POLITECNICO
MILANO 1863

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

GIS-BASED STRATEGY FOR DISTRIBUTION GRID PLANNING IN RURAL AREAS

Master Thesis by:
Vinicius Gadelha Teixeira Filho

Advisor:
Prof. Marco Merlo

Co-Advisor:
PhD Student Silvia Corigliano
Eng. Massimo Bolognesi

Academic Year 2019

Acknowledgments

I would first like to thank my advisor Professor Marco Merlo, for being the one who made all this work possible and guiding me in my academic and professional career.

I would also like to thank my co-advisors. PhD student Silvia Corigliano for sharing all her knowledge in the kindest way and giving me support, from the very first day, in every part of my work. Eng. Massimo Bolognesi for being an open link of communication between me and the partner company Enel GI&N, providing me the information necessary to carry on my activities and also support my professional career. I could not have asked for better guidance.

A grateful thanks to all the professors from the School of Industrial and Information Engineering of Politecnico di Milano, particularly the ones in the Smart Grid master's degree who contributed to my formation and helped me to be a better engineer.

I thank all the friends that directly or indirectly supported me during these years. My two roommates Kevin and Henrique who shared with me the burden and the glory of it all. My lovely colleagues Aleksandar, Angela, Hristina, Ivana and Mateus. The road would not be as fun without all of you.

Finally, I would like to reserve a special place for my family, the most important people of my life. My sisters Beatriz and Thais and my parents Vinicius and Zélia for allowing me to pursue my dreams even when it involves sacrifices, being the best example of human beings I could follow. My love Ana, for staying by my side, supporting me through every obstacle and showing me how to persevere. You show me everyday what true love means. This is for you.

Thank you all.

Abstract

Access to electricity is considered a necessity to human dignity and many would argue that it should be a basic human right. Since 2015 the United Nations adopted the access to affordable, reliable and sustainable energy as one of the 17 Sustainable Development Goals, which are an urgent call for action by all countries in a global partnership. This led to an increasing effort with approximately 120 million people gaining access to electricity each year, but by 2018 there are still 860 million to be reached (*IEA*, 2019a). Most of this population is located in rural remote rural areas, mainly in the sub-Saharan Africa, where projects of electrification are often prohibitively costly. Differently from urban electrification, which is generally carefully planned to achieve increased reliability, rural electrification lean towards economical aspects due to the high investment costs these projects demands. This requires, therefore, new strategies to be developed that can deal with the specificity of the problem. The strategy here proposed makes use of Geographic Information System (GIS) and terrain analysis to create the best electric network topology, and apply its results to a real case study, with the collaboration of *Enel GI&N*, in the municipality of Cavalcante in a rural area in Brazil.

This evaluation requires a deep literature review on the rural electrification strategies that have been adopted worldwide. The most conventional form of electricity transport, three-phase systems, has been constantly evolving. One of the innovations developed in the recent years is the use of compact transformers that offers a more economical solution for supplying auxiliary services (*Orsini*, 2013). Due to their high versatility in terms of voltage level, they can improve rural electrification by allowing connection between LV loads to HV lines and reducing the total length of MV lines that would otherwise be required to connect to HV/MV substations. More suitable for low power applications, like in rural areas, single-phase systems has also been widely used. By reducing the number of cables, these system are often the most economical ones. Strategies such as the Single Wire Earth Return (SWER), which uses a single MV single-phase conductor and the earth as the return ground wire, are reported to be able to reduce up to 30% the total capital costs (*Karhammer et al.*, 2006). Other less conventional strategies, such as the Shielded Wire Systems (SWS), uses the shielded wires of HV transmissions towers as conductors and creates a MV system capable of supplying local loads (*Iliceto*, 2016). Lastly, the boom of Distributed Generation (DG) and Renewable Energy Sources (RES) in the last decades, provided an economical off-grid solution for electrification besides the connection to an existent electric grid.

Besides deciding which type of electric system will be used, a rural electrifica-

tion project must find the best network topology for the area considered. What path the electric lines must take, which terrains it should avoid, and how to group loads in the most efficient way, are some of the questions to be answered. The GIS for electrification (GISEle) tool was developed as an effort to improve the planning of rural electrification in developing countries (*Carnovali and Edeme*, 2019). It is an open source Python-based tool that uses GIS and terrain analysis to model the area under study, groups loads using a density-based clustering algorithm called DBSCAN and then it uses graph theory to find the least-costly electric network topology that can connect all the people in the area. The methodology of GISEle consists in three main steps: data analysis, clustering and grid routing. During the initial phase of data gathering and analysis, GIS data sets are created in order to properly map several information about the area to be electrified. Some of these information are: population density, elevation, slope and roads. They are all processed using a weighting strategy that translates the topological aspect of the terrain to the difficulty of line deployment. Then, DBSCAN is used to strategically aggregates groups of people in small areas called clusters. The output is a set number of clusters, partially covering the initial area considered, in which the grid routing algorithm is performed. Finally, GISEle uses the concept of Steiner tree to create a network topology connecting all the aggregated people in each cluster, and then, if necessary, it makes use of Dijkstra's algorithm to connect each cluster grid into an existing distribution network.

This thesis contribution involves improving the initial concept of the GISEle tool, and apply it on a case of rural electrification, with the support of the partner company Enel Global Infrastructure and Networks S.r.l (*Enel GI&N*). The results obtained by the least-cost topology of GISEle was considered too simplistic when compared to the real distribution grids. The goal is to better represent the hierarchical structure of an electric network that is usually composed by a high power main branch that connects densely populated areas, and its low power derivations (called collaterals) that connects sparse populated areas. Figure 1 represents a sketch of the standard topology (a) and the proposed topology (b).

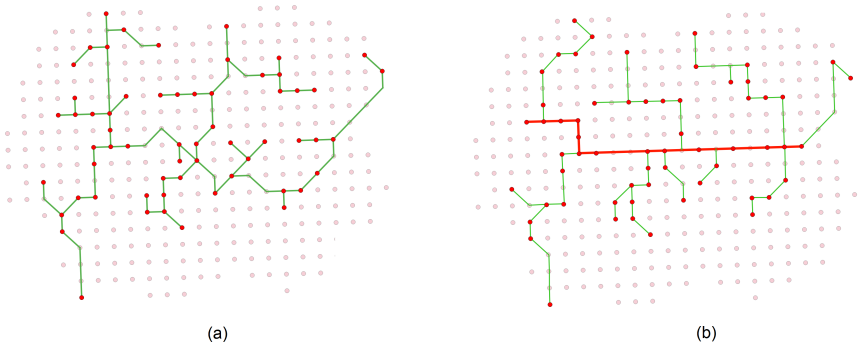

Figure 1: *Example of (a) a simple least-cost topology and (b) a least-cost topology considering a hierarchical structure composed of a main branch and collaterals*

To achieve this more realistic topology a two-step procedure using two different resolution grids was developed. By using a lower resolution grid of points, the population can be aggregated and the grid routing algorithm will consider only the densely populated areas, creating the main branch. Then using a higher resolution

grid of points, a weight reassignment strategy is used so that the the Steiner tree created exploits the presence of the main branch, creating this way the collaterals. Using this approach, each of these collaterals can be identified and accordingly sized based on the peak power of the loads they supply. Besides that, other improvements were made to GISEle to enhance its capability: substation typification and assignment, clustering sensitivity, cluster merging and human-machine interface are among the elements that were improved. Furthermore, a new python routine for improving the substation connection was developed. It evaluates the cost of each connection and chooses the least-cost one allowing for each group of loads (clusters) to share the same MV substation.

The new approach proposed for GISEle was used in a real case study in Brazil. The main goals of this thesis work, which were decided together with the partner company Enel GI&N's engineers that supported the study, can be summarized as:

- Create electric grids connecting people in the municipality of Cavalcante in Brazil, allowing for an increase number of customers that can be connected with on-grid solutions and lay the foundations for a future trade-off comparison with off-grid solutions;
- Connect those grids to the existing distribution network presented in figure 2;
- Generate a topological representation based on GIS of the items mentioned above;
- Size the cables and report the costs.

The results of this rural electrification analysis, which focus only on the topological aspect and does not consider other important factors such as system reliability and quality of service, suggests that that it is possible to reach up to 100% of the local population through an expansion of the distribution network. The new main branch and collaterals approach that was developed, managed to reduce up to 47% the total investment cost in line deployment in respect to the initial GISEle approach. With the previous assumptions, the total length of lines necessary to achieve 100% electrification was 1635 km, with a total investment cost of 33.93 million euros. These costs are related to the line deployment only, other costs such as the MV/LV transformers and protection equipment were not considered. The average cost per person connected went from 5212 euros in the standard approach to 2785 euros using the main branch and collaterals (if households are considered, the average cost is 17199 euros in the standard approach and 9190 with the main branch and collaterals approach).

This cost reduction suggests that an optimized electrification strategy through better routing could shift the balance point between on-grid and off-grid solutions such as the use of microgrids and PV generation. Within this thesis work, the trade-off between the percentage of people connected and the cost per person connected and, due to DSO request, the cost necessary to achieve 100% electrification was assessed. The results validate not only the technicality of the approach proposed, by the cost optimization achieved, but also the topological aspect of the grid created by GISEle, which is similar to the existent MV distribution grid. Figures 2 and 3

present an overall view of the distribution grid expansion achieved by this thesis work, where this topological similarity can be verified.

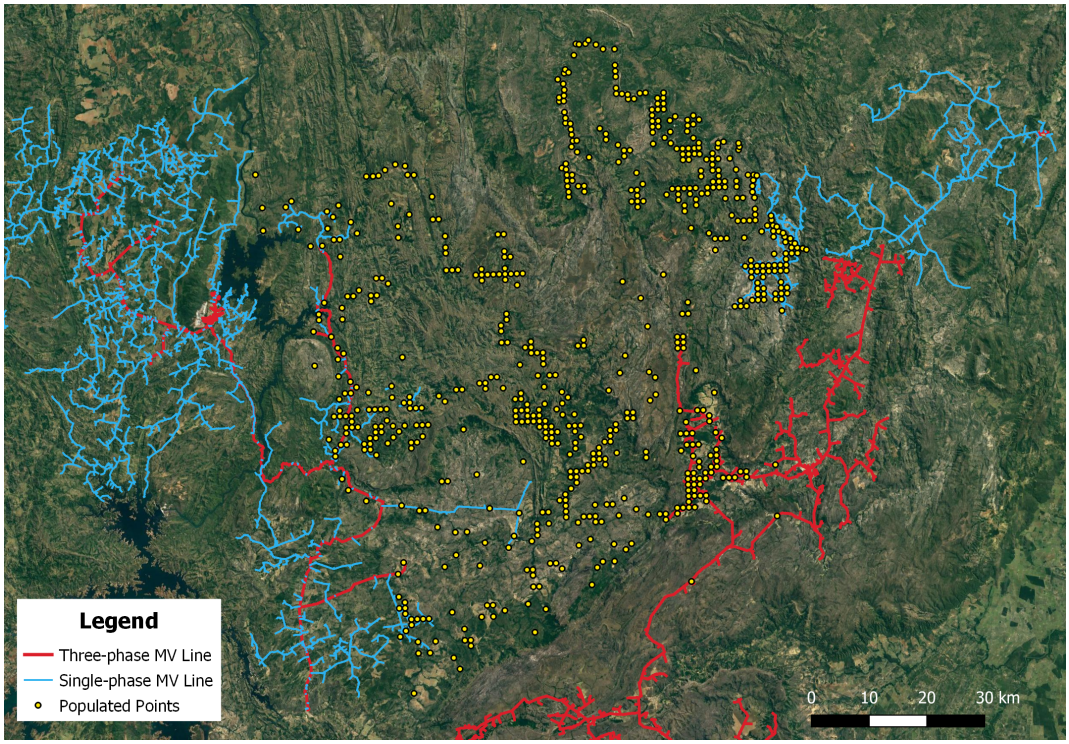

Figure 2: *Image of the distribution grid before expansion*

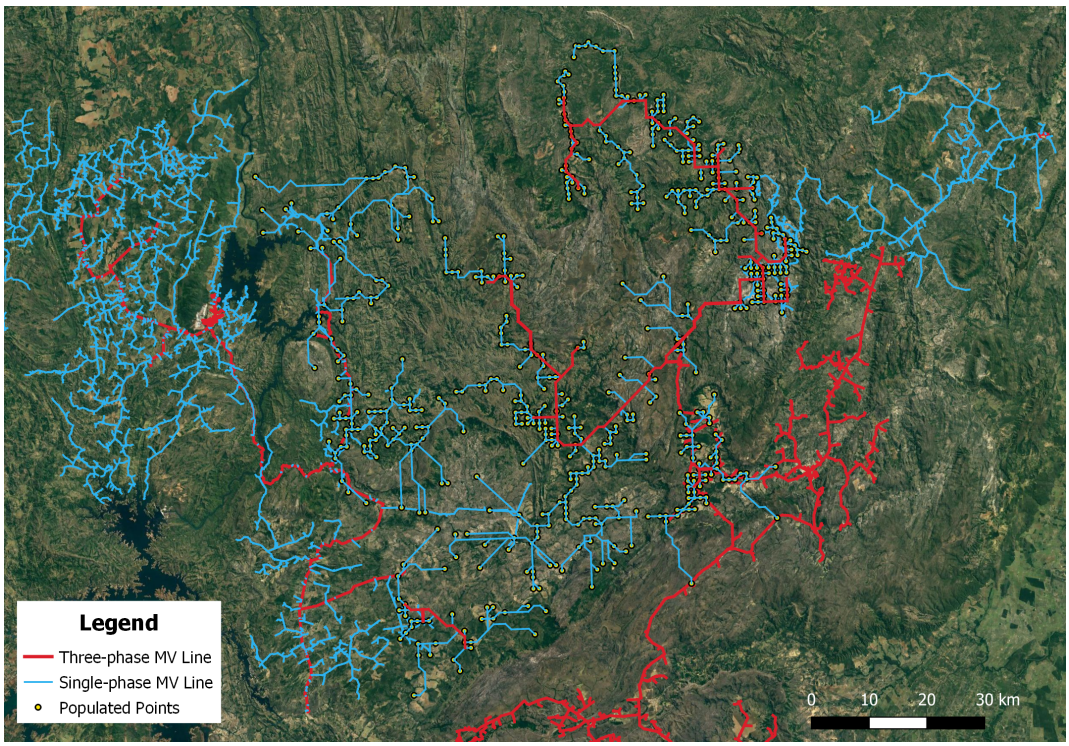

Figure 3: *Image of the distribution grid after the expansion*

Riassunto

L'accesso all'elettricità è considerato una necessità per la dignità umana e molti evidenziano come questo dovrebbe essere un diritto umano fondamentale. Dal 2015 le Nazioni Unite hanno adottato l'accesso all'energia accessibile, affidabile e sostenibile come uno dei 17 obiettivi di sviluppo sostenibile, che sono un invito urgente all'azione da parte di tutti i paesi in un partenariato globale. Ciò ha portato a uno sforzo crescente con circa 120 milioni di persone che ottengono l'accesso all'elettricità ogni anno, ma all'anno 2018 ci sono ancora 860 milioni da raggiungere (*IEA*, 2019a). La maggior parte di questa popolazione si trova in remote aree rurali rurali, principalmente nell'Africa sub-Sahariana, dove i progetti di elettrificazione sono spesso proibitivi in termini di costi. A differenza dell'elettrificazione urbana, che è generalmente attentamente pianificata per ottenere una maggiore affidabilità, l'elettrificazione rurale tende ad aspetti economici a causa degli elevati costi di investimento richiesti da questi progetti. Ciò richiede pertanto lo sviluppo di nuove strategie in grado di affrontare la specificità del problema. La strategia qui proposta si avvale del Geographic Information System (GIS) e dell'analisi del terreno per creare la migliore topologia di rete elettrica e usare i risultati in un caso di studio reale, con la collaborazione del *Enel GI&N*, nel comune di Cavalcante in una zona rurale in Brasile.

