance between a set of POIs. The Orienteering Problem (OP) was first introduced by Tsiligirides [Tsiligirides1984]. The objective of OP is to determinate a set of POIs to visit in a specific order that maximizes a score, [Gunawan et al.2016]. This score is related to the problem domain and its constraints. Usually, the score comes from a value associated with the edges that join the POIs. But some domains required to consider values associated with the POIs. When that situation happen, it is called a Mixed Orienteering Problem (MOP) [Vansteenwegen et al.2011].

The TTDP approach chosen should be derived from the domain of the real problem, where the set of selected POIs must consider the tourist's preferences. In this dissertation are used the Time-Dependent, Time-Windows and Mixed concepts. And a new approach of the TTDP was created, that we called Mixed Time-Dependent Orienteering Problem with Time-Windows (MTDOPTW).

3.3.2 Graphs and Algorithms

In 1741, Leonhard Euler solved the *Konigsberg* bridge problem [Euler1736], it gave birth to the graph study. In *Konigsberg*, capital of Eastern Prussia, a merchant city crossed by the river Pregel. It owned seven bridges connecting all the city, what gave origination to the problem: Can one walk across all seven bridges and never cross the same one twice? The problem appeared to not have solution, but there was no proof of that conclusion. So Euler represented the city through a graph with four nodes, each corresponding to a portion of land, and seven edges, each corresponding to a bridge (figure 3.5).

With that representation, it was possible to show that there was no path connecting all areas walking through each bridge only once. Euler's proof was the first

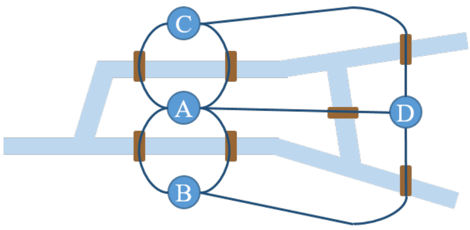
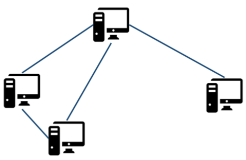


FIGURE 3.5: The *Konigsberg* Bridge Problem

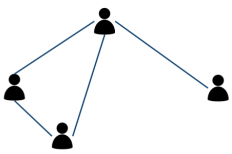
mathematical problem solved using a graph, showing that with a graph representation it can be easier to solve some issues.

Graphs represent a modular structure or a network, and the following explanations are based on Albert-László Barabási online book, *Network Science*[4], and Dieter Jungnickel book [Smith and Jungnickel1999].

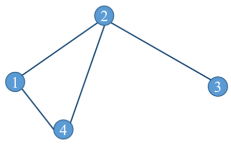
A network is a complex system that connects various components. An example of a network is the Internet (Figure 3.6 (a)) that connects routers (computers).An example for a social network is illustrated in figure 3.6 (b), where each component is a persona and two personas are connected if they are "friends".



(a) Internet Network



(b) Social Network



(c) Graph

FIGURE 3.6: Network *vs* Graph

Although the two networks examples seem different, with distinct concepts and domains, both can be represented by a simple structural model, that is, the same graph, as illustrated by Figure 3.6 (c). Each component in the network corresponds to a node in the graph structure. The links or connections between components (personas/routers) are edges in the graph. A network is directed if it has links that have traversal directions, and it is undirected if all of its links can be traversed in both directions. Some networks can have simultaneously present directed and undirected links. An example of a directed network is the mapping of a location roads. In contrast, for a social network that represents friendships each link is undirected.

[4]http://networksciencebook.com/

Most of the interest networks are weighted, that is, values are associated either with the links, or with the edges, or with both. For example, for a graph model of car roads, each link has an associated travelling distance. The physical distance is important between components of physical systems, not only to the car road example. But, in networks, distance is a more complex notion. Take the example of Figure 3.6(b): what is the distance between two persons who do not know each other? The distance here is not relevant, the important information is that they have friends in common, and that it is possible to trace a path of friendships to get to each other node through the node two.

In the aim of this dissertation, a graph is denoted by $G = (N, E)$, where $N = \{1, \ldots, n\}$ is the set of nodes and $E = \{1, \ldots, m\}$ the set of edges. If an edge has a specified direction it is represented as $e = (n_i, n_{i+1})$ with $n_i, n_{i+1} \in N$. If it is bi-directional, it is represented as $e = \{n_i, n_{i+1}\}$.

A path in a graph is an ordered list of edges between two given nodes. The length of a path corresponds to the number of connections. Let p be the path between the nodes n_0 and n_k, $p = \{(n_0, n_1), (n_1, n_2), \ldots, (n_{k-1}, n_k)\}$. The length of the path p is k, since this is the number of edges in path p.

The more important definitions, in terms of paths, for this dissertation are:

- Shortest Path: the path with the shortest "distance" between two nodes (does not need to be unique);
- Average Path Length: the average of the shortest paths between all pairs of nodes;
- Cycle: a path with the same start and end node;
- *Eulerian* Path, a path that traverses each edge exactly once;
- *Hamiltonian* Path, a path that visits each node exactly once.

When the problem at hand aims to look for the "best" paths (best in some relevant and mensurable aspect), an algorithm of the, so-called, Shortest Path algorithms class (SP) should be used, [Sedgewick and Wayne2011]. The typical use is to search the shortest route or the fastest, sometimes the cheapest or even the safest. The SP algorithms have been extensively used and studied. For example, when the goal is to be the best in one singular aspect, for example the fastest, it is called the *single-objective SP*. Some examples of single-objective SP well known algorithms are: Dijsktra's algorithm [Sniedovich2006] and the Astar (A*) heuristic [Hart et al.1968].

Some problems require that more than one best path is found, the k best paths in that aspect, which is known as the *K Shortest Paths* (KSP) [Hoffman and Pavley1959]. Some more complex problems require finding the best path in more than one aspect, for example the fastest and cheapest path. In this case, since it intends to optimise two different objectives, it is denominated *biobjective* (or bicriteria) *shortest path* (BSP), [Henig1986]. If it extends to more than two objectives, is called *multiobjective* (or multicriteria) *shortest path* (MSP), given birth to the *multiobjective path problem* (MPP), [Clímaco and Pascoal2012].

Let f^k be the function that assigns a real value for an objective k, P is the set of all feasible solutions, $x \in P$, with $k = 1, 2, ..r$, and $r \geq 1$ the total number objectives. The MPP can be formulated as:

$$min f(x) = (f^1(x), f^2(x), \ldots, f^r(x)) \tag{3.1}$$

Note that the BSP is a particular case of the MSP with two objectives and can be solved through enumerative approaches the labelling methods or ranking methods, [Raith and Ehrgott2009]. The multiobjective path problem aims to find the solution that is optimal in all the objective functions $f^1, \cdots, f^r$. However, in general, there is not one and only solution optimal for all the objective functions, and the concept of optimality is replaced by the concept of non-dominance (efficiency), [Clímaco and Pascoal2012]. Let P denote for the set of all solutions. A feasible solution $x \in P$ is efficient if does not exist any $y \in P$ with $f(y) \leqslant f(x)$ and $f(x) \neq f(y)$.

For the tour generator algorithm in TRS (Section 2.5), we will be using SP algorithms for the search of efficient solutions. In fact, this dissertation's main problem can be defined as: find the path that minimizes the crowding values and maximizes the total sustainable values associated with the POIs (which will be nodes in our graph model). Is obvious that there are two different objectives to be optimized: crowding and sustainability of POIs, so it can be stated as a biobjective shortest path problem. The problem formulation and its implementation are described in Chapter 5.

[This page has been intentionally left blank]

Chapter 4

Web Platform

To disperse tourists through less congested areas by suggesting routes going through more sustainable POIs, we need to have specific data. As previously explained, a data set of POIs, crowding points and sustainability values for a given location is a crucial part of the TRS. Due to the nonexistence of such a data set for Lisbon POIs, there is the need to build one. To fulfil that need a web platform was implemented and a data provided by the *Santa Maria Maior* parish council workers was saved into a new database . The construction of the web platform is explained in Section 4.1. The database linked to it is described and documented in Section 4.2. Lastly, in Section 4.3, the REST API designed to achieve the connection between the web platform and the database is described.

4.1 Requirements, Structure and Functionalities

Parish workers needed easy access to the database so they can add the POIs of interest together with the respective sustainability classification. This information is crucial in this project, because one of its goals is to maximize local sustainability. Moreover, the mobile application needs historical information associated with the POIs. That information is the central point of the app's gamification, so the web platform was extended to fulfil that requirement.

A web platform was implemented with the actors and functionalities presented in the use case diagram at Figure 4.1. Note that the web platform was implemented in HTML and Javascript.

The actors presented in the diagram are: User, Parish Council Worker, Sustainability Manager and Historian. The last two are Parish Council Workers and inherit the respective permissions. The actor User represents any person that has access to the web platform interface and has permission to try the use case authentication. An authenticated user is a Parish Council Worker and it can have especial permissions if is a Sustainability Manager or an Historian.

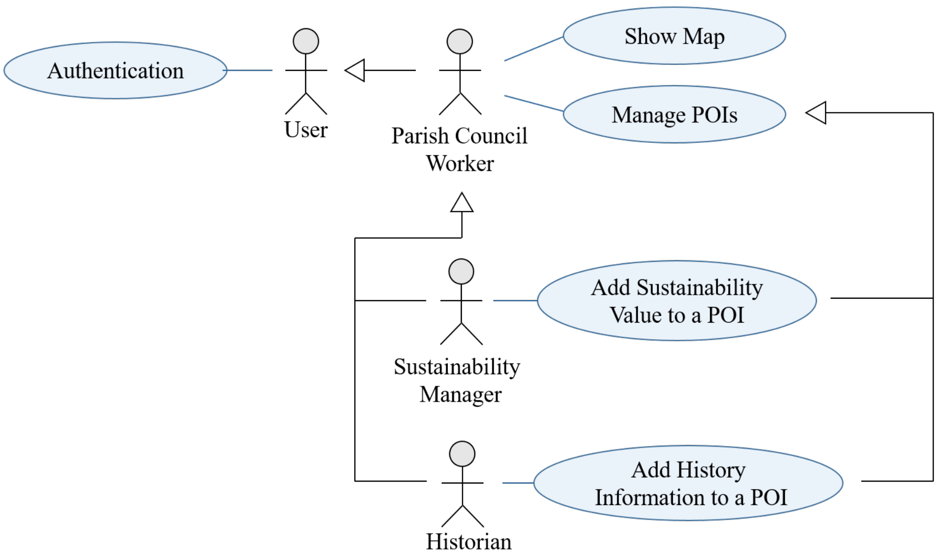


FIGURE 4.1: Use Case Diagram of Web Platform

The use cases are: Authentication, Show Map, Manage POIs, Add Sustainability Value to a POI and Add History Information to a POI. The process of authentication is a basic login, where the user provide an username and password. In the sequence diagram, Figure 4.2 illustrated the interaction that occurs in that process.

The message *login(username, password)* between the User and the Interface illustrates that action. Then, it is validated in the database of users, where it is archived the username, password and type of user. In the database a select to search the user with that credentials is triggered, message *getUser(username, password)*. If the user is registered the database provide the User and the respective Type of User. The return of the Type of User is important to show in the interface the functions permitted. If returns null, it means that the user does not exist or does not have allowance to enter in the web platform. When the user is authenticated, it means that is a Parish Council Worker and is shown the map.

The map view is implemented using a open-source javascript library called *leaflet*[1],

[1] https://leafletjs.com/

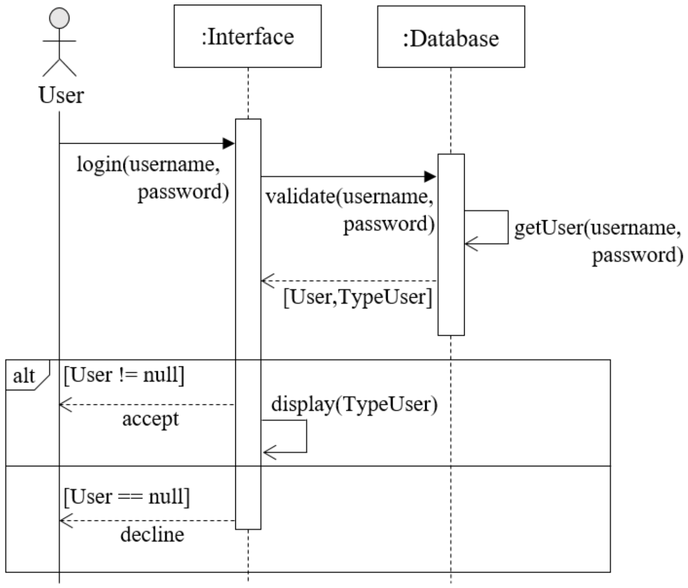


FIGURE 4.2: Authentication Sequence Diagram

for interactive maps that uses the *OpenStreetMaps*[2] as map source. In Figure 4.3 is a screen-shoot of the web platform in map view mode, what appears after the authentication.

