

POLITECNICO DI MILANO

Scuola di Ingegneria Civile, Ambientale e Territoriale

Master of Science in Geoinformatics Engineering



**Innovative Web GIS functionalities for the promotion
of natural parks: User experience improvement by
Multi-Criteria and Spatial Algorithms blending.**

Master Graduation Thesis of:

Matteo Rizzi

Matr. 905016

Supervisor: **Prof. Maria Antonia Brovelli**

Co-Supervisor: **Dr. Daniele Oxoli**

Academic Year 2019-2020

Matteo Rizzi

Innovative Web GIS functionalities for the promotion of natural parks: User experience improvement by Multi-Criteria and Spatial Algorithms blending.

Master Thesis in Geoinformatics Engineering, December 31, 2019

Supervisors: Maria Antonia Brovelli and Daniele Oxoli

Politecnico di Milano

Master of Science in Geoinformatics Engineering

Scuola di Ingegneria Civile, Ambientale e Territoriale

Piazza Leonardo 32

20133 and Milano

Acknowledgement

With this thesis, I conclude my studies at Politecnico di Milano. Before I go, I would like to extend my sincere thanks to all persons who have made this possible:

- My supervisor Professor Maria Antonia Brovelli, for sharing her passion for what she does and she teaches. She has been a real source of inspiration for my academic journey;*
- My co-supervisor Dr Daniele Oxoli, for supervising this project with his insightful ideas, useful advice and continued support. He provided encouragement and guidance throughout the project, both from a professional and human perspective;*
- My parents Pietro and Carmen, for their encouragement and unconditional love during the last years. They have always reminded me that no matter what happens, they will always be there for me;*
- My sister Eleonora, for always being my greatest certainty. She has been the example that I hope to become in a few months for her child, my future little grandchild;*
- My friends, roommates, colleagues and relatives, for accompanying me both in the best and worst moments. Each of them has left, in its own way, an imprint in my heart;*

Here I am, and it is mainly thanks to all of you.

Milan, April 2020

Abstract

Nowadays, sustainable development and preservation of territories present several challenges which innovative and advanced technological tools aid to address and accomplish. According to that, it is presented here the design and the development of a Web Application for enriching the tourism management system of natural parks. The system is conceived for the Insubria Region within the INSUBRI.PARKS project, funded by the Interreg program of the European Union. The Insubria is a historical-geographical area spread between Northern Italy and Southern Switzerland embedding an estimable historical and natural heritage. Nevertheless, the tourism potential of the region is not fully exploited due to a complex political context within the geographical area which extends across different local and national jurisdictions. The main goal of the INSUBRI.PARKS project is to improve the tourism attractiveness of the Insubria Region natural parks by also by supplying a modern IT infrastructure to empower territorial marketing operations. In this context, this thesis work aims at developing a Web Geographic Information System (WebGIS), providing the visualisation of geospatial data together with novel functionalities to improve the users' experience. These include interactive plugins blending Multi-Criteria Decision Making techniques and spatial algorithms, allowing users to interact with the WebGIS to plan their experiences in the parks according to their expressed preferences. The development of the experience planning plugin and its integration into a standard client-server WebGIS architecture represents the main outcome of this work. Both the design and implementation of a prototype of the plugin are presented. Discussion on benefits deriving from the adoption of the experience planning plugin in the context of the INSUBRI.PARKS project is provided together with future directions and improvements for this work. The applications here discussed leverage exclusively Free and Open Source Software thus providing to the tools the possibility to be empowered, replicate and improved.

Sommario

Al giorno d'oggi, lo sviluppo sostenibile e la conservazione dei territori presentano numerose sfide che strumenti tecnologici innovativi aiutano ad affrontare e realizzare. A questo proposito, viene qui presentato lo sviluppo di una applicazione Web studiata per arricchire un sistema di gestione del turismo nei parchi naturali. Il sistema è concepito per la regione dell'Insubria nell'ambito del progetto INSUBRI.PARKS, finanziato dal programma Interreg dell'Unione Europea. L'Insubria è un'area storico-geografica che si estende tra il Nord d'Italia e la Svizzera meridionale incorporando un patrimonio storico e naturale di inestimabile valore. Tuttavia, il potenziale turistico della regione non è pienamente sfruttato a causa del complesso contesto politico dell'area geografica di interesse, la quale si estende attraverso diverse giurisdizioni locali e nazionali. L'obiettivo principale del progetto INSUBRI.PARKS è quello di potenziare l'attrattiva turistica dei parchi naturali Insubrici anche attraverso lo sviluppo di una moderna infrastruttura IT per supportare le operazioni di marketing nel territorio. In questo contesto, questo lavoro di tesi mira a sviluppare un WebGIS per la visualizzazione di dati geospaziali insieme a nuove funzionalità per migliorare l'esperienza degli utenti. Tra queste è incluso un plugin interattivo che sfrutta tecniche decisionali a criteri multipli e algoritmi spaziali per consentire agli utenti di interagire con il WebGIS per pianificare esperienze di visita personalizzate. L'integrazione del plugin di pianificazione dell'esperienza all'interno di un'architettura WebGIS client-server rappresenta il risultato principale di questo lavoro. Vengono dunque presentati sia il design che l'implementazione di un prototipo del plugin. Successivamente, insieme ad una discussione sui vantaggi derivanti dall'adozione del plugin nel contesto del progetto INSUBRI.PARKS, vengono fornite indicazioni e miglioramenti futuri per questo strumento. Le applicazioni qui discusse sfruttano esclusivamente software gratuiti e open source che forniscono a chiunque la possibilità di migliorare e riutilizzare in altri contesti gli strumenti proposti.

Contents

1	Introduction	1
2	The INSUBRI.PARKS Project	3
3	State-of-Art	7
4	Multi-Criteria Decision Making	11
4.1	Decision Making Process	12
4.1.1	Decision Making Models	16
4.2	MADM Model	17
4.2.1	AHP-Revisitation for the INSUBRI.PARKS	18
5	Path Algorithms	25
5.1	Background in Graph Theory	25
5.1.1	Walk, Trail and Path	28
5.2	Shortest Path Algorithms	30
5.3	K-Shortest Paths	33
5.3.1	Yen's algorithm	34
5.3.2	The integration between AHP and KSP	37
6	Data and Software	39
6.1	Creation of the Geo-Database	39
6.1.1	Data Collection	39
6.1.2	Data Exploration and Processing	41
6.1.3	Database implementation	42
6.2	Introduction to WebGIS	45
6.2.1	Standard Architecture	47
7	INSUBRI.PARKS WebGIS	51
7.1	Architecture	51
7.1.1	Python	56
7.1.2	Django	57
7.1.3	pgRouting	59
7.2	Requirements Engineering	62
7.2.1	Goals	63

7.2.2	Requirements	63
7.3	Main Functionalities	64
7.3.1	Map	65
7.3.2	Tourist Experience Planner tool	67
8	Conclusion	73
	List of Figures	75
	List of Tables	77
	List of Acronyms	79
	Bibliography	83

Introduction

” *Look deep into nature, and then you will understand everything better.*

— **Albert Einstein**
(Theoretical Physicist)

Nowadays, the management of protected areas and natural parks have become a critical task since the tourism promotion, necessary for the economic sustainability of the management operations, always has to deal with the sustainable exploitation of their environmental resources. In this context, the development of Information Technology (IT) applications that builds a bridge between tourists and geographical data has a crucial role.

A meaningful example is the INSUBRI.PARKS [1], an Interreg [2] funded project aiming at the cooperation between natural parks along the cross-border area between Italy and Switzerland. The final purpose of this project is to develop a network of touristic experiences and facilities through cooperation among project partners.

The introduction of innovative technologies in the parks tourism promotion is the primary focus of the INSUBRI.PARKS project. To achieve the above, in this thesis work a number of activities are carried out which encompasses the creation of the project database, the design and implementation of ad-hoc IT applications and their deployment on the Web. The project database is designed to include geolocated information relating to touristic attractions and the pedestrian mobility of each park [3]. A census activity of the data owned by the partners is presented to discover, harmonise, and to digitise information of strategic interest for this project.

The data infrastructure aims at the implementation of specific Web applications to both interact with the project area through the access to the digital cartography of the area as well to plan custom experiences on the territory aided by smart IT tools [4]. Indeed, the primary goal of this thesis work is the development and the integration of an experience planning plugin into a standard server-client WebGIS architecture. This peculiar functionality is made available through an interface that will guide the user in querying the database according to their preferences, which are considered and weighed through the integration with Multi-Criteria decision models and spatial algorithms.

The integration of GIS and Multi-Criteria Analysis is the preferred way to develop this spatial decision support system [5]. The use of GIS contributes to the collection of information, production of derived information and handling a large volume of data, which helps to consider the problem in all its complexity and describe the alternatives [6] [7]. In the specific case of this thesis, a revisit of a well-known Multi-Criteria Model (the Analytical Hierarchy Process) is proposed. In contrast to the conventional approach, in this work, the result is achieved almost in real-time by providing a suitable solution through the help of a spatial graph algorithm (K Shortest Paths) [8]. This association of GIS and MCDA can be thought of as a process that transforms and combines geographical data and value decision maker's preferences to obtain information for spatial decision making [9].

A prototype of the IT infrastructure mentioned above is developed by using the Spina Verde Park, which is one of the INSUBRI.PARKS project areas, as a case study. The IT infrastructure development takes advantage exclusively of Free and Open Source Software (FOSS) components. Besides the design of the system architecture, both goals of the system and user requirements are discussed. The assets provided by the use of FOSS technologies for the development of this standardised IT infrastructure is considered by highlighting the benefits deriving from the use of open and shared software platform in the context of the INSUBRI.PARKS project.

Focusing on the key points mentioned above, the rest of the document is organised as follows. The INSUBRI.PARKS project is briefly introduced in **Chapter 2**. Then, **Chapter 3** contains the State-of-Art of the current progress of GIS in the field of tourism. **Chapter 4** outlines the Multi-Criteria Decision Making topic. **Chapter 5** focus on the Path Algorithms exploited in this work. **Chapter 6** describes the whole data structure together with the standard WebGIS solutions. A prototype on Spina Verde Park is presented in **Chapter 7**, while, finally, in **Chapter 8**, both key conclusions and the future directions of the work are discussed.

The INSUBRI.PARKS Project

” *You may have the universe if I may have Italy.*

— **Giuseppe Verdi**
(Composer)

In this chapter, a brief overview of the INSUBRI.PARKS project is outlined together with the description of the area concerned. This is followed by a discussion on the main objectives of the project.

INSUBRI.PARKS is a project financed by the Interreg transnational cooperation program between Italy and Switzerland 2014-2020 (see <https://interreg-italiasvizzera.eu>). The project proposal was formulated by some of the Lombardy Region (Northern Italy) natural parks belonging to the Insubria Region, from which the project derives its name. The Insubria is a historical-geographical region extending approximately between the Adda river (East) and the Sesia river (West), and between the St. Gotthard Pass (Southern Switzerland) in the north and the Po Valley in the south (see **Figure 2.1**). This area embraces many interconnected landscapes extending from the plain to the Alpine surroundings, crossing the sub-alpine lakes, such as the Como Lake, and including many urban centres of both the Lombardy Region and the Canton of Ticino (Southern Switzerland).

The INSUBRI.PARKS project aims at overcoming the title of ”green islands” that is usually entitled to natural parks located along urbanised regions, such as the Insubria Region. This aim can be achieved by designing and implementing strategies to integrate the natural, cultural and historical patrimony characterising the project area and - at the same time - to leverage the peculiarities of each natural park and neighbouring areas. Having this in mind, the project team is composed primarily by the natural parks of the Insubria Region. It is characterised by its specific identity and oneness straddling both Italy and Switzerland. The Italian Parks that are part of the project partners have pursued, since from their institution thirty years ago, based on their statutory objectives, strategies to enhance their territory and cultural heritage, landscape and natural resources present in the individual protected areas. In support of the activity of enhancement has been built, over time, a dense network of paths and paths themes that constitute the backbone of the green tourist offer of the Parks. In the strategic areas of its competence, the parks have strengthened the structures

accommodation able to offer additional services to hikers who, individually or in a group, are present in the protected area. The parks involved in the project are:

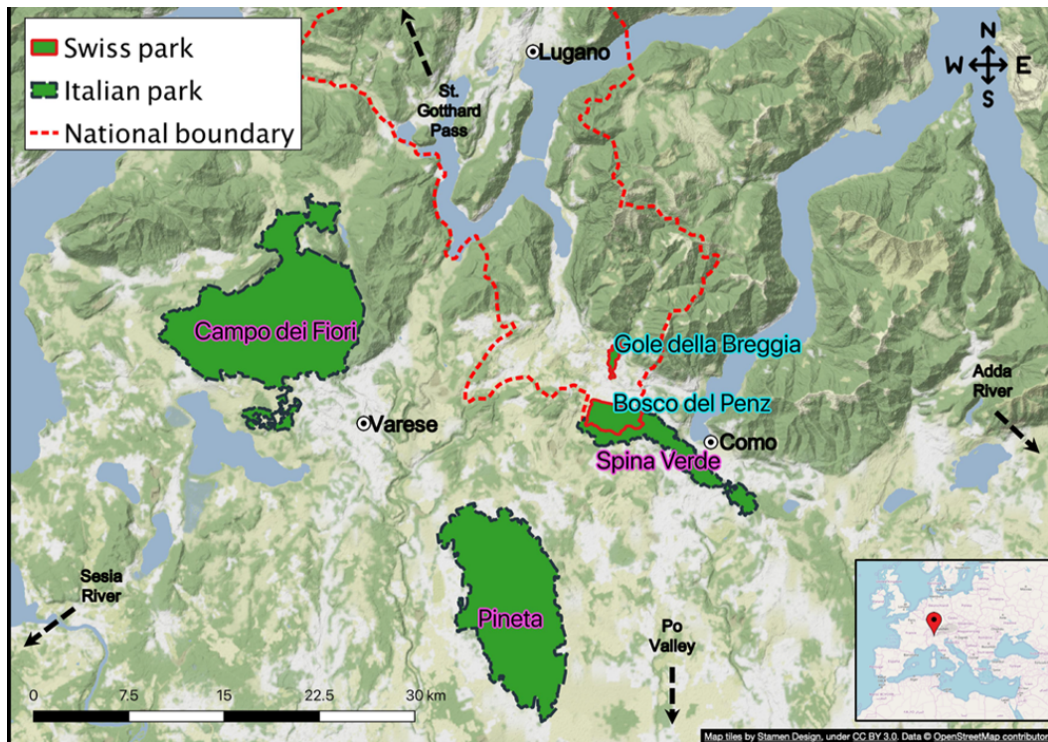


Fig. 2.1: The INSUBRI.PARKS project area [1].

- **The Spina Verde Regional Park** (Como, Italy) is a true outdoor museum: along the 80 kilometres of trails, it is possible to observe the archaeological sites of the protohistoric era, the monumental medieval remains, the Torre del Baradello up to most recent trenches of the First World War (see <https://www.spinaverde.it/main>);
- **The Campo Dei Fiori Park** (Varese, Italy) includes two important massifs, the Campo Dei Fiori and the Martica, separated from the Valle Della Rasa, from where the Olona is born. The particular location geographical and geological features have favoured the establishment of much vegetation varied, with chestnut and beech woods, areas of rupicolous flora and wetlands, rich in faunal presences. Relevant historical-architectural presences, such as the Sacro Monte (Unesco Heritage), the Grande Albergo, the Liberty villas, the Badia di Ganna and the Rocca di Orino (see <http://www.parcocampodeifiori.it>);
- **The Pineta Park** (Como, Italy) is a unique example of almost 48 square kilometres of piedmont pinewood sheltered from the largest plain of Europe West. La Pineta, in its uniqueness, represents a particular piece in the panorama

"local" forest, being the largest and most dense arboreal expanse of the area. This characteristic makes it rich in species typically forest animals to the point of being an important centre of diffusion towards smaller tree-lined surfaces (see www.parcopineta.org);

- **Gole Della Breggia Park** (Chiasso, Switzerland) represents, from a scientific point of view, a jewel naturalistic and geological. In particular, the geological outcrops constitute an exceptional document that covers events almost continuously geological events over about 100 million years, between the Jurassic and the Tertiary. From another point of view, it is an area with high tourist-recreational use in a heavily inhabited context (see <https://www.mendrisiottoturismo.ch/it/commons/details/Parco-delle-Gole-della-Breggia/52602.html>);
- **Collina del Penz** (Chiasso, Switzerland) has been working for some time to be like a real park, full of peculiarities historical architectural and naturalistic biotopes of national interest, in physical and geographical continuity, among other things, with the Spina Verde Regional Park on the Lombard side (see https://it.wikipedia.org/wiki/Collina_del_Penz).

Parks are also main stakeholders of the project that is expected to trigger the actual tourism potential of this border area by creating better physical and virtual connections among the local tourist attractions. The no-profit organisation Pro Val Mulini ONLUS ([10]) is committed regarding the historical and cultural aspects of the project. The Swiss Regional Tourism Office (OTR) Mendrisiotto Turismo ([11]) is in charge of providing the know-how on best tourism management practices. Finally, the scientific partners Politecnico di Milano ([12]) from the Italian side, and The University of Applied Sciences and Arts of Southern Switzerland (SUPSI, [13]) from the Swiss side, are responsible for the technological component of the project that incorporates the digital data collection and analysis as well as the development of the IT infrastructure.

Therefore, the main objectives of INSUBRI.PARKS are to promote the peculiarities mentioned above in the interests of the Insubria Region tourism potential. Three main strategies are considered (see **Figure 2.2**). The first one is to improve and expand the available touristic experiences offered by the parks. Several touristic packages, intended as thematic itineraries, will be designed and empowered with new infrastructures as well as extended in terms of cultural content. The second strategy consists of the development of a centralised communication and marketing office for the promotion of the project to the external audience through the widespread use of emerging Web and social media support. In parallel, a central management office will be set up to ensure coordination among partners activities

as well as control over both the administrative and technical edges of the whole project. Finally, the whole system will take advantage of the IT infrastructure that is designed both to deal with geospatial data collection and to connect the integrated management of the parks with the virtual exploration of the project area by the tourists.

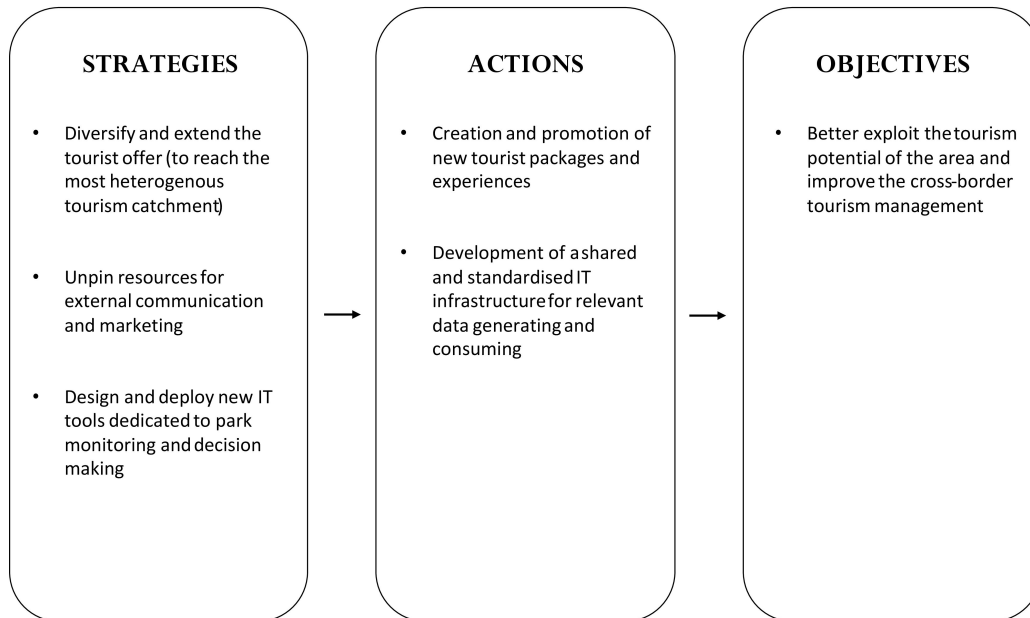


Fig. 2.2: INSUBRI.PARKS project strategies, actions and objective

State-of-Art

” *Study the past if you would define the future.*

— **Confucius**
(Philosopher)

Tourism has closely been connected to the progress of Information and Communication Technologies (ICTs) for over 30 years. Internet expansion in the late 1990s has mutated operational and strategic practices in tourism [14] [15]. The tourism business at first focused on utilising computerised systems to increase efficiency in the processing of private information and managing distribution. Now, the Internet and ICTs are essential on all operative, structural, strategic and marketing levels to promote global cooperation among suppliers, intermediaries and consumers around the world (Buhalis & Law, 2008; [16]). E-tourism grants chances for business expansion in all geographical, marketing and operational senses. As a result of Internet developments, several new tools and methods have come into tourism field for helping the comprehension of user needs and demands [17]. Among them, the Decision Support System (DSS) can be defined as an interactive computer-based information system which is created to support and provide resolutions to decision problems, with an interactive ability to understand the problem through the use of models and data processing [18] [19]. A study by Michael S. Scott Morton (1971) illustrated that managers benefited from applying a computer-based management decision system. The 1970s was a period of conceptual and technology evolution for DSS while, in the mid-1980s, executive information systems became popular in companies and more adopted to the scope of decision support. In the early 1990s, a transformation occurred from mainframe-based DSS to client/server DSS.

DSS and planning are problematic tasks in tourism development since it connects various stakeholders, thus requires tools that support effective decision making to come to terms with the economic, social, and environmental demands. Tourism activity depends heavily on information about the tourism destination such as locations, accommodations, restaurants, routes and attractions [20]. The DSS can be modelled as a Multi-Criteria Decision Making system (MCDM) in which the information mentioned above act as multiple and independent criteria. In the literature, different authors carried out researches on the integration between MCDM and GIS. For example, in [21] a GIS-based Multi-Criteria Analysis (MCDA)

useful to select priority areas for preservation in the Ahaggar National Park in Algeria is proposed. In this paper, the association of GIS and MCDA is seen as a process that transforms and combines geographical data and value decision maker's preferences to obtain information for spatial decision making. Similar to the above, [22] presented a research on a GIS-based Multi-criteria analysis in site selection of water reservoirs near to Malaysia. In this document, there is a significant analysis of the study area, which highlighted the importance of analysing the area deeply to understand different points of views in a better way. In fact, each specific study area has its own set of criteria and alternatives which should be identified in a preliminary phase of MCDM analysis. More recently, [23] provided a description of how GIS and MCDM are strictly related and interconnected with tourism: according to the author, the success of any tourism business is determined by tourism planning, development research and marketing. As it could be assessed from previous researches, GIS tools offer excellent opportunities for the development of modern tourism applications using maps [24].