Questa valutazione richiede una profonda revisione della letteratura sulle strategie di elettrificazione rurale che sono state adottate in tutto il mondo. La forma più convenzionale di trasporto di elettricità, i sistemi trifase, è in costante evoluzione. Una delle innovazioni sviluppate negli ultimi anni è l'uso di trasformatori compatti che offrono una soluzione più economica per la fornitura di servizi ausiliari (*Orsini*, 2013). Grazie alla loro elevata versatilità in termini di livello di tensione, possono migliorare l'elettrificazione rurale consentendo la connessione tra carichi BT e linee HV e riducendo la lunghezza totale delle linee MT che sarebbero altrimenti necessarie per collegarsi alle sottostazioni HV / MV. Più adatti per applicazioni a bassa potenza, come nelle aree rurali, sono stati ampiamente utilizzati anche i sistemi monofase. Riducendo il numero di cavi, questi sistemi sono spesso i più economici. Strategie come il Single Wire Earth Return (SWER), che utilizza un singolo conduttore monofase MT e la terra come filo di terra di ritorno, sono in grado di ridurre fino al 30% dei costi totali di capitale (*Karhammer et al.*, 2006). Altre strategie meno convenzionali, come gli Shielded Wire Systems (SWS), utilizzano i conduttori schermati delle torri di trasmissione ad alta tensione come conduttori e creano un sistema MT in grado di fornire carichi locali (*Iliceto*, 2016). Infine, il boom di Distributed Generation (DG) e Fonti di energia rinnovabile (RES) negli ultimi decenni, ha fornito una soluzione off-grid economica per l'elettrificazione oltre alla connessione a una rete elettrica esistente.

Oltre a decidere quale tipo di sistema elettrico verrà utilizzato, un progetto di elettrificazione rurale deve trovare la migliore topologia di rete per l'area considerata. Quale percorso devono seguire le linee elettriche, quali terreni deve evitare e come raggruppare i carichi nel modo più efficiente, sono alcune delle domande a cui rispondere. Lo strumento GIS per l'elettrificazione (GISEle) è stato sviluppato con l'obiettivo di migliorare la pianificazione dell'elettrificazione rurale nei paesi in via di sviluppo (*Carnovali and Edeme*, 2019). È uno strumento open source basato su Python che utilizza GIS e analisi del terreno per modellare l'area in studio, raggruppa i carichi utilizzando un algoritmo di clustering basato sulla densità chiamato DBSCAN e quindi utilizza la teoria dei grafi per trovare la topologia della rete elettrica meno costosa che può collega tutte le persone nella zona.

La metodologia adottata GISEle consiste in tre fasi principali: analisi dei dati, clustering e routing della rete di distribuzione. Durante la fase iniziale di raccolta e analisi dei dati, vengono creati set di dati GIS per mappare correttamente diverse informazioni sull'area da elettrificare. Alcune di queste informazioni sono: densità di popolazione, altitudine, pendenza del terreno e strade. Sono tutti elaborati utilizzando una strategia di ponderazione che traduce l'aspetto topologico del terreno in difficoltà nella distribuzione della linea. Quindi, DBSCAN viene utilizzato per aggregare strategicamente gruppi di persone in piccole aree chiamate cluster. L'output è un numero impostato di cluster, che copre parzialmente l'area iniziale considerata, in cui viene eseguito l'algoritmo di routing della griglia. Infine, GISEle utilizza il concetto di albero Steiner per creare una topologia di rete che collega tutte le persone aggregate in ciascun cluster e, se necessario, utilizza l'algoritmo di Dijkstra per connettere ciascuna griglia del cluster a una rete di distribuzione esistente.

Questo contributo di tesi si pone come obiettivo il miglioramento delle funzionalità dello strumento GISEle e l'applicazione di detto Strumento su un caso di elettrificazione di un area rurale, con il supporto della società partner Enel Global Infrastructure and Networks S.r.l (*Enel GI&N*). La versione originaria del codice GISele è stata considerata troppo astratta nella ottimizzazione della topologia di rete, ci si è quindi posti l'obiettivo di rappresentare meglio la struttura gerarchica di una rete elettrica che di solito è composta da un ramo principale ad alta potenza che collega aree densamente popolate e le sue derivazioni a bassa potenza (chiamate connessioni laterali) che collegano aree popolate sparse. La figura 4 rappresenta uno schizzo della topologia standard (a) e della topologia proposta (b). Per ottenere questa topologia più realistica è stata sviluppata una procedura in due fasi che utilizza due diverse griglie di risoluzione. Utilizzando una griglia di punti a risoluzione più bassa, la popolazione può essere aggregata e l'algoritmo di routing della griglia prenderà in considerazione solo le aree densamente popolate, creando il ramo principale. Quindi, utilizzando una griglia di punti a risoluzione più elevata, viene utilizzata una strategia di riassegnazione del peso in modo che l'albero di Steiner creato sfrutti la presenza del ramo principale, creando così i derivazioni.

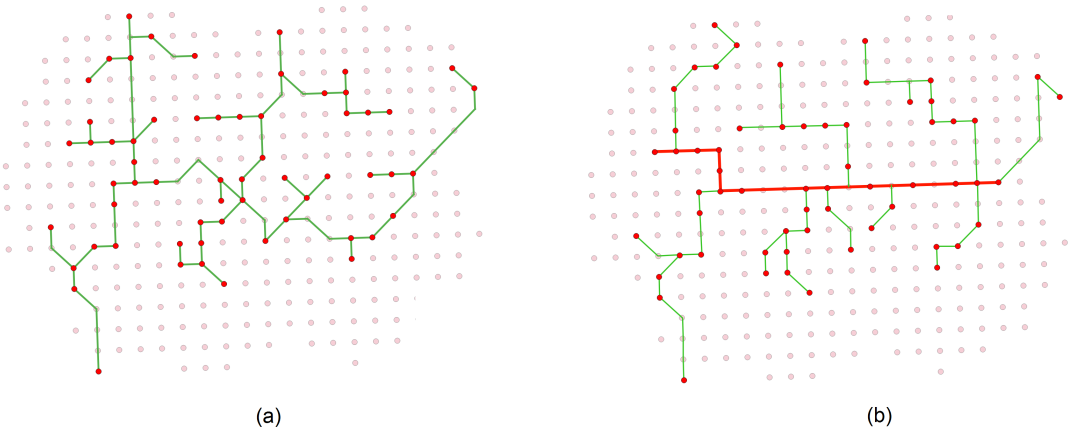

Figure 4: *Esempio di (a) una semplice topologia a basso costo e (b) una topologia a basso costo considerando una struttura gerarchica composta da una dorsali e derivazioni.*

Utilizzando questo approccio, ciascuno di questi feeder laterali può essere identificato e di conseguenza dimensionato in base alla potenza di picco dei carichi che forniscono. Oltre a ciò, sono stati apportati altri miglioramenti a GISEle per migliorarne le capacità: tipizzazione e assegnazione della sottostazione, sensibilità del cluster, fusione del cluster e interfaccia uomo-macchina sono tra gli elementi che sono stati migliorati. Inoltre, è stata sviluppata una nuova routine Python per migliorare la connessione alla sottostazione. Valuta il costo di ogni connessione e sceglie quella meno costosa consentendo a ciascun gruppo di carichi (cluster) di condividere la stessa sottostazione MT. Il nuovo approccio proposto per GISEle è stato utilizzato in un caso di studio reale in Brasile. Gli obiettivi principali di questo lavoro di tesi, che sono stati decisi insieme alla società partner Enel GI&N, possono essere sintetizzati come:

- Creare reti elettriche che collegano le persone nel comune di Cavalcante in Brasile, consentendo di aumentare il numero di clienti che possono essere alimentati con soluzioni on-grid e ponendo le basi per un futuro confronto di trade-off con soluzioni off-grid;
- Collegare questa rete alla rete di distribuzione esistente presentata in figura 5;
- Generare una rappresentazione topologica basata sul GIS degli elementi sopra menzionati;
- Dimensionare i cavi e quantificare i costi.

I risultati di questa analisi di elettrificazione rurale, che si concentrano solo sull'aspetto topologico e non prendono in considerazione altri fattori importanti come l'affidabilità del sistema e la qualità del servizio, suggeriscono che è possibile un'espansione della rete di distribuzione che raggiunge il 100% della popolazione locale. Il nuovo approccio principale per dorsali e derivazioni sviluppato, è riuscito a ridurre fino al 47% del costo totale dell'investimento nella distribuzione in linea rispetto all'approccio GISEle iniziale. La lunghezza totale delle linee necessarie per raggiungere l'elettrificazione al 100% era di 1635 km, con un costo di investimento totale di 33,93 milioni di euro. Questa riduzione dei costi suggerisce che una strategia di elettrificazione ottimizzata attraverso un migliore instradamento potrebbe spostare il punto di equilibrio

tra soluzioni on-grid e off-grid come l'uso di microgrid e la generazione di PV. Questi costi sono correlati solo alla distribuzione della linea, altri costi come i trasformatori MT / BT e le apparecchiature di protezione non sono stati considerati.

Il costo per persona collegato è passato da 5212 euro nell'approccio standard a 2785 euro utilizzando le dorsali e derivazioni. Se si considerano le cliente, l'approccio standard ha portato a 17199 euro per cliente connesso, mentre l'approccio principale di dorsali e derivazioni 9190 euro. Esiste un punto di ottimo tra la percentuale di persone connesse e il costo per persona connessa che è stato valutato nell'ambito di questo lavoro di tesi; inoltre, su richiesta del DSO, è stato valutato anche il costo necessario al raggiungimento dell'elettrificazione al 100%. I risultati confermano non solo la tecnicità dell'approccio proposto, dall'ottimizzazione dei costi raggiunta, ma anche l'aspetto topologico della rete creata da GISEle, che è simile alla rete di distribuzione MT esistente. Le figure 5 e 6 presentano una visione d'insieme dell'espansione della rete di distribuzione raggiunta da questo lavoro di tesi, in cui è possibile verificare questa somiglianza topologica.