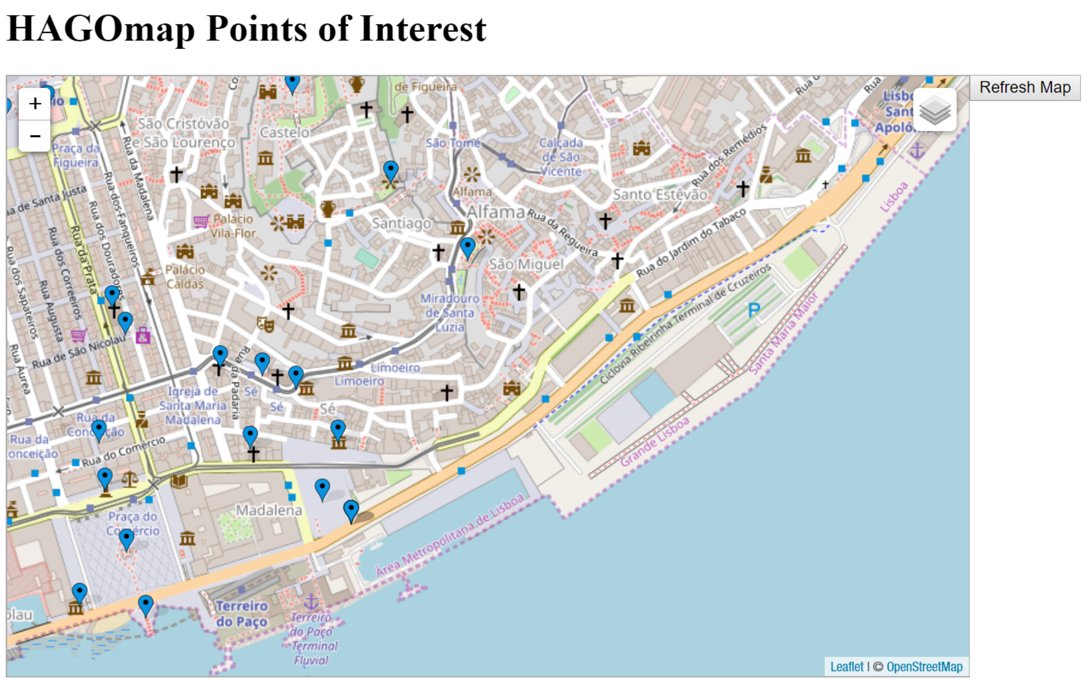


FIGURE 4.3: Map View of Web Platform

That library allows to add markers on the map. In this case, each blue marker represents a POI stored in the database. The interface sends a message of get to

[2]https://www.openstreetmap.org/

the database, which selects all the POIs stored and gives a list with them. That interaction is illustrated in the sequence diagram Refresh Map, Figure 4.5(b). That library also permits to listen to the HTML clicks on the map, and know if it is on the empty map or in a marker that represents a POI. With that listener, it is possible to add new POIs, update or delete the existing ones. That leads to the Manage POIs use case, the principle function of this platform, that agglomerates those three previous concepts. A possible user behavior for that use case is illustrated in the sequence diagram of Figure 4.4.

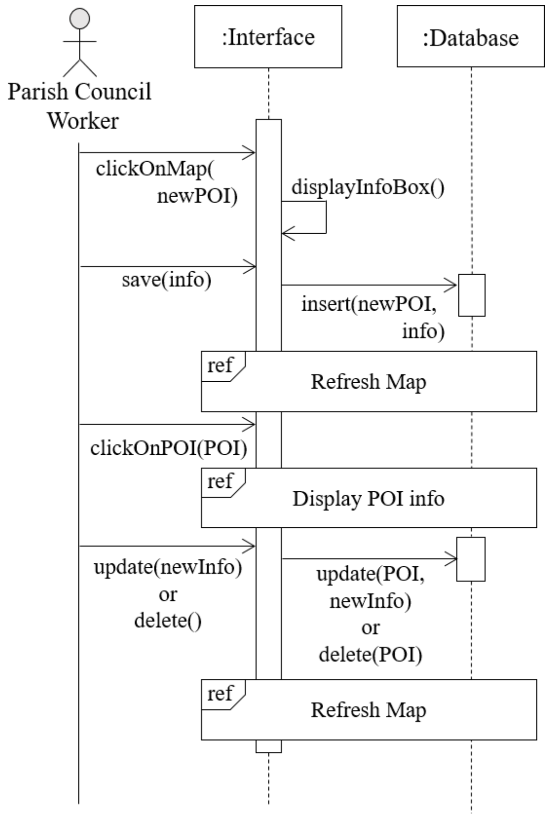
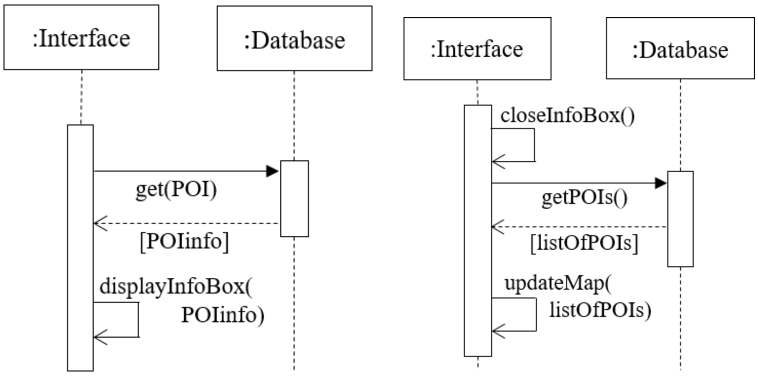


FIGURE 4.4: Manage POIs Sequence Diagram

When the user wants to add a new POI to the database it only needs to click on the map, message *clickOnMap(newPOI)*, then it appears a red marker and a box to fill with the information of that new POI, message *displayInfoBox()*. The color red for the new marker was chosen to distinguish from the blue ones already presented on the map. The box fields are different according to the user. If it is a Sustainability Manager it appears a field to add the value of sustainability. Otherwise, if it is a Historian appears a field to add the history. But every parish council worker can fill the same standard POI information, only shows more fields if it has that permission. That standard fields are shown in the screen-shoot in Figure 4.6. Where it is also possible to visualize the red marker of the new POI



(a) Display POI Information (b) Refresh Map

FIGURE 4.5: Sequence Diagram of ref fragments

on the map. Once the fields are filled, the user can click on the Save button, then the box of information is closed. In the map is automatically added the new POI, now in a blue marker. That sequence of messages is represented in Figure 4.5(b).

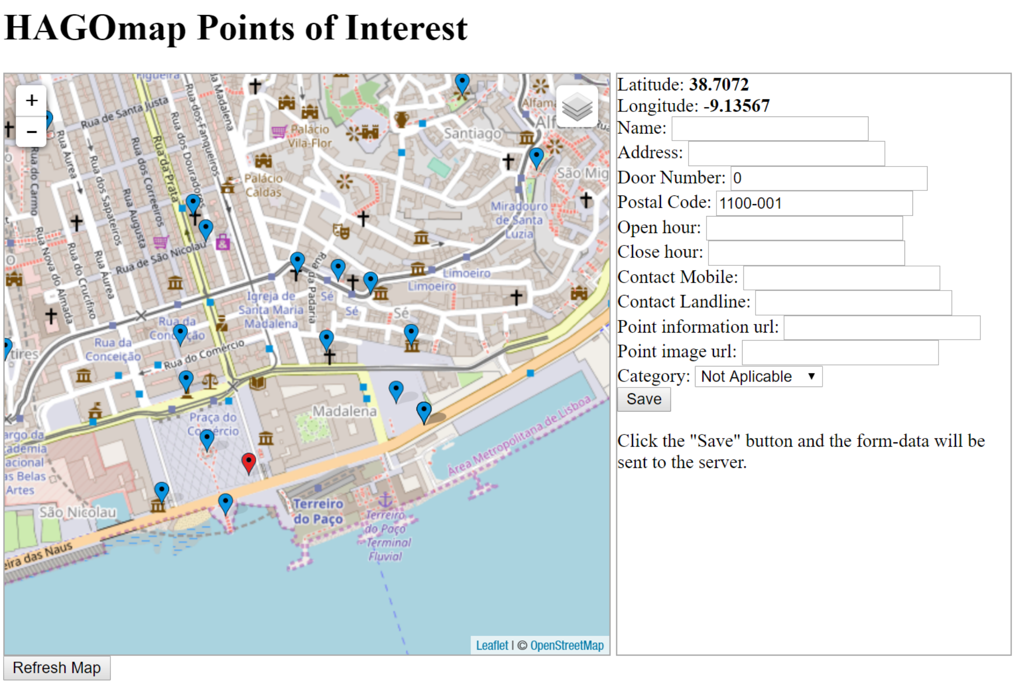


FIGURE 4.6: Insert POI View of Web Platform

When the user wants to update or delete a POI it only needs to click on a blue marker, message *clickOnPOI(POI)*. The corresponding box with the information form, but already filled with the data stored of that POI, sequence diagram 4.5 (a). The interface gets the POI data from the database and fill the form with it. After that the user can edit the present information, this flexibility is important because a Parish Council Worker can add the POIs with its standard information and then an expert can add the sustainability or the history afterwards. When the user is satisfied with the information, it is possible to click on the update button, and the modifications are saved in the database. If the intention is to delete the POI the process is identical: the POI information shows up and bellow it two buttons, the delete and the update one. The map is then refreshed with the new changes made by the user.

4.2 Points of Interest Database

The database that stored the POIs referred to in the previous section is structured following the class diagram illustrated in Figure 4.7. It was implemented in *MySQL*[3] on a *phpmyadmin*[4] server, a free software tool to handle the administration of *MySQL* databases. The POIs database is a vital part of the microservice POINT, that the ROUTE one consumes, section 3.2, which proposes to recommend tours that suggest sustainable POIs. The fields presented in the web platform match with the attributes of the class diagram. The attributes necessary to suggest a POI in a touristic tour are: the coordinates to position the POI: *longitude*, *latitude* and *altitude*; the *point_name* to identify the POI to the tourist; the *category* to align the tourist preferences; and the *sustainability* to guarantee a sustainable tour.

The category is a special attribute, the possible categories are already predefined in the table Category, also listed in Table 4.1. Maintaining the categories fixed makes possible to filter the POIs suggested by category. A POI can only have one category associated, that association is mandatory, and a category can be associated to many POIs. To compute the total time of a tour there is the need to know how long a tourist stay in each POI. A generalisation for the time spent in each type of POI was made and stored in the database. That time aims to be an approximation of the real time that the tourist stays at the POI (in minutes).

As explained in the Chapter 3, the tour must only suggest open POIs. That information is stored in *open_hours* and *close_hours*, it can have values between 0 and 23 simulating the hours. If a POI is a place that is always open, for example a square or a viewpoint, those variables hold the value -1, so it can always

[3]https://dev.mysql.com/

[4]https://www.phpmyadmin.net/

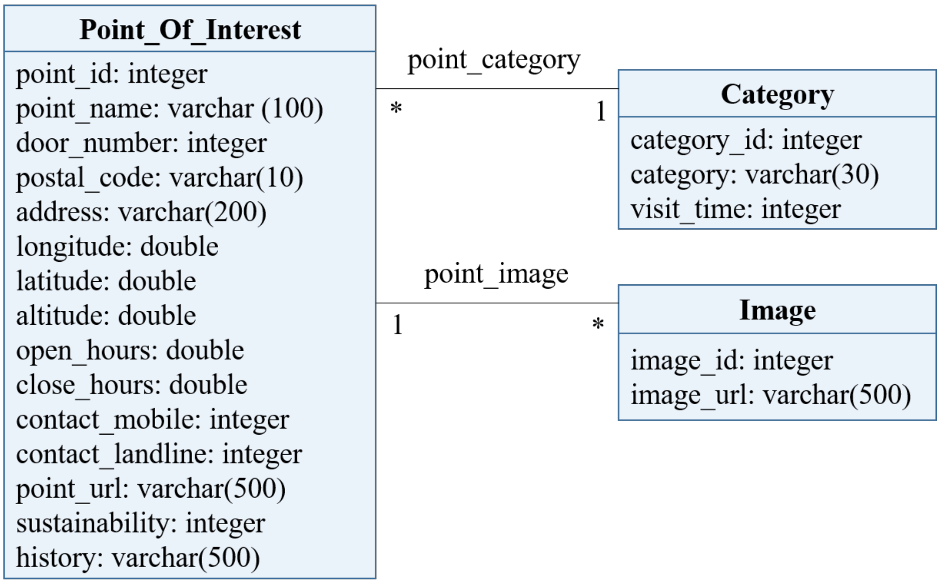


FIGURE 4.7: Class Diagram of POIs Database

TABLE 4.1: POIs Categories

Category_Id	**Category**	**Visit_Time**
0	NA	0
1	Local Store	15
2	Religious Spot	30
3	Viewpoint	15
4	Restaurant	70
5	*Tasquinha*	45
6	Museum	70
7	Monument	30
8	Square	15

be suggested in the tour. Proceeding on the remaining POIs' attributes, there are a *address*, *door_name* and *postal_code* to provide more geographic information. And a *point_url* if exists a web page with more information for the tourist consults. Lastly, in POI information, it is possible to stores two phone contacts, *contact_mobile* and *contact_landline*. The attribute *history* stores the local history, it was a requirement of the mobile app. As referred in the literature review, tourist enjoy to see POI's pictures in mobile apps that provide tours. The class Image was implemented to save the *image_url*, that for testing propose was a link of a public cloud, like *Dropbox*[5].