Unfortunately, in the Italian context, many actors in the tourism market adopt basic Websites to showcase their natural heritage, and only a few leverage WebGIS applications [25]. This is the current situation of the parks involved in the INSUBRI.PARKS project that have not already implemented such technologies allowing proper management and promotion of their products. For examples, the Spina Verde Website [26] offers a traditional list of activities complemented by some PDF documents in which the map of the park is just an image. What comes from this old-way approach is a non-accurate real-time experience because the user is not allowed to navigate geolocated information describing the park. An interactive map encourages users both to use it and to be curious about whatever they can find inside it. The previous considerations shall remain valid for the vast majority of the Italian National Parks: in fact, the backwardness both in ideas and technologies has led to the creation of a consistent gap concerning other European or worldwide case studies. Indeed, some foreign operators provide a complete and interactive Web experience in which users may get in touch with a user-friendly geographic information system. Finland shows how much a well-functioning Web GIS service may be helpful to attract tourists. In particular, it has an all-embracing integrated portal done in collaboration among all the regional parks. It displays pathways, suggested activities, facilities and any other relevant information of all the parks through a dynamic map in which users can track their location in real-time without the risk of being lost in the middle of nowhere [27].

The gap above mentioned is even more significant considering the National Park Service (NPS) Web portal [28]. Since 1916, the NPS has been entrusted with the care of the American national parks. With the help of volunteers and partners, these unique places have been safeguarded and shared with more than 330 million visitors

every year. The IT infrastructure allows to digitally explore each national park by map or by topic of interest. Visitors can easily find the information they need about help centres, events, lodging, places to eat and shop, and services throughout the park. It is an excellent way to find where things are in the park, or what is near to tourists when they are in the park. On the practical level, the NPMMap is a suite of Web map tools built using innovative mapping techniques and technologies. Among those, the NPMMap Builder, Park Tiles, NPMMap.js, and the NPMMap Symbol Library make these places come alive for visitors to the national parks.

- **NPMMap Builder:** NPMMap Builder walks users through the process of building elegant, accessible, and usable maps step-by-step and deploying them to either NPS.gov or a stand-alone Website. The Builder uses the other NPMMap tools listed above, wrapping them in a simple interface designed to be used by National Park Service operators who have no experience writing code or building maps. Maps built using this tool make extensive use of the powerful Leaflet Web mapping library. Maps developed with Builder can be found all across NPS.gov (see <https://www.nps.gov/npmap/tools/npmap-builder>);
- **Park Tiles:** Park Tiles is a suite of online basemaps produced to match the National Park Service's graphic identity. These basemaps can be used as stand-alone reference maps or customised Web maps with additional data overlays. On the one hand, Park Tiles uses data from NPS national enterprise datasets to outline features within park boundaries. On the other hand, outside of park boundaries, Park Tiles relies on OpenStreetMap data to provide additional context (see <https://www.nps.gov/maps/tools/park-tiles/#4/39.03/-95.98>);
- **NPMMap Symbol Library:** The NPMMap Symbol Library is the National Park Service's Web map icon set. The Symbol Library has a clear, distinctive and unique graphic identity based on several specific map symbols. The attention to detail helps to provide a massive number of facilities, services, and attractions offered by national parks (see <https://www.nps.gov/maps/tools/symbol-library>).

According to the above, the INSUBRI.PARKS project comes on the top of this context, having the aim to fill in the existing gap between Italy and the rest of the world.

The next chapter reports details of the proposed strategies to fill this gap by proposing the development of an interactive WebGIS plugin embedding the MCDM analysis on geospatial data.

Multi-Criteria Decision Making

“ *Decision is the ultimate power. Decisions shape destiny.* ”

— **Tony Robbins**
(Philanthropist)

This chapter is organised as follows. First, an overview of the topic, together with a brief literature review, is reported. **Section 4.1** provides a general disclosure of the decision making processes. Finally, **Section 4.1.1** describes the research methodology and the procedure applied in this study.

The Multi-Criteria Decision Making (MCDM) refers to the evaluation of a set of possible alternatives according to multiple, often conflicting, criteria that are adopted for the selection of the most suitable option in a particular condition [29]. The evaluation of the recommended alternatives, which is formulated as a Multi-Criteria group decision-making problems, generally concerns a large number of approaches and techniques introduced by many authors in the literature [30]. During the last decades, MCDM methods have received a great deal of attention from practitioners and researchers. This chapter attempts to document the exponential growth in MCDM methods and models. Many former studies have reviewed MCDM techniques in different field of application. However, few studies have been carried out to review the role of MCDM techniques in applications that combine tourism and GIS [31][32].

MCDM is a way of dealing with complex problems by separating the problems into smaller parts. After considering some problems and making judgements on its sub-components, these pieces are reassembled to present an overall picture of the starting problem [33]. Most of MCDM techniques deal with discrete alternatives, which are specified by a collection of criteria [34]. Criteria values can be defined as a cardinal or ordinal information. Latest MCDM methods allow decision-makers to deal with all types of information, but one of the problems faced during the MCDM process is the choice of the aggregation procedure for determining the final decision problem as results of the analysis of all the available information. Despite that, multiple criteria decision analysts provided with a variety of aggregation procedures [35]:

- **Formal models:** algorithms, procedures and selection paradigms;
- **Evaluation:** theories assumptions about preferences and organised representations of them;
- **Assessment procedures:** elicitation, estimation and scaling of individuals' preferences, utilities and biased probabilities in MCDM situations.

MCDM methods include a wide range of divergent approaches which can be generally classified into two main categories. These are the discrete MCDM and continuous MCDM methods [36][37]. Many publications have been produced to provide information about these MCDM methods, their development and applications into several disciplines. The pillars of modern MCDM were uncovered in the 1950s and 1960s. The growth of MCDM research accelerated straddling the 1980s and 1990s and seemed to have extended its exponential growth. Köksalan, Wallenius, and Zionts [38] presented a brief chronicle of the evolution of MCDM methods. They concisely outlined the development of the topic, along with the history. Keeney, Raiffa, and Rajala (1979) [39] developed the basics of Decision with Multiple Objectives. Hwang and Masud (1979) [40], provided improvements to MCDM methods in a wide range of applications. Later, Tzeng and Huang [41] reexamined the MCDM methods by adding a weighting procedure (SAW), a technique for ranking preferences (TOPSIS), a choice-expressing reality method (ELECTRE), and finally a linear programming approach for multidimensional analysis of preference (LINMAP). Saaty [42], published an exhaustive study on the Analytic Hierarchy Process (AHP), on which is based most of the model used in the current work. Later, he promulgated a study on the additional development of the Analytic Network Process (ANP) method [43].

Recently, the use of hybrid and modular methods is becoming progressively popular. Hybrid and modular methods are founded on well-known traditional methods. So, there is not a unique and well-defined methodology that one could follow step-by-step from the beginning to the end of a decision process. **Section 4.1** will explain better this quote because different parts of different methods have been assembled to achieve a functional hybrid MCDM model for the experience planner plugin of the INSUBRI.PARKS project.

4.1 Decision Making Process

"Decision making is a process of identifying and choosing alternatives based on the values and preferences of the decision-maker" [44]. This is the definition of decision

making given by János Fülöp and can be summarised in the diagram reported in **Figure 4.1**.

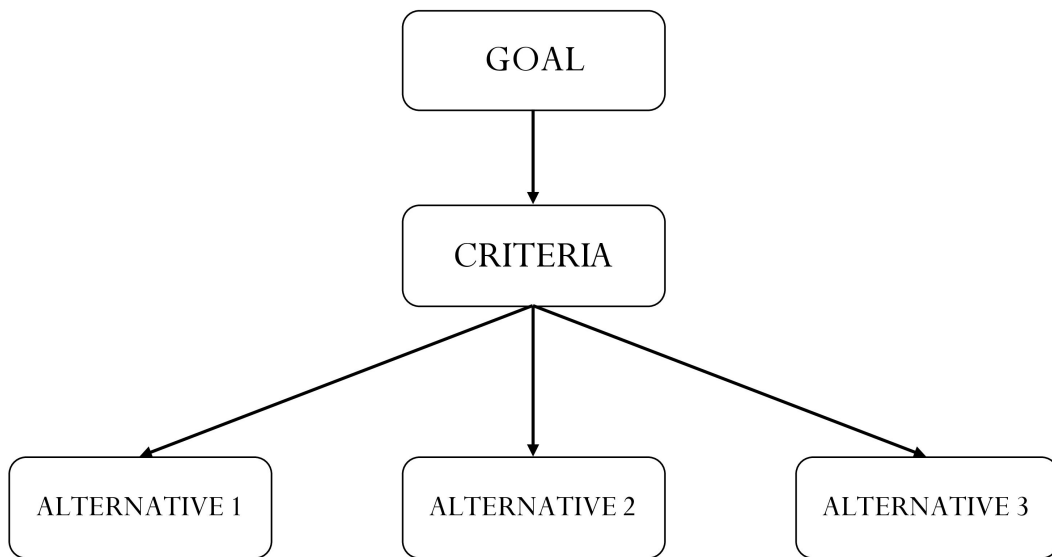


Fig. 4.1: Decision Making block diagram.

Decision-making is applied systematically and logically. The essential steps involved in the decision process are (see **Figure 4.2**):

1. **Determine the decision problem:** Decision-makers must ultimately be aware of the decision problem. It is crucial to identify, understand and set the problem before making a decision. This process allows for identifying the root causes by limiting assumptions;
2. **Identify the criteria:** Identifying and determining criteria which can accurately describe alternatives. A decision problem which contains a large number of criteria is particularly useful to yield better alternatives. An ideal set of criteria should be operational, significant and non-redundant.
3. **Identify alternatives:** A significant part of decision making involves the analysis of a finite set of alternatives. All the possible alternatives are evaluated, and then any alternatives that fail to meet the requirements are eliminated until there remain only one or more alternatives, thus achieving the desired goal;
4. **Assign weights to each criterion:** The weights to the criteria are assigned accordingly to the pairwise comparison matrix;
5. **Score criteria for each alternative:** A matrix is formed by scoring criteria for each alternative, and this matrix is used to the decision rules;

6. **Apply the decision rules:** Based on the input from criteria weights vector and scores of criteria from each alternative decision rules must be applied to determine the potential and suitable alternative;
7. **Evaluate alternatives:** After the evaluations, the decision-making tool can be utilised to rank the alternatives or allowing choosing a more promising alternative from a set of defined alternatives;
8. **Identify the most suitable alternative:** The goal is achieved when a suitable alternative is identified.

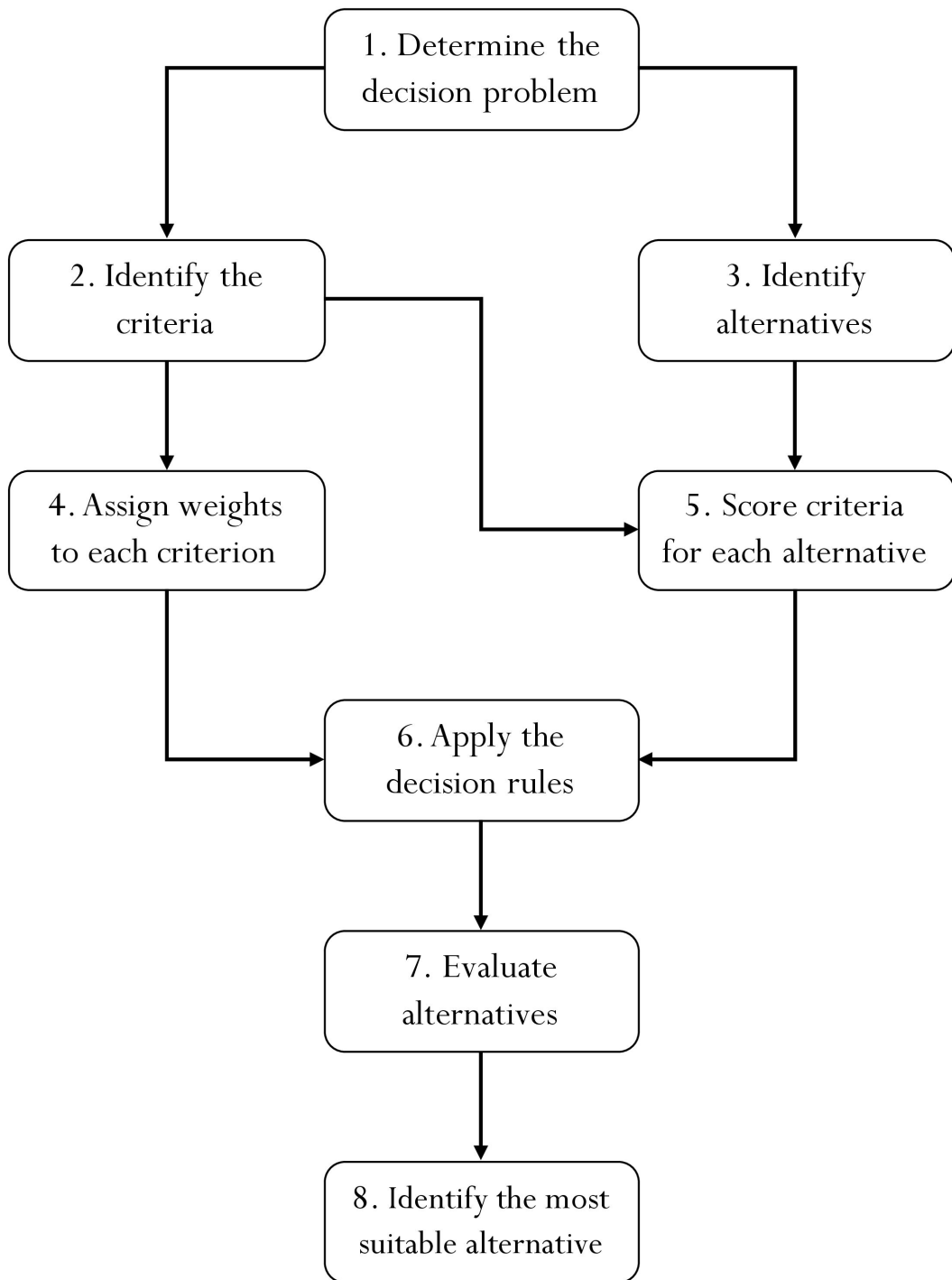


Fig. 4.2: Step by Step Decision Making Process.

4.1.1 Decision Making Models

Depending on the decision problem, decision making is divided into two types of single criterion and Multi-Criteria decision making. In the context of this work, the decision problem is applied to the Tourist Experience Planner tool. It aims to find a list of *most suitable* paths within a Park. Each path has a score depending on both the proximity to tourist attractions and the user's interests. Therefore, the Tourist Experience Planner tool allows users to express his interest among many different touristic categories (e.g. History or Religious points of interest). Consequently, in the field of Decision Making Models, the analysis is focused on the Multi-Criteria Decision Making since there are multiple criteria to consider.

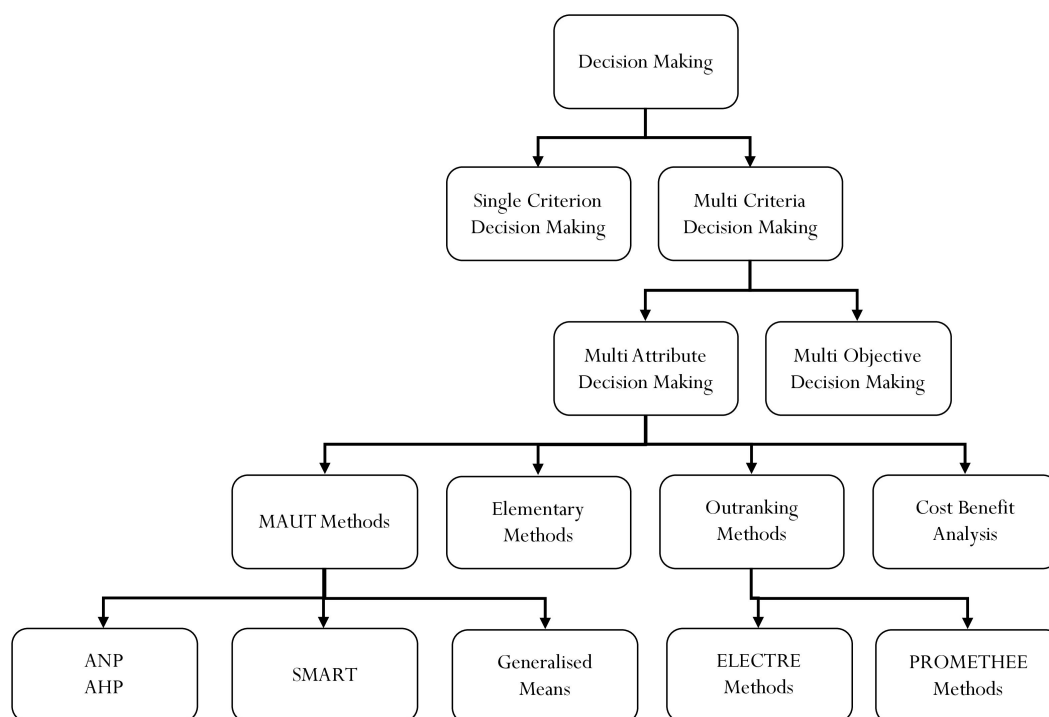


Fig. 4.3: Classification of Decision Making Models.

Then, according to the type of criteria, MCDM is classified into two types (see Figure 4.3), namely Multi-Attribute Decision Making (MADM) and Multi-Objective Decision Making (MODM). The subtle but fundamental difference between MADM and MODM lies in the approach to the problem. Based on the assumption that both models are applied to the same problem (they have the same criteria and alternatives), these are the two possible approaches:

- **MADM:** first of all, the criteria are weighted according to the user preferences, then the alternatives are weighted based on the criteria weights. Finally, the output is the rank of *the best alternatives*;

- **MODM:** the first step, as before, consists of weighting the criteria. Then, the optimal weight for an unknown alternative is computed according to the criteria weights. In the end, the process returns the alternative that comes closest to the optimal weighted computed in the previous step.

Due to this, the MADM model lends itself mostly to the requirements of the IN-SUBRI.PARKS Tourist Experience Planner tool.

The following section presents a detailed analysis of some methods belonging to the *chosen* model: the MADM.

4.2 MADM Model

After providing a background on the Decision-Making process and general models, it is possible now to go into details on Multi-Attribute Decision Making model. As the word itself suggests, the MADM concerns the structuring and solving decision problems which involve multiple criteria. The primary purpose is to support decision-makers facing such problems. There does not exist a unique solution for such problems, and it should be necessary to differentiate between solutions. MADM problems can be solved according to three different goals:

- Choosing the "best" alternative from a set of available alternatives;
- Choosing a set of valid alternatives;
- Choosing all alternatives which respect specific requirements.

In this work, the selected way to give the solution corresponds with choosing a small set of reasonable alternatives.

As previously evidenced in the **Figure 4.3**, several MADM methods have been created. Based on the literature reviewed, the noticed advantages and disadvantages, as well as fields of application for each method, are summarized in **Table 4.1**. Most of them have seen a typical pattern of evolution, such as the transition from MAVT to MAUT or from AHP to ANP. Outranking methods, like ELECTRE and PROMETHEE, which were accepted early on the development of the MCDA, were overtaken by the use of value-measurement approaches such as AHP, ANP, and MAUT. In recent years, because of the progress of technology, combining various methods has become common in Multi-Criteria Decision Analysis [45].

Method	Advantages	Disadvantages	Fields of Applications
Multi-Attribute Utility Theory (MAUT)	Takes uncertainty into account; can incorporate preferences.	Needs a lot of input; preferences need to be precise.	Economics, finance, actuarial, water management, energy management, agriculture
Analytic Hierarchy Process (AHP)	Easy to use; scalable; hierarchy structure can easily adjust to fit many sized problems; not data intensive.	Problems due to interdependence between criteria and alternatives; can lead to inconsistencies between judgment and ranking criteria; rank reversal.	Performance-type problems, resource management, corporate policy and strategy, public policy, political strategy, and planning.
Data Envelopment Analysis (DEA)	Capable of handling multiple inputs and outputs; efficiency can be analyzed and quantified.	Does not deal with imprecise data; assumes that all input and output are exactly known.	Economics, medicine, utilities, road safety, agriculture, retail, and business problems.
Fuzzy Set Theory	Allows for imprecise input; takes into account insufficient information.	Difficult to develop; can require numerous simulations before use.	Engineering, economics, environmental, social, medical, and management.
Simple Multi-Attribute Rating Technique (SMART)	Simple; allows for any type of weight assignment technique; less effort by decision makers.	Procedure may not be convenient considering the framework.	Environmental, construction, transportation and logistics, military, manufacturing and assembly problems.
PROMETHEE	Easy to use; does not require assumption that criteria are proportionate.	Does not provide a clear method by which to assign weights.	Environmental, hydrology, water management, business and finance, chemistry, logistics and transportation, manufacturing and assembly, energy, agriculture.
ELECTRE	Takes uncertainty and vagueness into account.	Its process and outcome can be difficult to explain in layman's terms; outranking causes the strengths and weaknesses of the alternatives to not be directly identified.	Energy, economics, environmental, water management, and transportation problems.

Tab. 4.1: Advantages and Disadvantages of MADM Methods [46].

However, as regards the model, this work proposes a revisitation of the Analytical Hierarchy Process (AHP), which is the MADM well-known technique developed in the 1970s by Thomas Saaty.

4.2.1 AHP-Revisitation for the INSUBRI.PARKS

As stated in **Chapter 1**, the INSUBRI.PARKS Tourist Experience Planner required a specific and "not-common" decision model. Going into the details, this model follows the standard AHP method for the majority of the steps [47].

Step 1: Developing the Model

In this section, it is important to have clear in mind the definition of the following entities:

- **Paths Network:** it is the set of all the roads that are labelled as *path* inside the park;
- **Path:** it is a specific and numbered path inside the park;
- **Edge:** it is a part of a *path*; it has an *entry* and an *exit* point. Every time a *path* intersects another path, a new edge is formed;

- **Suggested Path:** this is a *real-time computed path*; it is built by many *edges* belonging to some different *paths*;
- **Path Gate:** it is the path access point or exit point.

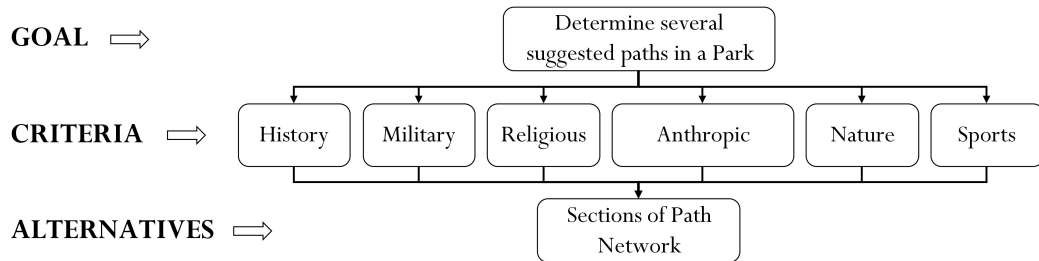


Fig. 4.4: Tourist Experience Planner model.