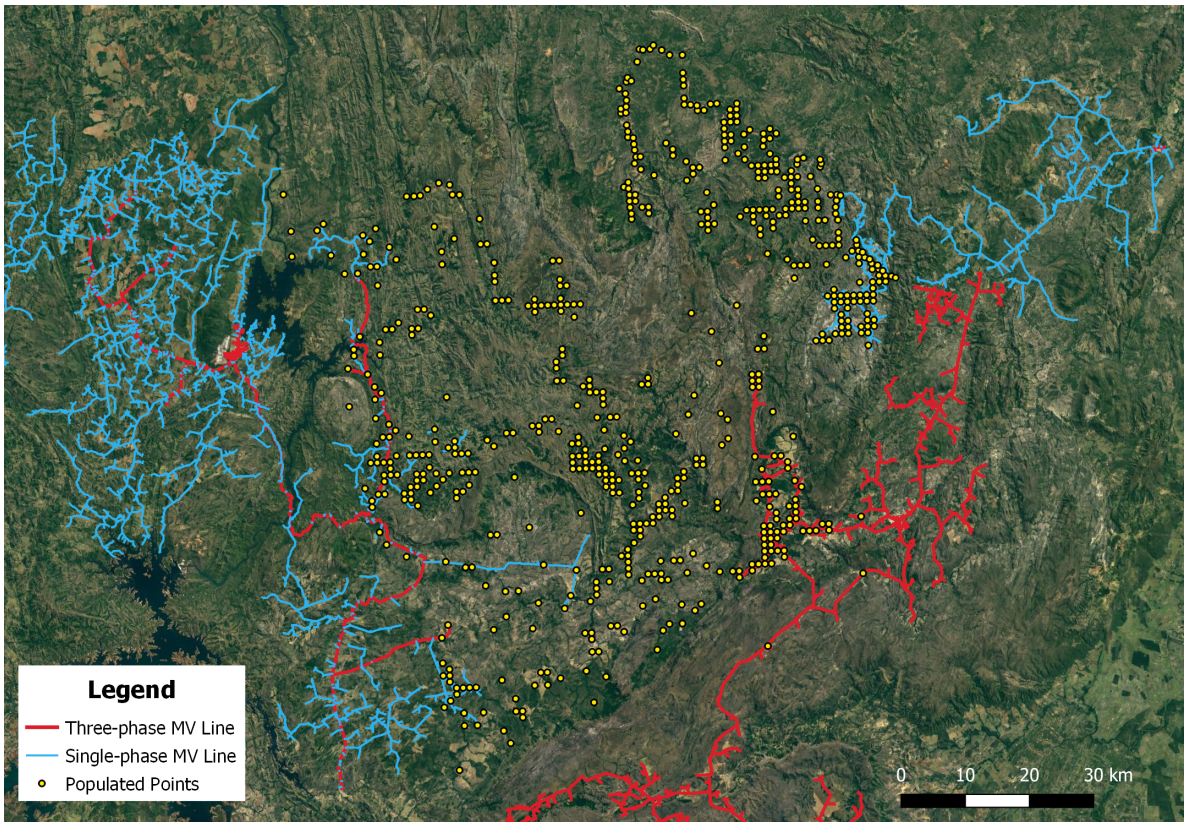

Figure 5: *Immagine della rete di distribuzione prima dell'espansione*

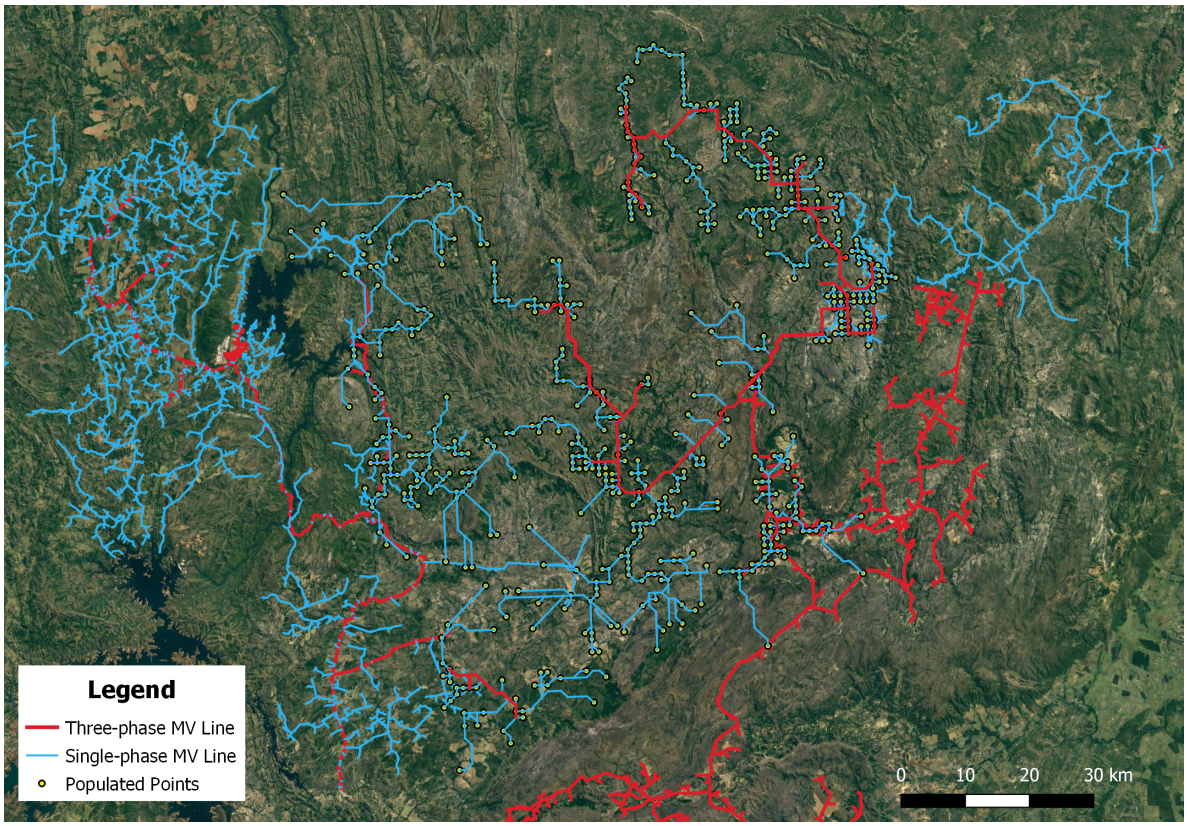

Figure 6: *Immagine della rete di distribuzione dopo dell'espansione*

Contents

List of Figures

List of Tables

1 Introduction

By the year 2018 the number of people without access to electricity was 860 million, from which around 600 million were in the sub-Saharan Africa zone (*IEA*, 2019a). In a world of ever-increasing inequality, the disparities between the so-called developed countries and the developing ones are undeniable. Electricity and energy consumption are no exception. In USA, the state of California alone uses more energy powering TV's and warming swimming pools than entire countries in Africa (*Moss*, 2019). With the introduction of new high energy demanding technologies such as electric vehicles and cryptocurrencies, the energy consumption tends to only increase with electricity playing a major role. The development of Renewable Energy Sources (RES) and new electrification strategies are crucial to achieve not only a worldwide better living standard but also to prevent the global collapse in face of the imminent effects of climate change.

1.1 Access to electricity: global outlook

The access to electricity is considered a necessity to human dignity and many argue that it is a basic human right. It is known that electricity is the main propellant of a country's development, and has been intrinsically related to economic growth (*WorldBank*, 2018). China is a clear example of a country that had a big economic growth in the last decades, which translates into the huge increase in the country's electricity consumption per capita that went from 998 kilowatt-hours (kWh) in 2000 to 4905 kWh in 2018. In 1979 the Convention on the Elimination of All Forms of Discrimination against Women (CEDAW) under Article 14 stated that state parties should *"take all appropriate measures to eliminate discrimination against women in rural areas ... and, in particular, shall ensure to such women the right ... to enjoy adequate living conditions, particularly in relation to ... electricity"* (*Tully*, 2006). More recently in 2015 the adoption of the new United Nations(UN) Sustainable Development Goals (SDG's) marked a new level of political recognition of the importance of energy to development. For the first time this included a target to ensure to everyone access to affordable, reliable and sustainable energy, also known as Sustainable Development

Goal 7 (SDG7).

Since then, the number of people gaining access to electricity has been around 120 million per year. Even though the vast majority of people without access to electricity are in the sub-Saharan Africa, South Asian countries such as Indian and Pakistan have over 40 million each. The overall outlook of access to electricity in the world is shown in figure 1.1.

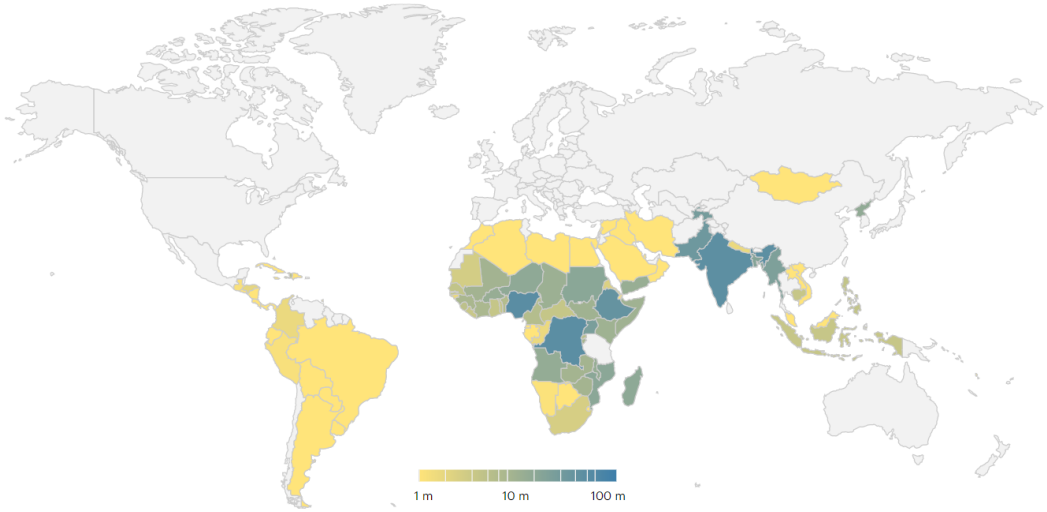

Figure 1.1: *People without access to electricity by 2018 (IEA, 2019a)*

In Latin America the countries Colombia, Peru and Guatemala all have over 1 million people who could gain access to electricity while Brazil have 0.4 million. Most of this population lives in sparse rural areas far from the urban centers where usually the cost for electrification is prohibitive. According to the World Energy Outlook 2019 (*IEA*, 2019b), urban areas have an electricity access ratio of 96% compared to only 79% of rural areas. Providing a better quality of life for people in rural areas through rural electrification could also help to mitigate other problems, such as the effects of the rural exodus. Figure 1.2 presents the ratio of access to electricity between these two areas, rural and urban, for a variety of countries. Countries in the top right have a high rural and urban electrification ratio while countries in bottom left lack in providing electricity access, therefore it is not a surprise that industrialized countries with higher Human Development Index (HDI) belong to the former category.

1.2 Electrification strategies

After discussing the importance of rural electrification and the overall access to electricity, it is important to point out solutions to achieve that goal. Historically the main strategy to bring electricity to sparse rural areas has been the grid expansion. National grid expansions requires high capital investments, notably in rural environments where many terrain obstacles are present and long lines have to be deployed to connect significantly light loads. In its base the problem of rural electrification is

financially not attractive, therefore developing good strategies for grid extension is necessary.

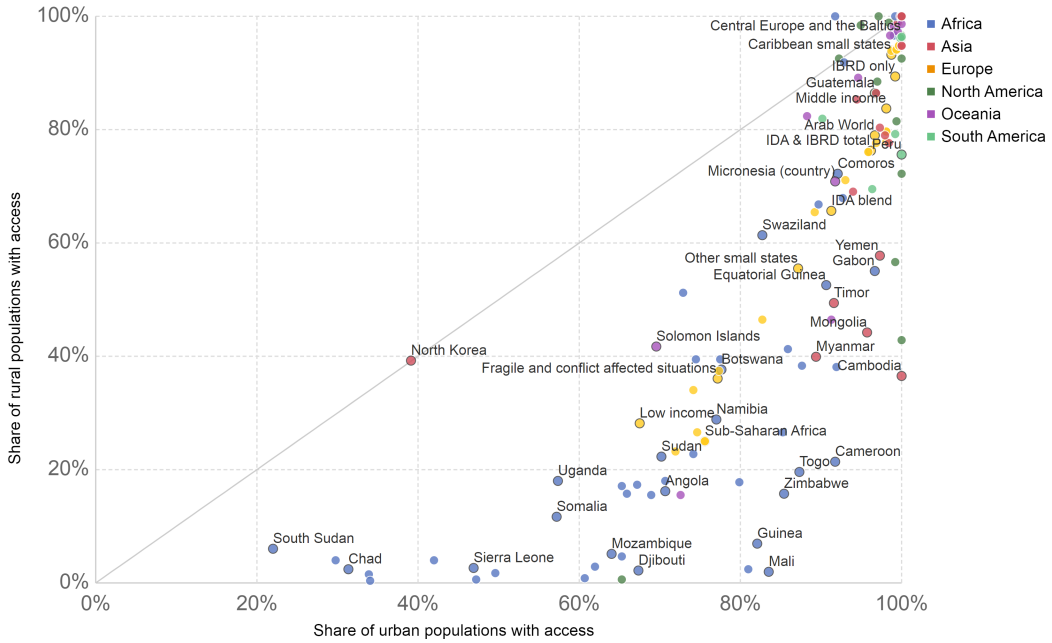

Figure 1.2: *Share of rural population with electricity access vs share of total population with electricity access (IEA, 2019b)*

The development of RES and the new possibilities that came together with Distributed Generation (DG) provided new solutions such as integrated microgrids and off-grid systems that can rely on other energy sources besides diesel generators. In 2018 Electricity generation from renewables increased by 450 TWh, which accounts for 7% when compared to the previous year, reaching for more than a quarter of total power generation. Growth in output from solar photovoltaic (PV), wind and hydro accounted for 90% of this increase. Around 180 GW of new renewable power capacity was added in 2018 only (*IEA*, 2019b). For a sustainable development scenario in the future, additional measures to motivate investment in renewable-based electricity will push the share of RES to two-thirds of electricity generation output and up to 37% of final energy consumption by the end of 2040.

The constant increase of RES in the world's electricity production requires System Operators (SO) from the whole world to adapt to this new form of generation and the effects of it . Care must be taken in order for the electrification process to not cause a detriment in the electric power system reliability. Challenges such as the bi-directional power flow, an effective demand response and a economical viable energy storage deployment have to be overcome in order to unlock the full potential of renewable solutions, and allow society to reach the sustainable development scenario for 2040. To help with that, new tools for planning power systems have been studied. As an example the usage of the Geographic Information System (GIS) for mapping and territorial analysis has been proved to be a powerful instrument in helping countries to develop their national grids (*Kaijuka*, 2007). Also new software have been developed in order to enhance the capabilities of load estimation

and sizing of generators, identifying in great detail the best combinations of energy technologies to be installed based on several criteria. To compare different energy generation solutions, generally an index called Levelized Cost of Energy (LCOE) is used, which is defined as the ratio between the total costs and the total energy production over the lifetime of the generation technology.

Aiming to deliver a tool capable of managing GIS data and provide the best electric grid design for a given area, the GIS for rural Electrification (GISEle) application was developed in 2019 (*Carnovali and Edeme*, 2019). GISEle was initially designed for:

- Collect morphological, social and energy data from the target area;
- Identify load centers by means of cluster analysis;
- Model the least cost electric grid interconnecting all the entities constituting the energy system, with an adequate estimation of costs and length;
- Trace the optimal path for the electric connection of the cluster with the national grid;
- Build the load profile of the community and design the energy generation system necessary to supply it;
- Present the resulting electric grid topology on a visual geo-referenced map.

The goal is to help government planners and off-grid electricity system entrepreneurs to make better decisions about how to plan and implement electrification efforts. It was firstly used in a case study in the area of Namanjavira in Mozambique, showing promising results in creating grid designs and evaluating energy resources. The objective of this thesis is to enhance GISEle, perform a different case study in Brazil and apply its results to real-case rural electrification examples performed by the company, and partner of this thesis project, Enel Global Infrastructure and Networks S.r.l (*Enel GI&N*). The idea is to better represent the hierarchical structure of an electric network that is usually composed by a high power main branch that connects densely populated areas, and its low power derivations (called collaterals) that connects sparse populated areas. Collaborating with Distribution System Operators (DSO) in four states of Brazil, Enel GI&N proposed a case study in the rural area in the municipality of Cavalcante where the connection of few customers very distant from the existing network is in nearing completion. GISEle application has been since adapted and improved to fulfill the demands of this new scenario and achieve an even more realistic result using technical data inputs provided by Enel GI&N.

1.3 Outline of the thesis

The thesis is divided in seven chapters and here is a brief summary of the next ones:

Chapter 2 performs a literature review of the wide spectre of solutions for rural electrification in the last decades. It starts describing the most consolidate power system scheme with three-phase systems and pointing the newest technologies that

have been developed in the market. Other unconventional solutions are also analysed such as single-phase systems and shielded wire systems. In the end, off-grid solutions are analysed and cost comparison studies are shown.

Chapter 3 describes the state-of-the-art on spatial analysis, which are the elements behind the GISEle development. These elements include a detailed description of the history and the basis of graph theory, an overview of the geographic information system, terrain modelling and spatial analysis and clustering techniques.

Chapter 4 precisely demonstrates how the previous elements were used in order to develop GISEle. Every phase from the starting gathering of data to the final substation connection and cost analysis are deeply detailed. This constitutes the methodology on which the case study is founded.