Table 4.2 exemplifies the metadata for a Point of Interest stored in the database. It has the name *Sé Catedral de Lisboa*, opens at nine hours of the morning and

[5]https://www.dropbox.com/

closes ate seven hours of afternoon. Also it has a sustainability value of 95 and belongs to the category 2, is a Religious Spot (table 4.1).

TABLE 4.2: Example of metadata for a POI

Attributes	**Values**
point_id	116
point_name	Sé Catedral de Lisboa
door_number	1
postal_code	1100-585
address	Largo da Sé
longitude	-9,133335
latitude	38,709795
altitude	35,0
open_hours	9,0
close_hours	19,0
contact_mobile	-1
contact_landline	21 886 752
point_url	https://www.patriarcado-lisboa.pt/site/
sustainability	95
history	"... A sua construção teve início na segunda metade do século XII, após a conquista da cidade aos Mouros por D. Afonso Henriques, ..."
category_id	2

4.3 REST API

REST, Representational State Transfer, is a web architecture to manage information over the internet, [Masse2011]. API, Application Programming Interface, is a set of rules (create, read, update and delete) to provide communication over the HTTP protocol. The web platform, section 4.1, implemented a REST API in the PHP[6], where the information was structure in JSON[7]. In Figure 4.8 is illustrated the REST API structure used.

In the present work the web platform, explained in section 4.1, is the client and the POIs database is the server. The platform sends HTTP Requests (GET, POST and DELETE) the API execute SQL queries in the DB and with the result return information in the JSON format. Next in this section is provided a simple documentation of that implementation.

[6]https://www.php.net/
[7]http://json.org/

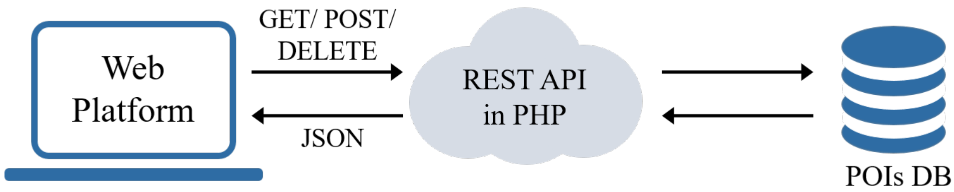


FIGURE 4.8: REST API Illustration

The format JSON to represent a POI is the following, the example is the same as in the table 4.2.


```
{
  "point_id": "116" ,
  "point_name":"Sé Catedral de Lisboa" ,
  "door_number":"1" ,
  "postal_code":"1100-585" ,
  "address":"Largo da Sé" ,
  "latitude":"38,709795" ,
  "longitude":"-9,133335" ,
  "altitude":"35,0" ,
  "opens_hours":"9,0" ,
  "closes_hours":"19,0" ,
  "contact_mobile":"-1" ,
  "contact_landline":"21 886 752" ,
  "point_url":"https://www.patriarcado-lisboa.pt/site",
  "sustainability":"95" ,
  "history":"... A sua construção teve início na segunda
  metade do século XII, após a conquista da
  cidade aos Mouros por D. Afonso Henriques,..." ,
  "category_id":"2"
}
```


In order to add the POIs in to the map in the web platform, it is made a GET for the specified URL and with a path parameter = "point_of_interest". That GET generate the MySQL Query (1), that provides a list of POIs further encode into a list of JSON POIs format by the API.

Where in the web platform an user aims to update or delete a POI marked in the map, it clicks directly on it. Doing that provides the coordinates of the marker

mySQL Query 1 Gathering POIs for map insertion.

```
SELECT * FROM Point_Of_Interest;
```

that corresponds to the coordinates of the POI stored in the database. Then, to show the information of that POI, the interface make a GET for the URL in this format: `URL/point_of_interest/latitude/longitude`. In that URL the latitude and the longitude are the coordinates of that POI, that are used in the MySQL query (2). Again the API gets a POI that encode into JSON to provide to the web platform that will present that information to the web platform user.

mySQL Query 2 Gathering POI with a specific coordinates, to update/delete.

```
SELECT * FROM Point_Of_Interest
WHERE latitude = $path_params[2]
     AND longitude = $path_params[3];
```

Following the sequence, when the user wants to update it fill a box form with all the informations required. Then the platform encode a new POI in JSON format with that new values, and send in an UPDATE request to the API. With that JSON the MySQL query (3) is executed.

mySQL Query 3 Updating POI with a specific coordinates.

```
UPDATE point_of_interest
    SET point_name = $point_name ,
     category_id = $category_id ,
     door_number = $door_number,
            postal_code = $postal_code ,
     address = $address ,
     opens_hours = $opens_hours ,
     closes_hours = $closes_hours ,
     contact_mobile = $contact_mobile ,
     contact_landline = $contact_landline ,
     point_url = $point_url ,
     image_url =  $image_url ,
     sustainability = $sustainability ,
     history = $history
WHERE latitude = $latitude
    AND longitude = $longitude;
```

Otherwise, if the user intend is to delete the POI, the platform send a DELETE request. That request has linked a simpler JSON format only with the coordinates, the ones relatives to the POI that the user wants to remove. Finally the API performs in the database the MySQL query (4).

mySQL Query 4 Deleting POI with a specific coordinates.

```
DELETE FROM Point_Of_Interest
    WHERE latitude = $latitude
    AND longitude = $longitude;
```

An example of that JSON format is the following:


```
{
  "latitude":"38,709795" ,
  "longitude":"-9,133335" ,
}
```

[This page has been intentionally left blank]

Chapter 5

Walking Tour Recommendation

As previously explained (Section 3.1) the walking tour recommendations system is the core of the microservice ROUTE. A tour recommendation system needs a method that is able to generate tours. In the present case study, it is intended to suggest customised tours to the tourist, that is, a tour that obeys the domain restrictions and the stated user preferences. The tour generator algorithms proposed in this chapter aim to solve the MTDOPTW defined in Section 3.3.1). In order to describe the thesis' research behind the tour generation, this chapter is organised as follows. Section 5.1 describes the chosen structure for the graph model and its components. Section 5.2 explains how the tour is generated and defines all the problem's restrictions and criteria (user preferences and domain constraints). There were three different approaches for the MTDOPTW optimisation algorithm that are described in Section 5.3. Finally, Section 5.4 presents the results of the tour generation tests.

5.1 The Graph

The graph represents a structure of a network, in this case, the downtown Lisbon network. Each graph node represents a network component: either a street intersection or a POI. Connections (e.g., the streets) between components are represented by the edges of the graph. The graph is based on a real map of Lisbon for the location used by this case study, specifically the *Junta de Freguesia Santa Maria Maior* area. The map's origin is the *Open Street Maps*, an open-source map provider. It allows us to freely download files with the extension *.osm* with all the map's metadata. The implementations for the tour generator next described were coded using Java Language and there are a few java libraries that use *osm* file maps. To represent the graph structure the java library *Graphstream*[1] was used.

The graph itself is composed by the nodes' set, comprising 16 248 nodes (only for street's intersections) and 18 806 edges (for the streets) and it's illustrated by Figure 5.1. Notice that this is a bidirectional, or non-oriented, graph because Lisbon streets can be walked by pedestrians in both directions.

FIGURE 5.1: Lisbon Graph Illustration

In the figure, each dot represents a node and the lines between nodes are the edges. As referred to in Section 3.3.2, most graph models have weights. The weights represent requirements, constraints or even data stored in the graph structure. In this case, both the nodes and the edges have weights, primarily because of crowding data (stored in the edges) and the sustainability (for the nodes).

5.1.1 The Nodes

As previously referred, each node may represent either an intersection or a POI. In the former case, an intersection, the geographical coordinates and the crowding value are stored, together with the respective altitude. With these values, it is possible to calculate the slope associated with the adjacent edges and, consequently, the physical effort needed for walking through that edge towards the node.

[1]http://graphstream-project.org/

If it is a POI, this means that it is an attraction of the city, that is, a place that may be visited by a tourist. In this case, besides the coordinates, altitude and crowding value, some other values are stored. To promote sustainable POIs, the sustainable value associated with the POI is also stored. Moreover, tourists want to visit POIs of their interest which, as already analysed in previous chapters, need one category. For the POI to be visited, it must be open so that the tourist may enjoy its visit, which means that the visiting hours are also an important metadata. The opening and closing hours are also stored. The estimation of the visiting duration time, i.e., the time expected for the tourist to stay at the POI and visit it, is another needed value, detailed in Table 4.1. Note that the expected visiting duration is important to calculate the total time duration for the tour to be suggested. In summary, the weights associated with the nodes that represent POIs (which includes the weights for the non-POI nodes) are:

- coordinates: x_i-longitude, y_i-latitude and z_i-altitude (degrees);
- wn_i: crowding value (percentage);
- g_i: sustainability value ($\{0, \ldots, 100\}$);
- c_i: category, in the range $\{0, \ldots, 8\}$ (see Table 4.1);
- v_i: visiting time (minutes);
- o_i: opening hour ($\{0, \ldots, 23\}$);
- f_i: closing hour, ($\{0, \ldots, 23\}$).

where $i = 1, \ldots, N_i$, and N_i is the initial number of nodes. Note that, for the sustainability values, zero means that the POI is not sustainable and 100 means that it achieves the maximum level of sustainability.

When the graph is created from the map, it only has nodes representing intersections. Then a connection with the database described on Section 4.2 is made using a MySQL jdbc driver. A SQL query (5) is executed to provide all the POIs stored, and represent them in the graph as nodes, with the respective weights.

Next, another SQL query (6) is made, this time to the table Category, to store the expected visiting time associated with each category. For this, a Hash-map

mySQL Query 5 Gathering POIs for graph node insertion.

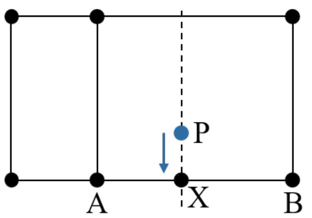
```
   SELECT point_id, longitude, latitude, altitude,
        sustainability, opens_hours, closes_hours,
        category_id
    FROM Point_Of_Interest;
  }
}
```

mySQL Query 6 Gather POIs' category and visiting durations.

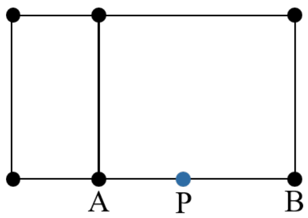
```
    SELECT category_id, visit_time FROM Category;
  }
}
```

that links the category_id with the visit_time is needed so that the latter can be associated to every POI's node.

Finally, the POIs are inserted as a node with the respective weights. To add a new node in the graph, there is the need to approximate the POI's location in the graph structure. So as to minimize the alterations made to the graph and to use the streets to walk straight into (and from) the POI, it needs to be inserted into the closest edge. This step approximation is illustrated by Figure 5.2.



(a) POI out of Graph



(b) POI as Node in Graph

FIGURE 5.2: Poi to Node Illustration

Figure 5.2 represents an excerpt of a graph, with the two nodes A and B representing the POI's nearest intersections, with coordinates (x_a, y_a, z_a) and (x_b, y_b, z_b). In Figure 5.2 (a), the point P - (x_p, y_p, z_p) - represents the geographical coordinates of the POI to be insert into the graph structure. As such, the POI's new node is inserted at the edge nearest to the perpendicular projection of the POI's real location, in this case at point X, the new node, with coordinates (x_x, y_x, z_x). This approximation is computed using Algorithm 1 next described, here the closest edge is obtained by using function *closestEdge(latitude,longitude)* of *graphstream.*

Algorithm 1 Insertion of POIs as graph nodes:

1. Find the closest edge, *closestEdge*$(y_p, x_p) -> AB; AB slope$:
   $m_{AB} = \frac{y_a - y_b}{x_a - x_b};$
3. *AB linear equation:*
   $y = m_{AB} \times x + b_{AB};\ \ b_{AB} = m_{AB} \times x_a - y_a;$
4. *Calculate the slope of PX perpendicular to AB):*
   $m_{PX} = \frac{-1}{m_{AB}};$
5. *Formulate the PX line equation:*
   $y = m_{PX} \times x + b_{PX};\ \ b_{PX} = m_{PX} \times x_p - y_p;$
6. *Calculate* x_x*:*
   $m_{AB} \times x_x + b_{AB} = m_{PX} \times x_x + b_{PX};\ \ x_x = \frac{b_{PX} - b_{AB}}{m_{AB} - m_{PX}};$
7. *Calculate* y_x*:* $y_x = m_{AB} \times x_x + b_{AB};$
8. *Calculate* z_x*:*
   $z_z = \frac{z_a + z_b}{2};$
9. *Add node X and respective weights to the graph.*

Hereinafter, the total number of nodes in a graph (including both intersections and POIs) will be denoted by N.

5.1.2 The Edges

Each edge represents a street or, as seen above, an excerpt of it. It has two associated end-nodes: an "initial" node n_i and a "final" node n_{i+1}. Also, to fulfil the project main aims, it needs to store a crowding value, representing the congestion level of that street, hence making it possible to suggest tours with edges presenting low crowding values.

In order to calculate the effort of walking down the street, the edge needs to store its slope value and the walking distance. As seen in Algorithm 1, the slope c an be calculated using the altitude of the respective end-nodes.