The model used in the Tourist Experience Planner is represented in **Figure 4.4**, and it can be summed up as follow:

- The **goal** is to find a list of suggested paths which are built using several edges of different predefined paths;
- The **set** of criteria are defined according to the different categories in which the touristic points are split;
- The **alternatives** are all the edges that are part of the whole paths network of the park.

Step 2: Derive weights for each criterion

The approach can be used to help decision-makers for prioritising alternatives and determining the optimal alternative using pairwise comparison judgments [48][49]. The scale used for the pairwise comparisons is outlined in **Table 4.2** [50].

Since the users provide an interest in the criteria from 1 to 5, it was necessary to add an extra-step before using **Table 4.2**. A value from 1 to 9 is assigned based on a "forced-comparison" between two expressed preferences.

Example: if the criteria *history* has an interest of 5 and the criteria *religion* has an interest of 1, the ratio between *history* and *religion* is 5/1. It means that, following the fundamental scale table, *history* has "Extreme importance" concerning *religion* and so, the value for *history* compared with *religion* is 9 (and, of course, the value of *religion* compared with *history* is 1/9).

Intensity of Importance	Definition	Explanation
1	Equal Importance	User has the same preference for the two categories
3	Moderate importance of one over another	User slightly prefers one category over another
5	Essential importance	User prefers one category over another
7	Very strong importance	User significantly prefers one category over another
9	Extreme importance	The evidence favouring one category over another is of the highest possible order.
2,4,6,8	Intermediate values between two adjacent judgments	When a compromise is needed

Tab. 4.2: The Fundamental Scale by Saaty

Let's assume that:

a_{ij} = numeric value coming from the comparison between criteria i and j

n = number of criteria.

The results of all the comparison ($\frac{n(n-1)}{2}$) will produce the $A_{n \times n}$ matrix which will be used to generate the vector with the weight of each criterion.

The value a_{ij} of matrix A follows the following properties:

- if $a_{ij} = a$, then $a_{ji} = \frac{1}{a}$ with $a > 0$;
- if A_i has same importance of A_j , then $a_{ij} = a_{ji} = 1$. In practice, each value of the main diagonal of A is equal to 1

where the i index represents the rows of the pairwise comparison matrix, while the j the columns.

In the following example, for making it as clear as possible, only three criteria are used as a sample: *history, religion, sport*. The A matrix (see **(Figure 4.5)**) is built in the triangle area above the main diagonal, and then the below triangle is simply its reciprocal.

Once the A matrix is built, it is possible to derive from A the criteria weight vector w (n -dimensional column vector):

		CRITERIA		
		<u>HISTORY</u>	<u>RELIGION</u>	<u>SPORT</u>
CRITERIA	<u>HISTORY</u>	1	a	b
	<u>RELIGION</u>	1/a	1	c
	<u>SPORT</u>	1/b	1/c	1

Fig. 4.5: Pairwise comparison A matrix.

$$w_i = \sqrt[n]{\prod_{k=1}^n a_{ik}}$$

Finally, the criteria weight vector w is normalised to obtain a vector \bar{w}_i in which the sum of the weights is equal to 1:

$$\bar{w}_i = \frac{w_i}{\sum_{k=1}^n w_k}$$

Step 3: Consistency Check

Once the vector of weights has been determined, it is essential to understand whether the pairwise comparison matrix is consistent or not, which means to "measure" the subjective judgments of the decision-maker at each comparison [51]. To better understand, suppose there are three people (A , B and C) from different height and suppose that, following a series of pairwise comparisons, they can be sorted by decreasing height as follows: $A > B > C$. It is evident that A is higher than B and B is higher than C , so it is entirely consistent to say (for the transitive property) that A is higher than C .

Unfortunately, sometimes it is not possible to obtain a consistent comparison as before. This is why it is necessary to determine consistency indices and tolerance values to be able to manage and consider even less immediate and straightforward cases. To make an example of inconsistent comparison, assume that there are three teams of football who compete in a tournament and want to determine which is the strongest formation. Now, team A beat team B (A is stronger than B) and team B has beat team C (B is stronger than C). This does not necessarily imply that team A can beat team C , thus creating inconsistency. So, it is important to define the indices used to determine the consistency of a matrix and above all the tolerance thresholds adopted in the literature to determine whether a pairwise comparison matrix may be well formulated or not.

The first-required index is the Consistency Index (*CI*), a value obtained by the following formula:

$$CI = \frac{\lambda - n}{n - 1}$$

where λ is the maximum eigenvalue of the matrix *A* and *n* represents the dimension of the matrix itself.

The second-required index is the Random Consistency Index (*RCI*) (see **Table 4.3**) where the dimension *n* of matrix *A* is associated with the relative value of *RCI*:

RCI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49
n	1	2	3	4	5	6	7	8	9	10

Tab. 4.3: Random Consistency Index

Finally, the Consistency Ratio (*CR*):

$$CR = \frac{CI}{RCI}$$

In the less restrictive case, in order to consider the matrix *A* consistent, the threshold is placed at 0,1 so *CR* must be < 0.1 (10%). To determine more precisely and according to the size of *A* the threshold under which to consider the matrix consistent, it is possible to use the values in the **Table 4.4**:

Threshold	n
0.05	3
0.08	4
0.10	≥ 5

Tab. 4.4: Thresholds values

As can be understood by analysing the *RCI* values as a function of the size of the matrix (see **Table 4.3**), it is useless to define one consistency threshold in the case of pairwise comparison matrix of size $n = 1$ and $n = 2$. In the first case, the result is

a 1×1 identity matrix, while in the second case, comparing only two criteria, it is not possible to obtain inconsistency.

Step 4: Weighting the Edges of Path Network

Once the consistency test is done, the standard AHP method expects to multiply the criteria weight vector by the score assigned to all the criteria of each alternative [52]. In the INSUBRI.PARKS Tourist Experience Planner, the alternatives are all the edges in the path network. So, they are entered in a P_{ec} matrix in which each row is an edge (e), and each column is a criterion (c) score (see **Table 4.5**). This score is assigned according to the number of touristic points close enough to the corresponding edge, divided into categories.

Edge ID	History	Religion	Sport
1	3	8	1
2	4	2	6
3	1	1	9

Tab. 4.5: Example of P_{ec} matrix

At this point, the AHP method resumes and compute the edge weight vector S (m -dimensional column vector where m is the number of edge alternatives) as follow:

$$S_i = \sum_{k=1}^m p_{ik} \times w_i$$

In this way, as shown below, the criterion-score of each edge of the path network in P_{ec} is multiplied by the weight of the corresponding criterion in the criteria weight vector w :

$$\begin{matrix} P_{ec} \\ \begin{pmatrix} 3 & 8 & 1 \\ 4 & 2 & 6 \\ 1 & 1 & 9 \end{pmatrix} \end{matrix} \times \begin{matrix} w \\ \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \end{matrix}$$

Step 5: Graph Algorithm

The big difference between the standard AHP and this revisitation is in the final ranking of the alternatives. In the standard case, the method is already able to

return the result of the MCDM, but the revisited method not yet. Since the requested output must be a rank of suggested paths (they should match as much as possible with the user preferences), a graph algorithm for finding k-shortest paths is applied to the paths network of the park. This topic is addressed in the next chapter.

Step 6: Sort and rank the suggested paths

At this point, there is a list of suggested paths which are in turn built by many edges. The weight of each edge (computed at the end of Step 4) in each suggested path is added up to obtain a final score for all the suggested paths.

Finally, the goal of the model is achieved, and a sorted rank of suggested paths is returned.

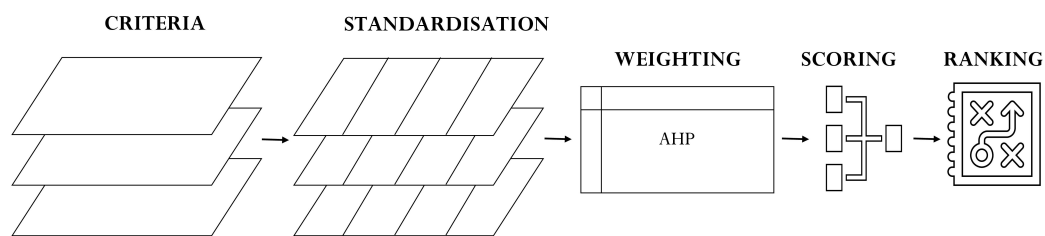


Fig. 4.6: The structure of INSUBRI.PARKS MCDM model.

Path Algorithms

” *You must be interested in finding the best way,
not in having your own way.*

— **John Wooden**
(Coach)

As mentioned in the previous chapter, once each section of the whole paths network has its weight, a graph algorithm to find a list of available routes between two points is executed. As can be seen from **Figure 5.1**, the paths network of Spina Verde Park can be interpreted, from a combinatorial geometry perspective, as an undirected graph where:

- **Nodes:** represent gates/intersection between two or more edges of paths;
- **Edges:** represent edges of paths that connect gates and intersection points.

This chapter, at first, aims to give a theoretical background about graph theory. The second section proposes a round-up of the shortest path algorithms which will help to understand how they work and what kind of reasoning is behind them. Then, there is a brief but essential discussion about the motivation behind the choice of a specific algorithm among an ocean of alternatives. Finally, the implementation of the algorithm mentioned above is approached and discussed in detail.

5.1 Background in Graph Theory

In mathematics and computer science, graph theory studies the combinatorial objects called graphs. They are mathematical structures which model pairwise relations between objects [53]. Conceptually, a graph is formed of vertices (also called nodes) and lines (the so-called edges) that connect them (see **Figure 5.2**).

Formally, a graph is a pair of sets (V, E) , where V is the set of vertices and E is the set of edges, formed by pairs of vertices. In E , every element can occur more than

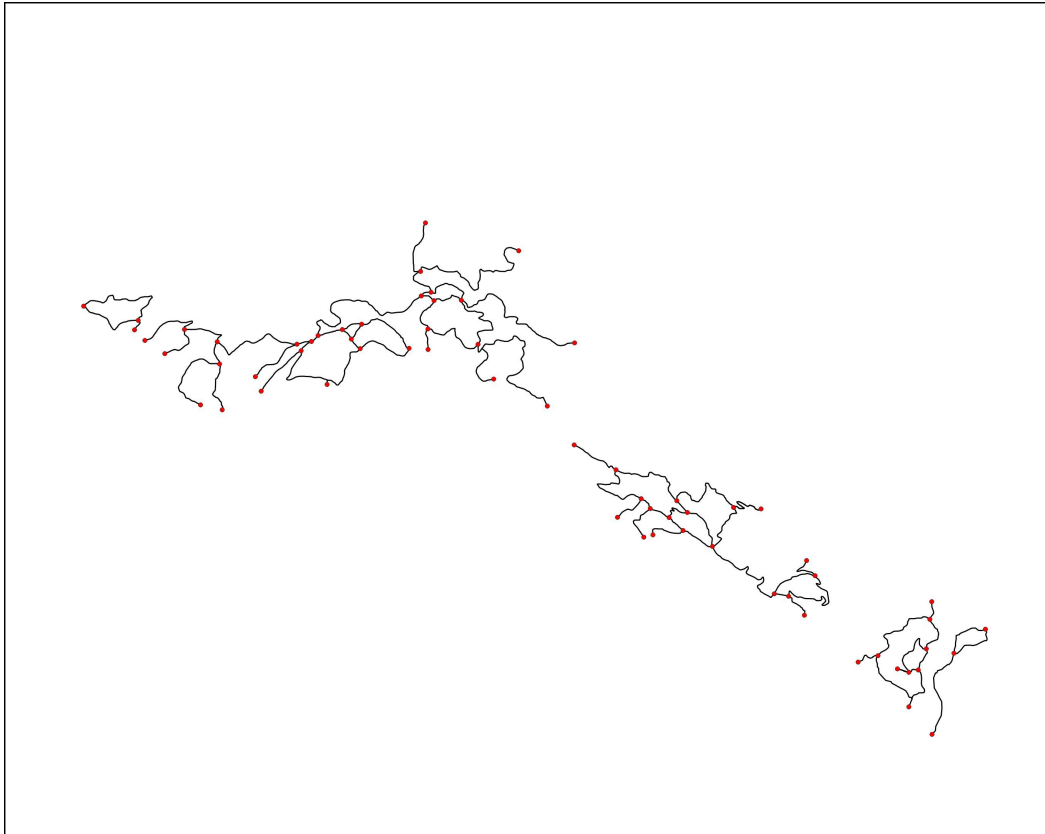


Fig. 5.1: Spina Verde Park paths network as an undirected graph.

once so that each one has its multiplicity. Usually, the labels of the vertices and edges are with letters (e.g. v_1 , e_2) (see **Figure 5.3**).

About the next sections of this chapter, it is very important to have clear in mind the following terminologies [54]:

- The two vertices a and b are end-vertices of the edge (a, b) ;
- Edges that have equal end-vertices are parallel;
- An edge of the form (a, a) is a loop;
- A graph is simple if it has no parallel edges or loops;
- A graph with no edges is the so-called empty graph;
- Edges are adjacent if they share a common end-vertex;
- A graph where edges point in a direction is called directed, otherwise, it is undirected;

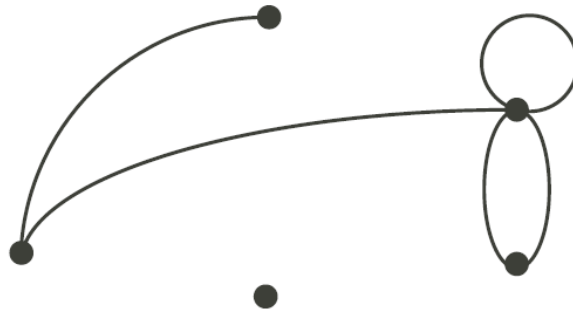


Fig. 5.2: Simple graph example.

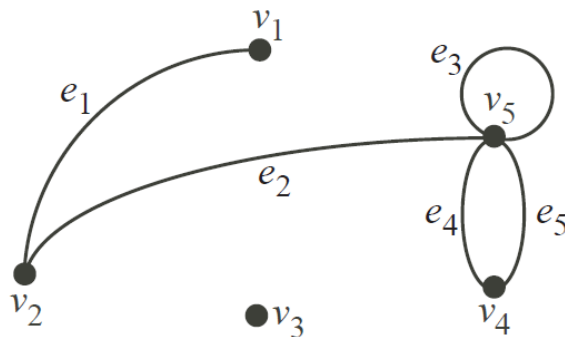


Fig. 5.3: Simple graph with labels.

- Two vertices a and b are adjacent if an edge connects them ((a, b) is an edge);
- The degree of the vertex b , written as $d(b)$, is the number of edges with b as an end-vertex;
- A pendant-vertex is a vertex whose degree is 1;
- An edge that has a pendant vertex as an end vertex is a pendant edge;
- An isolated vertex is a vertex whose degree is 0.

According to this and coming back to **Figure 5.1**, it is possible to define some properties of the Spina Verde paths network:

- It is an aggregation of four different and independent graphs;

- Each standalone graph is simple and undirected;
- The maximum degree of a vertex is (check the number);
- Each standalone graph has no pendant vertex or edges.

In this regard, it is worth emphasizing the difference among walks, trails and paths.

5.1.1 Walk, Trail and Path

Walk

A walk in the graph $G = (V, E)$ is a finite sequence of alternating vertices and edges of G . The walk starts at a vertex. It is allowed to visit a vertex or go through and edge more than once.

Example: In the graph:

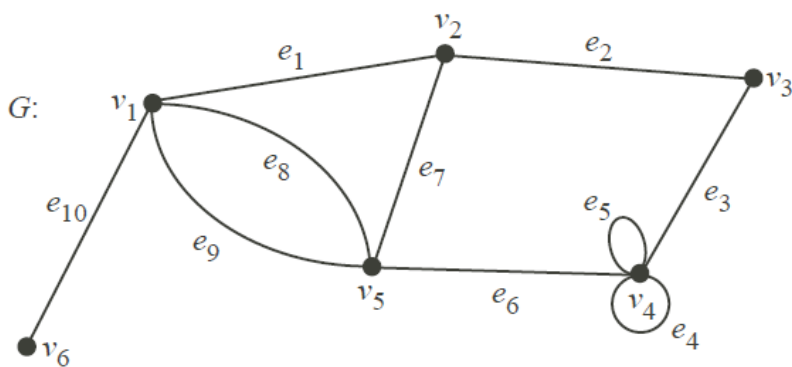


Fig. 5.4: Graph used to define walks, trails and paths.

The sequence:

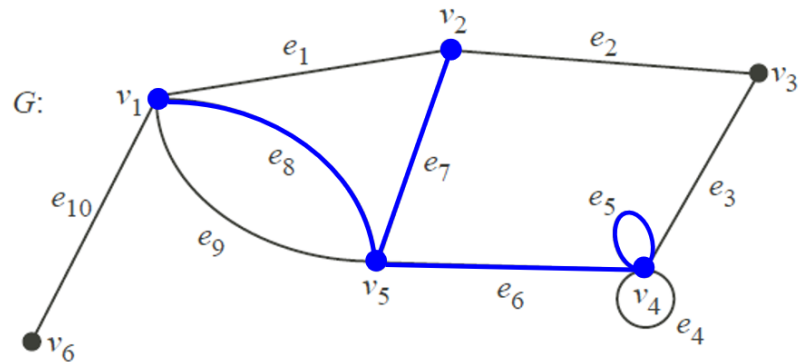
$$v_2, e_7, v_5, e_8, v_1, e_8, v_5, e_6, v_4, e_5, v_4, e_5, v_4$$


Fig. 5.5: Walk example

is a walk.

Trail

A walk is a trail if any edge is traversed at most once.

Example (Continuing from the previous example) The walk:

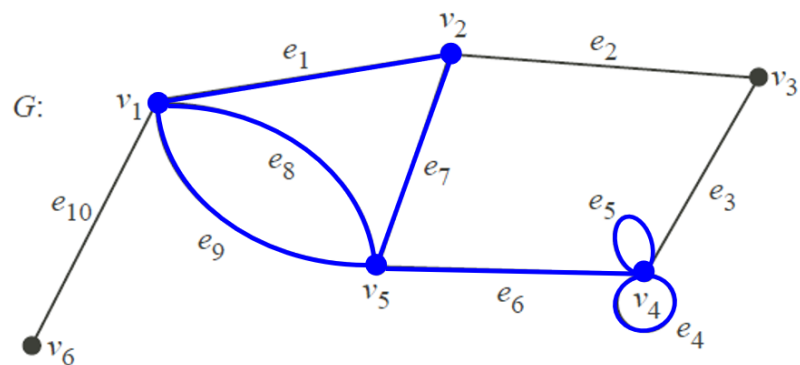
$$v_1, e_8, v_5, e_9, v_1, e_1, v_2, e_7, v_5, e_6, v_4, e_5, v_4, e_4, v_4$$


Fig. 5.6: Trail example

is a trail.

Path

A trail is a path if any vertex is visited at most once except possibly the initial and terminal vertices when they are the same (if it happens, the closed path is a circuit).

Example (Continuing from the previous example) The trail:

$v_2, e_7, v_5, e_6, v_4, e_3, v_3$

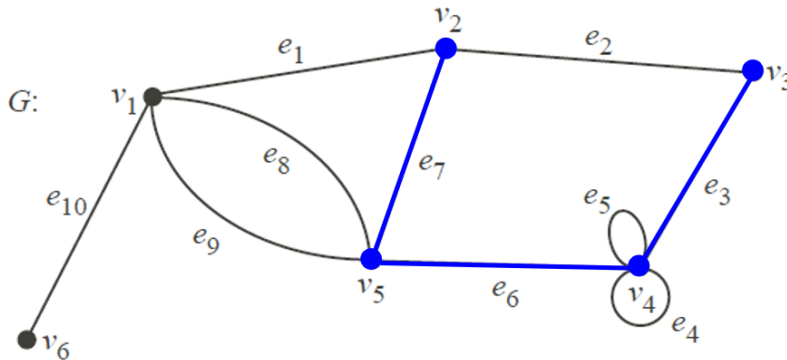


Fig. 5.7: Path example

is a path.

As stated in the previous chapter, a crucial phase of the AHP method is to find a list of shortest paths between two points in order to proceed with the scoring of all the edges part of the path. For this reason, in the next section, the focus shifts on the algorithms employed in finding the shortest path between two vertices in a graph.

5.2 Shortest Path Algorithms

The so-called shortest path problem is a problem of finding the shortest path or route from a starting point to a final destination. Commonly, in order to represent the shortest path problem, graphs are the most suitable solution [55]. On the edges of a graph, it is possible to "walk" by going from one vertex to other vertices. Besides, lengths of edges are often called weights, and the weights are usually used for computing the shortest path from two points. There exist many kinds of algorithms applied in the shortest path problem. However, only several of them are going to be discussed in this section because of their affinity with the algorithm used in the INSUBRI.PARKS experience planner, namely:

- Dijkstra's Algorithm

- Floyd-Warshall Algorithm
- Bellman-Ford Algorithm

In the algorithms mentioned above, the very first step is to represent the graph as an adjacency matrix A . The dimension of A is equal to $n \times n$, where n is the number of vertices in the graph. The element of matrix $a[i][j]$ is classified as an edge that connects the i^{th} and j^{th} vertices; the value here represents the weight of the corresponding edge. Nevertheless, if between vertices i and j do not exist any edge, the value in $a[i][j]$ will be equal to infinity. Another common representation useful in this context is an array of edges. If m is the number of edges in a graph, then an $m \times 3$ two-dimensional arrays can be used; in each row, the starting vertex, the ending vertex, and the edge that connects them are stored. The advantage of using an array of edges in comparison to the adjacency matrix is visible when there is more than one edge that connects two vertices. In fact, in this situation, it is not possible to use the adjacency matrix in order to represent the graph.

Dijkstra's Algorithm

For each vertex within a graph, the algorithm assigns a label $d[]$ which determines the minimal length from the starting-points to the other vertices v of the graph. The algorithm works sequentially, and in each step, it decreases the value of the label of the vertices. When all the vertices have been visited, the algorithm stops [56]. If on the one hand, the label of the starting point s is zero $d[s] = 0$; on the other hand, labels in other vertices v are equal to infinity ($d[v] = \infty$). It means that, at the very early stage, the length from the starting point s to the other vertices is unknown. Moreover, each vertex v has an attribute concerning the fact that it was visited or not yet (a boolean value $u[v] = True/False$). Of course, when the algorithm starts, all vertices are assigned as unvisited.