Chapter 5 presents the improvements that were developed within this thesis work and incorporated into GISEle. It also describes the context under which those improvements were decided, and how the collaboration with the partner company Enel GI&N was given.

Chapter **??** presents the real life case study of Cavalcante in which the methodology proposed will be tested. The choices behind the selection of the area, the context behind the Brazilian electrification policies and the main factors that drove the study are addressed here.

Chapter **??** reports the results obtained, highlighting the accomplished objectives and obstacles faced. It shows the distribution grid expansion proposed and cost necessary to give access to electricity to all people in the study area.

2

Literature Review on Rural Electrification Strategies

Rural electrification projects require specific techniques according to the situation. Differently from urban electrification, which is generally carefully planned to achieve increased reliability, rural electrification leans towards economical aspects due to the high investment costs these projects demand. The evaluation of the trade-off between investment costs and reliability or quality of service of rural electric system is not an easy task. Literature about this topic is vast as it is of crucial importance, being present since the beginning of the first electrification planning studies. The following sections describe the main techniques that have been proposed in the last decades and applied at different scales worldwide.

2.1 Three-phase systems

The first poly-phase electrical power systems date from late 1880s, since then this technology has been used in every power system, mostly due to its higher capacity of transfer power that can be up to 3 times higher than a single-phase system. These system are particularly efficient in transporting energy being widely used in high voltage (HV) and medium voltage applications. Nowadays the usage of high voltage direct current (HVDC) has been a new alternative for the conventional three-phase system, in cases of transporting energy through a very long distance and very high voltages up to 1100kV.

For the purpose of this work only medium voltage (MV) and low voltage (LV) systems will be analysed, particularly in the case of creating a derivation of an existent distribution network system to supply unattended loads. HV and MV networks are connected through substations. These are composed of several equipment such as disconnectors, circuit breakers and transformers, which the most important and expensive equipment. Figure 2.1 presents a standard HV/MV substation layout used in existing three-phase systems.

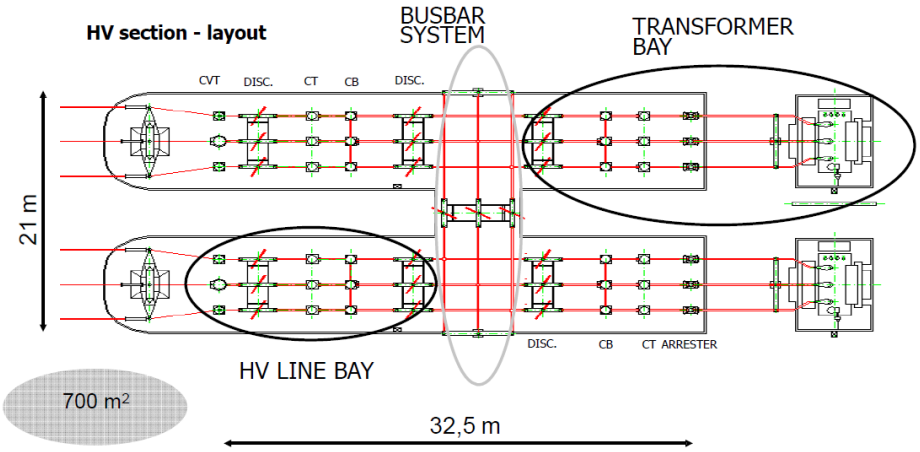

Figure 2.1: *Traditional HV/MV substation layout used by the distribution company Enel*

Costs of deploying a new medium voltage three-phase line

The costs of implementing lines depend on different factors:

- Length

It is natural that the cost of the line will be dependent on its length, the longer the line the higher the number of poles and the amount of material and work necessary for its deployment.

- Type

Overhead lines are usually preferably instead of underground cables which can be up to 50% more expensive than using bare conductors

- Voltage and power

The higher the voltage and power levels, the higher is the cost of insulation material for cables. The design and cost of the towers is also more expensive, since better insulators are necessary.

- Terrain

The location where the medium voltage line will be deployed strongly influences the costs of a medium voltage line. Auxiliary works such as excavations for underground cables and cleaning the path from natural obstacles for overhead lines, have a big impact on the work cost. Deploying lines along a paved road has significantly lower costs than through a dense forest or a mountain.

Table 2.1 summarizes these factors and provide the average costs of two Italian distribution companies, A2A and Enel.

Type of Line	Cost
Overhead cable line	min 45 (Al 35mm^2) - max 60 (Al 150mm^2)
Overhead cable line on rough ground	132 (Al 150mm^2)
Overhead bare conductor	89.3 (Al/Acc 150mm^2)
Underground cable on natural ground	55
Underground cable on paved road	min 80 - max 129.5 (Al 185mm^2)
Underground cable on rough and rocky ground	192.9 (Al 185mm^2)

Table 2.1: *Average cost of different types of medium voltage lines per unit length in k€/km (Orsini, 2013)*

As expected it can be seen that the type and power level of the line is a huge aspect regarding its costs, going up to 60000 €/km for overhead lines and 129500 €/km for underground cables.

Important to notice that these costs are related to the medium voltage line alone, assuming that it is a derivation of an existing line from the distribution network and does not consider the usage of other substations or circuit breaker, meaning that a fault in the new deployed line would cause part of the distribution system to go out of service. For this reason it is highly recommended that the derivation to the new medium voltage line derives from an existing substation instead of a line, where protection equipment and transformers can be implemented to separate electrically the new load to be supplied from the rest of the system. In this scenario additional costs must be considered.

Costs of conventional MV/LV transformer

Transformers are a consolidated technology in the market, consequently their price has been reducing along the decades as new different types of transformer technologies arise. These technologies mostly differ in the type of insulation used in their windings. This insulation can be of a liquid type, as oil transformers, or a dry type like resin transformers or vacuum.

Power [kW]	Resin Transformer	Dry Transformer	Oil Transformer
50	4,800	3,400	2,500
100	5,900	5,500	3,900
250	8,000	7,200	5,500
400	10,400	8,500	7,200

Table 2.2: *Price of a conventional 20 kV- 0.4kV transformer for different technologies in €*

These costs are related only to the installation of the transformer, but other costs must be taken into account when the life span of 30 years of a transformer is considered. Costs such as maintenance and losses can, at the end of the lifetime of these equipment, more than double the initial installation cost (*Orsini*, 2013).

Other solutions: TIP and T-PASS

In face of the high costs related to the deployment of a new conventional medium voltage line together with a transformer substation, new alternative solutions have been proposed in recent years. The *service voltage transformer* , named TIP, is a SF6 gas

insulated single-phase transformer that was developed by ABB Adda. This transformer has incredible versatility being able to connect to a wide range of voltage levels, from 72.5 kV to 550 kV, while having its secondary at either low voltage (230 V) or medium voltage (11.5 kV).

As the name suggests, its original purpose was to supply the auxiliary services of a conventional substation directly from the high voltage busbar, without having the necessity to connect the substation to external medium voltage lines. Studies show that the TIP solution is more economical if there are no near medium voltage lines to be connected to supply the auxiliary services. The breakeven distance has been shown to be about 2.7 km (*Orsini*, 2013). Even when the substation has a medium voltage section, where in this case only a conventional MV/LV transformer is necessary to supply the auxiliary services, the TIP solution offers a high reliability and reduction of the total amount of out-of-service hours. This possibility can be advantageous for distribution operators even though the cost of a TIP can be up to 20 times the cost of a traditional transformer

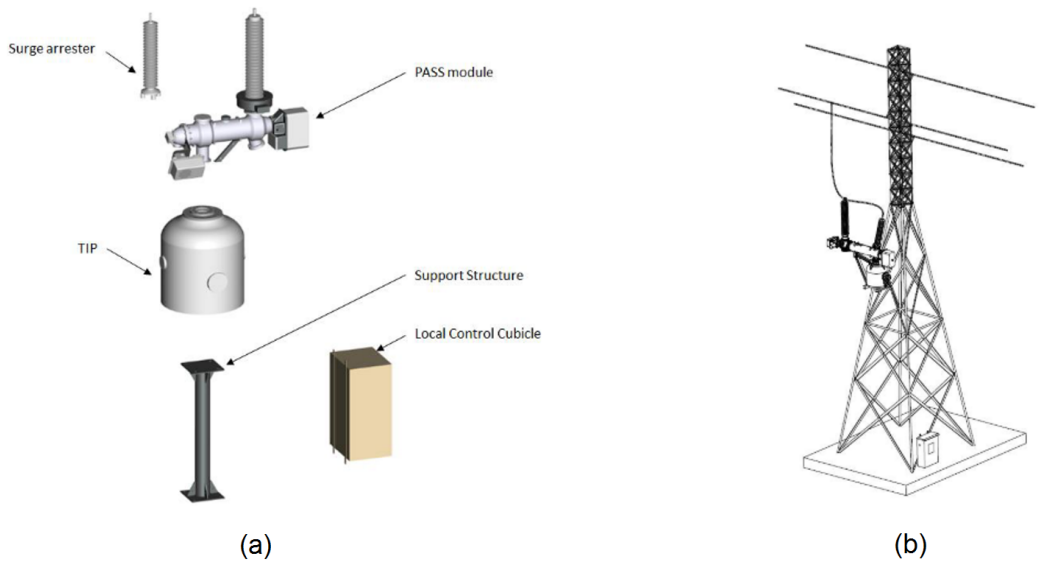

Figure 2.2: *Components of a T-PASS (a) and an example of a T-PASS connected to a HV tower (b)*

Making use of the incredible compactness of the TIP, a multi-functional T-PASS module was developed and it is composed of: the TIP transformer, a high voltage breaking and switching device (PASS), a surge arrester and a local control cubicle (LCC). Comparing the layouts of figure 2.3 and figure 2.1, it is evident that using the T-PASS makes the substation smaller and more compact. Also, this technology creates an attractive solution particularly for rural electrification purposes, since it allows an easy connection of low voltage network directly to a high voltage transmission line.

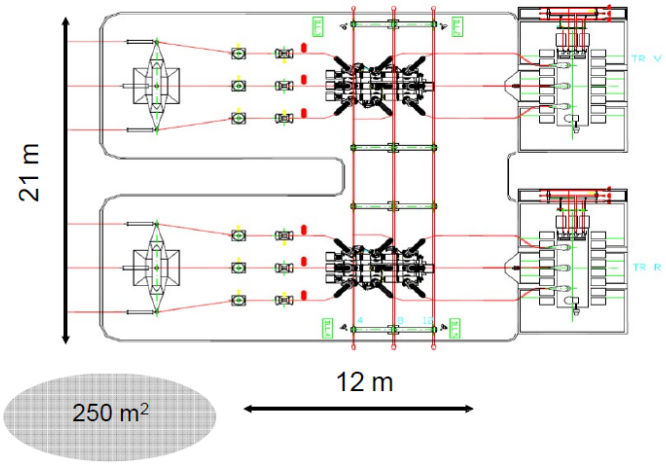

Figure 2.3: *HV-MV substation layout using the T-PASS*

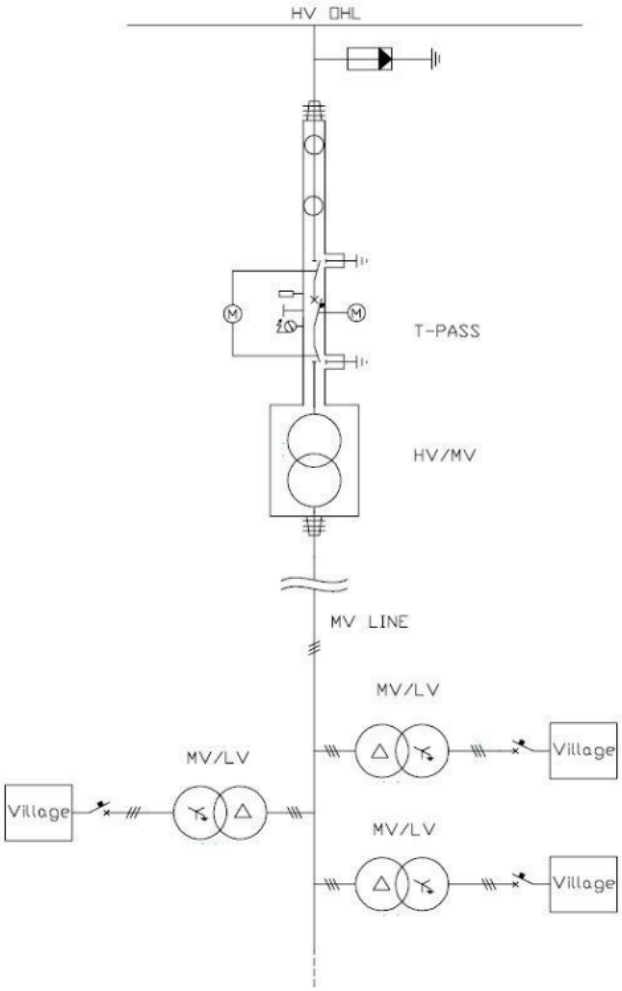

Figure 2.4: *Single diagram of a rural network supplied using a T-PASS configuration (Bongiorni and Civardi, 2010)*

Case study in Sudan

Bongiorni and Civardi (2010) describes an study made in a rural area of Sudan which compares the costs of three rural electrification options (dedicated MV line, diesel generators and RES) to the T-PASS solution. The rural area to be electrified has a population density of 50-100 people per km^2 covering a total of 90 km^2 area with 13 villages. The area is near to a HV transmission line but it is far from the nearest HV substation, over 70 km, making a suitable case study for the T-PASS solution. The annual consumption is estimated at 6750 MWh.

Dedicated MV distribution line

The standard solution will be composed of several equipment such as: HV/MV oil transformer, surge arresters, circuit breakers, current and voltage transformers and disconnector switches. The substation designed has an installed power of 5 MVA, with a distribution line voltage level of 11 kV three-phase. The total length of the lines is 72 km with a 150mm^2 conductor.

Renewable energy source

Covering all energy requirements the solution for supplying the area with renewable energy will be made strictly based on solar PV. Auxiliary equipment required for this solutions are: inverters, battery bank and a backup-only diesel generator. With this solution each village will be equipped with: 70 kW PV panels, 70 kW diesel generator with reduced fuel tank with lifespan of 6 years, 170 kWh battery bank with lifespan of 15 years.

T-PASS solution

The T-PASS solution will allow the villages to be connected to the nearest HV line, reducing the total length of MV lines required to supply the area. The installed power of this system will be of 1MVA with a distribution line length of 29 km at 11 kV voltage three-phase system. Alternatively an extra solution adding 70kW of solar PV generation together with the T-PASS solution was evaluated.