The time can be estimated using another java library - *Graphhopper*[2] - that works with the *osm* file and provides a routing server. Since estimated tour overall time should not overlap the tourist's designated visiting time, an average time for travelling through the edges must also be strored. The weights associated with the edges are listed below, where $j = 1, \ldots, M$, and M denotes the total number of edges:

- we_j: crowding value (percentage);
- s_j: slope (percentage);
- d_j: distance (meters);
- t_j: time (milliseconds);
- e_j: effort (METs [Jetté et al.1990]).

Note that, initially, that is, when the case study location graph was first generated, all the crowding weights were set at zero, since, at the time, there was no information about crowding values.

The effort is measure in METs, Metabolic Equivalents Tasks, one MET is defined as the amount of oxygen consumed at rest (approximately $3,5ml$ of $O_2/kg/min$). For example, if a task requires 2 METs it means it requires twice of the resting effort. This measure is divide in three levels of intensity: light for values smaller than three, moderate for levels between three and six, and heavy values of METs higher than six, Table 5.1. This classification is based in the world health organization[3], and allows to adjust the tour to the physical tourist needs as a user preference.

TABLE 5.1: MET classification into levels of effort

Level	**MET**
1) Light	< 3
2) Moderate	[3,...,6]
3) Heavy	> 6

[2]https://www.graphhopper.com/

[3]https://www.who.int/dietphysicalactivity/physical$_{a}ctivity_{i}ntensity/en/$

The formula of effort, e_i, in METs, is the one use in treadmills[4], equation 5.1.

$$e_j = \frac{(\frac{d_j}{t_i} \times 43,275) \times 0,1 + (\frac{d_i}{t_j} \times 43,275) \times 1,8 \times s_j + 3,5}{3,5} \tag{5.1}$$

$i = j, \ldots, M$

5.2 The Tour

The tour its the result of the TRS and such as the tourist's preferences and domain restrictions are respected. A tour is represented by an oriented path, p, composed by nodes (intersections and POIs) and edges that connect the nodes.

$$p = \{(n_0, n_1), (n_1, n_2), \ldots, (n_{k-1}, n_k)\}$$

and where $n_i, i = 0, 1, \ldots, k$, represent the nodes in the path and (n_j, n_{j+1}) each of the path edges.

Let N be the number of nodes in a path, N_p the number of POIs, where $N_p \leq N$, and M the number of edges in the path. Each tourist specifies an available time to visit, in other words, the path always has a time upper bound - TM. The tour's total time is calculated by adding the visiting times for all nodes (that are POIs) and the walking time in each edge, that is:

$$\sum_{i=1}^{N_p} v_i + \sum_{j=1}^{M} t_j \leqslant TM \pm \delta_r \tag{5.2}$$

where δ_r is a pre-determined time tolerance.

Regarding effort, the tourist can choose a maximum level of effort, EF. Thus, each edge of the recommended tour must obey:

$$e_j \leqslant EF, \quad j = 1, \ldots, M \tag{5.3}$$

To customize the tour's POIs for the tourist, a set of favorite (pre-defined) categories, C, must be selected and therefore, the tour's nodes (if POIs) must obey at

[4]http://www.csecho.ca/wp-content/themes/twentyeleven-csecho/cardiomath

least one category in the set:

$$\exists_{i=1}^{N} c_i \in C \tag{5.4}$$

And finally, the tour's initial time, IT, and the expected final time, $FT = IT + TM$, must be feasible with the opening and closing hours of every POI. If i is a POI with an o_i and a f_i, then, to suggest i in the path, o_i must be smaller than $IT + tt_i$ and f_i higher than $IT + tt_i$, that is,

$$o_i \leqslant IT + tt_i \leqslant f_i \tag{5.5}$$

where tt_i is the time of travel since the first node in the path until the POI i:

$$tt_i = \sum_{j=1}^{M_i} t_j + \sum_{f=1}^{N_{p_i}} v_f \tag{5.6}$$

with M_i the number of edges in the path until i, and N_{p_i} the number of POIs in the path until i.

The previous constraints aim to fulfil the tourist's expectations. However, the recommendation must also insure the goals of promoting sustainability and dispersing the tourists. The main objective is to recommend tours that avoid already congested areas and try to prevent new congestion. In other words, we need to minimize the total crowding value in the tour:

$$\begin{aligned} min \quad & \sum_{j=1}^{M} we_j + \sum_{i=1}^{N} wn_i \\ s.t. \quad & c_i \in C,\ i = 1, \dots, N_p \\ & o_i \leqslant IT + \sum_{j=1}^{M_i} t_j + \sum_{f=1}^{N_{p_i}} v_f \leqslant f_i,\ i = 1, \dots, N_p \\ & e_j \leq EF,\ j = 1, \dots, M \end{aligned} \tag{5.7}$$

The other objective consists in recommending tours with sustainable POIs and to maximize the total sustainability value:

$$max \sum_{i=1}^{N_p} g_i, \tag{5.8}$$

However, since the sustainability is an integer measure, we can turn it around and measure it between 0 (highest sustainability) and 100 (lowest sustainability), obtaining a minimisation problem.
Hence, the overall problem becames a bicriteria (min-min) optimisation problem:

$$\begin{aligned} min \quad & \Big(\sum_{j=1}^{M} we_j + \sum_{i=1}^{N} wn_i \ , \ \sum_{i=1}^{N_p} g_i\Big) \\ s.t. \quad & c_i \in C, \ i = 1, \ldots, N_p \\ & o_i \leqslant IT + \sum_{j=1}^{M_i} t_j + \sum_{f=1}^{N_{p_i}} v_f \leqslant f_i, \ i = 1, \ldots, N_p \\ & e_j \leq EF, \ j = 1, \ldots, M \end{aligned} \tag{5.9}$$

where M_i is the number of edges in the path until i, and N_{p_i} the number of POIs in the path until i.

5.3 Tour Generator Algorithm

This section describes the iterations occurred in the implementation of the bicriteria minimization problem, with paths obeying all the previous restrictions (Equation 5.9). Those iterations lead to the final solution, each one aim to solve a limitation of the previous one. All the code developed can be found on the GitHub Repository[5].

5.3.1 First Iteration

With the aim of solving an SP graph problem to suggest walking tours, for this initial approach we decided to use the pre-implemented search function *AStar* of *graphstream*, which is the implementation of the well-known *A-Star* search - A^*, [Hart et al.1968]. It works by using a cost function that calculates the cost to move from one node to another and a heuristic to avoid exhaustive search. In this case, no heuristic function was given, so the algorithm is, in fact, using the *Dijkstra's* algorithm, [Sniedovich2006]. Moreover, only one criteria (cost) function may be

[5]https://github.com/arhpo-iscteiulpt/TourGeneratorAlgorithm

defined by the user, which implied that we were forced to use a linearisation of the objectives defined in the previous section (Section 5.2).

Figure 5.3 represents two nodes, a and b, and the edge e. that connects them. For each trio $e = (a, b)$, the sum of the crowding values is $w_{ab} = w_a + w_b + w_e$, the sustainability value $g_{ab} = g_a + g_b$, and the total time of walking and visiting that trio, $tt_{ab} = v_a + v_b + t_e$. The cost of suggest the trio, applied in this iteration is the following:

$$Cost_{ab} = w_{ab} + g_{ab} + tt_{ab} \tag{5.10}$$

Note that if a or b are not a POI, then the value of sustainability g_i and of visiting v_i are zero in the function, with $i = a, b$.

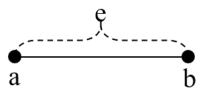


FIGURE 5.3: Node-Edge-Node Example

Thus, this straightforward approach searches for the tour that minimizes the sum of crowding, sustainability and time, that is, the optimisation problem to be solved is to find the tour $p = \{(n_0, n_1), (n_1, n_2), \ldots, (n_{N-1}, n_N)\}$ that solves

$$min(\sum_{i=1}^{N} w_{n_{i-1}n_i} + \sum_{i=1}^{N_p} g_{n_{i-1}n_i} + \sum_{i=1}^{N_p} tt_{n_{i-1}n_i}) s.t. 5.3 \wedge 5.4 \wedge 5.5 \tag{5.11}$$

However, when adding the time in the function it makes a high impact on the cost. The visit time of the POIs is higher than the time of walking each edge, for example a tour with two POIs can have a walking time of 10 minutes and each POI with 30 minutes of visit time, producing the overall tour time of 70 minutes. That causes a problem to the algorithm, suggesting tours with less POIs, then less sustainability. Also preferring to suggest paths that cross crowd areas over spent time in POIs. The time dimension is overwhelming the crowding and sustainability scales, generating tours with few or no POIs because the total value of sustainability of each POI does not compensate the impact of the visiting time. To solve this issue, rather than using the cost function 5.11, a possible solution is to provide a parametric cost function, like the linear convex combination proposed in 5.12. The linear combination is formed by using weights for each variable that add up to one. In this case, we decided to emphasise crowding and sustainability over the total time. Thus, the chosen weights were 40% to the crowding and sustainability and 20% to the time, Equation 5.12.

$$min(0,4\sum_{i=1}^{N} w_{n_{i-1}n_i}) + 0,4\sum_{i=1}^{N} g_{n_{i-1}n_1} + 0,2\sum_{i=1}^{N} tt_{n_{i-1}n_i}) \tag{5.12}$$

However, that does not solve the problem in its totality, the time is a very important constraint for the tourist. And the tourist available time, the TM, was not taking into account, so a second approach arises. The tests performed to both cost equations are described in subsection 5.4.1.

5.3.2 Second Iteration

This second iteration derives from the first in order to find a better solution to the constraint time inefficiency described. It also employs *Dijkstra's* algorithm version offered by *graphstream* and the cost function interface. The total crowding w_{ab}, and sustainability g_{ab} remains, while the variable time will be used in later as a selector of feasible paths.

This approach find feasible paths by working in two phases:

1. Find the k best tours, i.e., the tours optimising both criteria: crowding and sustainability;
2. Select the tours that obey the TM time bound and chose the more adequate.

$$Cost_{ab} = w_{ab} + g_{ab} \tag{5.13}$$

In the first phase the cost function of the trio $e = (a, b)$ (Figure 5.3) is the linear sum of the total crowding value with the total sustainability value (Equation 5.13). In terms of the path's cost function we get Equation 5.14.

$$min(\sum_{i=1}^{N} w_{n_in_{i+1}} + \sum_{i=1}^{N} g_{n_in_{i+1}}) \tag{5.14}$$

In the end of the first phase is created a collection of the k best tours, that were obtained by minimising Equations 5.14 subject to the original constraints.

In the second phase, the set of the alternative k tours is analyzed. Towards this end, for each one of the tours, the total time is computed, that is, the walking time over the edges and the visiting time in each of the nodes that are POIs are added. Then, the paths for which the total time is less or equal to the available time upper bound, TM, are selected (Equation 5.15). From that new set, is chosen the best one, in other words, the one with smallest cost.

$$(\sum_{j=1}^{M} t_j + \sum_{i=1}^{N_p} v_i) \leqslant TM \tag{5.15}$$

The tests performed to this new approach are described in subsection 5.4.2.

5.3.3 Third Iteration

The third approach adds two innovations to the second one:

1. Uses an implementation based in Yen's algorithm to find the k shortest paths by solving a bicriteria problem [Yen1971];
2. The minimization of sustainability and crowding are solved via a ranking approach [Martins and Pascoal2003].

Yen's algorithm was chosen because of the proven efficiency on finding k shortest paths, thus filling in for the gap created by the fact that *Dijkstra*'s algorithm finds only one path (the shortest) linking a given source node with a given target node, that was used in the previous iterations.

The crowding and sustainability criteria are independent, that is, uncorrelated. The cost function taken as a linear sum makes the criteria compensate for each other. As an example, if the tour is crossing higher crowded areas, it searches for lower (null) sustainability nodes to suggest, in order to stabilise the overall cost. But since this problem is a multicriteria shortest path algorithm, in this case bicriteria (BSP), this implies that the best solution for both criteria, crowding and the sustainability, is a non-dominated one (Section 3.3.2).

This third iteration proposes the implementation of a ranking bicriteria algorithm, based on the author explanations found in [Martins and Pascoal2003]. The goal is to find a path p that minimizes the pair of cost functions: sustainability, $f_g(p)$, and crowding, $f_w(p)$ (Equation 5.16), that is,

$$min f(p) = (f_g(p), f_w(p)) \tag{5.16}$$

To achieve this goal, we propose to use a k shortest path algorithm, Yen's algorithm, to find the k best paths for both criteria: sustainability and crowding. The result is a list of paths that optimize each criteria.

Assume that, p_g^* is the best sustainable path and p_w^*, is the best path in terms of crowding and let $f_g(p_g^*) = f_g^*$ and $f_g(p_g^*) = f_w^*$. In this case, the optimal costs are:

$$f(p_g^*) = (f_g^*, f_w(p_g^*)) \tag{5.17}$$

$$f(p_w^*) = (f_w^*, f_w(p_w^*)) \tag{5.18}$$

Those pairs of values are represented by the blue dots, p_g^* and p_w^*, in the graphic of Figure 5.4, representing the best path for each each of the criteria. With those points, we can delimit the area where the potential "optimal" paths for both criteria may be: a square, whose faces are defined by the segment $g = [f_g^*, f_g(p_w^*)]$ and the segment $w = [f_w^*, f_w(p_g^*)]$ for crowding path values.