Once all the vertices have been visited, the Dijkstra's algorithm finishes; otherwise, from the list of unvisited vertices, it chooses the vertex which has the smallest value at its label (which is the smallest distance) considered neighbours of the starting-point. For each unvisited neighbour, the length changes into the sum of the label's value at the initial vertex v and the length of the edge which connects them. If the resulting value is less than the value at the label, that value changes into the new obtained value: $d[neighbors] = \min(d[neighbors], d[v] + l)$.

After considering all of the neighbours, the initial vertex is labelled as visited. After repeating this step n times, all vertices of the graph will be visited, and the algorithm ends. The vertices not-connected with the starting point remain assigned to infinity.

Algorithm 1 Dijkstra's Algorithm

```
1: procedure Dijkstra( $v$ )
2:    $d[i] = inf$  for each vertex  $i$ 
3:    $d[v] = 0$ 
4:    $s = new\ empty\ set$ 
5:   while  $s.size() < n$  do
6:      $x = inf$ 
7:      $u = -1$ 
8:     for  $i$  in  $V - s$  do ▷  $V$  is the set of vertices
9:       if  $x \geq d[i]$  then
10:         $x = d[i]$ 
11:         $u = i$ 
12:    insert  $u$  into  $s$  ▷ The process from now is called Relaxing
13:    for  $i$  in  $adj[u]$  do
14:       $d[i] = \min(d[i], d[u], w(u, i))$ 
```

Floyd-Warshall Algorithm

Consider the graph G , where vertices are numbered from 1 to n . The notation d_{ijk} indicates the shortest path from i to j , which also passes through vertex k . If there is an edge between vertices i and j , it will be equal to d_{ij0} ; otherwise, it can be labelled as infinity. However, for other values of d_{ijk} there are two choices: If the shortest path from i to j does not pass through the vertex k then the value of d_{ijk} is equal to $d_{ij0} - 1$; If the shortest path from i to j passes through the vertex k , it means that it goes first from i to k , and then from k to j . In this situation, the value of d_{ijk} will be equal to $d_{ikk} - 1 + d_{kjk} - 1$. The shortest path will be the minimum of the following two statements:

- $d_{ij0} =$ the length of the edge between vertices i and j ;
- $d_{ijk} = \min(d_{ij0} - 1, d_{ikk} - 1 + d_{kjk} - 1)$

Algorithm 2 Floyd-Warshall's Algorithm

```
1: procedure Floyd – Warshal()
2:    $d[v][u] = inf$  for each pair  $(v, u)$ 
3:    $d[v][v] = 0$  for each vertex  $v$ 
4:   for  $k = 1$  to  $n$  do
5:     for  $i = 1$  to  $n$  do
6:       for  $j = 1$  to  $n$  do
7:          $d[i][j] = \min(d[i][j], d[i][k] + d[k][j])$ 
```

Bellman-Ford Algorithm

In contrast to Dijkstra's algorithm, the Bellman-Ford one accepts the edges with negative weights. That is why a graph can contain cycles of negative weights, which will generate numerous paths from the starting to the ending point, where each cycle will minimize the length of the shortest path. The array $d[]$ stores the minimal

length from the starting point s to other vertices. The algorithm consists of several cycles, where each phase minimize the value of all edges by replacing $d[b]$ to the following statement: $d[a] + c$. In this expression, a and b are vertices of the graph, while c is the corresponding edge which connects them. In order to calculate the length of all the shortest paths, the algorithm requires $(n - 1)$ phases. For those unreachable vertices, the value of remain assigned to infinity.

Algorithm 3 Bellman-Ford's Algorithm

```

1: procedure Bellman – Ford(int v)
2:    $d[i] = inf$  for each vertex  $i$ 
3:    $d[v] = 0$ 
4:   for step = 1 to  $n$  do
5:     for all edges  $e$  do
6:        $i = e.first$ 
7:        $j = e.second$ 
8:        $w = e.weight$ 
9:       if  $d[j] > d[i] + w$  then
10:        if step ==  $n$  then
11:          return "Negative cycle found"
12:         $d[j] = d[i] + w$ 

```

In the specific case of this research, the INSUBRI.PARKS project needs a shortest path algorithm, but it is not enough because, having a single shortest path as a result, does not fit with the idea behind the Tourist Experience Planner. In fact, once the user expresses his preferences and criteria, the system is supposed to return multiple suggested paths and not a unique one. For this reason, in the following section, it is introduced the more specific problem of computing the K shortest paths connecting a given source-destination pair in a graph [57].

5.3 K-Shortest Paths

The K shortest paths (KSP) algorithm has a long history and many applications to a diverse range of problems. This is primarily attributable to the two reasons cited by Eppstein [58]: the K best solutions to an optimization problem might be found by calculating the K shortest paths between two given vertices. In this case, the approach is to look for the first path in the ranking of the shortest paths that satisfies a set of constraints or preferences which may be challenging to specify formally or arduous to optimize [59][60]. Moreover, finding many different shortest paths allows a sensitivity analysis of the optimal solution for many problem parameters [61][62].

The K shortest paths problem was examined as early as 1959 by Hoffman and Pavley [63]. Numerous algorithms have been proposed for many variations of the problem. An extensive classification of algorithms for shortest paths problems was provided

by Deo and Pang [64], while complete online literature for the K shortest paths problem is available from Eppstein [65]. Several K shortest paths problems impose restrictions in the definition of a path. For instance, the specific version of the algorithm used for this work, only considers the shortest paths starting and ending with a vertex defined as an entry or exit gate of the Park. Others restrictions are that the paths should be simple, and they are not allowed to have loops.

The restriction that the paths are simple, which means no vertex can be repeated, makes the K shortest paths problem considerably harder. This problem was investigated, among others, by Clarke [66], Yen [67][68], Lawler [69], Fox [70], Shier [71], Katoh et al. [72], and Perko [73]. The principal weakness of the method proposed by Clarke is the huge number of intermediate paths which must be stored before concluding. On the contrary, Yen's algorithm requires a generally smaller number of intermediate paths, has an $O(Kn^3)$ time-bound, and it works both on directed and undirected graphs.

Due to its simplicity and efficiency in a wide range of situations, the Yen's algorithm has been used to solve the KSP problem in the experience planner plugin of the INSUBRI.PARKS project.

5.3.1 Yen's algorithm

As stated before, the KSP algorithm used in this work is completely based on Yen's algorithm. It computes single-source K shortest paths in a graph with non-negative edge cost. The algorithm dates back to 1971 and employs, more or less, any other shortest path algorithms. Before starting with the description of the algorithm, there are some necessary notations [74]:

- N : the size of the graph, i.e., the number of nodes in the network;
- (i) : the i^{th} node of the graph, where i ranges from 1 to N . This means that 1 is the source node of the graph and N is the sink node of the graph;
- d_{ij} : the cost of the edge between (i) and (j) , assuming that $(i) \neq (j)$ and $d_{ij} \geq 0$;
- A^k : the k^{th} shortest path from (1) to (N) , where k ranges from 1 to K . Then $A^k = (1) - (2^k) - (3^k) - \dots - (Q_k^k) - (N)$ where (2^k) is the 2nd node of the k^{th} shortest path and (3^k) is the 3rd node of the k^{th} shortest path, and so on;

- A_i^k : a deviation path from A^{k-1} at node (i^k) , where i ranges from 1 to Q_k . Note that the maximum value of i is Q_k , which is the node just before the sink in the k shortest path. This means that the deviation path cannot stray from the $k - 1$ shortest path at the sink. The paths A^k and A^{k-1} follow the same path until the i th node, then $(i)^k - (i + 1)^k$ edge is different from any path in A^j , where j ranges from 1 to $k - 1$;
- R_i^k : the root path of A_i^k that follows that A^{k-1} until the i th node of A^{k-1} ;
- S_i^k : the spur path of A_i^k that starts at the i th node of A_i^k and ends at the sink.

The algorithm can be broken down into two different parts [75]:

- Determining the first k shortest path A_k ;
- Determining all the other k shortest paths which will be stored in the container B .

Regarding the first step, to determine the shortest path from the source to the sink A^1 , every efficient shortest path algorithm explained in the previous section can be used. In the Yen's algorithm, the chosen one is Dijkstra because, in case there are no negative edges, it is the best in terms of computational efficiency (this perfectly fits with the paths network seen as a graph). Looking for A^k , where k at this point ranges from 2 to k , the algorithm assumes that all paths from A^1 to A^{k-1} have previously been found. The k iteration can be split into two processes, the first one for finding all the deviations A_i^k and the second one for choosing a minimum length path to become A^k . At this iteration, i ranges from 1 to Q_k^k . The first process can be again subdivided into three operations:

- Choose the R_i^k ;
- Find S_i^k ;
- Add A_i^k to container B .

The root path, R_i^k , is chosen by finding the subpath in A^{k-1} that follows the first i nodes of A^j , where j ranges from 1 to $k - 1$. Then, if at least one path is found, the cost of edge $d_{i(i+1)}$ of A^j is fixed to infinity. Next, the spur path, S_i^k , is found by computing the shortest path from the spur node i , to the sink. By removing the previously used edges from (i) to $(i + 1)$, the spur path is different. $A_i^k = R_i^k + S_i^k$, the addition of the root path and the spur path, is added to B . Next, the edges that

were removed (their cost was set to infinity) are restored to their initial values. The second process identifies a suitable path for A^k by finding the lowest-cost path in container B . This path is removed from its container and inserted into container A . Then, the algorithm proceeds to the next iteration.

Before showing the pseudo-code of this algorithm, a small but crucial topic must be faced: the time complexity. Yen's algorithm time complexity depends mainly on the shortest path algorithm used in the computation of the spur paths. So, based on Dijkstra's algorithm, the complexity is $O(N^2)$. Yen's algorithm makes Kl calls to Dijkstra, where l is the length of the spur paths, so the expected value of l is $O(\log N)$. At the end, the complexity becomes $O(KN(M + N \log N))$.

Algorithm 4 Yen's KSP algorithm

```

1: procedure KSP(Graph, source, sink, K)
2:    $A[0] = \text{Dijkstra}(\text{Graph}, \text{source}, \text{sink})$ 
3:    $B = []$ 

4:   for k from = 1 to  $K$  do
5:     for i from = 0 to  $\text{size}(A[k - 1]) - 2$  do
6:       spurNode =  $A[k - 1].\text{node}(i)$ 
7:       rootPath =  $A[k - 1].\text{nodes}(0, i)$ 

8:       for path  $p$  in  $A$  do
9:         if rootPath ==  $p.\text{nodes}(0, i)$  then
10:          remove  $p.\text{edge}(i, i + 1)$  from Graph

11:          for node rootPathNode in rootPath except spurNode do
12:            remove rootPathNode from Graph

13:          spurPath =  $\text{Dijkstra}(\text{Graph}, \text{spurNode}, \text{sink})$ 

14:          totalPath = rootPath + spurPath

15:          if totalPath not in B then
16:             $B.\text{append}(\text{totalPath})$ 

17:          restore edges to Graph
18:          restore nodes in rootPath to Graph

19:          if B is empty then
20:            break
21:           $B.\text{sort}()$ 
22:   return A

```

5.3.2 The integration between AHP and KSP

To better clarify how the KSP is integrated with the Multi-Criteria Decision System, each component of the Tourist Experience Planner tool of the INSUBRI.PARKS project is acting a specific role as shown in **Figure 5.8**. Accordingly to this, it is possible to synthesize the whole process: once the KSP algorithm found all the k shortest paths, each edge belonging to these shortest paths is mapped with its corresponding weight (which comes out from the Multi-Criteria Decision Making process). Then, each edge-weight inside each shortest path is summed up to return a final weight for each shortest path in the list. Finally, after sorting the k shortest paths according to this final weight, a sorted list of suggested paths is ready to be filtered and then returned to the user.

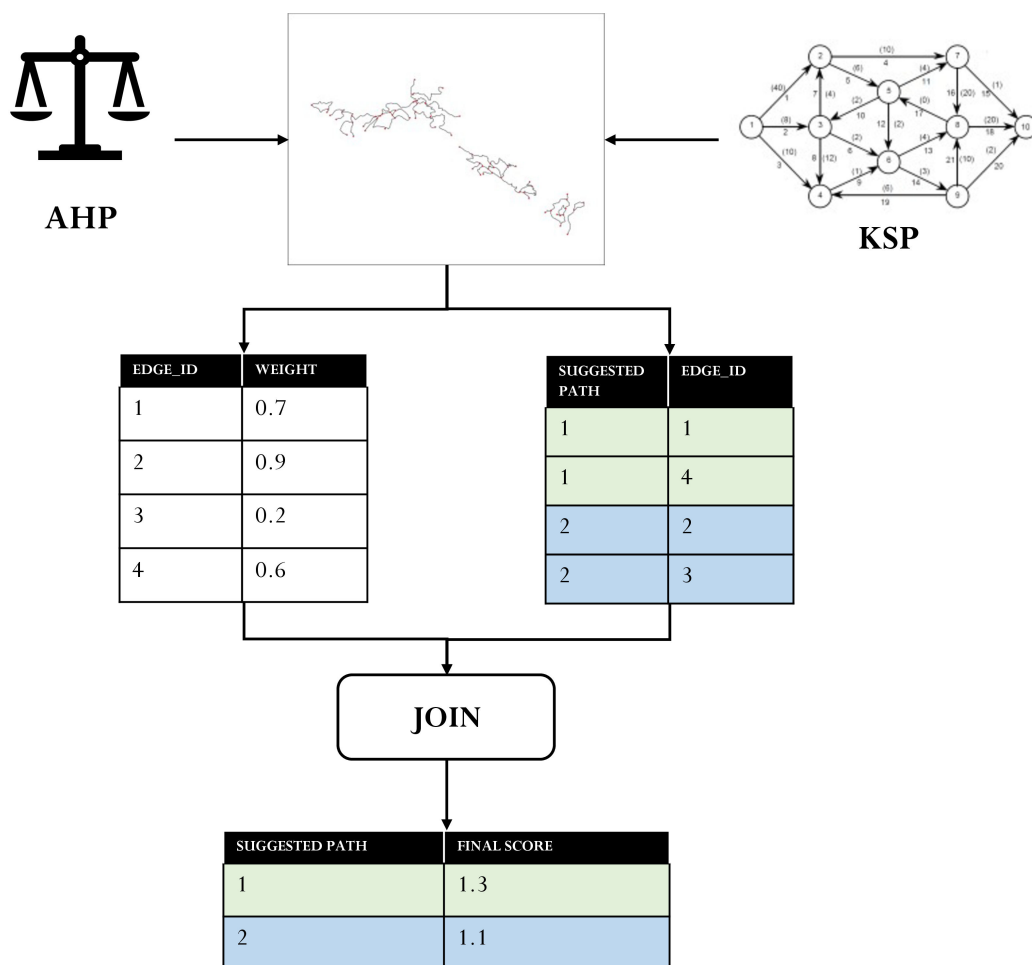


Fig. 5.8: The structure of the Tourist Experience Planner tool

Now, the focus of this thesis moves from the Tourist Experience Planner tool analysis to the description of the geographic dataset and the explanation of the whole INSUBRI.PARKS WebGIS infrastructure.

Data and Software

” Without data, you are just another person with an opinion.

— W. Edwards Deming
(Statistician)

One of the critical aspects of this work is the design and the implementation of an efficient geographic database that includes many types of *mappable* information: boundaries, areas, points of interest and paths network. On the one hand, these geographic entities should be stored in a database; on the other hand, they should also be available for online viewing. The solution to this problem is the previously-mentioned Web Geographic Information System (WebGIS), a more accessible and widespread way of sharing geographic data.

In **Section 6.1.1**, the process of data collection is thoroughly explained as the starting point of the creation of the GIS database. **Section 6.1.2** addresses the details of the exploration and preparation of the data. The implementation of the database mentioned above is addressed to the **Section 6.1.3**. Lastly, **Section 6.2** provides an overview of the general characteristics of a WebGIS together with an analysis of the standard software tools used in the development.

6.1 Creation of the Geo-Database

6.1.1 Data Collection

The first step to be considered in the creation of a proper geographic database is the data collection. A preliminary data census was performed by searching for suitable data own by the INSUBRI.PARKS project partners. Unfortunately, as stated in **Chapter 2**, it emerges that these actors did not own appropriate and detailed dataset concerning their natural heritage. From this, derives the decision to focus the attention on Spina Verde park only. The idea was to create a replicable data model applicable to all the other parks partner of the project.

With this in mind, the collection of useful data for the Spina Verde park has been performed by merging data from different sources:

- **National and Administrative boundaries** of the Park directly from the Spina Verde archive;
- **Paths network** has been recreated starting from both printed map of all the available paths and raw data acquired with GPS (see **Figure 6.1**);
- **Points of Interest** have been collected both from Spina Verde staff and from OpenStreetMap (OSM) [76].



Fig. 6.1: Printed map of the Spina Verde Paths Network.

Once all the required geodata has been collected, those data have to be explored and processed to verify their accuracy, to add extra essential features or to remove redundant and unnecessary features [77]. This task was accomplished using QGIS Desktop [78]. This free and open-source geographic application allows the visualisation, managing, editing and analysing of geospatial data.

6.1.2 Data Exploration and Processing

The graphical representation of spatial data is done in the form of vector data. The vector geometries are rendered through a mathematical manipulation referenced by geographic coordinates. Vector data are mainly used to represent discrete spatial features, which are georeferenced points, lines or polygons. **Figure 6.2** shows all the layers that compose the geo dataset rendered and styled in QGIS [78]. It is a free and open source GIS application enabling the user to visualize, manage, edit, analyse data, and compose printable maps. It runs on Linux, Unix, Mac OSX, Windows and Android and can be downloaded free of charge (<https://qgis.org/en/site>).

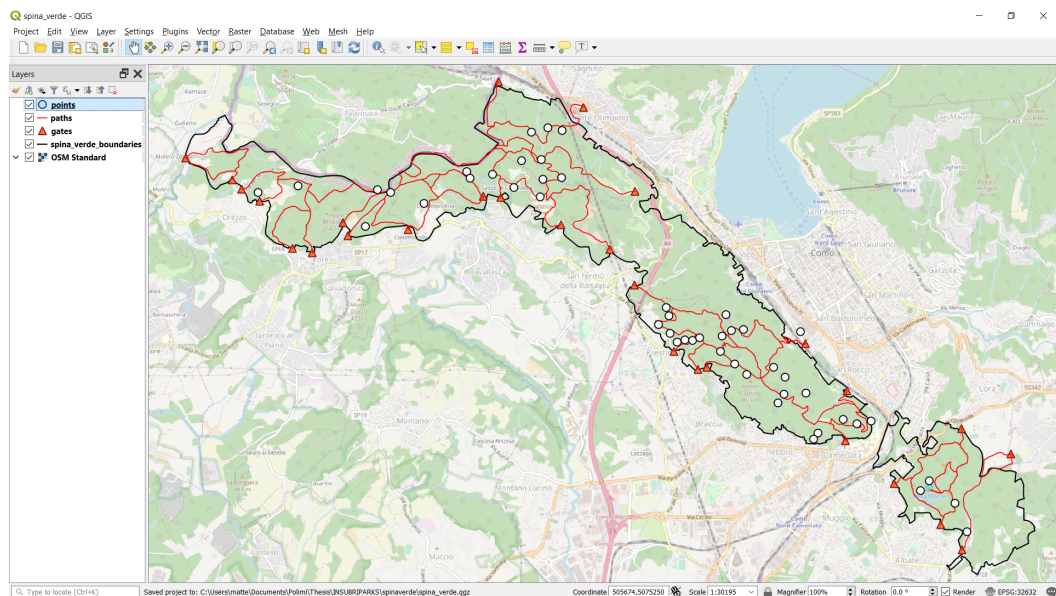


Fig. 6.2: Spina Verde Park with all its layers.

By exploiting QGIS powerful editing tools, each of these layers is processed with the aim of preparing the data to be stored in the database [79].

Re-projection

First of all, to ensure that each layer has the same Coordinate Reference System (CRS), it is set for all the layers to *WGS 84 / UTM zone 32N (EPSG:32632)*, which is the official Lombardy Reference System.

Clipping

Almost every layer has an extent far greater than the actual area of interest. Saving those extra features in the database would be a waste of space; therefore, each layer is clipped according to the park to which it belongs.

Topological adjustment

For the determination of the suitable suggested paths, the paths network must be represented by a set of nodes and arcs that indicate their connection. Therefore, once the paths network is a complete graph, the KSP algorithm can be run to find the K shortest paths in the graph. Since the original paths network vector layer does not provide the right topology, it became necessary to make a morphological manipulation. In fact, in the beginning, the paths network contains the single paths as lines with overlapping segments or with close but separate lines. Moreover, intersections between paths do not correspond to nodes. Therefore, this required a pre-processing on the layer to eliminate duplicate edges and break the lines of the paths at the right intersections.

Buffering and Spatial Join As stated in Chapter 4, each edge in a path has a score for each different category of points of interest. The score of the edge a in the category *Sport* is nothing but the number of points belonging to the *Sport* category which are at a maximum distance of 100 meters from the edge a . This score can be computed by combining a buffer operation of the paths layer with a spatial join operation:

1. Compute a buffer around the paths of sufficient size to cover all the points of interest. Two or more overlapping buffers are accepted because a point can be accessible from 1 to n edges (see **Figure 6.3**);
2. After the processing of both sidewalks (left and right), the next step is to use the Join Attribute by Location tool (a QGIS processing tool) to assign to each point the corresponding edges using their IDs;
3. The last step aims at computing summary statistics for each point category at each edge. In fact, the Statistic by Categories tool (another QGIS processing tool) is applied to the point layer resulting from the spatial join.

After all these transformations, the dataset is ready to be stored in a database. This guarantees the consistency of the data independently from any software or application.

6.1.3 Database implementation

A Relational Database Management System (RDBMS) is a collection of tools that enable to create, update and otherwise interact with a relational database [80].