Cost analysis results

Taking as reference the lowest initial cost solution, the set of diesel generators, the net present value (NPV) of each proposed alternative was evaluated over 20 years. The yearly interest rate was assumed at 5% and the OM costs are assumed as a percentage of the initial investment, being 1.5% for the T-PASS and conventional MV lines and 2% for the PV.

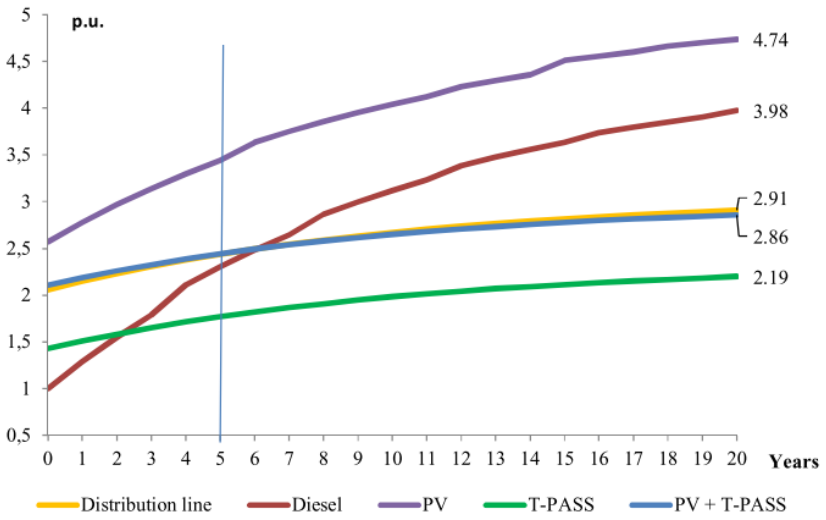

Figure 2.5: *NPV of each solution taking as base value the investment cost of diesel generators*

The results, summarized by the figure 2.5 show that due to the high operation costs of the diesel generator, mainly related to the fuel, this solution becomes more expensive compared to others even though it has the smallest initial capital investment. The most expensive alternative was the renewable solution, mostly due the high costs of the battery bank and operation cost of the diesel generator used. It has been concluded that the renewable energy solutions, which will be further discussed in section 2.4, are more financially competitive in isolated rural communities and in a low demand scenario. In the end the T-PASS offered the best solution for this application, having the lowest cost out of all possibilities after the second operating year, while having the second lowest initial capital cost.

2.2 Single-phase systems

Since three-phase systems main advantage is to transfer a high volume of energy, the rural electrification is one of the fewer cases where the usage of this systems are not the best economical solution. In the rural environment a specific scenario is present: low density of consumers, low power, long distance between consumers and low foreseeable demand increase. These aspects results in a case where the full potential of a three-phase may not be necessary and the burden of costs from using more wires overcome the advantages of the method. Distribution companies have reported that for a low demand (approximately 2 kVA per consumer) and low density (approximately 10 to 15 consumers) rural area, deploying three-phase lines are not economic viable (*CEPEL*, 2002).

Conventional single-phase systems

This system is the most widely used worldwide for connecting single-phase customers at the low voltage and medium voltage distribution network. In urban areas the distribution network is composed of a 4-wire system with the traditional three-phases available plus a neutral wire connected to the centre of the star of the sec-

ondary winding of the transformer at the secondary substation. In sparse areas and low density communities a single-phase system composed of only phase and neutral wires, named phase-to-neutral systems, can be deployed connecting all the customer to a single phase of the three-phase system. In any distribution system the balance of the three-phases must be preserved when connecting loads.

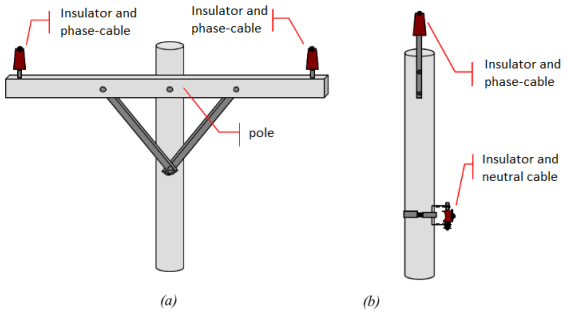

Figure 2.6: *Phase-to-phase (a) and phase-neutral (b) design of poles used in distribution systems. Adapted from Fandi (2013)*

This topology has a cost saving when compared to a typical three-phase system regarding the reduction of material involved: less wires, simple poles, less labour force. Also single-phase equipment tend to be cheaper than their three-phase counterparts. Summing all the cost savings it has been reported that a conventional 2-wire single-phase-to-neutral system, as shown in figure 2.6 (a), costs 40% to 60% less than a comparable three-phase system (*CEPEL*, 2002; *Bertollo*, 2008).

If there is a foreseeable increase of demand in the area to be supplied, where the upgrade to a traditional three-phase medium voltage system is required, other topologies can be deployed such as phase-to-phase systems. In this configuration in order to upgrade to a three-phase system the addition of only a single-phase cable is necessary. However it is more expensive compared to phase-to-neutral systems, since two MV insulators is necessary and the pole has a more complex structure as can be seen in the Figure 2.6 (a). The cost of single-phase-to-phase systems has been reported as 20% to 30% less of three-phase systems (*Bertollo*, 2008).

Single wire earth return (SWER)

The SWER technique consists in supplying power to small loads from the medium voltage grid using only one energized conductor and the earth as a return ground wire. It has been used to supply power to rural loads worldwide in different types of variations from South Africa to Russia, Brazil, New Zealand and Australia; especially due to its reduced investment and maintenance costs as well as fast and simple deployment of lines (*Bertollo*, 2008).Originally the SWER system was intended to supply mainly small single-phased loads, however due to the widespread of induction motors and other three-phase loads used in agriculture, as well as constantly increase in energy demand, new concerns rise and many studies are already evaluating the economical solutions of upgrading the traditional SWER systems, so it may be able to supply three-phase loads (*Bertollo*, 2008).

Also in the past the SWER systems were connected at the medium voltage level, which ranges from 10 kV to 40 kV, however with new built-in transformers (T-PASS) technology the possibility for connection at the high voltage grid is real. Studies with the T-PASS technology shows that new transformers can be connected to voltages up to 550 kV while still providing safe and high quality low voltage secondary connections (*Bertollo*, 2008). There are two main variations of SWER systems: direct connected to the MV grid and connected through an isolating transformer.

Direct connection SWER

In this variation each single-phase load is connected to a branch of the three-phase medium voltage grid. For that reason the presence of a ground connection at the substation is mandatory, which means that the transformer must be connected in a star grounded configuration at the secondary side. In this case the current flowing through the ground will be reflected to the distribution system, which may cause imbalances, reliability loss and safety issues if not properly planned. This issue is aggravated by the fact that in the direction connection SWER the sensitive earth protection cannot be implemented, since there is a permanent nominal current flowing through the ground. For this reason the grounding system, the study of soil resistivity and humidity will strongly influence the applicability and efficiency of this type of configuration. The simplicity of the method, however, results in the most economic results.

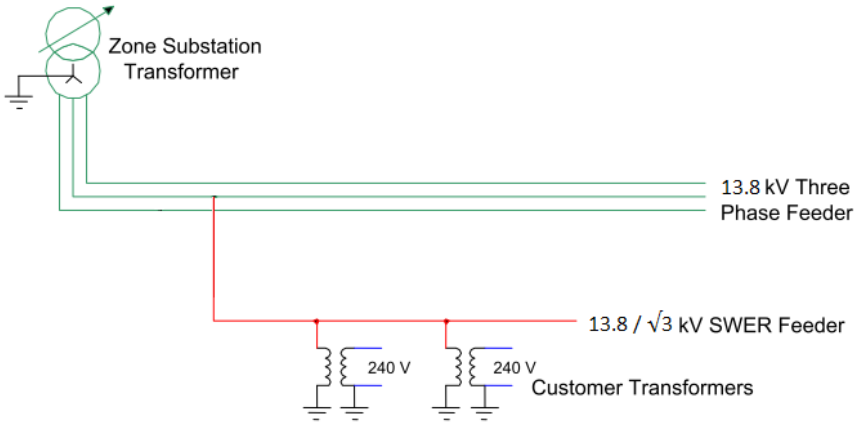

Figure 2.7: *Example of SWER configuration without isolating transformer used in Brazil. Adapted from Hosseinzadeh et al. (2011)*

SWER with isolating transformer

The isolating transformer is the most technically complex and expensive component in the SWER system, for that reason usually this method is mainly adopted to long distance grids. The concept is to isolate the medium voltage three-phase network to the SWER network, in a way that the isolating transformer is the one that supplies the grounded path to the loads. This removes the previous condition that the substation transformer must be in a star-grounded connection at the secondary side. From the many functions and advantages that the isolating transformer brings to the SWER system, some can be highlighted:

- Possibility to select a proper voltage level exclusively to the SWER network and different from the medium voltage grid. This could be used, for example, to increase the voltage level of the SWER grid in order to supply a bigger area.
- Restriction of the earth currents area to the area between the SWER distribution transformers and the supplying isolating transformer. Also helping to minimize the interference with open wire communications caused by earth currents.
- Limits the short-circuit current inside the SWER network while also allowing the usage of sensitive earth fault protection schemes on the medium voltage three-phase network. Without using the isolating transformer, the medium voltage feeder earth protection would detect the nominal earth currents of the SWER system as a permanent fault.
- Reduces the costs of the SWER distribution network transformers. Since the isolating transformer can provide voltage control, the distribution transformer can have a fixed tap instead of a more expensive variable tap configuration.

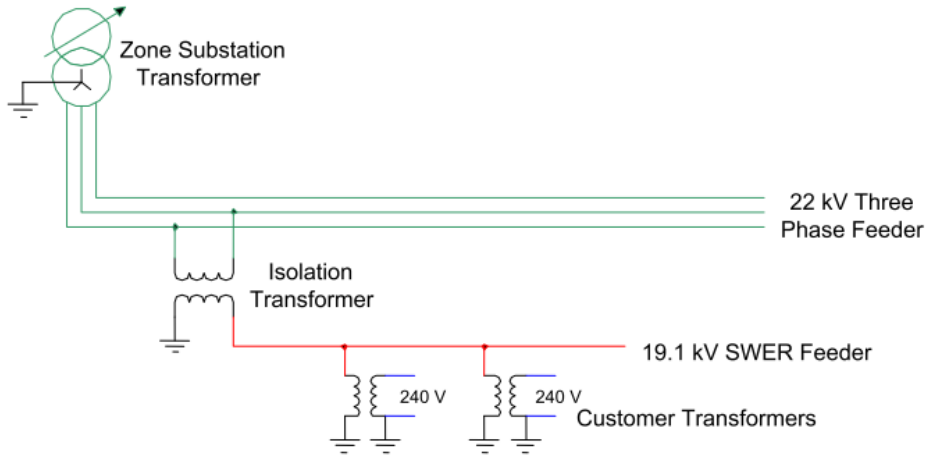

Figure 2.8: *Example of SWER configuration with isolating transformer used in Australia. Adapted from Hosseinzadeh et al. (2011)*

Challenges in using a SWER system:

Safety

Grounding is the most important aspect of any SWER system. It requires a very carefully planning of the grounding system in order for the whole environment be safe and reliable. Differently from traditional systems, where the earth is used mainly as an alternative path to occasional currents, in the SWER system there is a continuous load current. The main goal of the planned grounding system would be to provide good quality electricity to rural customers, using the earth as a stable conductor, while keeping the overall safety and reliability of the whole electrical power system.

The flow of current through the earth may result in dangerous voltage levels around the area which are usually described as touch potential and step potential.

Step potential being the voltage between the feet of a person standing on top of an energized surface, while the touch potential is voltage between the energized object and the feet of a person in contact with such object. In order to keep people and animals safe, those voltage levels should not be higher than 25V (*Brooking and Van Rensburg*, 1992). This value is a product of the load current and the earth resistance, which means that it is also a limitation to the amount of load that can be supplied. The lower the soil resistivity the higher the amount of load that can be connected to the SWER network. A maximum SWER capacity of 480 kVA limited by 25A at 19.1 kV is suggested, which can be delivered for up to 100km at a maximum voltage drop of 10% (*Karhammer et al.*, 2006). If loads or prospective loads within the next 10 years are likely to exceed this capacity, then a two-wire or three-wire option should be investigated. Keeping the soil resistivity, or the ground resistance, within the acceptable standards is one of the main attributes that will answer whether the SWER system implementation would be viable or not. In rare cases the costs of keeping a low soil resistivity or a complex grounding system will overcut the biggest advantage of using a SWER system in the first place: the economically attractiveness.

Adding to the complexity of the grounding system in SWER networks is the fact that the soil resistivity is not constant and may vary widely within short distances inside the same area. In general the soil resistivity depends on the type of soil and the amount of soil moisture. There is also a relation between the temperature and soil resistivity, which means that it also varies during the periods of the year. This relationship is important due to power losses in the form of heat to the earth, which implications could result into an increasing temperature of the soil, causing the earth to dry and hence increasing its resistivity. This creates an continuous event that may keep drying the earth while constantly increasing the soil resistivity which end up causing violations to the safety parameters (*Brooking and Van Rensburg*, 1992).

Voltage, current and load imbalances

Another concern while deploying SWER networks is the balance of the loads and its effects of the electrical power system. The supply of single-phase SWER loads from a three-phase system will eventually lead to negative sequence currents. When the isolating transformer is not present, those imbalances may be mitigated by a good load balance of each separated phase, which could increase the total amount of loads supplied. Since the isolating transformer is connected to two phases of the three-phase medium voltage network, the negative sequence currents are inevitable. Those imbalances are even more notable to motor loads in which it can lead to higher losses and reduction of the life expectancy of these equipment.

Voltage regulation

Together with the soil resistivity, another major limiting factor of SWER systems is the voltage rises due to the Ferranti effect. This effect is responsible for the rising in voltage levels along the transmission line in conditions of light loading. It occurs when the capacitive characteristics of the line become more predominant than the inductive one. This effect is particularly strong in very long lines feeding small loads, which is most likely the scenario of the rural areas the SWER system is meant to

supply. In the absence of any reactive compensation, the voltage levels at the end of the SWER line will always be higher than at the beginning.

The most widespread solution is the usage of shunt reactors which can be deployed along the line increasing the inductive compensation, however fixed reactors will contribute to the voltage fall in heavy load conditions resulting in even lower voltage levels on these scenarios. A controllable shunt that will be switched on the line during light loading and switched off the line during heavy loading could minimize the Ferranti effect impact. Others modern solutions such as the usage of distributed generation (DG) to provide voltage regulation and improve the voltage profile (*Kashem and Ledwich*, 2004), as well as the deployment of monitoring units (*Song et al.*, 2017) have been proposed.