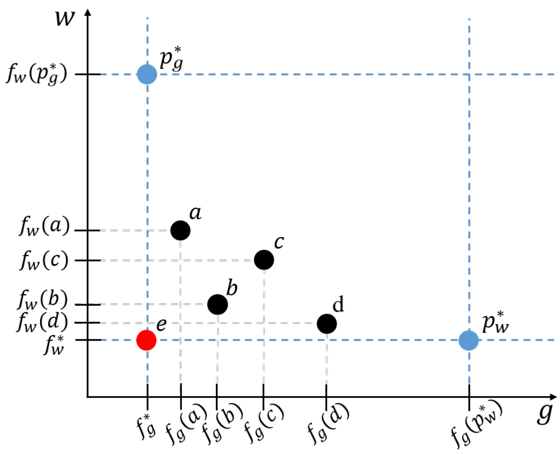


FIGURE 5.4: Ranking Bicriteria Example

To explain how the ranking bicriteria finds the non-dominated paths set, we will use the example in Figure 5.4. Let $P = \{p_g^*, a, b, c, d\}$ be the list of the k first shortest paths ordered by increasing value of sustainability, f_g and e an hypothetical (that is, non-existent) optimal path, the one able to optimise both criteria simultaneously.

A path $p \in P$ is said to be *non-dominated* if there is no other path $q \in P$ such that $f(q) \preceq f(p)$, that is,

$$f_g(q) \leq f_g(p)) \wedge f_w(q) \leq f_w(p) \tag{5.19}$$

If such a q exists, than we say that q *dominates* p.

To evaluate the non-dominance in P the minimum value of the crowding cost found - m_w, must be stored, and in all the P iterations the paths will be compared to that minimum, and if it needs update it. The iteration of P, starts with the path p_g^* that fixes $m_w = f_w(p_g^*)$, and save the path in the list F that store the feasible paths. Then, the path a, it has a smaller cost of crowding what updates the m_w, and it is saved in E, for path b the same process occurs. Next, the path c, it has a higher value than the m_w, so it is discarded. Finally, the path d, has smaller value of m_w, it is stored in F and the iteration of P stops. The set $F = \{p_g^*, a, b, d\}$ it is an approximation of the *Pareto* curve, Figure 5.5. All the paths in F will be sorted by the distance to the perfect and non-existent path e, that has the greatest values for each criteria, and the best path is b.

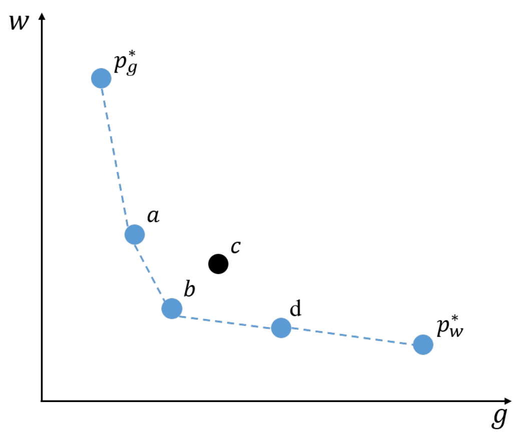


FIGURE 5.5: *Pareto* curve illustration

In Algorithm 2, all the steps of the ranking bicriteria implemented are described.

The set of efficient or non-dominated paths composes the *Pareto* curve [Henig1986]. Then a criteria (coordinate) is fixed for sorting the paths in increasing order of values. For our particular case, we choose to sort using the sustainability function values since it presents less possibilities for different values, making the algorithm more efficient. The list of the sustainable ksps, P, already computed, will be iterated, analysing each path in terms of crowding.

Algorithm 2 Ranking Bicriteria Algorithm

1. Find the k best paths using a SP algorithm for the following criteria: (a) $min \sum_{i=1}^{N} g_{n_i n_{i+1}}$ (b) $min \sum_{i=1}^{N} w_{n_i n_{i+1}}$;
2. Let p_g^* be the best path for criteria (a) with cost $f_g^* = f_g(p_g^*)$ and p_w^* be the best path for criteria (b) with cost $f_w^* = f_w(p_w^*)$;;
3. Define $m_{g_w} = f_w(p_g^*)$;
4. Let $F = \emptyset$ be the set of feasible bicriteria paths;
5. Insert p_g^* in F and Remove p_g^* from P;
6. For each path $p \in P$:
   if $f_w(p) \leqslant m_{g_w}$
   Insert p in F;
   Define $m_{g_w} = f_w(p)$;
   Remove p from P;
7. Sort the the paths in F by the distance - $d(p)$ with $p \in F$ - to the pair (f_g^*, f_w^*), where $d(p) = \sqrt{(f_g(p) - f_g^*)^2 + (f_w(p) - f_w^*)^2}$

The tests performed to this third, and last, algorithm iteration are described in subsection 5.4.3.

5.4 Tests and Validation

The following test results were obtained using a Java environment, *eclipse 2018-09* running in a computer with a Intel Core i5-4200U processor and 8 GB RAM.

The web platform and its inherent database were implemented but, since the parish council experts did not supply the needed information on time for the experiments, the POIs in the graph were randomly generated. The POIs were randomly created in a range $\{0, 100\}$ percentage of the number of nodes in the graph. The opening and closing hours were random too, opening in a range $\{7, 10\}$ hours and closing in a $\{19, 23\}$ hours. The sustainable value varies in a range $\{50, 100\}$, high values to only have POIs that may be suggested.

Note that, as previously explained, the other microservices were not completed at the time thus the nodes and edges values are static. On the other hand, the present results intend to experiment with the different approaches for tour generation. As such, the effort, e_j (see Subsection 5.1.2) were not tested, in view of the fact that this restriction is dependent on the personal user's preferences. Notwithstanding, these should also be tested in a future version of the current implementation.

For the following tests an excerpt of the graph presented in Section 5.1, representing the parish council *Santa Maria Maior* was used, it owns 3650 nodes and 4025 edges, and is illustrated in Figure 5.6.

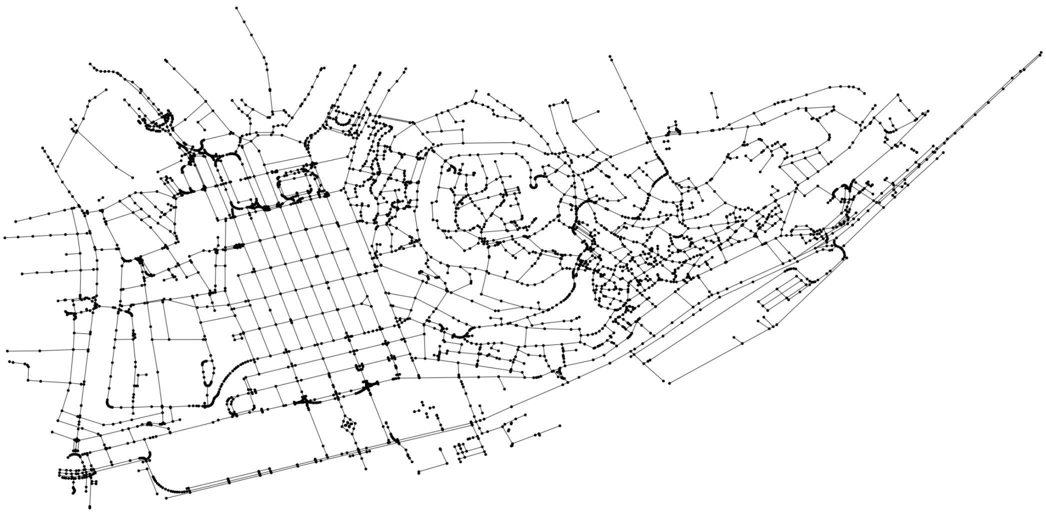


FIGURE 5.6: Graph representing Parish Council *Santa Maria Maior*

Most of the tests presented in this section are visualization tests. Those visualization tests follow the next color code: graph nodes painted in green color are represent POIs, nodes and edges in colors between orange and brown represents the crowded areas, nodes and edges in color red represents the tour generated, and the green nodes with a red stroke represent the POIs suggested in the tour. Color code is illustrated in Figure 5.7.

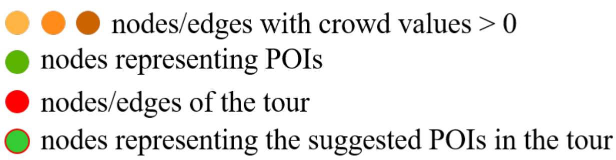


FIGURE 5.7: Color Code Legend

Finally, this section is divide in three subsections, one for each algorithm iteration. Each iteration will be tested with the same scenario to analyse the outputs between them. A test in this context is a call to the algorithm, in order to provide a tour with a specify preferences and domain.

5.4.1 First Iteration Tests

In this subsection, the same test will be performed to each cost equation formulated in Section 5.3.1. Those equations are 5.11 and 5.12, both are a linear sum of the criteria crowding, sustainability and time, but in 5.12 each criteria has different weights.

For this test POIs were randomly created as it was previously explained, with a percentage of 10% of the graph nodes. The initial node for the tour was S and the final node was T, which coordinates are on Table 5.2. The available time was ten hours (TM) and the initial time seven hours of the morning (IT), and all the categories were selected, in order to compute possibilities with all the POIs.

TABLE 5.2: Coordinates of initial node S and final node T

Node	x (longitude)	y (latitude)
S	-9,140675	38,714618
T	-9,132108	38,708775

To the domain were added three crowded areas, those are based on the static crowded data provide by the parish council, after a transformation, as the one explained in Section 3.2, those areas were (A), (B) and (C). They have an epicenter, a crowding value and a radius (in meters), stated on Table 5.3.

TABLE 5.3: Crowd areas for tests

Crowd Area	**Latitude**	**Longitude**	**Radius (meters)**	**Crowding**
A	38,713987	-9,138593	800	70
B	38,711042	-9,138535	500	25
C	38,707842	-9,136744	1000	100

In Figure 5.8 is represented the result of the test described using the cost equation 5.11, the one where the cost is the sum minimization of the three criteria. And in Figure 5.8 is represented the result of the one using the cost equation 5.12, where the cost equation has weights for each criteria. Those weights are 40% for the crowding, 40% for sustainability, and 20% for time. The goal was to increase the value of those two criteria, to provide tours that do not overpass the most crowding areas and that suggest POIs.

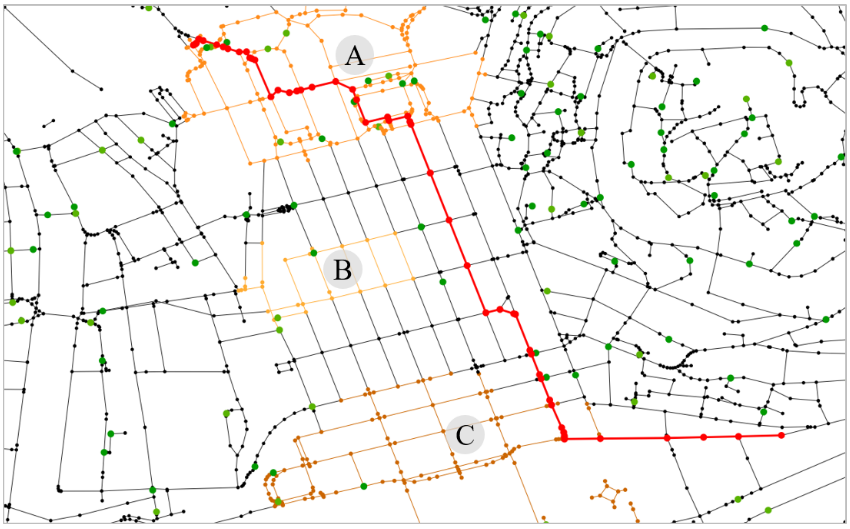


FIGURE 5.8: Graph representing result test with cost equation 5.11

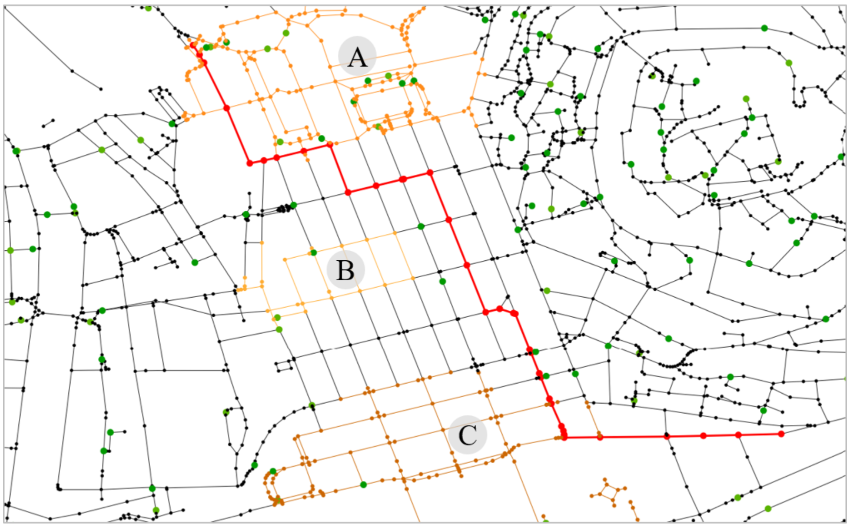


FIGURE 5.9: Graph representing result test with cost equation 5.12

The first result, with no weights, Figure 5.8, shows a tour that overpass the areas (A) and (C) and do not suggest POIs. Also, the second result, with weights, Figure 5.9, the recommended tour over pass the same areas and do not suggest POIs. Although, is possible to visualize that with the weights the tour adjusts, not crossing completely the crowded area (A).