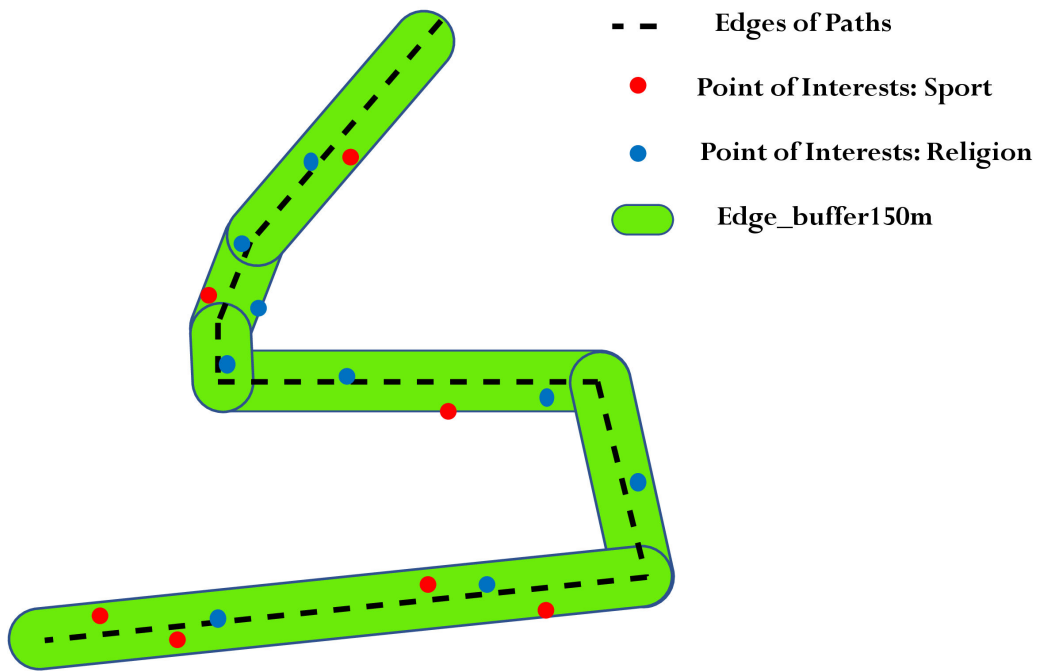


Fig. 6.3: Buffer around edges and points of interest.

RDBMSes store data in the form of tables, using (at most) Structured Query Language (SQL) to access the database.

In the field of GIS, the most popular and widely used RDBMS is PostgreSQL [81]. It is an open-source object-relational database system compliant with all major operating systems. It is provided under the PostgreSQL License, which can be defined as a very liberal license. The reason why it is so diffused among GIS insiders is attributable to PostGIS [82]. It is an extension to PostgreSQL that enables support for geographical objects in the database. This means that the database can store georeferenced points, lines or polygons.

Therefore, given the importance of having an RDBMS archive to guarantee centralised management of information within the INSUBRI.PARKS geographic dataset, all the vector data have been exported from QGIS to PostGIS (see Figure 6.4).

In the database, vector data are utilised to represent discrete spatial features in the shapefile format, an Esri [83] data format used to store location, shape and attributes of geographic features. All the layers currently in the database are presented below (please, remind that they all refer to Spina Verde park:

- **Administrative boundaries:** this layer shows all the administrative boundaries of the park;

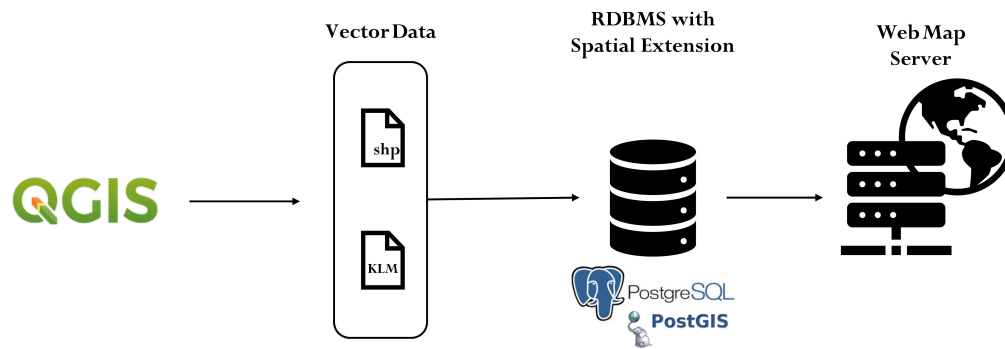


Fig. 6.4: Data flow.

- **Area:** it contains the whole area of the park;
- **Gates:** this layer is composed of all the gates through which a tourist can enter in the park;
- **Paths Network:** it is the core of the Tourist Experience Planner tool. This layer provides the whole paths network of the park built by more than 70 single edges;
- **Historical - Archaeological:** this layer includes both historical and archaeological points of interest;
- **Military:** this layer shows some military weapons, buildings or memorials;
- **Religious:** a collection of religious places (e.g. churches);
- **Nature:** this layer shows all the points characterised by landscape views or natural heritage;
- **Sports:** a collection of facilities, courts and sports equipment;
- **Anthropic activities:** this layer includes all the points of interest in which memories of human activity are preserved.

As previously stated, QGIS is a powerful tool to visualise the layers stored in the geo-database. Since QGIS allows users to query and edit the datasets only from their machines, it must be installed on the computer. The program, besides, needs a specific knowledge to be handled despite its user-friendly interface. For these reasons, amateur users, who desire to browse the map without a reasonable knowledge or possibility to utilise QGIS, may be discouraged. The solution to this problem is

the so-called WebGIS, a more convenient, affordable, and intuitive way for sharing INSUBRI.PARKS data. The next section describes the solution mentioned above.

6.2 Introduction to WebGIS

A WebGIS can be considered as a unified virtual representation that mixes spatial, temporal and attributes information for each object of interest [84]. It allows elaborating analysis, interpretation and simulation in multiple ways that reveal similarities, patterns, and trends in the form of maps. World Wide Web has shown the considerable importance and unique ability of GIS to integrate spatial and non-spatial data with its advanced and flexible architecture [85]. Due to this, it has become possible to distribute geographic information to a worldwide audience with renovated IT infrastructures. WebGIS provides a fresh perspective on information and brings to innovative approaches with an intuitive user interface [86][87].

A WebGIS reaches these major features through standardisation thank to the open standards introduced by the Open Geospatial Consortium (OGC). Namely, the OGC Web Services (OWS) are services developed to make accessible geographical data to the Web through standard Web protocols. They include services for data access, data display and data processing. The standard work-flow for data retrieval is based on four different steps:

- a client demands a server for a functionality;
- the server replies with a .xml file containing these functionalities;
- the client requests the needed data from the server;
- the server provides those data.

As commented previously, a crucial aspect of ensuring the interoperability between the various components is the use of recognised standards and protocols. In this regard, the Web services specified by OGC play an essential role [88]. The most popular OWS implementations are the Web Map Service (WMS), the Web Feature Service (WFS), and the Web Coverage Service (WCS) [89].

WMS

The Web Map Service (WMS) is a standard protocol for retrieving cartographic images produced by a map server. **Figure 6.5** explains how a WMS transforms

geographic information into a georeferenced map image. The request from the client-side sets parameters (e.g. which layer on which area) that the map server takes into account to generate an image that will be finally returned to the client. This standard specifies three methods:

- **GetCapabilities:** returns service-level metadata about the accessible data and the valid request parameters;
- **GetMap:** returns a map with all the requested geographic data;
- **GetFeatureInfo:** returns information about particular features taken from the attribute table of the corresponding layer.

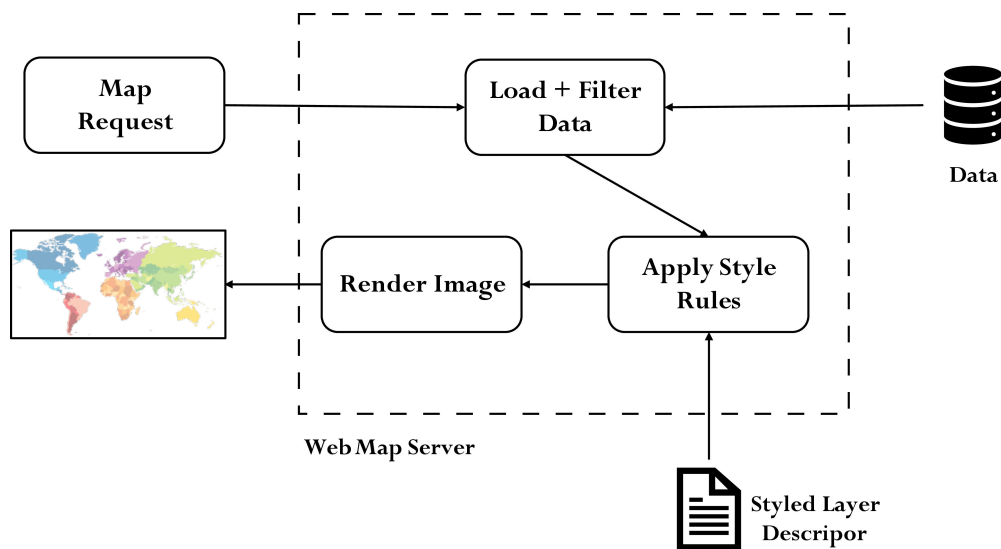


Fig. 6.5: WMS architecture.

WFS

Additionally, the map image can also be rendered by a Web mapping server. The OGC protocol used for it is Web Feature Service (WFS). Through this protocol, users generate their maps and application using the available data (see **Figure 6.6**). This standard defines eleven operations among which the most commonly used are:

- **GetCapabilities:** gives a readable explanation of the available features;
- **DescribeFeatureType:** a description of feature types;
- **GetFeature:** replies with one or more features;

- **Transaction:** allows users to manage the data creation.

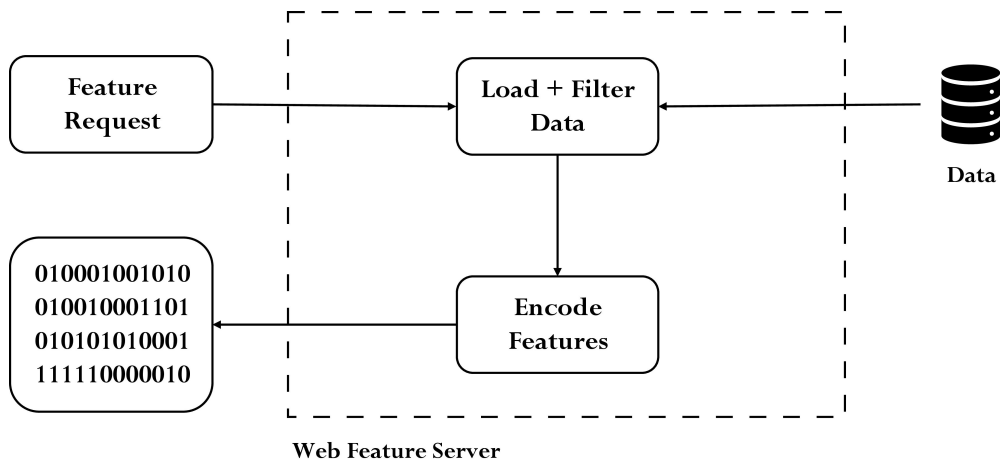


Fig. 6.6: WFS architecture.

WCS

It is a service that encourages electronic retrieval of geospatial coverages, which are geospatial information representing space-varying phenomena. Unlike the WMS, which portrays spatial data to return static maps (rendered as pictures by the server), the Web Coverage Service (WCS) provides available data together with their detailed specifications. It returns data with its original semantics and numerical features (instead of pictures) which may be interpreted, extrapolated and not just represented. Three basic functions compose the WCS service, supported by three interfaces

- **GetCapabilities** returns an XML document describing the service;
- **DescribeCoverage:** lets clients request a full description of one or more coverages served by a specific WCS server;
- **GetCoverage:** returns a coverage (e.g. values or properties of a set of geographic locations), encoded in common coverage format.

6.2.1 Standard Architecture

In performing the GIS analysis tasks, WebGIS is similar to the client/server typical three-tier architecture [90]. The geo-processing is broken down into server-side and client-side tasks. **Figure 6.7** shows a high-level view of a standard WebGIS architecture.

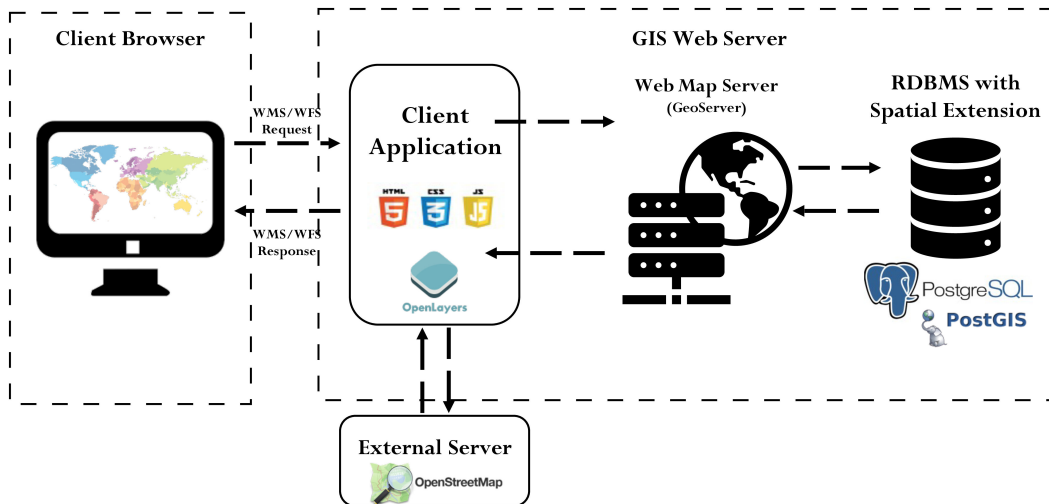


Fig. 6.7: Standard WebGIS Architecture

The main components are:

- **Database & Desktop GIS:** as discussed in the first section of this chapter, the spatial data of the project are stored in a database. The server of the WebGIS connects directly to this database to obtain all the required data. Therefore, any alteration to the database tables is reproduced in real-time on the server and consequently to the client;
- **Web Map Server:** it dynamically produces maps of spatially referenced data from geographic information. It is nothing but a simple interface for Web-based mapping applications. A map server after receiving a request for the desired layers returns a map as a standard picture (GIF, JPG, PNG) or as a vector format (e.g. KML, GML, shp file);
- **External Server:** the WebGIS provides the user with the possibility to select different basemaps as background for the layers. These maps can be retrieved through many External Servers which render maps in the form of rasters composed by tiles. A tile is a square bitmap graphic (normally in .png format) that matches to a particular section of the map. This approach ensures speed and performance since the image is broken up into tiles accordingly to the zoom level. If the map moves to a new location or to a new zoom level, the demanded tiles are retrieved, and the previous ones are stored in the cache of the browser;
- **Client Application:** it is responsible for producing what will be made available to users. It is usually written in HTML, CSS, JS, and then a specific library for displaying maps in the browsers (e.g. LeafLet, Openlayers);

- **Client:** it is the end-user of the system. It has to interact with all the functionalities and the data returned by the Server.

The INSUBRI.PARKS WebGIS is built based on the standard architecture described above. The next chapter goes into details of it.

” *A map is the greatest of all epic poems. Its lines and colors show the realization of great dreams.*

— **Gilbert H. Grosvenor**
(Editor of National Geographic)

In the Spina Verde Park case study, the procedure and the tools outlined in the previous chapters are tested for developing a prototype of WebGIS thanks to which tourists can interact and surf the INSUBRI.PARKS geodatabase.

The next sections focus on all the details of the whole platform: starting from the specific high-level architecture to its main functionalities, but also including an analysis of both goals and user's requirements.

7.1 Architecture

For the development of the project, various tools and technologies are used. Each of them is included in **Figure 7.1** and then discussed in the following paragraphs. Other tools as the ones for deploying on a remote server are not elaborated as they go beyond the specific focus of the thesis.

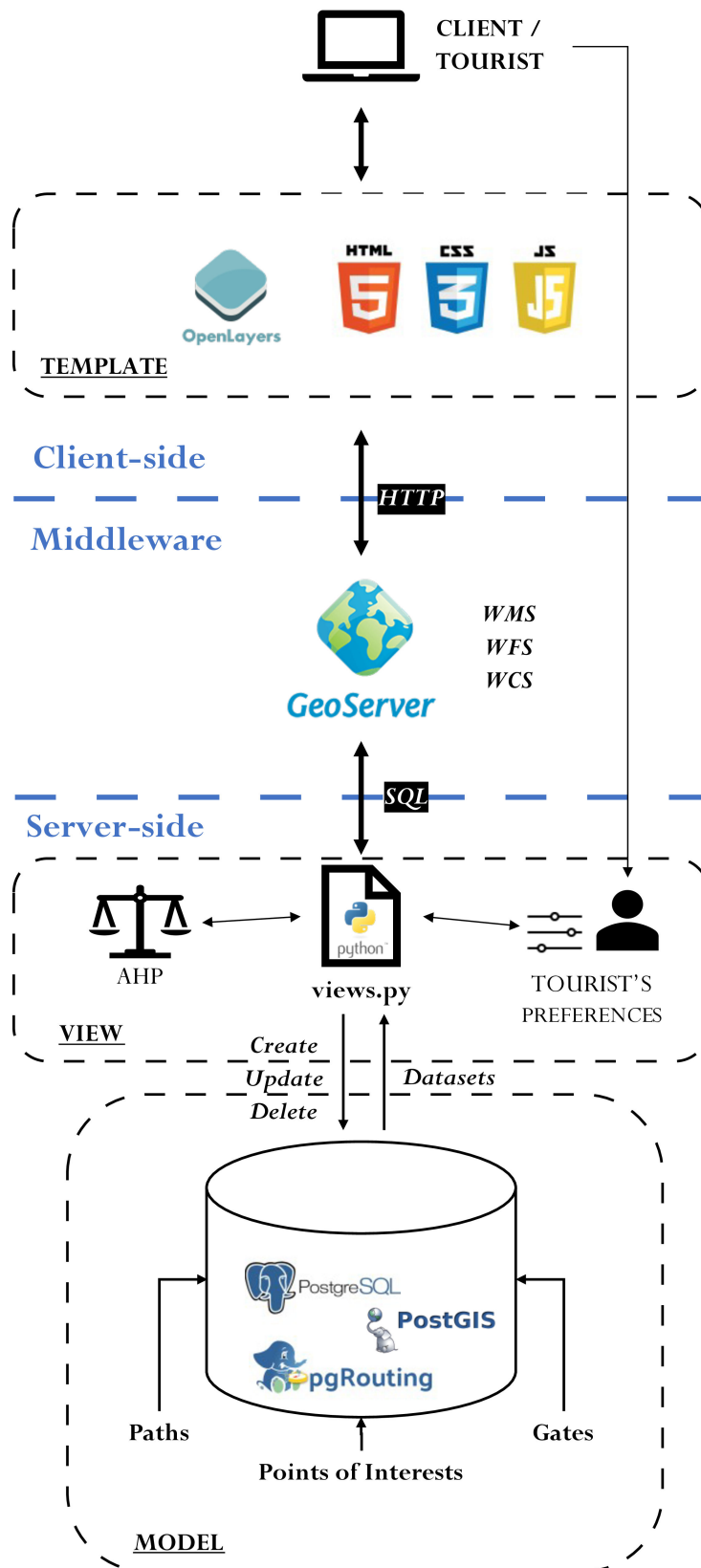


Fig. 7.1: INSUBRI.PARKS WebGIS High-level architecture.

The software architecture of the WebGIS consists mainly of three significant components: client-side, middleware and server-side.

Client-side

This component contains the end-user of the whole platform (the user) together with all the front-end code produced. As stated in the previous chapter, the three coding languages used to design the user interface are:

- **HTML (HyperText Markup Language):** it is the skeleton of a Website. It constitutes the structure, that means the division of where each component should be placed. It uses tags to define this structure [91];
- **CSS (Cascading Style Sheets):** it is the *beauty parlour* of HTML. It uses its defined properties and its syntax to provide a specific style to the selected tags of the HTML components [92];
- **JavaScript:** it is the *nature* of the Website. It defines the behaviour of a Website and all its components. For example, the reaction after moving the mouse pointer over the button is managed by a JavaScript piece of code [93].

The core of the JavaScript scripts can be found in OpenLayers [94]. It is a free and open-source map viewing framework which provides an API to combine maps and a variety of useful components. OpenLayers can be considered as agnostic of server-side technologies because it works very similar to servers, but it has no server coding language at all. It allows, in an easy way, to build an application using layers from different sources and providers through many standards from OGC. The client of INSUBRI.PARKS WebGIS relies mainly on OpenLayers to display both the data hosted on GeoServer and the basemaps retrieved from the External Services (as explained in the previous chapter). This geospatial library provides all the elements which allow users to interact with the map, like the zoom buttons or the full-screen functionality (all the provided functionalities are explicitly described in Section 7.4). The principal motivation behind the decision of using OpenLayers for this project is because of its strong connection with GeoServer.

Code 7.1 shows how OpenLayers load a vector layer through a WFS request to GeoServer. For this reason, the following example can be seen as a bridge between the Client-side component and the Middleware one.

```

/**
 * Vector layers for Spina Verde Points of Interest
 */
var points_Source = new ol.source.Vector({
  format: new ol.format.GeoJSON(),
  url: function() {
    return 'http://insubriparks.como.polimi.it:8080/
      geoserver/Insubriparks/ows?service=WFS&' +
      'version=1.0.0&request=GetFeature&' +
      'typeName=Insubriparks:points&maxFeatures=1000&' +
      'outputFormat=application%2Fjson' +
      '&srsname=EPSG:32632';
  },
  strategy: ol.loadingstrategy.bbox
});

var spinaverde_points = new ol.layer.Vector({
  title: 'Points of Interest',
  source: points_Source,
  displayInLayerSwitcher: false,
  visible: true,
  style: new ol.style.Style({
    image: new ol.style.Icon({
      src: '/static/webgis/img/generic.png',
      scale: 0
    })
  })
});

```

Listing 7.1: WFS request to GeoServer

Middleware

GeoServer [95] is an open-source, cross-platform Web Server written in Java which allows users to manipulate geospatial data. Nowadays, GeoServer has reached a high level of maturity, especially regarding the OGC Standards. It allows publishing spatial data from many sources, including PostGIS database where INSUBRI.PARKS data are stored. Moreover, GeoServer offers a browser-based administration interface (see **Figure 7.2**) that facilitates the configuration of each aspect of the server: adding, styling, publishing layers, or changing in environmental settings. For publishing data, GeoServer offers a system composed of many levels of *containers*. The

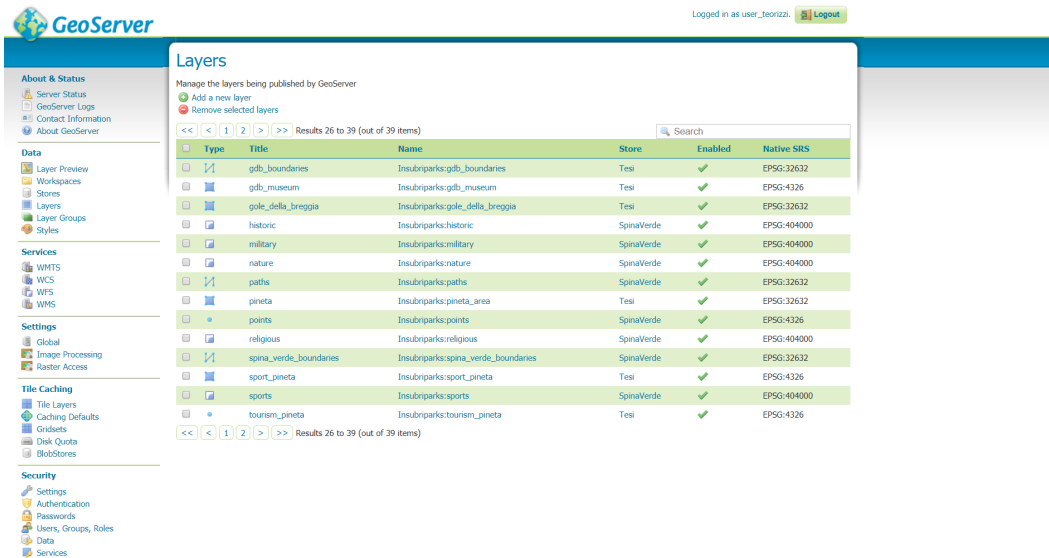


Fig. 7.2: GeoServer Web interface

more widespread is the workspace which is used to put together similar layers. A workspace can contain one or more stores which represent the link to the geospatial data source (e.g. a table in a database, a file). Finally, a store is a collection of layers that is nothing but the collection of all the geographic features. The GeoServer instance of the INSUBRI.PARKS project has one workspace which contains a store connected to the PostGIS database. In turn, each table of the database is published as a layer.