Cost analysis and benefits

Studies all over the world have been showing that using a SWER network instead of a traditional three-phase distribution system is overall more economical. In Brazil, at specific scenarios, a direct connection SWER system can cost up to 10% of a traditional three-phase system (*Fandi*, 2013; *Bertollo*, 2008). Utilities in Australia and New Zealand have reported savings of 30% of the capital costs related to three-phase systems (*Karhammer et al.*, 2006). The main difference of those two approaches are the presence of the isolating transformer in the Australian SWER. According to *Bakkabulindi et al.* (2012) the addition of the cost of the isolating transformer makes the Australian SWER cost-efficient only for lines lengths longer than 8.5km. The Brazilian SWER being the cheapest method is cost-efficient for any line longer than 0.6km. For lines shorter than that, the cost savings of using only one conductor do not offset the disadvantages and challenges that these systems require.

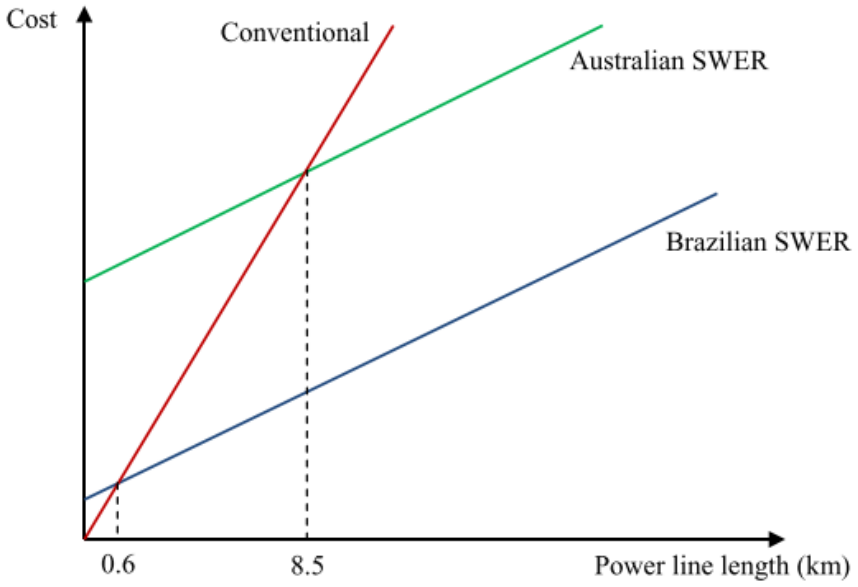

Figure 2.9: *Cost comparison and breakeven distance of conventional lines and SWER systems (Brooking and Van Rensburg, 1992)*

2.3 Shielded Wire Systems (SWS)

The SWS technique consists in insulating the shielded wires from the towers of high voltage transmission lines for a medium voltage operation and energize these conductors to supply the loads. Typically the SWS operates at medium voltage between 20kV and 34kV replacing an otherwise traditional MV three-phase distribution system. It was first proposed in the 1980s to be deployed in Ghana, where according to the report from the Energy Sector Management Assistance Program (ESMAP) (*Karhammer et al.*, 2006) the SWS system had about 526km of 161kV lines, serving up to ten thousand households being in commercial operation for over 15 years by the time the report was released in 2006.

Regarding the performance indexes it has been reported that the SWS experience in Ghana had better or the same results of a conventional supply. Other countries that deployed SWS were Laos, Ethiopia and Brazil. In Brazil the system was deployment in the north of the country providing energy to up to forty thousand residents (*Ramos et al.*, 2009) in a medium voltage SWS system of 34.5kV derived from a transmission line of 230kV. The system was in operation for more than 20 years and the results show that the usage of the SWS does not deteriorate the performance of the transmission line to withstand lighting strikes (*Ramos et al.*, 2018).

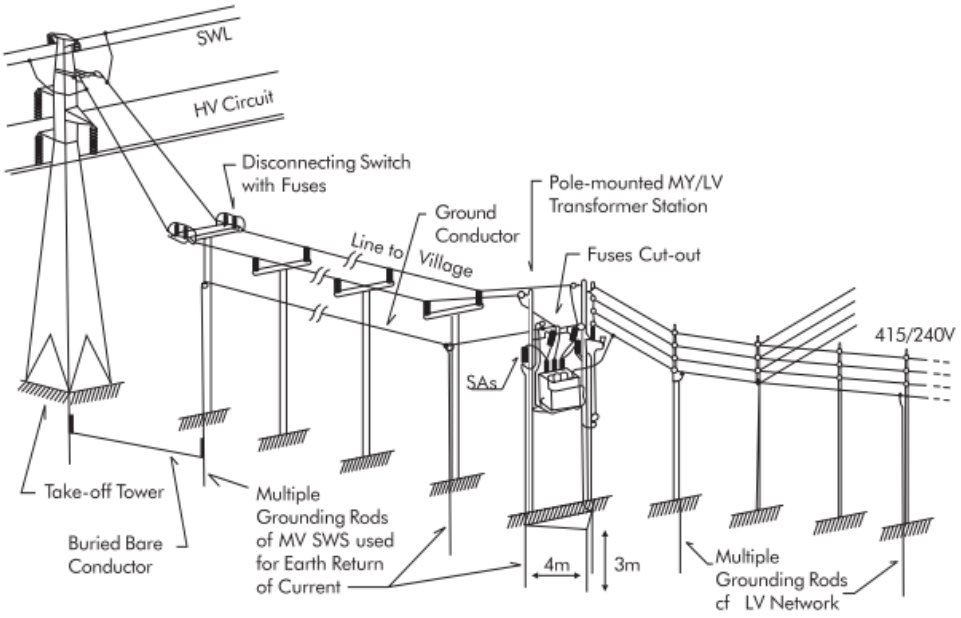

Figure 2.10: *Example of SWS proposed by Iliceto (2016)*

Different types of the schemes were developed by prof. Francesco Iliceto making the SWS adaptable for the various types of high voltage networks. In the most complete configuration it requires two shielded wires that will be energized at medium voltage and a ground wire as the third conductor. This creates a three-phase medium voltage grid that is capable of supply large three-phase loads such as induction mo-

tors of up to 200kW as experienced in the case of Brazil (*Iliceto*, 2016). If using the earth return conductor becomes a problem, a single-phase SWS using two shielded wires as conductors is also possible. On the other hand the most economical approach can be deployed using only a single shielded wire and the earth as a returning conductor to provide a single-phase medium voltage network, as in the SWER method.

Challenges of using SWS

Since most of the SWS also uses the earth as a conductor, being it a single-phase connected to one shielded wire or a three-phase network connected to two, many of the difficulties previously explained about the SWER systems are present. While the operation of SWS is simple and can be performed by regular distribution utility personnel, the planning of the SWS application is recommended to be made or at least supervised by professionals with experience in the area. In fact the unfamiliarity with the SWS technology caused failures during the commissioning in the Ghana experience. The load carrying capacity of the SWS, using a 76mm^2 ACSR shield wire is about of 9MW. For lines longer than 100km the load-carrying capacity is reduced to 4MW up to 3MW at the distance of 150km (*Iliceto*, 2016).

In the case of using two shielded wires to create a three-phase medium voltage distribution network, the rated voltage is resulted to be $\sqrt{3}$ times higher than a conventional line. This happens because in a SWS the phase-to-ground voltage is equal to the phase-to-phase voltage, since both shielded wires have rated voltage and the third conductor is connected to the ground. For this reason the equipment procurement and construction of the SWS using earth return as a three-phase supply requires higher attention to this effect, which is not present in conventional distribution.

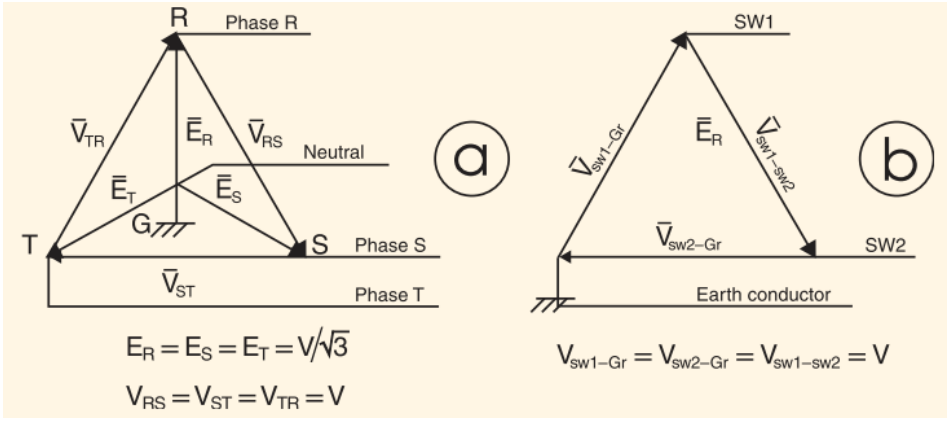

Figure 2.11: *Phasor diagram of a conventional three-phase medium voltage distribution system (a) and a three-phase system using SWS (b) (Iliceto, 2016)*

Cost analysis and benefits

The main benefit of using a SWS is to make a better use of an existent high voltage transmission line, using its shielded wires instead of building a complete new

medium voltage grid. By having this dual use of the shield wire, the cost of sub-transmission towers, conductors, grounding mats can be completely avoided. For this reason in comparison to the cost of an equivalent medium voltage line, the SWS can save up to 85% of the total costs (*Karhammer et al.*, 2006). However since it uses the same low voltage distribution system, there is no cost difference in the construction of the low voltage network. There is also an environmental benefit in avoiding the construction of new transmission lines, which depending on the area may require deforestation (*Ramos et al.*, 2009).

Usage of capacitive coupling in shield wires

Another way of using the shielded wires as an alternative medium voltage distribution grid is energizing these cables with an induced voltage defined by the capacitive coupling. The capacitive coupling effect is particularly strong in extra high voltage transmission lines (above 500kV), in which the energy stored in the shielded wires could be used to feed small off-grid communities nearby. This energy stored is in fact the energy loss of the transmission line due to natural capacitive behaviour (*Huertas and Tavares*, 2019). Since the energy used to supply the rural loads does not come from the main transmission system, the transmission line loading is unaffected.

The capacitances are dependent on the transmission line geometry, and the induced voltage depends on the transmission line rated voltage. Regarding the length of the shielded wire, it does not influence on the induced voltage but it is an important parameter to compute the maximum extractible power. For this reason transposition towers must be avoided. It has been shown that a single insulated shield wire of 100km was identified as the best configuration to feed a hypothetical village load of 500kW using an induced voltage of 28kV derived from a 500kV transmission line in Brazil (*Huertas and Tavares*, 2016).

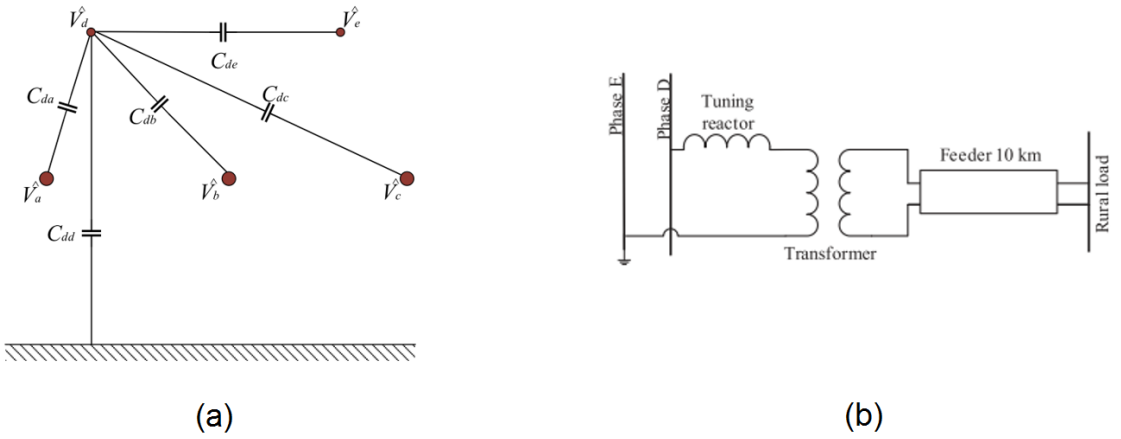

Figure 2.12: *Capacitance arrangement on a transmission line using SWS (a) and an example of feeder configuration using SWS (Iliceto, 2016)*

In order to achieve the maximum amount of power from the capacitive coupling as well as reducing the effect of the admittances, a tuning reactor is used to create a

resonant circuit that will be then connected to a feeder to supply the loads.

The main drawbacks of using this technique is that it requires precisely specific conditions in order for it to be implemented. Also as of today it has not been implemented in real case scenarios and even though the studies are promising, they are based on simulations and lack a real life experimentation. For this reason the capacitive coupling method have not been financially evaluated, despite the fact that it provide an alternative to medium voltage lines, the actual cost savings have not been estimated.

2.4 Off-grid solutions

Another alternative for the electrification of rural areas that is becoming more popular in recent years is the microgrid solution. A microgrid can be defined as a group of interconnected loads and distributed energy resources, with defined electrical boundaries, that form a local electric power system which could reach up distribution voltage levels. This entity acts as a single controllable unit and is able to operate in either grid-connected or in island mode (off-grid).

In the past, most off-grid systems were stand-alone, usually to supply a single household, and supplied by diesel generators. Today however, the huge widespread of renewable energy sources, especially solar photovoltaic and wind, is a result of a greatly decrease in the manufacturing costs of these technologies, to the point that small off-grid solutions with renewable energy sources can be technically feasible and financially competitive.

Microgrid topologies

There are many classes of microgrids, aiming at different types of problems and offering different solutions. For the purpose of electrification of rural areas usually a microgrid will be composed of a combination of the follow equipment: PV solar panels, wind turbines, Electrical Energy Storage Systems (EESS), diesel generator and converters.