5.4.2 Second Iteration Tests

In this subsection, three visualization tests were performed using the implementation of the second iteration. POIs were randomly created as it was previously explained, with a percentage of 10% of the graph nodes. The initial node for the tour was S and the final node was T, which coordinates are on Table 5.2. The available time was ten hours (TM) and the initial time seven hours of the morning (IT), and all the categories were selected, in order to compute possibilities with all the POIs. The tested scenarios were the following: 1) No crowding, 2) Crowd based on real crowding information and 3) Simulation of overcrowd.

Test (1) result is illustrated in Figure 5.10, where no crowd areas were added, this test works as the control tour. A control tour means, that the changes from the next two tests will be analyzed based on this result. In this case, the tour only suggests one POI, the node (i).

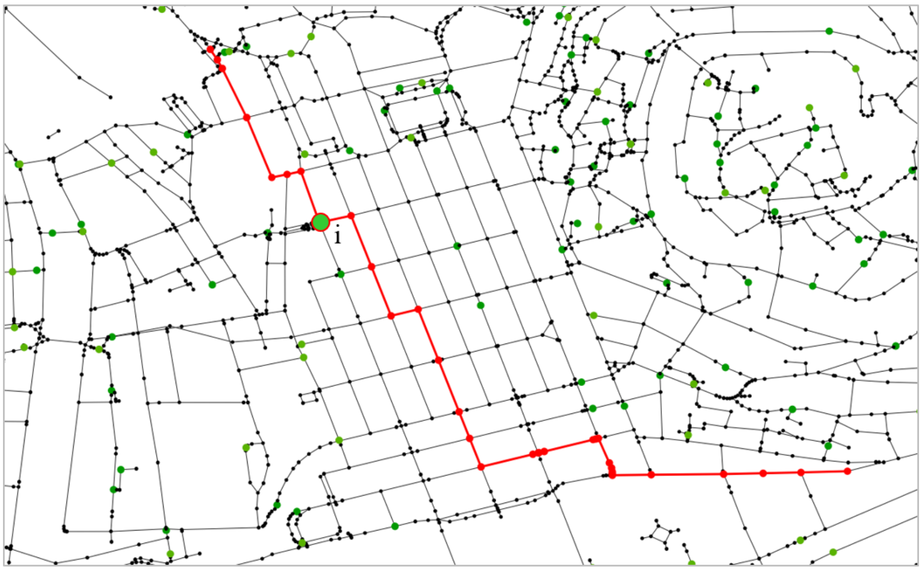


FIGURE 5.10: Graph representing the result of the test (1)

Test (2) is divide into two parts: 2.1) three crowded areas are added and a tour is compute, then 2.2) one crowded area is displaced.

The three crowded areas added on test (2.1) are based on the static crowded data provide by the parish council, after a transformation, as the one explained in Section 3.2, those areas were (A), (B) and (C). They have an epicenter, a crowding value and a radius (in meters), stated on Table 5.3.
The result of this test will also be compared with the results of the first iteration, Figure 5.9, as the domain test is the same. The tour generated, keeping the constraints of test (1) and added the crowded areas described, is illustrated in 5.11.

Comparing this result with the one of the first iteration, we conclude that both results avoid the crowded areas, but in this second iteration POIs are suggested.

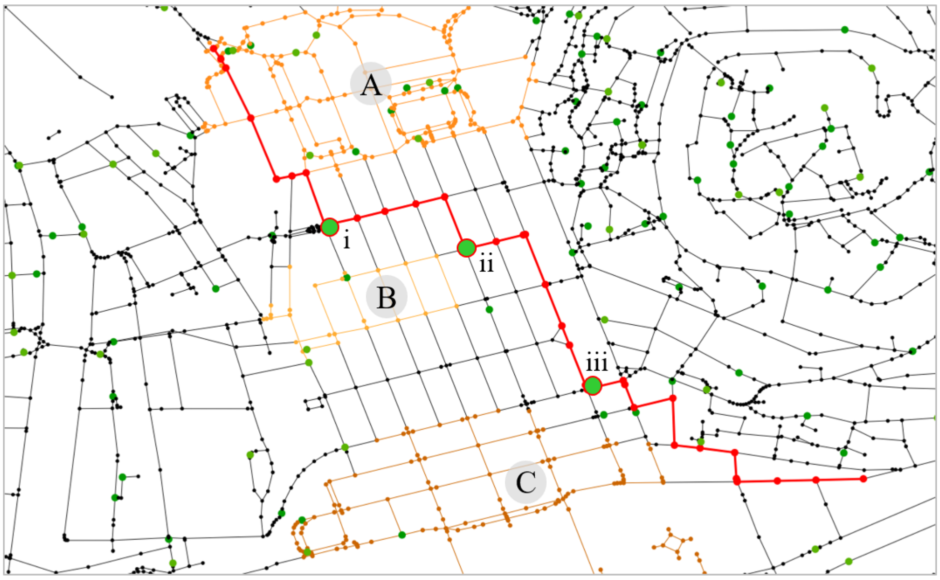


FIGURE 5.11: Graph representing the result of the test (2.1)

Correcting, with this new approach, the main inefficiency of the first iteration. Comparing the test (1) results and test (2.1), we conclude that the crowd area (A) has no impact on the tour, although areas (B) and (C) made the tour differ and adjust to the new domain. It is possible to visualize, comparing those figures, that the tour avoids the crowding, searching for zones where there is no crowding. Regarding the POIs suggested, in this test were three POIs (i),(ii) and (iii). This tour has more sustainability value than the test (1), where it was expecting to suggest more POIs because it did not need to adjust to the crowding values. It is possible to conclude that with a linear sum cost function of crowding and sustainability the algorithm has limitations. However, it seems that the weight of both criteria is equal, the algorithm uses one criteria to compensate the other. Then, when the tour passes through a crowded area, like in result (2.1) where it cross the area (A), it tends to suggest more sustainable POIs to reduce the overall cost.

To analyse the avoidance of crowding areas, the area (B) was shifted to the right. The goal was to visualise the tour adjusting to the new crowding definition. In Figure 5.12 is the result of the test (2.2). Comparing those last two results we conclude, that the tour adjusts to the crowding values, suggesting a different path that minimizes the crossed areas of crowding.

Finally, the test (3), tries to simulate a overcrowding situation by adding a crowd area exactly in the middle of the path suggested in test (1), the area (D) stated on Table 5.4. In Figure 5.13 is the result of the test (3), where is possible to visualize

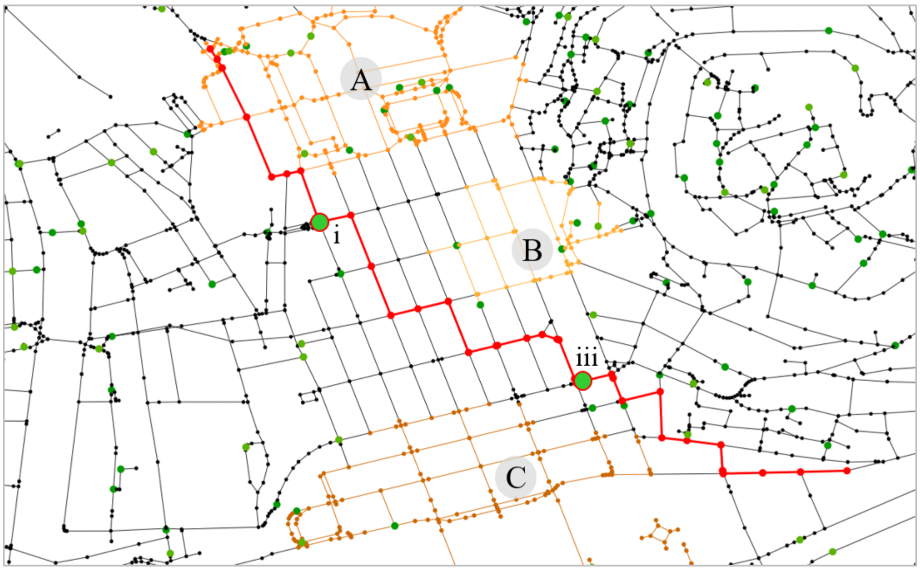


FIGURE 5.12: Graph representing the result of the test (2.2)

again the tour avoiding the crowd area, in this case on final part of the tour. A new POI is found and suggested, POI (iv), in that part of the tour that searches for no crowding areas, and uses the sustainable value to overcome the impact of the giant area.

TABLE 5.4: Crowd areas for the test 3 (second iteration)

Crowd Area	Latitude	Longitude	Radius (meters)	Crowding
D	38,710828	-9,136860	1500	170

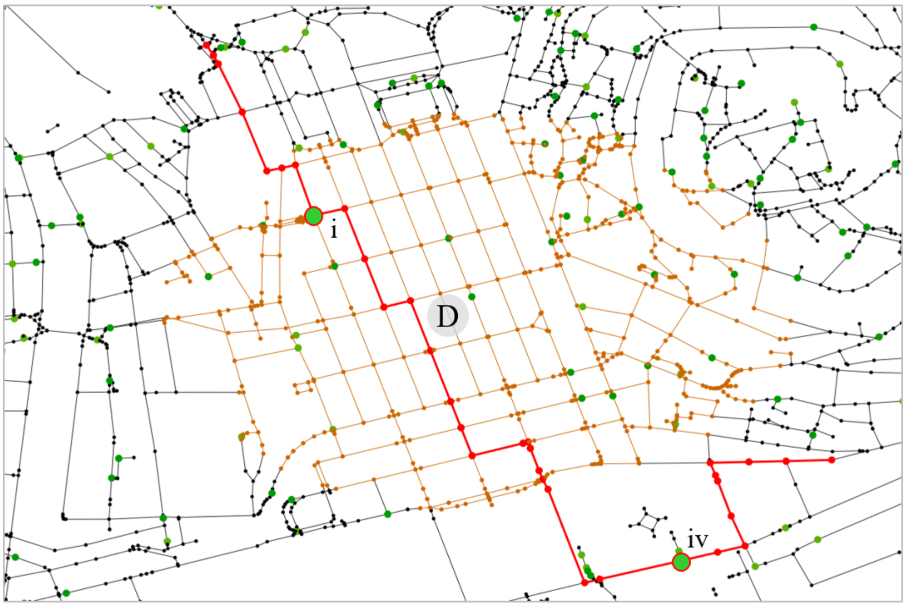


FIGURE 5.13: Graph representing the result of the test (3)

Using the results of this three phase visualization test, we conclude that the second iteration has the limitations stated in Section 5.3.3, and that a BSP (crowding and sustainability) as proposed would have better results. Those limitations are the use of each criteria to compensate the impact of the other, there is the need to stated their independency.

5.4.3 Third Iteration Tests

In this subsection the third iteration will be tested, for that, POIs were randomly created as it was previously explained, with a percentage of 10% of the graph nodes. The initial node for the tour was S and the final node was T, which coordinates are on Table 5.2. The available time was ten hours (TM) and the initial time seven hours of the morning (IT), and all the categories were selected, in order to compute possibilities with all the POIs. The three crowded areas added on this test are based on the static crowded data provide by the parish council, after a transformation, as the one explained in Section 3.2, those areas were (A), (B) and (C). They have an epicenter, a crowding value and a radius (in meters), stated on Table 5.3.

Using the scenario described, the paths that belong to the *Pareto* curve are in the Table 5.5, and their values are represented in the graphic 5.14. In this case, the hypothetical path e has a sustainable value of 8400 and a crowding vale of 70.