In the INSUBRI.PARKS architecture, GeoServer is responsible for mediating between the Server-side (specifically the database) and the Client layer.

Server-side

The Server-side contains all the required components which trigger the Tourist Experience Planner:

- **Database:** the object-relational database PostgreSQL consists of all the project geographic data. The PostGIS extension spatially enables it and makes possible running its sub-extension *pgRouting* to provide routing and other network analysis functionalities.
- **MCDM:** as outlined in Chapter 4, the MCDM is based on a revisitation of the AHP Multi-Criteria method. A popular and extensively used way to develop these kinds of methods is through the use of dedicated Python scripts. These

scripts allocate the preferences expressed by different involved users to produce a suitable result which will be the input for the pgRouting tool.

7.1.1 Python

Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for Web and app development [96]. It is perfect for Rapid Application Development because it allows dynamic typing and dynamic binding options. Python is accessible to everyone with a unique syntax that focuses on readability [97]. Developers can interpret and translate Python code much more natural than other coding languages. In turn, this reduces the cost of application maintenance and development because it enables teams to operate collaboratively without any significant knowledge barrier. Additionally, Python supports the use of modules, packages and frameworks, which signifies that programs can be designed in a modular style and code can be reused across multiple projects.

The AHP method has been implemented using a popular Python package for data science: Pandas [98]. It offers robust, flexible and extensible data structures that make data manipulation and analysis simple, among many other things. An example of these structures is in **Code 7.2**, where the criteria pairwise comparison matrix is declared as a Pandas DataFrame.

```
import pandas as pd

def criteria_pairwise_comparison(preferences_array):

    #Assign each value of the input array (containing the
        user's preferences of the criteria) to six different
        variables
    h = preferences_array[0] #History
    m = preferences_array[1] #Military
    r = preferences_array[2] #Religion
    a = preferences_array[3] #Anthropic
    s = preferences_array[4] #Sport
    n = preferences_array[5] #Nature

    #Criteria pairwise comparison matrix in a DataFrame,
        values are rounded at 2 decimals
    comparison_matrix = pd.DataFrame({
```

```

'H': [round(h/h,2), round(h/m, 2),
      round(h/r,2), round(h/a,2), round
      (h/s, 2), round(h/n,2), ],
'M': [round(m/h,2), round(m/m, 2),
      round(m/r,2), round(m/a,2), round
      (m/s, 2), round(m/n,2) ],
'R': [round(r/h,2), round(r/m, 2),
      round(r/r,2), round(r/a,2), round
      (r/s,2), round(r/n,2) ],
'A': [round(a/h,2), round(a/m, 2),
      round(a/r,2), round(a/a,2), round
      (a/s, 2), round(a/n,2) ],
'S': [round(s/h,2), round(s/m, 2),
      round(s/r,2), round(s/a,2), round
      (s/s,2), round(s/n,2) ],
'N': [round(n/h,2), round(n/m, 2),
      round(n/r,2), round(n/a,2), round
      (n/s,2), round(n/n,2) ]},
index = ['H', 'M', 'R', 'A', 'S', 'N
        ']
)

```

```
return comparison_matrix
```

Listing 7.2: DataFrame declaration

Then, with the help of the other powerful Pandas structures and tools, all the steps described in **Chapter 4** are carried out to return a DataFrame containing edges and *criteria-based* weights.

Since the AHP method has been entirely developed in Python, the idea of implementing the whole WebGIS platform within a Python environment was born. Among all the other options, the choice fell on Django.

7.1.2 Django

Django is a Python Web framework “for perfectionists with deadlines” which is a software tool used to give an easy way to build and run Web applications [99]. A Web framework usually follows the basic MVC (Model-View-Controller) design pattern for developing Web applications, but Django is a bit different in the right way. It implements the concept of Model-View-Template (MVT). MVT is lightly different

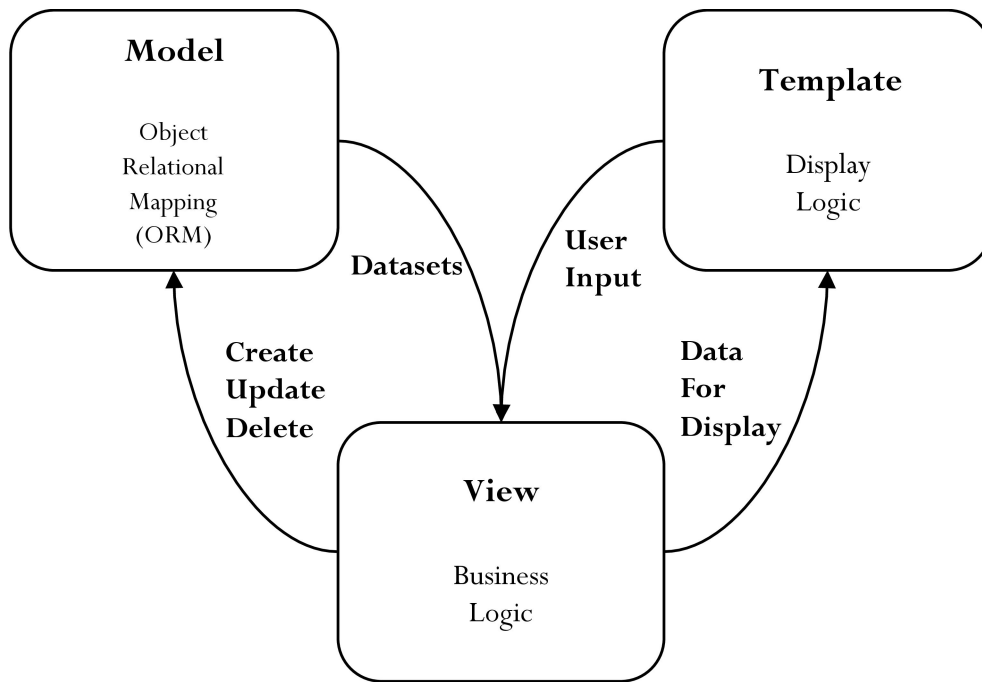


Fig. 7.3: Django MVT diagram

from MVC. The principal difference between the two architectural patterns is that Django itself takes care of the Controller part (which is nothing but the software code that controls the interactions between the Model and View), leaving the programmer with the templates.

Coming back to the begin of **Section 7.1**, in **Figure 7.1**, each component of the MVT pattern is identified as a rectangular dashed box. As can be seen from that picture, the INSUBRI.PARKS WebGIS architecture proposes a variation concerning **Figure 7.3**.

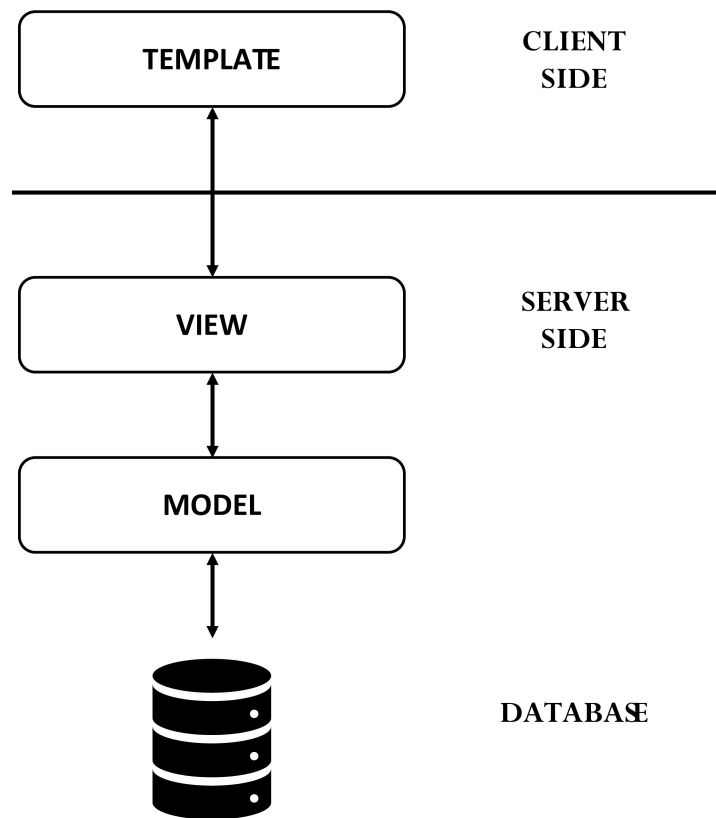


Fig. 7.4: A different view of Django MVT stack

In this variance (see **Figure 7.4**), there is a *line* between the Client and Server-side like all client-server architectures. Django handles both HTTP request and response objects to communicate between the client and the server. So, the View retrieves data from the database (PostgreSQL), which is part of the Model component. At this point, an HTTP request object sends the data to the Client-side (browser).

In other words, the View presents the Model to the client as an HTTP response in the Template.

7.1.3 pgRouting

The last tool to be analysed in the fig [big architecture] is the pgRouting [100] extension of the PostGIS database. pgRouting is an open-source geospatial routing library providing a variety of routing algorithms like *All Pairs Shortest Path (APSP)*, *Shortest Path*, *Driving Distance* and *Turn Restricted Shortest Path (TRSP)*. These routing algorithms are one a topic of interest in the GIS field [101]. pgRouting path algorithms process the full network to find the path between any two points. Theoretically speaking, this hurts the computational time, especially in case of substantial road networks [102].

pgRouting follows a SQL based architecture where the graph is stored in the form of a SQL table in a PostgreSQL database. Through SQL queries, the extension extracts the graph data for paths computation. **Figure 7.5** shows the high-level architecture of pgRouting.

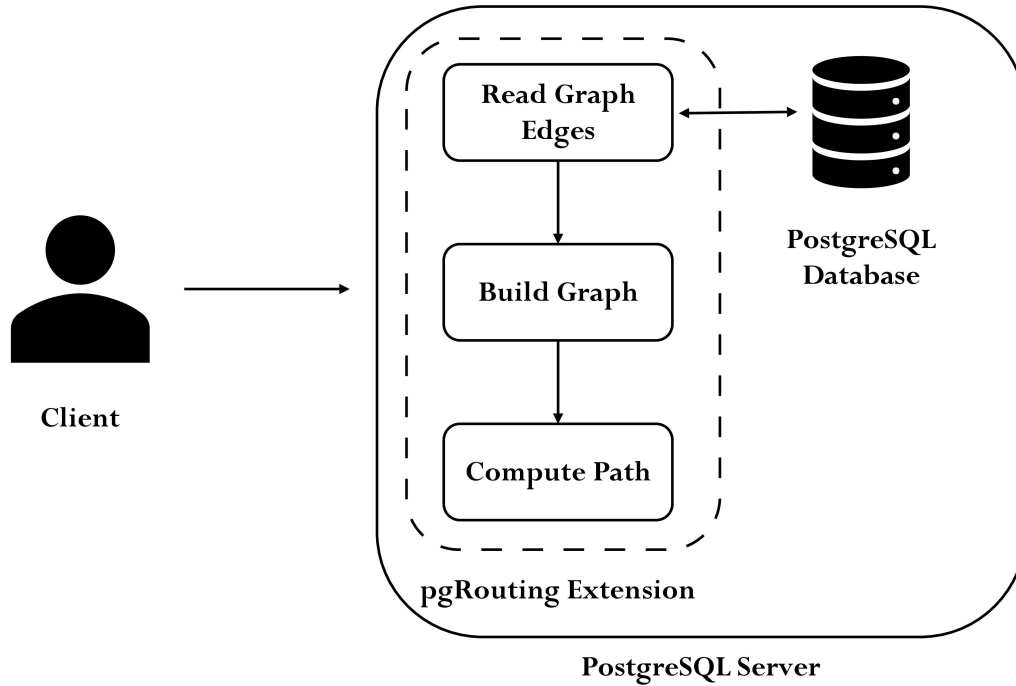


Fig. 7.5: pgRouting Architecture

Two of the components mentioned in the Figure above deserve particular attention:

- **PostgreSQL Database:** it contains the information of edges and vertices of the graph/network $G(V, E)$ stored as a SQL table (see **Table 7.1**)

Column	Data Type	Description
id	long_int	A unique identifier assigned to every edge
source	long_int	Identifier for the source node of the edge
target	long_int	Identifier for the target node of the edge
cost	real	The length of the edge
the_geom	geometry	A postGIS attribute which represents the geometry of the edge

Tab. 7.1: Table of Edges.

The first step for obtaining the table outlined above is to build a network topology of the Spina Verde paths network:

```
SELECT public.pgr_createTopology('paths', 0.001, 'id', 'source', 'target', 'cost', 'the_geom');
```

Listing 7.3: Create a topology query

- **pgRouting Extension:** at this point, the layer is ready to be processed by whatever routing algorithm. As mentioned in Chapter 5, the KSP algorithm is run using the *pgr_ksp* function, which returns the *K* shortest paths.

```
SELECT path_id, edge FROM pgr_ksp('SELECT_*_FROM_paths_network', a, b, n)
```

Listing 7.4: KSP algorithm in pgRouting

Code7.4 is the statement which takes as inputs, in addition to the *paths_network*, the following values:

- **a:** it is the entrance gate selected by the user, the entry-point;
- **b:** it is the exit gate where the shortest paths will end;
- **n:** it is the *K* value, that is the number of shortest paths to be found by the algorithm.

First of all, the pgRouting extension extracts the appropriate graph from the *paths_network* in the PostgreSQL database using a SQL query. Then, the *pgr_ksp* function in the server takes the above-outlined values as input and calculates the path between vertex *a* and vertex *b* and returns the path(s) to the client [103]. **Table 7.2** shows the output format.

Column	Data Type	Description
seq	INT	Sequential value starting from 1. It indicates to which path belongs to each edge
path seq	INT	Relative position in the path. Has value 1 for the beginning of the path.
start_vid	BIGINT	Identifier of the starting vertex
end_vid	BIGINT	Identifier of the ending vertex.
node	BIGINT	Identifier of the node in the path from start vid to end vid.
edge	BIGINT	Identifier of the edge used to go from node to the next node in path sequence. -1 indicates the last node of the path.
cost	FLOAT	Cost to traverse from node using edge to the next node in the path sequence.
agg_cost	FLOAT	Aggregate cost from start vid to node.

Tab. 7.2: pgRouting Output of KSP algorithm.

7.2 Requirements Engineering

After this detailed explanation of what lies behind the final product, it is possible now to shift the attention on the scope of the WebGIS, taking into account what are its specific goals and what are the users' needs.

The scope of the INSUBRI.PARKS WebGIS is to provide a handy and user-friendly platform to users with any level of competence. The application allows these users to navigate and interact with all the project layers. They can also enjoy the Tourist Experience Planner tool, becoming an active part of the whole system.

The primary measure of the success of a software system is the degree to which it meets the purpose for which it was intended [104]. This specific branch of Software Engineering is the so-called Requirements Engineering (RE) which is the process of discovering the purposes above mentioned, by identifying stakeholders and their needs (see **Figure 7.6**). In this context, it is helpful to differentiate the following terms:

- **Machine:** it is the system to be developed;
- **World:** it is the portion of the real-world affected by the Machine which is represented, in this thesis, by the WebGIS;
- **Goals:** they are prescriptive assertions formulated in terms of world phenomena;
- **Requirements:** they are prescriptive assertions formulated in terms of shared phenomena between World and Machine

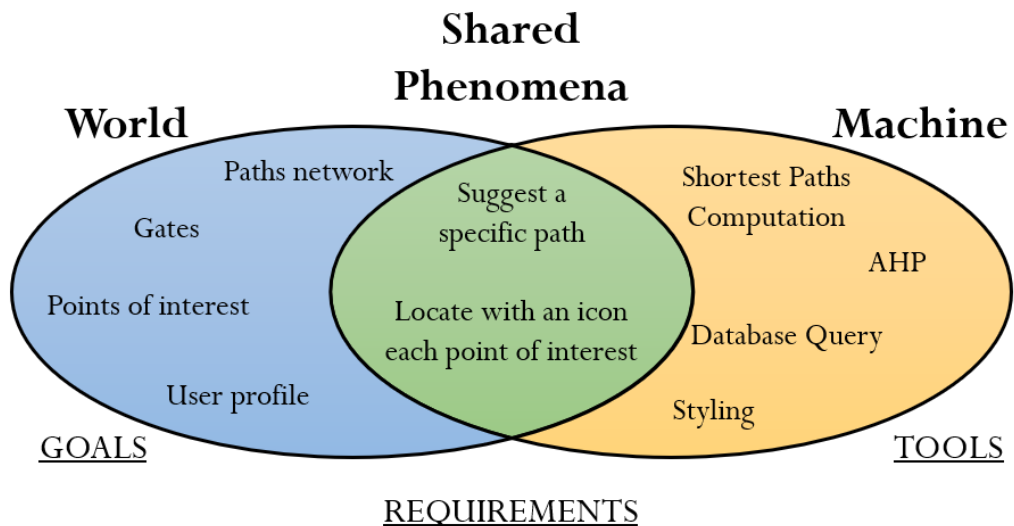


Fig. 7.6: World-Machine dualism.

7.2.1 Goals

The main goals of the system are the following:

- [G1] A user can visualize the position and the details of gates, points of interest and paths;
- [G2] A user can access the Tourist Experience Planner tool and express his preferences;
- [G3] A user can visualize the suggested paths and print them;
- [G4] A user can search a specific point of interest;
- [G5] A user can switch the area of interest very quickly;

7.2.2 Requirements

Once the goals are defined, each of them corresponds to one or more requirements.

- Each point of interest stored in the database is marked with a specific icon. Each category has a different icon to help the user to recognize the point as quickly as possible ([G1]). **Figure 7.7** shows the six icons, once for each category.

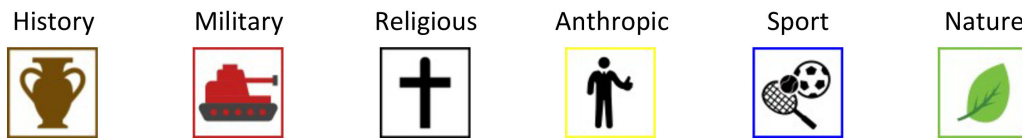


Fig. 7.7: Set categories of icons.

- Gates and Paths have a defined style which makes them identifiable when looking at the map ([G1]);
- All of the spatial data on the map can be queried by clicking on them for asking for more information ([G1]).
- When a user utilizes the Tourist Experience Planner tool, a user-friendly interface helps him to reach the end of the process smoothly ([G2]);
- When a user submits his preferences, then the AHP method is triggered together with the KSP routing algorithm. The result returns the user a list of ranked suggested paths ([G2], [G3]);
- The result of the user's preferences can be easily displayed on the map to facilitate the understanding of it ([G3]);
- The map can be anytime printed exploiting a specific functionality ([G3]);
- Each point of interest can be searched by the user using a search engine based on the name of the points stored in the database ([G4]);
- A user is providing by the capability to move the display frame easily over the map to examine different parks area of current interest ([G5]).

Taking into consideration the goals and the requirements listed above, the following section tries to give a detailed description of the main functionalities provided by the WebGIS.

7.3 Main Functionalities

The idea for this WebGIS is to keep always the map in the spotlight, so all the essential functionalities are collected in a side-bar. Therefore, in the following sections, the focus shifts on the Map and the Tourist Experience Planner tool.

7.3.1 Map

According to the requirements outlined above, each component on the map is now described in details.

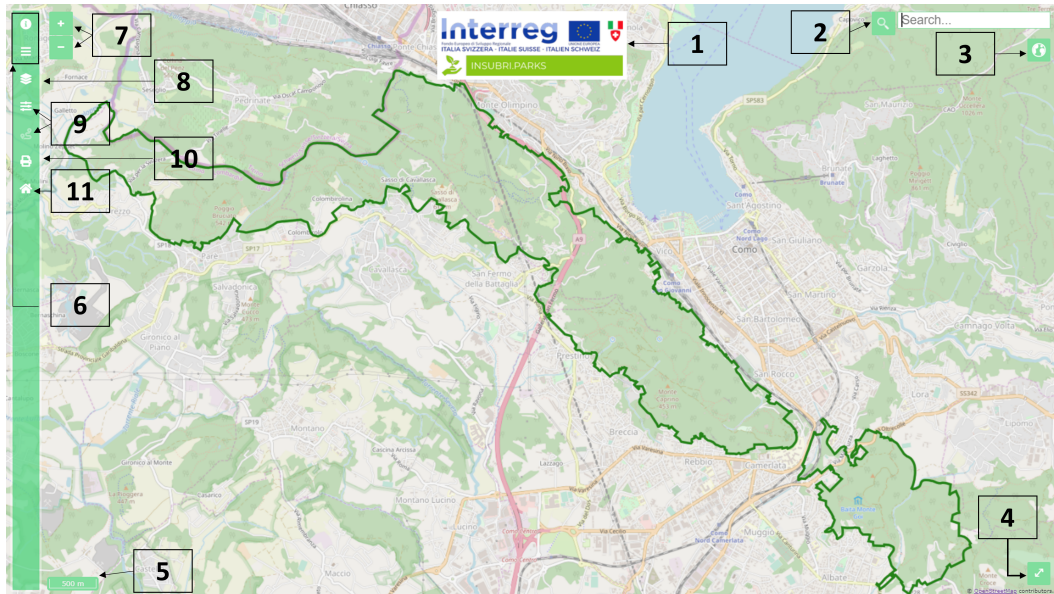


Fig. 7.8: INSUBRI.PARKS WebGIS Interface.