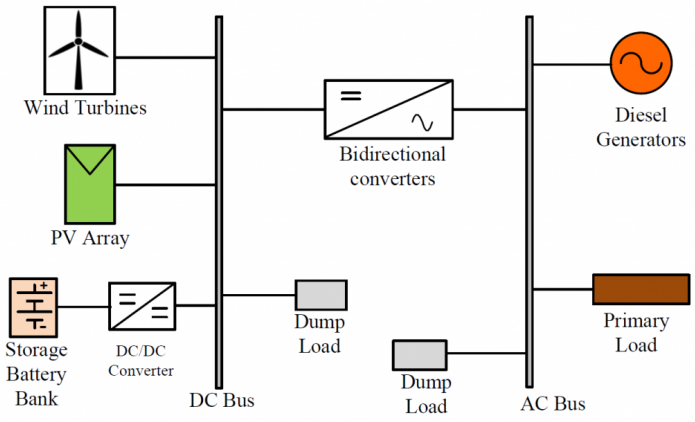

Figure 2.13: *Example of a standard microgrid design used for rural electrification*

The overall topology will mostly depend on application and presence of renewable resources available. The rural microgrid can be classified based on different characteristics:

- Grid connected or isolated

If grid connected, the microgrid will be a part of the distribution systems which means that the reliability, quality of service and security of the microgrid will have the same standards as the conventional system. However it does not provide much of cost reduction, since the grid has to be deployed nonetheless and microgrids requires special attention at their protection systems to avoid damages that can be caused by reasons such as internal faults or unwanted islanding. This results in better protection equipment that can concur in more costs. Isolated microgrids can offer cheap energy where the grid connection might be impossible or expensive. The main challenge is to create a reliable system that supplies the local loads with good quality of service, since all the regulation, protection, and demand control will be provided by the microgrid itself. Having a local controller that can manage the local generators together with optimizing the usage of the storage system has been a new object of study in academia for years now and some pilot projects have been proposed in under development countries such as Malaysia (*Fahmi et al.*, 2014).

- With or without EESS

The EESS can provide an alternative to the diesel generator in the time of the day where the renewable source is not enough to supply the load. With a proper coordination the EESS can secure that the diesel generator operates always at maximum efficiency, saving maintenance and fuel costs. Another advantage of using storage systems in off-grid applications is that they mitigate the intermittency of the renewable energy, reducing the frequency and voltage variation that are caused by these sources. In on-grid application the usage of EESS is usually optional, since these services can be provided by the main distribution grid.

- With or without a DC bus available

In recent studies, more and more microgrids are being equipped with a DC bus available for connection or in some cases without a AC bus at all. *Taufik* (2014) proposes an example of a house project supplied entirely based on different DC generators.

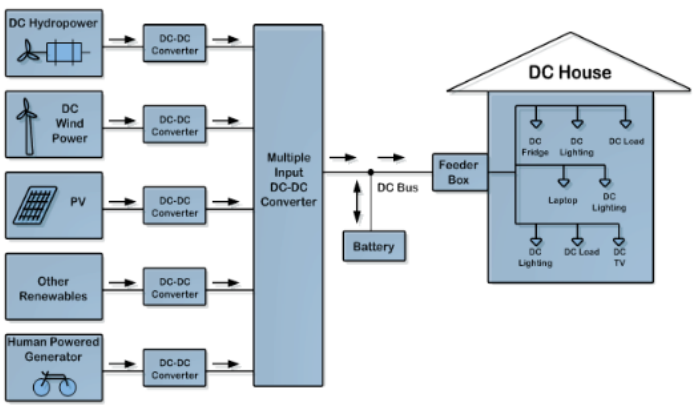

Figure 2.14: *Example of a DC house proposed by Taufik (2014)*

Figure 2.14 shows the house project proposed, that uses multiples DC-DC converters and a battery to control voltage at the DC bus. However, this approach is aimed to individual stand-alone households and not so useful for large rural communities.

Challenges of designing an off-grid microgrid

With all the different types of topology the design of a proper microgrid has to be carefully studied and adapted case by case. In doing so many factors must be taken into account when planning this systems. The main goal of an off-grid solution is to provide energy to the local loads by its own generators, independently to the main grid. This means that the demand and generation must match at all times, which may not be an easy task. Since the capital costs of generators depends on its size, and this cost will influence in the final price of energy, the risk of over-sizing exist and must be avoided. On the other hand reducing the size to achieve a better price may deteriorate the reliability of the power supply, having a risk of under sizing which should also be avoided. In general there is a trade-off between the price of energy that the microgrid will be able to supply and the power quality that must be carefully planned.

When using conventional controllable technologies, such as the diesel generator, the demand curve can be easily match by using more fuel, however the renewable energy sources are non-programmable and that is the reason the storage system is so important and hybrid system are preferable for off-grid applications. A real case example is a microgrid proposed by the Enel group in a pilot project for rural electrification in Vereda Altoredondo in Colombia. No external grid is present and an internal radial low voltage grid was designed to attend few customers, around 40, in a wide area of over 1km^2. The daily operation of the microgrid is shown in figure 2.15.

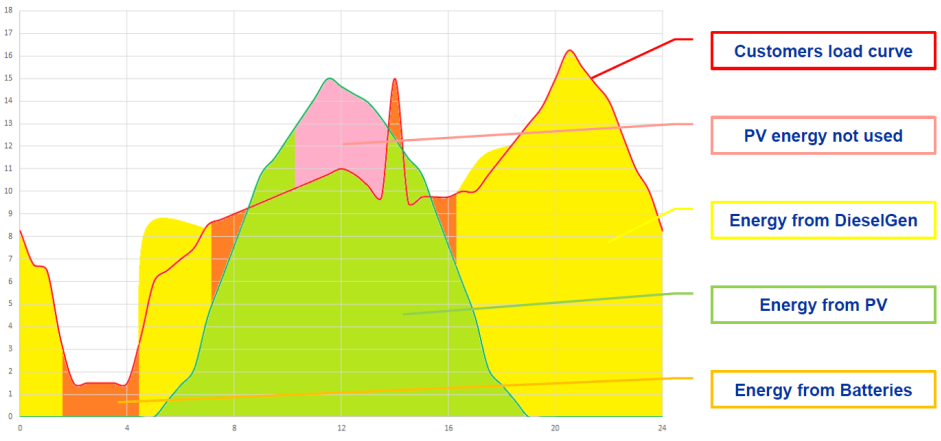

Figure 2.15: *Daily supply and demand curve of a microgrid installed by the company Enel in Colombia*

A hybrid power system composed of solar PV, diesel generators and an EESS is proposed using local controllers with simple telecommunication with the main distribution system. The controller provides the possibility for the distribution company to perform simple commands and read measurements. In a span of 24 hours

the load supply changes from storage system, diesel generator and PV.

Cost analysis

Some studies have been made to compare the microgrid solution as an alternative to grid connection. An economical study performed in 2014 in a rural community in Sri Lanka concluded that an off-grid system composed of a hybrid power microgrid could have a levelized cost of electricity (LCOE) of 0.3$/kWh providing overall financial benefits even in a scenario with increasing demand (*Kolhe et al.*, 2014). The village has 150 households resulting in an approximated demand of electricity of 270 kWh. Also the study assumes that there will be a grid extension that will connect to the microgrid after 10 years, going from isolated to grid-connected topology. Using the software HOMER Pro (*Farret and Simoes*, 2006) for optimizing the size of the components the sensitivity analysis concluded that the best combination would be: 30kW of solar PV, 40kW of wind, 25kW of diesel and a 222kWh battery bank. The net present cost of the project was 553k$ with an initial capital investment of 296k$.

Another study from 2016 evaluated the financial feasibility of different topologies of microgrids in Morocco, comparing the off-grid solution with a possible grid extension (*Ghiani et al.*, 2016). As well as in the Sri Lanka study, the software used to optimize the microgrid design was HOMER and the best configuration was also composed of solar PV panels, wind turbine, diesel generator and a storage system. The wind turbine and solar PV panels are assumed a lifespan of 20 years, being that also the timescale of the project with a yearly interest rate of 5%. A typical load profile was used for a community of around 1000 people and 200 households, with streetlights, community services and small shops. Together they have a peak power of 57kW and create an energy demand of 224 MWh/year. A conventional diesel generator of 60kW was included, being able to fully supply the village as a back-up source in case no renewable source is present. The fuel cost of 1.5€/L is assumed. The energy storage system is composed of 72 lead-acid batteries connected in 3 parallel strings of 24 batteries connected in series resulting in a 48V battery bank. The battery bank is connected to the microgrid through a DC/AC converter.

The microgrid proposed is compared to the grid extension solution. For that, a range of costs were assumed: the line deployment cost, varying from 5k€/km to 25k€/km; the operating and maintenance cost of the line assumed as 2% of the capital cost; and the grid power price cost, varying from 0.06€/kWh to 0.1€/kWh.

Case #	Microgrid configuration				NPC	LCOE	Renewable Fraction	Genset operation	Excess electricity	CO_2 emissions
	PV (kW)	Wind (kW)	Genset (kW)	Storage (Ah)	€	€/kWh	-	hours	%	kg/y
1	100	2x20	60	11700	838,211	0.299	0.96	1,025	28.1	21,600
2	100	2x30	60	11700	854,370	0.305	0.96	1,047	34.3	22,168
3	80	2x20	60	11700	857,785	0.306	0.94	1,335	21.5	28,319
4	100	2x20	60	9900	859,455	0.307	0.95	1,193	28.6	24,909
5	80	2x30	60	11700	872,426	0.311	0.95	1,343	28.8	28,690
6	80	2x20	60	9900	873,359	0.312	0.94	1,459	22	30,964
7	100	2x30	60	9900	876,797	0.313	0.96	1,230	34.7	25,756
8	100	2x20	60	7800	889,658	0.318	0.94	1,504	29.5	31,152
9	80	2x30	60	9900	891,478	0.318	0.94	1,496	29.2	31,807
10	80	2x20	60	7800	899,733	0.321	0.93	1,742	23	36,598

Table 2.3: *Microgrid design optimization (Ghiani et al., 2016)*

Table 2.3 summarizes the results of LCOE and net present costs (NPC) for different configurations of microgrid. The study compares the microgrid to a conventional off-grid solution of supplying the load with only the diesel generator, resulting in a LCOE of 0.713€/kWh and a NPC of approximately 2M€. This shows that using a hybrid microgrid with renewable energy sources and energy storage is always a more attractive financially solution. The breakeven distance is the distance in which the conventional grid extension becomes more expensive than the off-grid solution. This is an important parameter to evaluate whether or not the microgrid is the optimal solution or when it becomes attractive.

Capital cost for building new infrastructures €/km	O&M cost €/y/km	Grid power price €/kWh	Breakeven distance depending on wind resource availability (Average wind speed) km		
			(2.5 m/s)	(5 m/s)	(7.5 m/s)
5,000	100	0.06	174	107	72.4
		0.08	165	98.3	63.5
		0.1	156	89.3	54.5
8,000	160	0.06	108	67	45.3
		0.08	103	61.4	39.7
		0.1	97.2	55.8	34.1
15,000	300	0.06	57.8	35.8	24.1
		0.08	54.8	32.8	21.2
		0.1	51.9	29.8	18.2
20,000	400	0.06	43.4	26.8	18.1
		0.08	41.1	24.6	15.9
		0.1	38.9	22.3	13.6
25,000	500	0.06	34.7	21.5	14.5
		0.08	32.9	19.7	12.7
		0.1	31.1	17.9	10.9

Table 2.4: *Breakeven distances for different topologies and average wind speed (Ghiani et al., 2016)*

Based on the results reported in table 2.4, grid power price has a small influence at the breakeven distance, whereas the average wind speed and especially the cost for line deployment have a huge impact at the financial benefits of using a microgrid, changing significantly the breakeven distance.

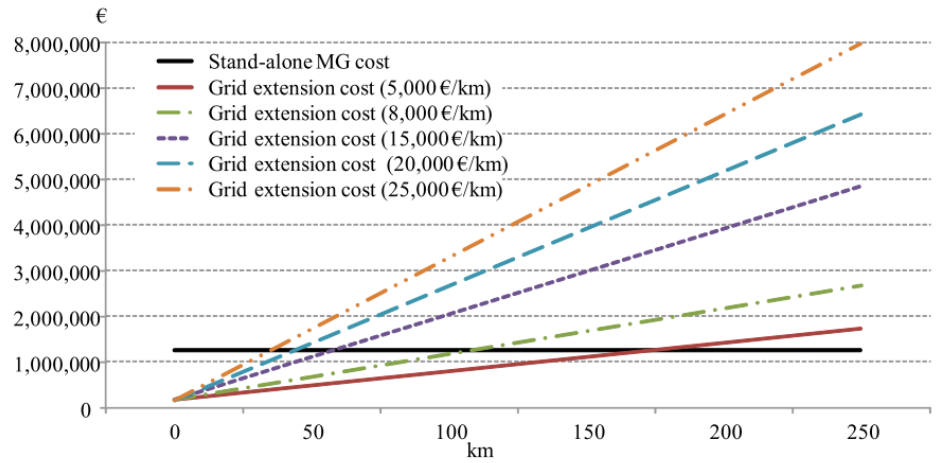

Figure 2.16: *Breakeven distance considering a constant grid power price of 0.06 euro per kWh (Ghiani et al., 2016)*

From the examples given by these two studies it is clear that using a microgrid solution for rural electrification can be a viable and better financial solution, if all the factors are taken into account and a proper microgrid design is made. To summarize, the main contributions to make an off-grid application feasible are: high solar irradiation and average wind velocity, high cost of line deployment, load forecast possibility for proper sizing of generators.

3

State-of-the-Art on Spatial Analysis

Besides deciding between each of the techniques described in chapter 2, rural electrification planning involves many other aspects in order to achieve the best topology of an electric grid. What path the electric lines must take, which terrains it should avoid, and how to group loads in the most efficient way, are some of the questions that arise during these projects, and they require more information than simply deciding whether to use a single-phase or a three-phase system. The approach of rural electrification proposed in this thesis aims to answer those questions, and in order to better understand how the solutions that will be later reported were developed, a theoretical background is necessary. The next sections will describe the state-of-the-art of a few topics that are relevant to the rural strategy proposed such as: terrain and spatial analysis, clustering algorithms and the basics of graph theory.

3.1 Graph theory

The simplest way to represent a network topology, such as an electric grid, is by making using of graphs. Society has been, for a long time, trying to solve real problems through mathematics. With graph theory it is no exception and its origin comes from a historically notable problem called *"The Seven Bridges of Konigsberg"* which was solved by the renowned mathematician *Euler* (1736). The problem revolves around the city of Konigsberg in Prussia that at the time had a small internal island that was separated from the two mainland portions by the Pregel River. These lands were connected through seven bridges and the problem proposed was:

> *"Is it possible to formulate a walk through the city that would cross each bridge once and only once?"*

This apparently simple question gave birth to one of the most fundamental mathematical theories. Nowadays graphs are used in a wide variety of application, from social sciences to biology and more notably in computer science where a whole field, called network science, derives from the graph theory.