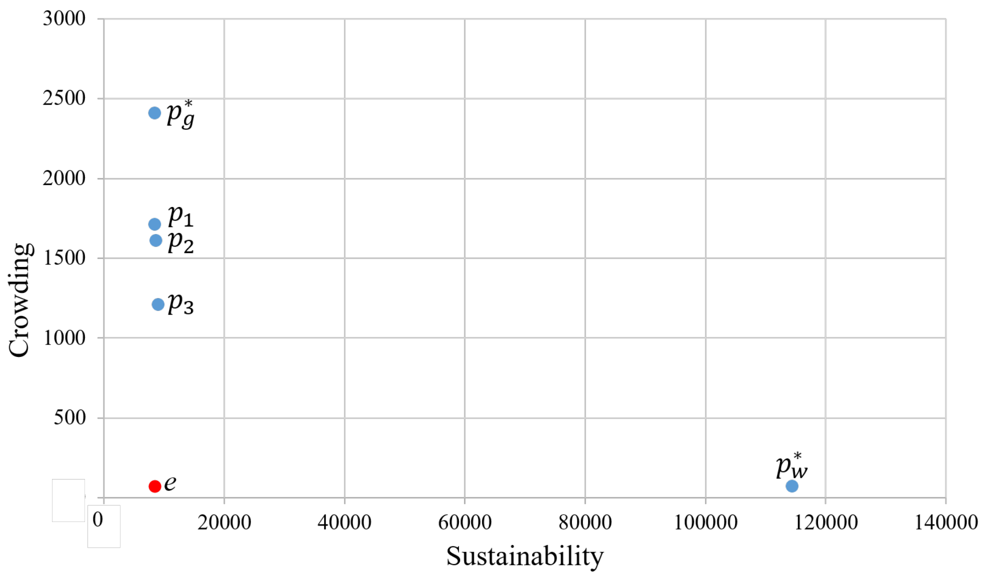


FIGURE 5.14: Graphic representing test results, *Pareto* curve approximation

TABLE 5.5: Bicriteria test results

Paths	Sustainability	Crowding	Distance to e	Number of POIs
p_g^*	8400	2410	2340	0
p_w^*	114446	70	106046	0
p_1	8414	1710	1640,06	1
p_2	8666	1610	1562,804	3
p_3	9014	1210	1294,834	1

The ordered list of the bests paths is $F = \{p_3, p_2, p_1, p_g^*\}$, in Figure 5.15 is represented the result of the best path - p_3 - the one with smaller distance to the hypothetical solution e. The path p_3 overpass two crowded area, (A) and (B), and suggests a POI (v). Comparing that result with the one of the test (2.1) of the previous section, it is possible to conclude that the present result is worse than the previous one. The result path p_3 cross more crowded areas and suggests less POIs.

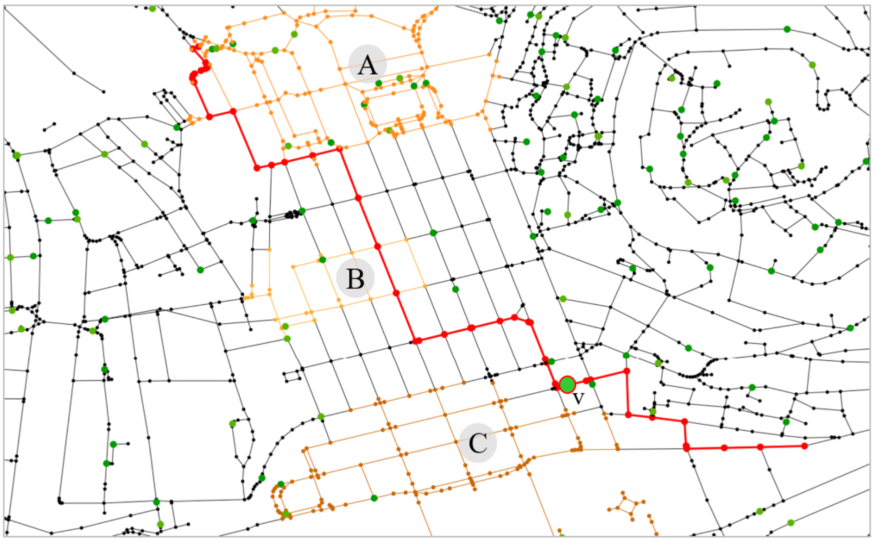


FIGURE 5.15: Graph representing the path p_3

Regarding the second path of the list F, the path p_2, represented in Figure 5.16, it suggests three POIs (i),(vi) and (v) and overpass the three crowded areas. Analysing the two paths, p_2 has three POIs and p_3 has only one, the sustainable value of p_2 should be better (less, in this case) than the sustainable value of p_3. What suggests that the sustainability criteria is not being consider in the correct or supposed way. Leading to the analyse of the path p_g^*, the path computed using only the sustainability has a criteria, the result is represented in Figure 5.17. The result shows a path that do not recommend any POI, suggesting that the ksp in use is minimizing the number of nodes. The ksp chosen should maximize the number of POIs in the path, in order to maximize the sustainability.

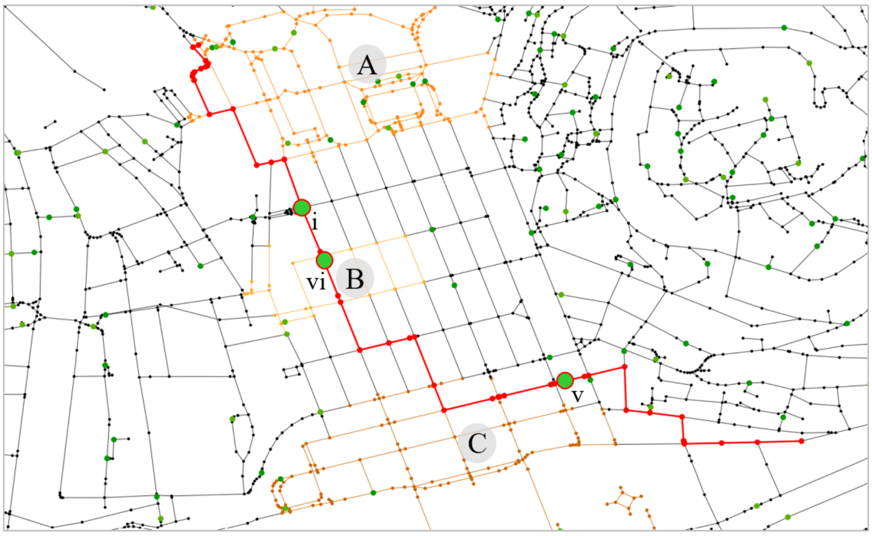


FIGURE 5.16: Graph representing the path p_2

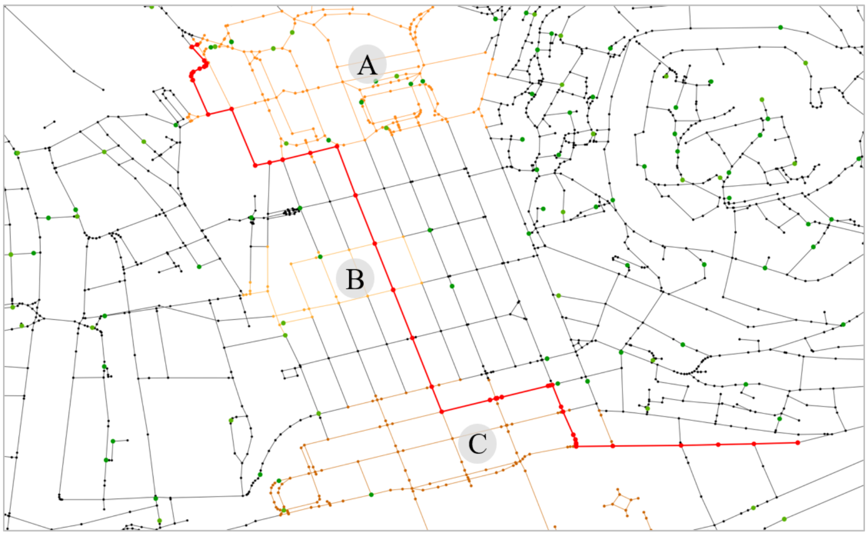


FIGURE 5.17: Graph representing the path p_g^*

5.4.4 Path Validation

The present test aims to validate the SP algorithms used in the previous tests, without the criteria and constraints specific of the case study domain. Using a comparison with a known routing server, the *Google Maps*[6], providing the same initial a final node, Table 5.2, and visualize if the tour recommended is similar. Figures 5.18, 5.19 and 5.20, represent the result of a test where there is no crowding and no POIs to each algorithm iteration.

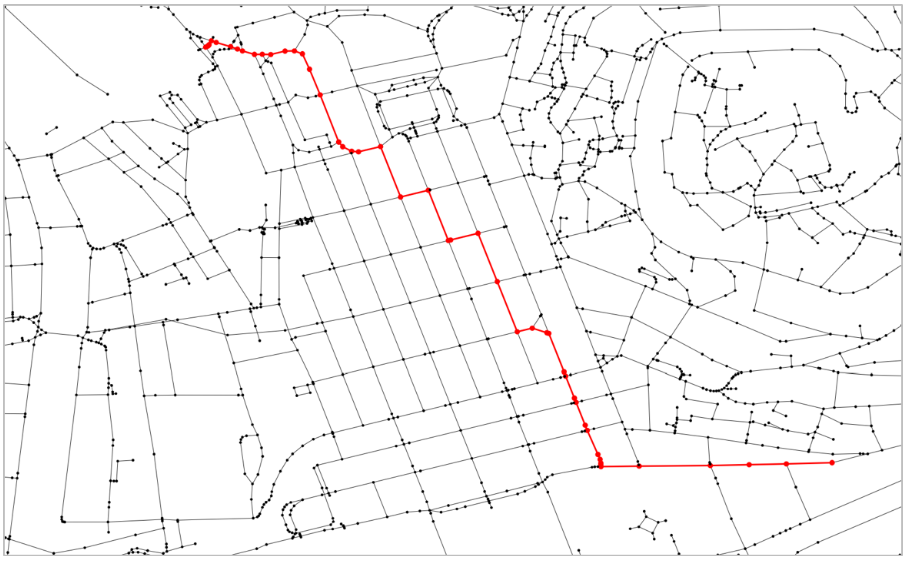


FIGURE 5.18: Representing the SP Validation, First Iteration

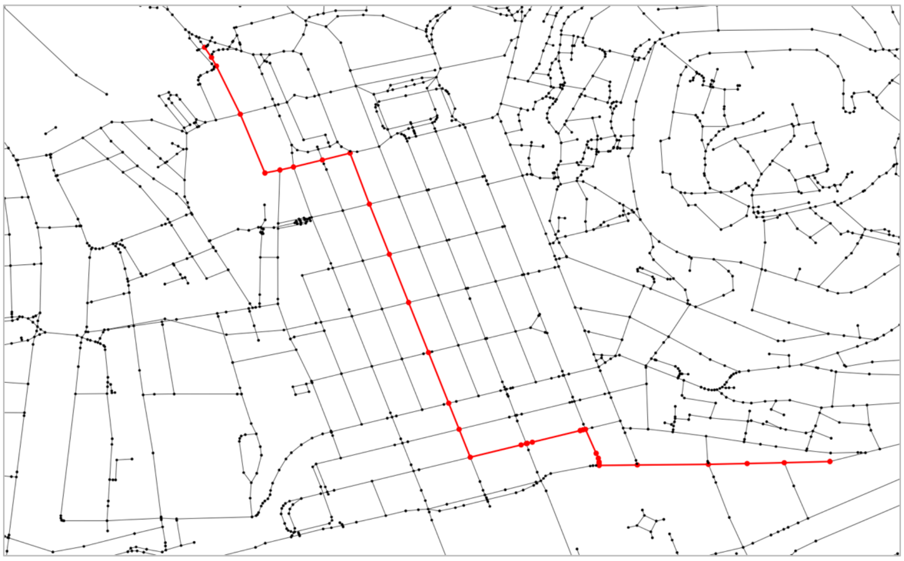


FIGURE 5.19: Representing the SP Validation, Second Iteration

[6]https://www.google.pt/maps

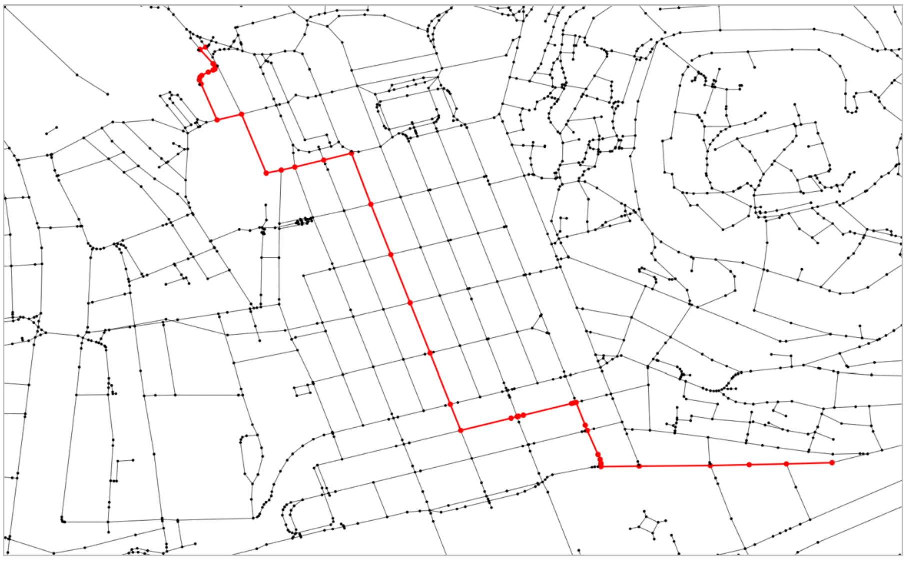


FIGURE 5.20: Representing the SP Validation, Third Iteration

Comparing those results, and the solutions presented in the screen-shoot of *Google Maps*, Figure 5.21, is possible to visualise that the tours suggested identical path choices. This similarity between the streets chosen validates the SP algorithm applied in the implementation of the approaches described in Section 5.3.