Figure 7.8 shows the map and all its essential components as buttons, panels and other useful widgets.

- **1 Logo:** it is the INSUBRI.PARKS project logo;
- **2 Search bar:** using this bar, the user can search the points of interest which are queried according to their name;
- **3 Map Zone Control button:** this button helps the system to achieve the requirement about the movement of the map display frame (see **Figure 7.9**)
- **4 Full-screen button:** it allows to enlarge the map canvas to its maximum size;
- **5 Scale bar:** it provides a visual indication of the size of each feature and the distance between two of them on the map;
- **6 Descriptive buttons:** they are used to inform the user on both the INSUBRI.PARKS project and how to use the WebGIS;

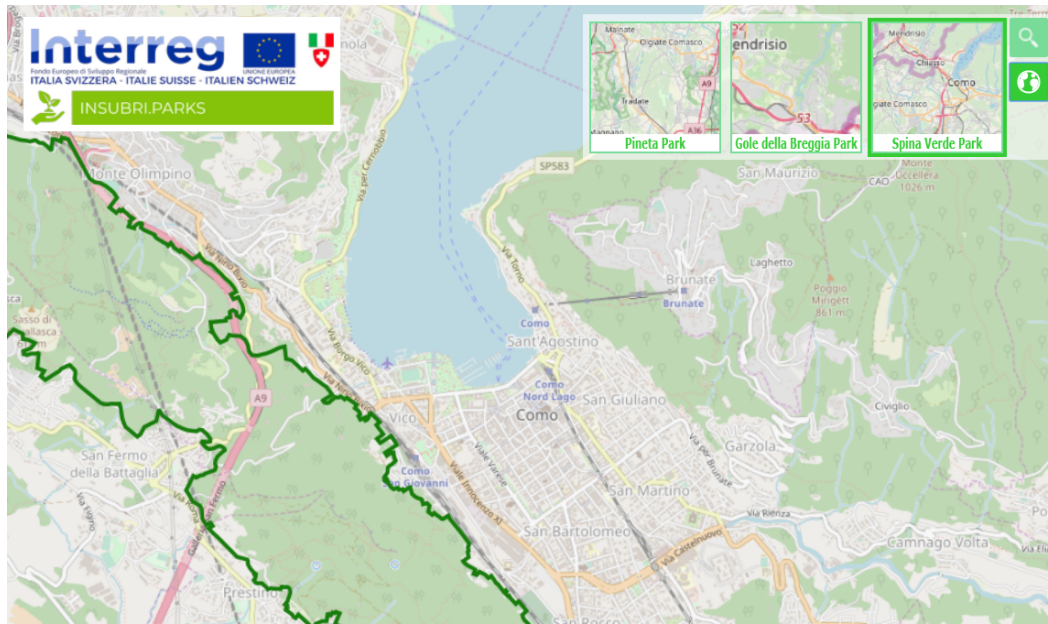


Fig. 7.9: Map Zone function.

- **7 Zoom buttons:** they allow zoom-in or zoom-out;
- **8 Layer switcher:** it is a list of the available layers and basemaps. They can be selected using the appropriate checkbox (see Figure 7.10)

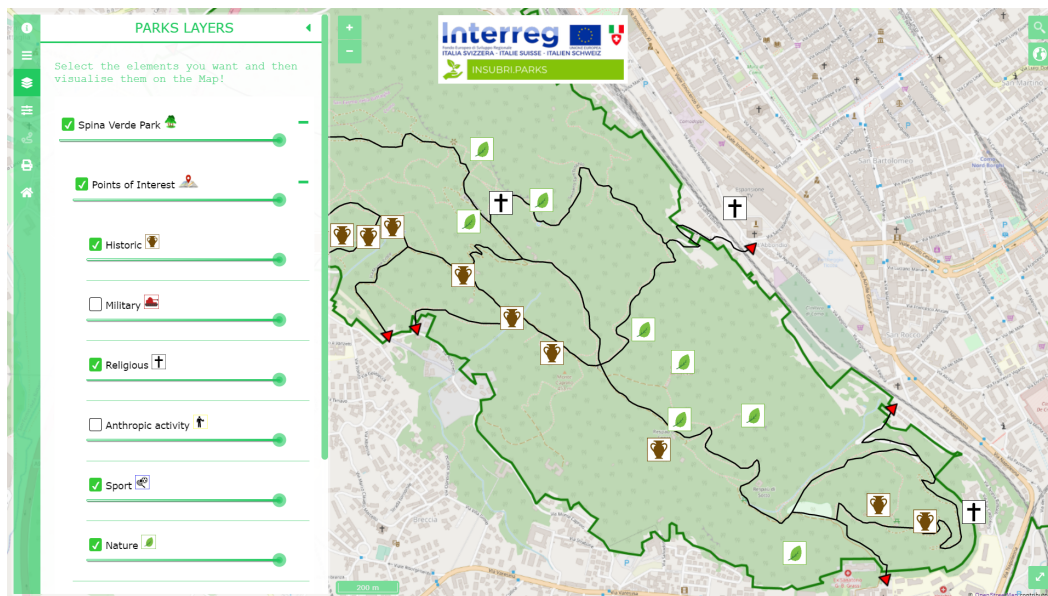


Fig. 7.10: Layers panel.

Each point of interest, if clicked, shows its basic information through a popup (see Figure 7.11)

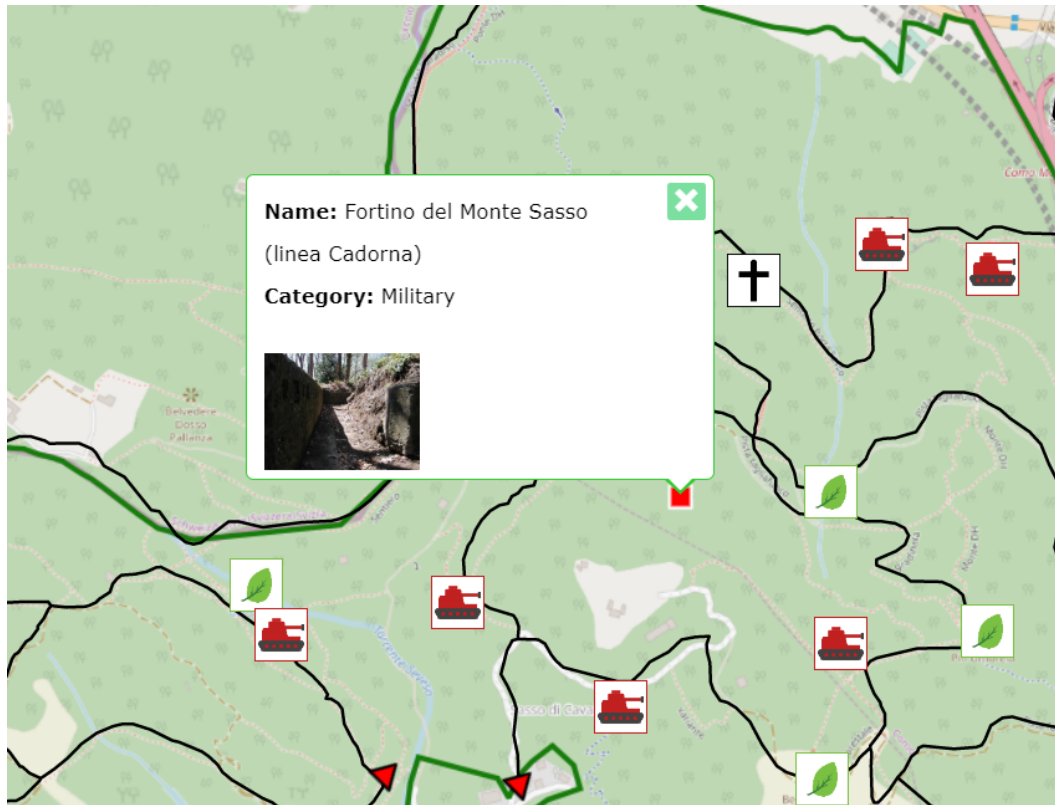


Fig. 7.11: Popup with basic information and image.

- [9] **Multi-Criteria buttons:** they are needed for the Tourist Experience Planner tool which is addressed in the next subsection;
- [10] **Print button:** by clicking it, the user can print whatever is displayed on the map and select his most suitable printing format;
- [11] **Home button:** it allows to come back to the Home page.

7.3.2 Tourist Experience Planner tool

Four panels build the Tourist Experience Planner tool. Each of them is analysed in the next paragraphs:

- **Criteria panel:** at this stage, the user has to rate from 1 to 5 each of the five criteria used in the Multi-Criteria Decision Making algorithm (see **Figure 7.12**)

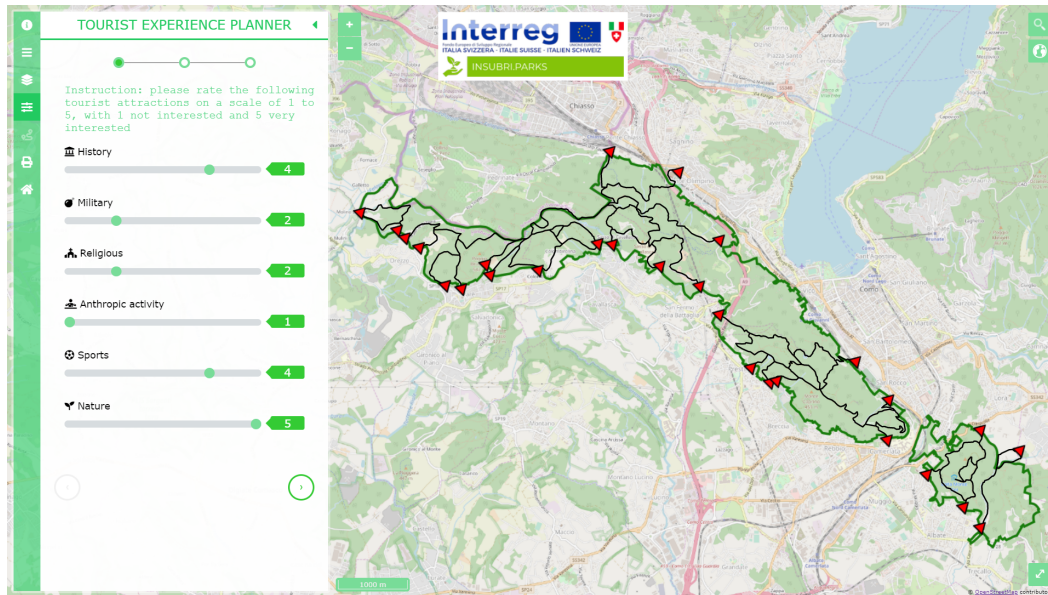


Fig. 7.12: Tourist Experience Planner criteria panel.

- **Constraints panel:** the user has to specify the maximum km to limit the length of suggested paths. In addition to this, he has to select a park (for this prototype the only available is Spina Verde). Finally, he has to indicate optionally specific entry and exit gates, which are relatively the starting and ending point of the suggested paths (see Figure 7.13)

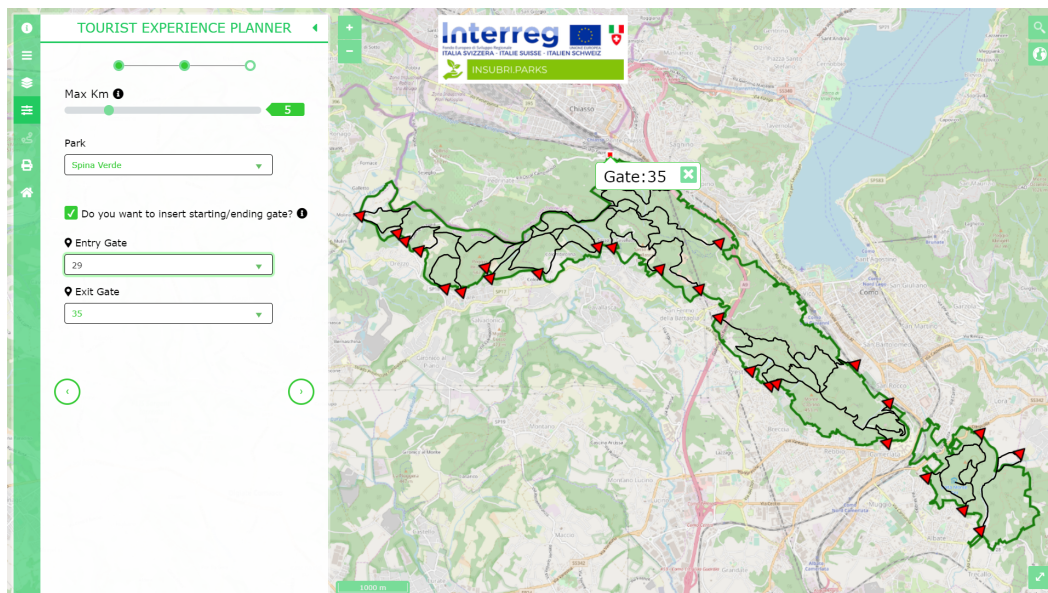


Fig. 7.13: Tourist Experience Planner constraints panel.

- **Specific-requests panel:** this panel provides to the user the opportunity to specify whatever request to be taken into account in the computing of the suggested paths (see **Figure 7.14**)

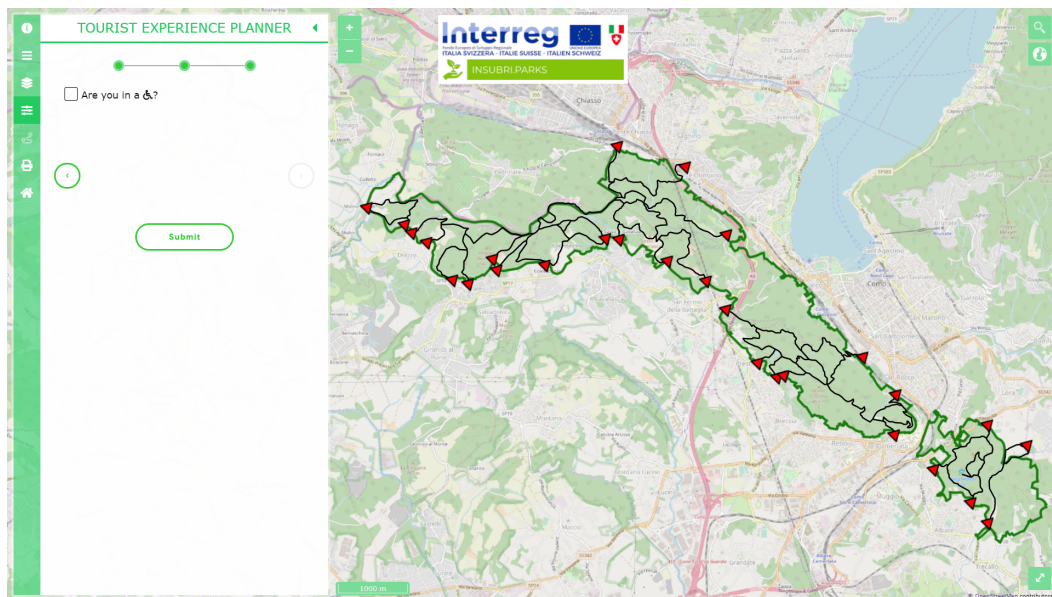


Fig. 7.14: Tourist Experience Planner specific-requests panel.

- **Results panel:** this is the very last panel of the tool. It displays to the user all the suggested paths which the algorithm found. Together with the suggested path, there is a smile icon which indicates the level of affinity with the user's preferences. Next to this, the length of the path is given. By clicking on the result, the suggested path is highlighted in red on the map. The user can obtain three different results:
 - **Predefined paths:** it is the result of the user's choice to do not indicate any gates. The algorithm chooses among the predefined paths of the Park (see **Figure 7.15**)

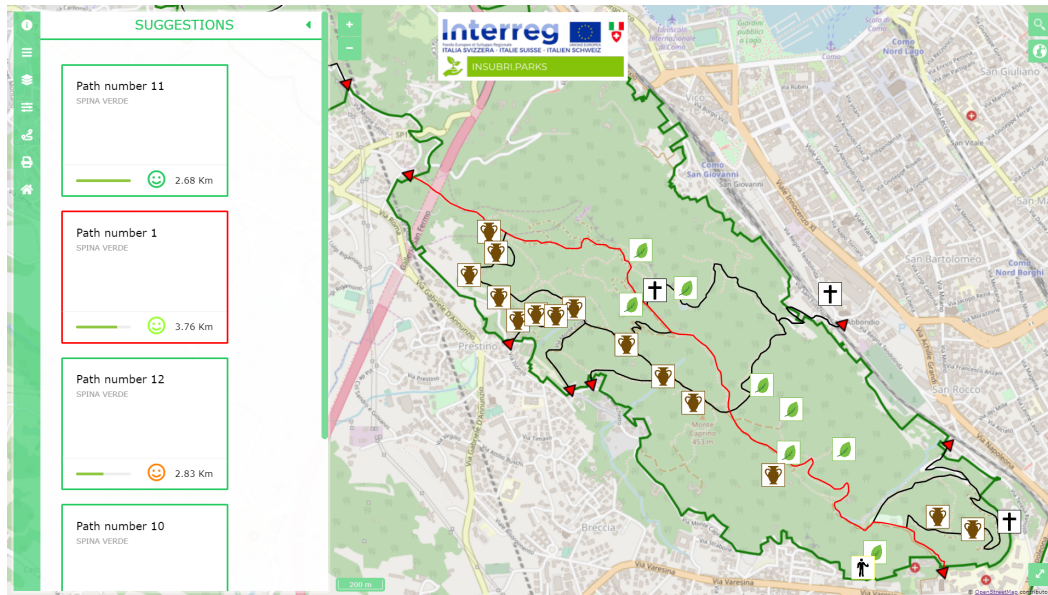


Fig. 7.15: Tourist Experience Planner result with predefined paths.

- **Suggested paths options:** if the user has specified an entry and exit gate, the result is a list of suggested paths computed by the algorithm (see Figure 7.16)

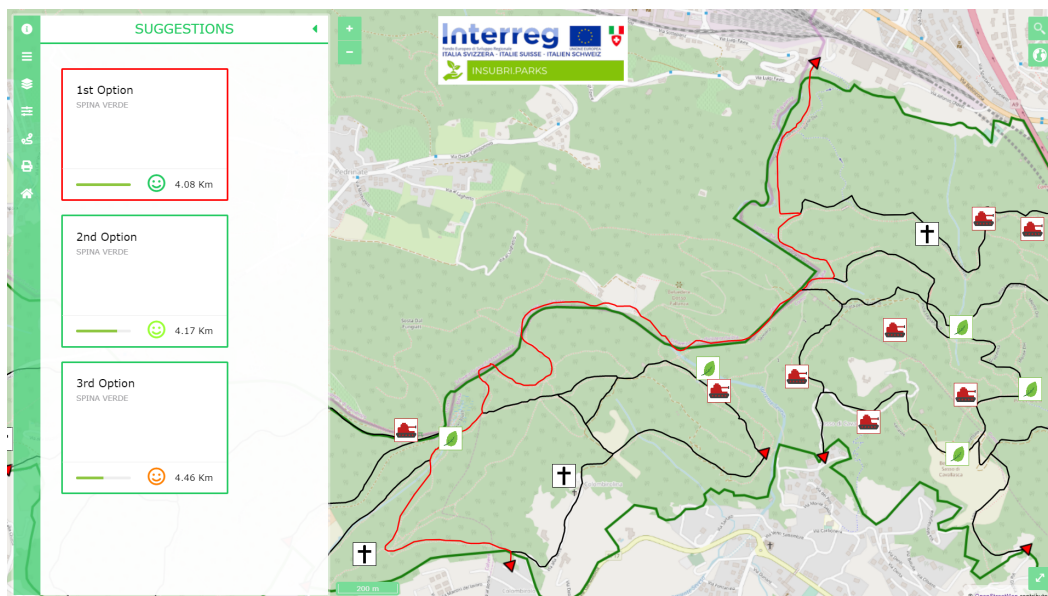


Fig. 7.16: Tourist Experience Planner result with different suggested path options.

- **Error message:** it is an error message (see **Figure 7.17**) which states that it is not possible to find a proper path. The reason behind this message is usually related to the length filter applied by the user in the constraints panel.

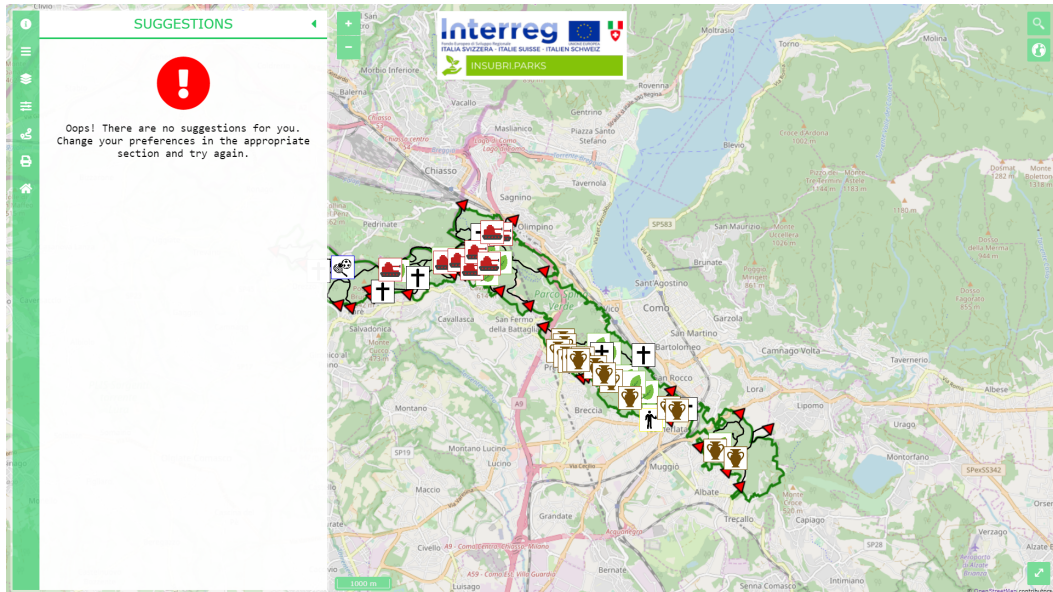


Fig. 7.17: Tourist Experience Planner result error message.

The first prototype of INSUBRI.PARKS WebGIS is available at the following address:
<http://insubriparks.com.polimi.it:8000/webgis>

Conclusion

” *The first step toward creating an improved future is developing the ability to envision it.*

— **Tony Dungy**
(Coach)

Most of the existing Italian natural parks do not have an adequate IT infrastructure, which implies of owing a heritage not fully exploited. This GIS platform will try to help INSUBRI.PARKS partners to reach their full potential by showing all the beauty and richness of our territory. Indeed, the present work demonstrates how the integration of MCDM to GIS can actively support users in performing geospatial decision-making tasks. This thesis, therefore, attests it through the application of finding the most suitable paths within a park performing both innovative methods (e.g. AHP revisitation) and well-known spatial algorithms (e.g. KSP algorithm).