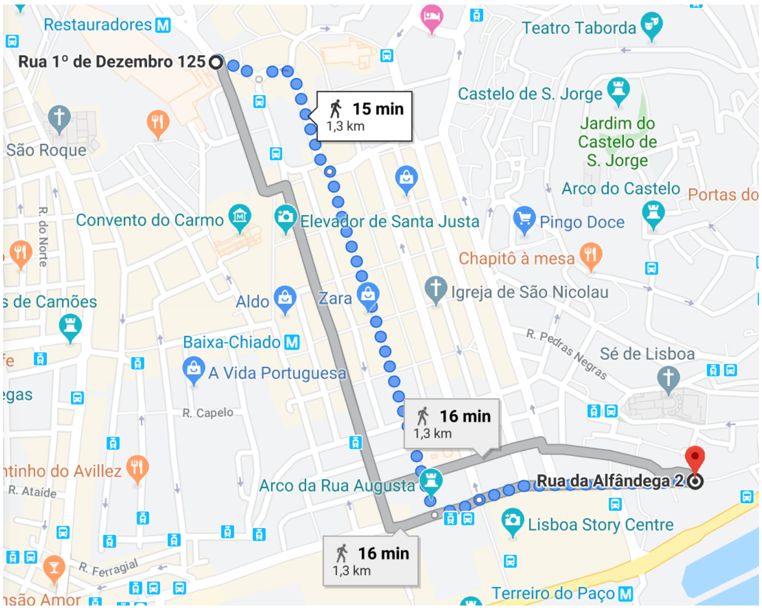


FIGURE 5.21: Representing the *Google Map* solutions

Chapter 6

Conclusion

Regarding the first research question of this dissertation: Is it possible to create a database of sustainable points of interest with knowledge from the parish council workers? We conclude that the construction of this type of database is possible and it was implemented in this dissertation.

The implementation of a sustainable POIs database filled with parish workers' knowledge is possible by using an intuitive web platform. In other words, to the experts' knowledge be reflected in the database they only need an easy way to insert it. The web platform was linked through the database with a REST API, Chapter 4 states that implementation.

Regarding the second research question of this dissertation: Is it possible to implement a tour generator graph-algorithm, that solves multicriteria shortest path with additional constraints (tour domain and tourist preferences)? We conclude that the implementation of a tour generator graph-algorithm with the addiction of complex constraints is possible and it was developed in this dissertation.

The second question is answered with the implementation of a BSP algorithm, over a graph structure, where it was added the tour domain constraints and tourist preferences. Although that implementation does not present perfect results, it allows concluding that a graph-algorithm can be used as a tour generator. Also, it can solve the BSP with complex constraints (tour domain and tourist preferences). The tour domain constraints were the time, time-dependent and time windows of POIs, and the avoidance of crowding areas. The tourist preferences were the time available and the categorized POIs. The implementation and tests are described in Chapter 5.

Chapter 7

Future Work

The future work for the solution presented is organized in three topics, concerning the tour generator algorithm implemented: 1) Future tests to the algorithm; 2) Evolution of the users preferences and tour constraints; and 3) Evolution of the algorithm approach.

(1) Future tests to the algorithm

The tests realized in Section 5.4 needed more details to be analyzed such as the POIs, where the tour passes and why. A tour output should be the the sustainable value of each suggested POIs by the tour. In order to analyze those values and made further conclusions of how the algorithm makes the tour decisions. The sustainable value is important to explain how it affects the cost function. Specially, when the POI is submitted in a crowded area.

A simulator of battery tests should be implemented, to proceed to scale and robustness tests of the algorithm. With the simulator, also a test simulating simultaneous different requests (that is, different user preferences) should be made. In order to understand how it works in a real case scenario, where many tourist are making calls to a new tour.

Some tests with a different graph topology should be done, to test if the graph structure affects the tour generator algorithm. In other words, if the study city were organized in a different way, for example the Manhattan Island, the result could or not differ.

(2) Evolution of the users preferences and tour constraints

Finish the implementation of the effort restriction, and proceed to tests with different levels of effort, and visualize the impact. Another user preference should be implemented, the possibility the tourist add some intermediaries POIs, that he/she really wants to visit. The tour constraint that has higher impact on algorithm was the crowding data. Then a naturally evolution of it should arise, called the dynamic crowding data. Crowd values that are associated with space and time, when the tourist arrives to a area the crowd value of it may differ from when he/she started the tour.

(3) Evolution of the algorithm approach

Adapt the bicriteria algorithm, Section 5.3.3, with a better SP and change the bicriteria to maximize the number of POIs instead of maximize the sustainability. Explore new tools and approaches for the algorithm.

Bibliography

[Andersson et al.2010] Andersson, M., Ntalampiras, S., Ganchev, T., Rydell, J., Ahlberg, J., and Fakotakis, N. (2010). Fusion of acoustic and optical sensor data for automatic fight detection in urban environments. *13th Conference on Information Fusion, Fusion 2010*.

[Blancas et al.2018] Blancas, F. J., Lozano-Oyola, M., González, M., and Caballero, R. (2018). A dynamic sustainable tourism evaluation using multiple benchmarks. *Journal of Cleaner Production*.

[Borràs et al.2014] Borràs, J., Moreno, A., and Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389.

[Bossel1999] Bossel, H. (1999). *Indicators for Sustainable Development : Theory , Method , Applications Indicators for Sustainable Development : Theory , Method , A Report to the Balaton Group*.

[Burton1985] Burton, I. (1985). The World Commission on Environment and Development. *Environmental Policy and Law*, 14(1):4–7.

[Cenamor et al.2017] Cenamor, I., de la Rosa, T., Núñez, S., and Borrajo, D. (2017). Planning for tourism routes using social networks. *Expert Systems with Applications*, 69:1–9.

[Clímaco and Pascoal2012] Clímaco, J. C. and Pascoal, M. M. (2012). Multicriteria path and tree problems: Discussion on exact algorithms and applications.

[Dichter and Gloria2017] Dichter, A. and Gloria, G. M. (2017). Coping with Success : Managing Overcrowding in Tourism Destinations. Technical report.

[Domínguez et al.2017] Domínguez, D. R., Díaz Redondo, R. P., Vilas, A. F., and Khalifa, M. B. (2017). Sensing the city with Instagram: Clustering geolocated data for outlier detection. *Expert Systems with Applications*, 78:319–333.

[Euler1736] Euler, L. (1736). Solutio problematis ad geometrian situs pertinentis.

[European Commission2016] European Commission (2016). *The European Tourism Indicator System*. Number March.

[Farrés2015] Farrés, J. C. (2015). Barcelona noise monitoring network. *Euronoise 2015*.

[Gavalas et al.2017] Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., and Vathis, N. (2017). Scenic route planning for tourists. *Personal and Ubiquitous Computing*, 21(1):137–155.

[Gavalas et al.2014] Gavalas, D., Konstantopoulos, C., Mastakas, K., and Pantziou, G. (2014). A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3):291–328.

[Glavič and Lukman2007] Glavič, P. and Lukman, R. (2007). Review of sustainability terms and their definitions. *Journal of Cleaner Production*, 15(18):1875–1885.

[Gomes et al.2008] Gomes, M., Marcelino, M., and Espada, M. (2008). *Sistema de Indicadores de Desenvolvimento Sustentável*.

[Gunawan et al.2016] Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering Problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.

[Hanai and Espíndola2012] Hanai, F. Y. and Espíndola, E. L. G. (2012). INDICADORES DE SUSTENTABILIDADE: CONCEITOS, TIPOLOGIAS E APLICAÇÃO AO CONTEXTO DO DESENVOLVIMENTO TURÍSTICO LOCAL. *Revista de Gestão Social e Ambiental*, 5(3).

[Hart et al.1968] Hart, P., Nilsson, N., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

[Henig1986] Henig, M. I. (1986). The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2):281–291.

[Hoffman and Pavley1959] Hoffman, W. and Pavley, R. (1959). A Method for the solution of the Nth Best Path Problem. *October*, 6(4):506–514.

[Ilavarasi and Joseph2014] Ilavarasi, K. and Joseph, K. S. (2014). Variants of travelling salesman problem: A survey. In *International Conference on Information Communication and Embedded Systems (ICICES2014)*, volume 62, pages 1–7. IEEE.

[Jetté et al.1990] Jetté, M., Sidney, K., and Blümchen, G. (1990). Metabolic equivalents (METS) in exercise testing, exercise prescription, and evaluation of functional capacity. *Clinical Cardiology*, 13(8):555–565.

[Koren et al.2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37.

[Kotiloglu et al.2017] Kotiloglu, S., Lappas, T., Pelechrinis, K., and Repoussis, P. (2017). Personalized multi-period tour recommendations. *Tourism Management*, 62:76–88.

[Lim et al.2015] Lim, K. H., Chan, J., Leckie, C., and Karunasekera, S. (2015). Personalized Tour Recommendation Based on User Interests and Points of Interest Visit Durations. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 1778–1784.

[Logesh et al.2018] Logesh, R., Subramaniyaswamy, V., and Vijayakumar, V. (2018). A personalised travel recommender system utilising social network profile and accurate GPS data. *Electronic Government, an International Journal*, 14(1):90.

[Martins and Pascoal2003] Martins, E. and Pascoal, M. (2003). A new implementation of Yen's ranking loopless paths algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2).

[Masse2011] Masse, M. (2011). *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces.*

[Mrazovic et al.2017] Mrazovic, P., Larriba-Pey, J. L., and Matskin, M. (2017). Improving Mobility in Smart Cities with Intelligent Tourist Trip Planning. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 897–907. IEEE.

[Park et al.2016] Park, S., Bourqui, M., and Frias-Martinez, E. (2016). MobInsight: Understanding Urban Mobility with Crowd-Powered Neighborhood Characterizations. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, volume 0, pages 1312–1315. IEEE.

[Peeters et al.2018] Peeters, P., Gössling, S., Klijs, J., Milano, C., Novelli, M., Dijkmans, C., Eijgelaar, E., Hartman, S., Heslinga, J., Isaac, R., Mitas, O., Moretti, S., Nawijn, J., Papp, B., and Postma, A. (2018). *Research for TRAN Committee - Overtourism: impact and possible policy responses*. European Parliament, Directorate General for Internal Policies, Policy Department B: Structural and Cohesion Policies, Transport and Tourism.

[Peng et al.2015] Peng, P., Tian, Y., Wang, Y., Li, J., and Huang, T. (2015). Robust multiple cameras pedestrian detection with multi-view Bayesian network. *Pattern Recognition*, 48(5):1760–1772.

[Raith and Ehrgott2009] Raith, A. and Ehrgott, M. (2009). A comparison of solution strategies for biobjective shortest path problems. *Computers & Operations Research*, 36(4):1299–1331.

[Ranneries et al.2016] Ranneries, S. B., Kalør, M. E., Nielsen, S. A., Dalgaard, L. N., Christensen, L. D., and Kanhabua, N. (2016). Wisdom of the local crowd. In *Proceedings of the 8th ACM Conference on Web Science - WebSci '16*, pages 352–354, New York, New York, USA. ACM Press.

[Ricci2002] Ricci, F. (2002). Travel recommender systems. *IEEE Intelligent Systems*, 17(6):55–57.

[Ricci et al.2015] Ricci, F., Shapira, B., and Rokach, L. (2015). *Recommender Systems Handbook*. Springer US, Boston, MA.

[Sedgewick and Wayne2011] Sedgewick, R. and Wayne, K. (2011). *Algorithms, Fourth Edition*.

[Smith and Jungnickel1999] Smith, D. K. and Jungnickel, D. (1999). *Graphs, Networks and Algorithms.*, volume 50.

[Sniedovich2006] Sniedovich, M. (2006). Dijkstra's algorithm revisited: the dynamic programming connexion. *Control And Cybernetics*, 35(3):599–620.

[Stahlschmidt et al.2016] Stahlschmidt, C., Gavriilidis, A., Velten, J., and Kummert, A. (2016). Applications for a people detection and tracking algorithm using a time-of-flight camera. *Multimedia Tools and Applications*, 75(17):10769–10786.

[Statistics Portugal2018] Statistics Portugal (2018). Estatísticas do turismo 2018. Technical report, Instituto Nacional de Estatística.

[Tsiligirides1984] Tsiligirides, T. (1984). Heuristic Methods Applied to Orienteering. *Journal of the Operational Research Society*, 35(9):797–809.

[Tudorache et al.2017] Tudorache, D., Simon, T., Frenț, C., and Musteaţă-Pavel, M. (2017). Difficulties and Challenges in Applying the European Tourism Indicators System (ETIS) for Sustainable Tourist Destinations: The Case of Braşov County in the Romanian Carpathians. *Sustainability*, 9(10):1879.

[Vaishnavi and Kuechler2004] Vaishnavi, V. and Kuechler, B. (2004). Design Science Research in Information Systems Overview of Design Science Research. *Ais*, page 45.

[Vansteenwegen et al.2011] Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. (2011). The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10.

[Vansteenwegen and Van Oudheusden2007] Vansteenwegen, P. and Van Oudheusden, D. (2007). The Mobile Tourist Guide: An OR Opportunity. *OR Insight*, 20(3):21–27.

[Yen1971] Yen, J. Y. (1971). Finding the K Shortest Loopless Paths in a Network. *Management Science*, 17(11):712–716.

[Zhou et al.2015] Zhou, X., Xu, C., and Kimmons, B. (2015). Detecting tourism destinations using scalable geospatial analysis based on cloud computing platform. *Computers, Environment and Urban Systems*, 54:144–153.

[This page has been intentionally left blank]

Appendices

AppendixA - Real Crowding Data

Figure 1: Partner Parish Council Estimation of Crowding Data