The MCDM and GIS blending has potential applications in various scientific as well as professional fields (e.g. tourism planning, logistic problems and emergency response), thanks also to the exclusive use of Free and Open Source Software and Open Data which minimizing development costs while ensuring performances and replicability of the proposed tools.

At the moment, the WebGIS is in a testing phase. Only when this phase is over, the users' feedback will be used to make improvements both in contents and style.

Possible future improvements regard the possibility for users to get more information about points of interest, maybe by improving the information stored in the database. Moreover, adding an altimetric profile of the suggested paths can be helpful for those users who need that information (e.g. mothers with baby strollers or people with disabilities). Finally, the implementation of a user profiling system could be helpful for collecting generic users' data which would allow optimising the Tourist Experience Planner tool. In fact, by exploiting those data through machine learning, new marketing campaign and technical enhancements can be done.

As a first case study, only the Spina Verde Park was considered. In the immediate future, all the other Parks partner of the project will be included.

In general terms, this research work aims to highlight the role of open data in empowering local tourism by implementing a practical solution for retrieving valuable knowledge from geospatial data by overcoming the existent gap between the available information and the users.

List of Figures

2.1	The INSUBRI.PARKS project area [1].	4
2.2	INSUBRI.PARKS project strategies, actions and objective	6
4.1	Decision Making block diagram.	13
4.2	Step by Step Decision Making Process.	15
4.3	Classification of Decision Making Models.	16
4.4	Tourist Experience Planner model.	19
4.5	Pairwise comparison <i>A</i> matrix.	21
4.6	The structure of INSUBRI.PARKS MCDM model.	24
5.1	Spina Verde Park paths network as an undirected graph.	26
5.2	Simple graph example.	27
5.3	Simple graph with labels.	27
5.4	Graph used to define walks, trails and paths.	28
5.5	Walk example	29
5.6	Trail example	29
5.7	Path example	30
5.8	The structure of the Tourist Experience Planner tool	37
6.1	Printed map of the Spina Verde Paths Network.	40
6.2	Spina Verde Park with all its layers.	41
6.3	Buffer around edges and points of interest.	43
6.4	Data flow.	44
6.5	WMS architecture.	46
6.6	WFS architecture.	47
6.7	Standard WebGIS Architecture	48
7.1	INSUBRI.PARKS WebGIS High-level architecture.	52
7.2	GeoServer Web interface	55
7.3	Django MVT diagram	58
7.4	A different view of Django MVT stack	59
7.5	pgRouting Architecture	60
7.6	World-Machine dualism.	63
7.7	Set categories of icons.	64
7.8	INSUBRI.PARKS WebGIS Interface.	65
7.9	Map Zone function.	66

7.10	Layers panel.	66
7.11	Popup with basic information and image.	67
7.12	Tourist Experience Planner criteria panel.	68
7.13	Tourist Experience Planner constraints panel.	68
7.14	Tourist Experience Planner specific-requests panel.	69
7.15	Tourist Experience Planner result with predefined paths.	70
7.16	Tourist Experience Planner result with different suggested path options.	70
7.17	Tourist Experience Planner result error message.	71

List of Tables

4.1	Advantages and Disadvantages of MADM Methods [46].	18
4.2	The Fundamental Scale by Saaty	20
4.3	Random Consistency Index	22
4.4	Thresholds values	22
4.5	Example of P_{ec} matrix	23
7.1	Table of Edges.	60
7.2	pgRouting Output of KSP algorithm.	62

List of Acronyms

AHP Analytic Hierarchy Process. 12

ANP Analytic Network Process. 12

CI Consistency Index. 22

CR Consistency Ratio. 22

CRS Coordinate Reference System. 41

CSS Cascading Style Sheets. 48

DSS Decision Support System. 7

FOSS Free and Open Source Software. 2

GIF Graphics Interchange Format. 48

GML Geography Markup Language. 48

HTML HyperText Markup Language. 48

HTTP HyperText Transfer Protocol. 59

ICT Information and Communication Technology. 7

JPG Joint Photographic Experts Group. 48

JS JavaScript. 48

KML Keyhole Markup Language. 48

KSP K Shortest Paths. 36

MADM Multi-Attribute Decision Making. 16

MAUT Multi-Attribute Utility Theory. 17

MCDM Multi-Criteria Decision Making. 7

MODM Multi-Objective Decision Making. 16

MVC Model-View-Controller. 57

MVT Model-View-Template. 57

NPS National Park Service. 8

OGC Open Geospatial Consortium. 45

OTR Swiss Regional Tourism Office. 5

PDF Portable Document Format. 8

PNG Portable Network Graphics. 48

RCI Random Consistency Index. 22

RDBMS Relational Database Management System. 42

RE Requirements Engineering. 62

SQL Structured Query Language. 43

SUPSI Scuola Universitaria Professionale della Svizzera Italiana. 5

TOPSIS Technique for Order of Preference by Similarity to Ideal Solution. 12

WCS Web Coverage Service. 45

WFS Web Feature Service. 45

WMS Web Map Service. 45

XML eXtensible Markup Language. 47

Bibliography

- [1] D Oxoli et al. "NATURAL HERITAGE MANAGEMENT AND PROMOTION THROUGH FREE AND OPEN SOURCE SOFTWARE: A PRELIMINARY SYSTEM DESIGN FOR THE INSUBRIPARKS PROJECT." In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2019).
- [2] *Interreg*. Accessed: 2020-03-19.
- [3] Isabelle D Wolf et al. "The use of public participation GIS (PPGIS) for park visitor management: A case study of mountain biking". In: *Tourism Management* 51 (2015), pp. 112–130.
- [4] Andrea Ballatore et al. "RecoMap: an interactive and adaptive map-based recommender". In: *Proceedings of the 2010 ACM Symposium on Applied Computing*. 2010, pp. 887–891.
- [5] Amor Laaribi. "Systemes d'information géographique et analyse multicritere: integration pour l'aide a la decision a reference spatiale." In: (1996).
- [6] Rodolphe Devillers, Robert Jeansoulin, and Michael F Goodchild. *Fundamentals of spatial data quality*. ISBN: 1905209568. ISTE London, 2006.
- [7] Yashon O Ouma, Emmanuel C Kipkorir, and Ryutaro Tateishi. "MCDA-GIS integrated approach for optimized landfill site selection for growing urban regions: an application of neighborhood-proximity analysis". In: *Annals of GIS* 17.1 (2011), pp. 43–62.
- [8] Sittichai Choosumrong, Venkatesh Raghavan, and Nicolas Bozon. "Development of Web-GIS Application for Emergency Route Decision and Planning using AHP analysis and pgRouting algorithm". In: *Proceedings of GIS-IDEAS* (2012), pp. 414–419.
- [9] Bernard Roy. *Méthodologie multicritère d'aide à la décision*. BOOK. Economica, 1985.
- [10] *Pro Val Mulini WebSite*. <http://www.provalmulini.eu>. Accessed: 2019-12-06.
- [11] *Mendrisiotto Turismo WebSite*. <https://www.mendrisiottoturismo.ch>. Accessed: 2019-12-06.

- [12] Polimi WebSite. <http://www.polimi.it>. Accessed: 2019-12-06.
- [13] SUPSI WebSite. <http://www.supsi.ch>. Accessed: 2019-12-06.
- [14] Dimitrios Buhalis. *eTourism: Information technology for strategic tourism management*. Pearson education, 2003.
- [15] Dimitrios Buhalis and Rob Law. "Progress in information technology and tourism management: 20 years on and 10 years after the Internet—The state of eTourism research". In: *Tourism management* 29.4 (2008), pp. 609–623.
- [16] R Egger and D Buhalis. *E-tourism case studies: Management & marketing issues in eTourism*. 2008.
- [17] Tarmiji Masron, Norhasimah Ismail, and Azizan Marzuki. "The conceptual design and application of web-based tourism decision support systems". In: *Theoretical and Empirical Researches in Urban Management* 11.2 (2016), pp. 64–75.
- [18] Ganesh Datt Bhatt and Jigish Zaveri. "The enabling role of decision support systems in organizational learning". In: *Decision Support Systems* 32.3 (2002), pp. 297–309.
- [19] Juan R González, David A Pelta, and Antonio D Masegosa. "A framework for developing optimization-based decision support systems". In: *Expert Systems with Applications* 36.3 (2009), pp. 4581–4588.
- [20] Peter Markus. "Design and development of a prototype mobile geographic information system for real-time collection and storage of traffic accident data". In: *Student thesis series INES* (2016).
- [21] MA Hamadouche et al. "GIS-based multicriteria analysis: an approach to select priority areas for preservation in the Ahaggar National Park, Algeria". In: *Arabian Journal of Geosciences* 7.2 (2014), pp. 419–434.
- [22] Bakhtyar Ali Ahmad, Himan Shahabi, and Baharin Bin Ahmad. "Application of GIS based multi-criteria analysis in site selection of water reservoirs (Case Study: Batu Pahat, Malaysia)". In: *Research Journal of Applied Sciences, Engineering and Technology* 9.11 (2015), pp. 995–1005.
- [23] Verka Jovanović. "The application of GIS and its components in tourism". In: *Yugoslav Journal of Operations Research* 18.2 (2016).
- [24] Tarmiji Masron, Badaruddin Mohamed, and Azizan Marzuki. "GIS base tourism decision support system for Langkawi Island, Kedah, Malaysia". In: *Theoretical and Empirical Researches in Urban Management* 10.2 (2015), pp. 21–35.
- [25] Franca IACOBELLIS et al. "Millesentieri: il portale/Webgis Opensource dei sentieri per la promozione delle risorse naturalistiche, architettoniche e storico-archeologiche". In: *Atti ASITA* (), pp. 1267–1273.

- [26] *Spina Verde WebSite*. <https://www.spinaverde.it/main/>. Accessed: 2019-12-06.
- [27] *Finland parks*. <https://www.nationalparks.fi/web/outdoors>. Accessed: 2019-12-06.
- [28] *NPS website*. <https://www.nps.gov/subjects/gisandmapping/npmap.htm#8/44.745/-110.814>. Accessed: 2020-01-20.
- [29] Cengiz Kahraman. *Fuzzy multi-criteria decision making: theory and applications with recent developments*. Vol. 16. Springer Science & Business Media, 2008.
- [30] Zhe Zhang and Kirsi Virrantaus. “Use of a Fuzzy Decision-making Approach in an Analysis of the Vulnerability of Street Networks for Disaster Management”. In: *Nordic Journal of Surveying and Real Estate Research* 11.2 (2016), pp. 7–19.
- [31] Khalil Gibran. “Evaluation of transportation projects and programs using multiple criteria”. In: (2007).
- [32] Theodosios RH Kritikos and Timothy RH Davies. “GIS-based Multi-Criteria Decision Analysis for landslide susceptibility mapping at northern Evia, Greece [GIS-basierte multikriterielle Entscheidungsanalysen zur Kartierung von Massenverlagerungspotenzialen im nördlichen Evia, Griechenland.]” In: *Zeitschrift der Deutschen Gesellschaft für Geowissenschaften* 162.4 (2011), pp. 421–434.
- [33] Ahmet Bozkurt. “Multi-criteria decision making with interdependent criteria using prospect theory”. PhD thesis. Master thesis, Middle East Technical University, Ankara, Turkey [online . . . , 2007.
- [34] BC Juan. “Scikit-Criteria Documentation”. In: (2018).
- [35] Peter C Fishburn. “A survey of multiattribute/multicriterion evaluation theories”. In: *Multiple criteria problem solving*. Springer, 1978, pp. 181–224.
- [36] Edmundas Kazimieras Zavadskas, Zenonas Turskis, and Simona Kildienė. “State of art surveys of overviews on MCDM/MADM methods”. In: *Technological and economic development of economy* 20.1 (2014), pp. 165–179.
- [37] Aditya Chauhan and Rahul Vaish. “Magnetic material selection using multiple attribute decision making approach”. In: *Materials & Design (1980-2015)* 36 (2012), pp. 1–5.
- [38] M Murat Köksalan, Jyrki Wallenius, and Stanley Zionts. *Multiple criteria decision making: from early history to the 21st century*. World Scientific, 2011.
- [39] Ralph L Keeney, Howard Raiffa, and David W Rajala. “Decisions with multiple objectives: Preferences and value trade-offs”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.7 (1979), pp. 403–403.

- [40] Ching-Lai Hwang and Abu Syed Md Masud. “Methods for multiple objective decision making”. In: *Multiple objective decision making—methods and applications*. Springer, 1979, pp. 21–283.
- [41] Gwo-Hshiung Tzeng and Jih-Jeng Huang. *Multiple attribute decision making: methods and applications*. CRC press, 2011.
- [42] Yoram Wind and Thomas L Saaty. “Marketing applications of the analytic hierarchy process”. In: *Management science* 26.7 (1980), pp. 641–658.
- [43] Thomas Lorie Saaty. *Decision making with dependence and feedback: The analytic network process*. Vol. 4922. RWS Publ., 1996.
- [44] János Fülöp. “Introduction to decision making methods”. In: *BDEI-3 workshop, Washington*. 2005, pp. 1–15.
- [45] Praveen Kumar Gade and Manjit Osuri. *Evaluation of Multi Criteria Decision Making Methods for Potential Use in Application Security*. 2014.
- [46] Mark Velasquez and Patrick T Hester. “An analysis of multi-criteria decision making methods”. In: *International journal of operations research* 10.2 (2013), pp. 56–66.
- [47] Emrah Önder, Bahadır Fatih Yıldırım, and Muhlis Ozdemir. “Multi criteria decision making approach for evaluating tourism destinations in Turkey”. In: *Academic Journal of Tourism and Management Researches* 1.1 (2013), pp. 1–15.
- [48] Matthew J Liberatore and Robert L Nydick. “Group decision making in higher education using the analytic hierarchy process”. In: *Research in Higher Education* 38.5 (1997), pp. 593–614.
- [49] Zhu Xian Cui et al. “Multi-criteria group decision making with fuzzy logic and entropy based weighting”. In: *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*. 2011, pp. 1–7.
- [50] Thomas L Saaty and Luis G Vargas. “The possibility of group choice: pairwise comparisons and merging functions”. In: *Social Choice and Welfare* 38.3 (2012), pp. 481–496.
- [51] Ungku Ahmad. “Application of multiple criteria decision making techniques in tourism and hospitality industry: A systematic review”. In: *Transformations in Business & Economics* 15.1 (2016), p. 37.
- [52] GO Odu. “Weighting methods for multi-criteria decision making technique”. In: *Journal of Applied Sciences and Environmental Management* 23.8 (2019), pp. 1449–1457.
- [53] Weiqiang Li. “Algebraic methods in graph theory”. PhD thesis. University of Delaware, 2015.
- [54] K Ruohonen. *Graph theory, Graafiteoria lecture notes, TUT*. 2013.

- [55] Kairanbay Magzhan and Hajar Mat Jani. “A review and evaluations of shortest path algorithms”. In: *International journal of scientific & technology research* 2.6 (2013), pp. 99–104.
- [56] Adeel Javaid. “Understanding Dijkstra’s Algorithm”. In: *Available at SSRN 2340905* (2013).
- [57] M Pascoal. “Implementations and empirical comparison for K shortest loopless path algorithms”. In: *The Ninth DIMACS Implementation Challenge: The Shortest Path Problem* (2006).
- [58] David Eppstein. “Finding the k shortest paths”. In: *SIAM Journal on computing* 28.2 (1998), pp. 652–673.
- [59] Eleni Hadjiconstantinou and Nicos Christofides. “An efficient implementation of an algorithm for finding K shortest simple paths”. In: *Networks: An International Journal* 34.2 (1999), pp. 88–101.
- [60] John Hershberger, Matthew Maxel, and Subhash Suri. “Finding the k shortest simple paths: A new algorithm and its implementation”. In: *ACM Transactions on Algorithms (TALG)* 3.4 (2007), 45–es.
- [61] Husain Aljazzar and Stefan Leue. “K*: A heuristic search algorithm for finding the k shortest paths”. In: *Artificial Intelligence* 175.18 (2011), pp. 2129–2154.
- [62] Qingsong Wen et al. “Finding top k shortest simple paths with improved space efficiency”. In: *Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems*. 2017, pp. 1–6.
- [63] Walter Hoffman and Richard Pavley. “A Method for the Solution of the N th Best Path Problem”. In: *Journal of the ACM (JACM)* 6.4 (1959), pp. 506–514.
- [64] Narsingh Deo and Chi-Yin Pang. “Shortest-path algorithms: Taxonomy and annotation”. In: *Networks* 14.2 (1984), pp. 275–323.
- [65] *K shortest paths algorithms*. <http://iinwww.ira.uka.de/bibliography/Theory/k-path.html>. Accessed: 2020-03-18.
- [66] S Clarke, A Krikorian, and J Rausen. “Computing the N best loopless paths in a network”. In: *Journal of the Society for Industrial and Applied Mathematics* 11.4 (1963), pp. 1096–1102.
- [67] Jin Y Yen. “Finding the k shortest loopless paths in a network”. In: *management Science* 17.11 (1971), pp. 712–716.
- [68] Jin Y Yen. “Another algorithm for finding the K shortest loopless network paths”. In: *Proc. of 41st Mtg. Operations Research Society of America* 20 (1972).
- [69] Eugene L Lawler. “A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem”. In: *Management science* 18.7 (1972), pp. 401–405.

- [70] Bennett L Fox. “Data structures and computer science techniques in operations research”. In: *Operations Research* 26.5 (1978), pp. 686–717.
- [71] Douglas R Shier. “On algorithms for finding the k shortest paths in a network”. In: *Networks* 9.3 (1979), pp. 195–214.
- [72] Naoki Katoh, Toshihide Ibaraki, and Hisashi Mine. “An efficient algorithm for k shortest simple paths”. In: *Networks* 12.4 (1982), pp. 411–427.
- [73] Aarni Perko. “Implementation of algorithms for K shortest loopless paths”. In: *Networks* 16.2 (1986), pp. 149–160.
- [74] *Basic notation of Yen’s algorithm*. https://en.wikipedia.org/wiki/Yen%27s_algorithm. Accessed: 2020-03-19.
- [75] Ernesto QV Martins and Marta MB Pascoal. “A new implementation of Yen’s ranking loopless paths algorithm”. In: *Quarterly Journal of the Belgian, French and Italian Operations Research Societies* 1.2 (2003), pp. 121–133.
- [76] *OSM WebSite*. <https://www.openstreetmap.org>. Accessed: 2020-03-22.
- [77] A Alamouri and M Gerke. “DEVELOPMENT OF A GEODATABASE FOR EFFICIENT REMOTE SENSING DATA MANAGEMENT IN EMERGENCY SCENARIOS.” In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 4 (2019).
- [78] *QGIS WebSite*. <https://www.qgis.org/it/site/>. Accessed: 2020-03-22.
- [79] P Montalto et al. “Il GeoDatabase dell’INGV della Sezione di Catania-Osservatorio Etneo: progettazione ed implementazione”. In: (2016).
- [80] *RDBMS*. <https://techterms.com/definition/rdbms>. Accessed: 2020-03-22.
- [81] *PostgreSQL WebSite*. <https://www.postgresql.org/>. Accessed: 2020-03-30.
- [82] *PostGIS WebSite*. <https://postgis.net/>. Accessed: 2020-03-30.
- [83] *ESRI WebSite*. <https://www.esri.com/en-us/home>. Accessed: 2020-03-22.
- [84] Ali Asghar Alesheikh, Hussein Helali, and HA Behroz. “Web GIS: technologies and its applications”. In: *Symposium on geospatial theory, processing and applications*. Vol. 15. 2002.
- [85] Tahsin Ettefagh. “Designing and implementing a Web GIS platform for visitors’ event experiences”. In: (2019).
- [86] Cristina Rosales Sánchez. *Prototype design for a Web-GIS disaster management system: enhancement of flood management systems by integration of crowdsourcing data*. 2013.
- [87] DANIELE OXOLI. “GIS-based analysis of the urbanization effect on territorial settings and surface runoff: the case of the buried torrents of Como city”. In: (2015).

- [88] Christopher D Michaelis and Daniel P Ames. *Web Feature Service (WFS) and Web Map Service (WMS)*. 2008.
- [89] EDOARDO PESSINA. “Landslide monitoring geographic information system”. In: (2019).
- [90] Agata LO TAURO, Giuseppe DI PAOLA, and Salvatore Enrico SPINA. “Preservation and Valorisation of cultural and environmental resources and information systems, an investigation into a Web GIS”. In: *Proceedings of the international conference on urban planning and regional development in the information society geomultimedia (CORP 2005), Vienna, Austria*. 2005, pp. 311–318.
- [91] *HTML Documentation*. <https://www.w3schools.com/html/default.asp>. Accessed: 2020-03-30.
- [92] *CSS Documentation*. <https://www.w3schools.com/css/default.asp>. Accessed: 2020-03-30.
- [93] *JavaScript Documentation*. <https://www.w3schools.com/js/default.asp>. Accessed: 2020-03-30.
- [94] *OpenLayers*. <https://openlayers.org/>. Accessed: 2019-12-06.
- [95] *GeoServer*. <http://geoserver.org/>. Accessed: 2019-12-06.
- [96] *Python Definition*. <https://djangobook.com/mdj2-django-structure/>. Accessed: 2020-03-25.
- [97] Rounak Banik. *Hands-On Recommendation Systems with Python: Start building powerful and personalized, recommendation engines with Python*. Packt Publishing Ltd, 2018.
- [98] *Pandas WebSite*. <https://pandas.pydata.org/>. Accessed: 2020-03-30.
- [99] Laxmi Thebe. “Community parenting platform: development and deployment using the Django framework”. In: (2016).
- [100] *pgRouting Documentation*. <https://pgrouting.org/>. Accessed: 2020-03-30.
- [101] Sittichai Choosumrong, Venkatesh Raghavan, and Nicolas Bozon. “Multi-criteria emergency route planning based on analytical hierarchy process and pgRouting”. In: *Geoinformatics 23.4* (2012), pp. 159–167.
- [102] Rohith Reddy Sankepally and KS Rajan. “IMPROVING PATHQUERY PERFORMANCE IN PGROUTING USING A MAP GENERALIZATION APPROACH.” In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* 42 (2018).
- [103] David Burstein and Leigh Metcalf. “The k shortest paths problem with application to routing”. In: *arXiv preprint arXiv:1610.06934* (2016).

- [104] Bashar Nuseibeh and Steve Easterbrook. “Requirements engineering: a roadmap”. In: *Proceedings of the Conference on the Future of Software Engineering*. 2000, pp. 35–46.