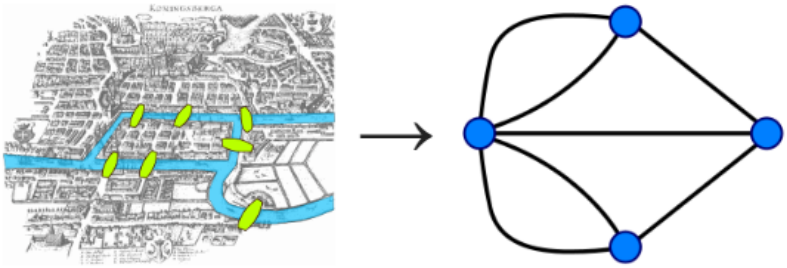

Figure 3.1: *City of Konigsberg transformed into the first graph created*

Definitions

There are various types of graphs and the term is often used loosely, in this work graph theory is used in order to proper represent an electric network topology. The following graph definition is the one that better suits this goal (*Williamson*, 2010):

Definition 3.1. Graph

A graph G is a pair $G = (V, E)$ *where:*

- *V is a finite set, called the vertices of G, and*
- *E is a subset of* P_2V *(i.e., a set E of two-element subsets of V), called the edges of G*

For the purposes of this thesis work which deals with electrification and grid planning, the aim is to connect loads and generation, which can be represented vertices, by means of a electric line, which can be represented as an edge, with lowest cost possible. Important to notice that based on the definition proposed it is not considered the presence of *loops* in the analysis. A *loop* is an edge that connects a vertex to itself, which would require that an edge can be made by a single element of the subset V. Since the idea is to representing a electric network as a graph, there is no real motivation to create a connection from a load or generator to itself. A graph can also be classified as *undirected* or *directed*, also called a *digraph*. A digraph is graph whose edges have a direction associated with them, i.e. the position order in which the pair of vertices are described represent the direction of that edge. In electrical power system analysis this concept is particularly useful when trying to represent a network with its power flow, in which the direction of power generated or absorbed is a valuable information.

Definition 3.2. Path

Let $e_1, e_2, ..., e_{n-1}$ *be a sequence of elements of E (edges of G) for which there is a sequence* $a_1, a_2, ..., a_n$ *of distinct elements of V (vertices of G) such that* $e_i = \{a_i, a_{i+1}\}$ *for* $i = 1, 2, ..., n-1$*. The sequence of edges* $e_1, e_2, ..., e_{n-1}$ *is called a path in G*

Definition 3.3. Connected graph

Let G = (V, E) be a graph. If for any two distinct vertices u and v of V there is at least one path connecting u and v, then G is a connected graph.

Definition 3.4. Tree

If G is a connected graph without any cycles then G is called a "tree" T. Therefore, a tree is a graph if for every pair of vertices $u \neq v$ in G, there is exactly one path from u to v.

Definition 3.5. Forest

A forest is a graph all of whose connected components are trees. In particular, a forest with one component is a tree.

Figure 3.2 shows an example of some of the items that were defined and the differences between them.

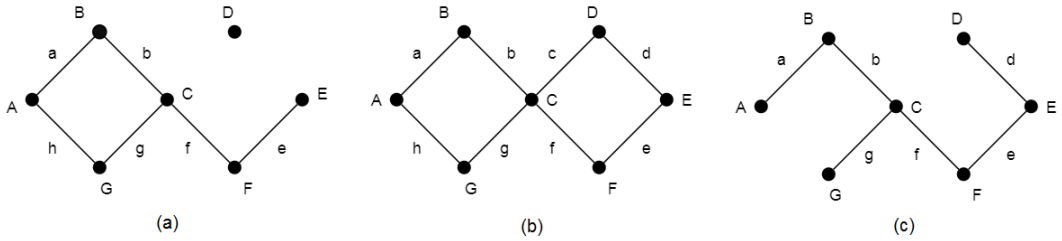

Figure 3.2: *Example of a graph (a), connected graph (b) and a tree (c)*

Definition 3.6. Spanning tree

A spanning tree of a graph G = (V, E) is a sub-graph T = (V, E') which is a tree and has the same vertices as G.

Observing the tree shown in figure 3.2(c) we can infer that this is a spanning tree of the graphs shown in figure 3.2 (a) and (b). If we consider that each edge e on the graph is associated to a weight $w(e)$ that will represent the cost of that connection, each different spanning tree will have a total cost assigned to it that will be the sum of all the weights. From all possible spanning trees from a graph G, the one with the lowest total weight is called minimum spanning tree (MST) and the definition is the following:

Definition 3.7. Minimum spanning tree

A minimum spanning tree T = (V, E') of a connected graph G = (V, E) is a spanning tree such that

$$\sum_{e' \in T} w(e') \leq \sum_{e \in T'} w(e'') \tag{3.1}$$

for every other spanning tree $T' = (V, E'')$

The *minimum spanning tree problem* revolves around finding the MST of a given graph and it is a well established mathematical dilemma. To find the solution several algorithms have been developed along the years, some of which will be discussed in the next section.

Solutions for the minimum spanning tree problem

Greedy algorithms follow an heuristic problem-solving approach of making locally optimal decisions with the intent that it will gradually arrive to a global optimum solution. For each step of the problem a decision is made that cannot be reverted and

if an incorrect decision is made at the beginning of problem-solving process, it can lead to a solution far from the optimum one. However, since these algorithms tend to require very low computational effort, they can provide fast sub-optimal answers that can be used as first step or comparison for simple analysis. One of the first solutions for the MST problem is called the Prim's algorithm, which was developed in 1930 following the greedy premise.

Prim's algorithm

Figure 3.3 shows an example of the Prim's algorithm step by step execution.

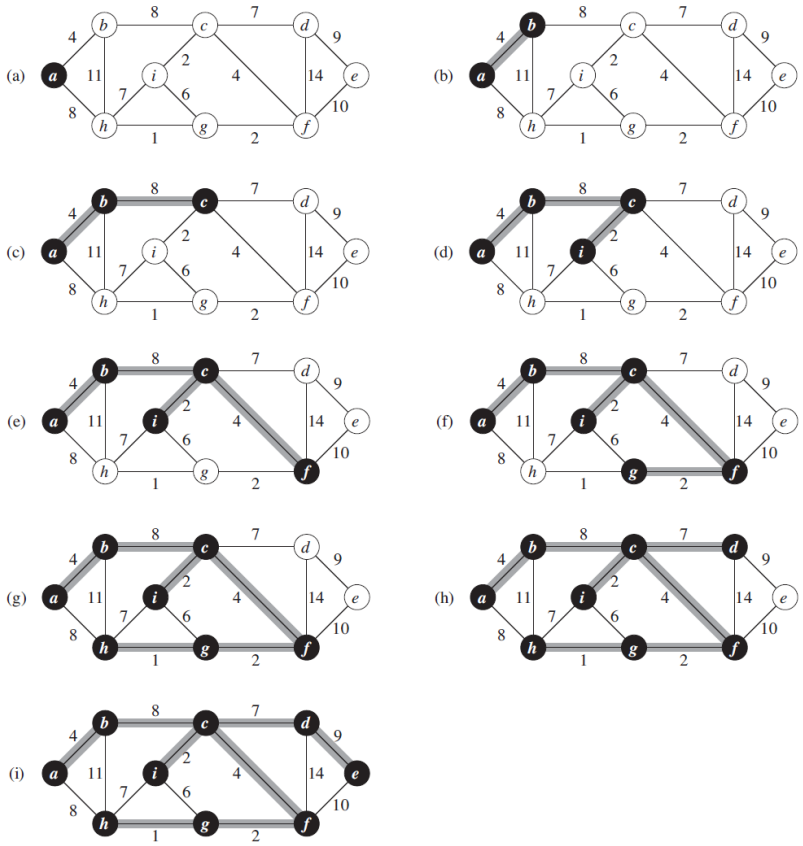

Figure 3.3: *Example of execution of the Prim's algorithm*

Consider an undirected weighted graph $G = (V, E)$ with n nodes and m edges. The Prim's algorithm starts building the MST from an arbitrary vertex and goes on adding the lowest weighted edge step by step. A short summary of the algorithm logic can be:

1. Select an arbitrary node in the graph to create a tree $T = (V', E')$ where $V' = v_0$, $v_0 \in V$ and $E' = \emptyset$.
2. While $V' \neq V$ do:
   (a) Identify all possible edges $e = (u, v)$ such that $u \in V'$, $v \notin V'$ and $e \subset E$.
   (b) Select the edge with lowest weight and add e to E' and v to V'.

(c) If all there is no possible edge, stop.

3. If the algorithm stops with $V' \neq V$, G has no spanning tree. Otherwise T is the MST of G.

The computational complexity and the time it requires to reach the solution depends on the amount of vertices and edges. The way in which the graphs are structured, if by matrices or lists, and the edges are ordered also influence on the complexity.

Kruskal's algorithm

A variation of the Prim's algorithm that is also well-known is called Kruskal's algorithm. In this variation all the edges of the initial graph are sorted in an ascending way related to their weights into a set. The algorithm proceeds by removing one edge at time from the set until the set is empty, i.e. all the edges have been evaluated. Figure 3.4 shows an example of the execution of the Kruskal's algorithm using the same graph of figure 3.3.

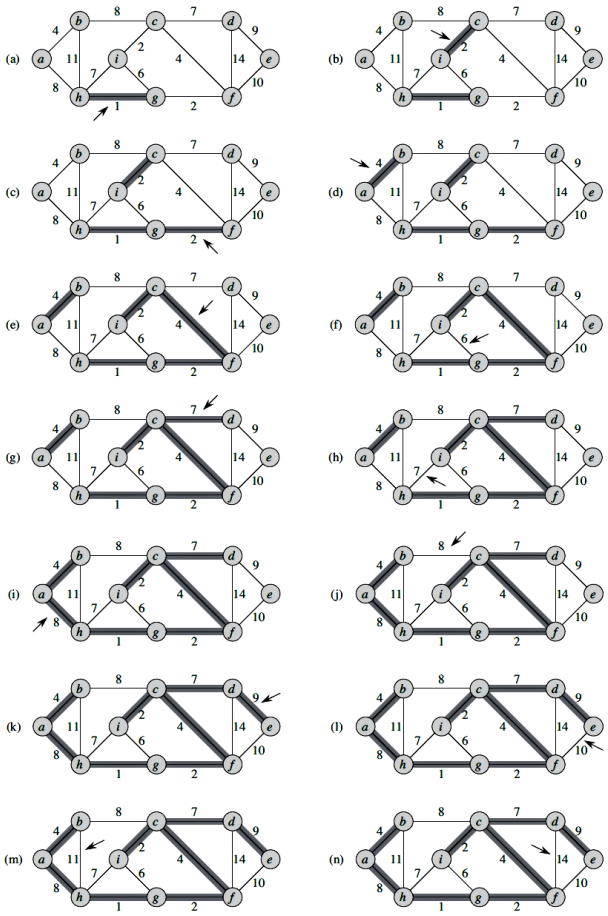

Figure 3.4: *Example of execution of the Kruskal's algorithm*

The premise is the same: consider an undirected weighted graph $G = (V, E)$ with n nodes and m edges. The logic for creating the MST is the following:

1. Create a forest F, where each node in the graph is a separated tree.

2. Create a set $S = (E)$ containing all edges in the graph sorted in an ascending order related to their weight.

3. While $S \neq \emptyset$, take the first and lowest weighted edge $e = (u, v)$ from S:

   (a) If u and v belong to different trees, then include the edge e into the forest F and combine the two trees into a single one.

   (b) If u and v belong to the same tree, then simply remove that edge from S.

The result of the algorithm is a forest F that will contain all the vertices of the graph G, if F is a connected graph then, by definition, F is also a minimum spanning tree of G. Both algorithms manage to correctly find a MST of the graph, which has two solutions due to the edges (b, c) and (a, h) having the same weight.

Besides the limitations and disadvantages of being a greedy algorithm, there is another limitation that makes them not quite suitable for our case study. Going back to the analogy of each node of our graph being a load or generation that must be connected, we can see that an electric grid following the MST structure will be composed only by straight lines and no intermediate connections such as substations. This solution seems too simplistic and rather unrealistic and the reason is that he minimum spanning tree problem is set to find the solution of the following question:

"How to go from a graph to a spanning tree with the lowest cost possible?"

Which is very similar, although still fundamentally different, to one of the question this thesis intend to answer, that is:

"How to connect isolated loads in a rural area in the most efficient way?"

This second question, instead of the *minimum spanning tree problem*, is perhaps more related to an also well-known mathematical dilemma called *shortest path problem*. The difference between these two problems and their implications will be discussed next.

Solutions for the shortest path problem

First consider the undirected weighted graph shown in figure 3.5(a) starting the analysis from point A. The MST of the graph will have the total cost of 10, and its path is highlighted in red. However if we consider another approach for the same graph, which is to find the shortest path from the point A to point C, the MST solution will still be the same even though it is clear that the shortest path would be to directly connect A to C even though it initially it has the highest cost. This showcases the fundamental difference between the two problems, which is that MST, by its definition, requires to connect every single point of the graph.

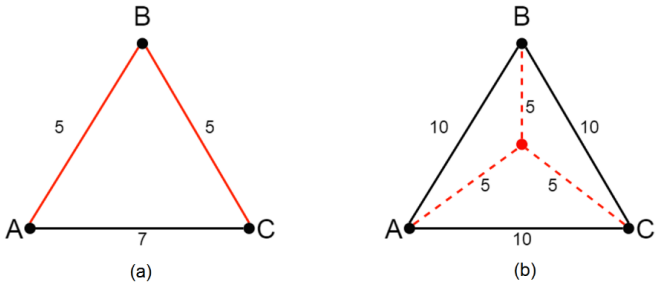

Figure 3.5: *Showcase of the differences between minimum spanning tree and shortest path*

Another showcase of the limitations of the MST solution and how it could be improved by adding intermediate optional points is given by figure 3.5(b). The addition of the center point demonstrates that a new, better and less costly spanning tree can be achieved by using optional targets or derivations. Therefore, the MST produces grids in which the node connection is straight, ignoring any obstacles that might be in the middle. For rural applications, deploying electric lines across a forest or a swamp is often way more costly then following a road around it. Solutions that can find an optimal path connecting few targets amidst a group of points are more suitable to our objective.

In order to proper comprehend the next algorithms that will be described, other definitions are important to be made:

Definition 3.8. Shortest path

Let $G = (V, E)$ be a connected weighted graph. The shortest path between two distinct vertices u and v of G, is the path that has a sum of weight less than or equal to any other path connecting u and v.

Definition 3.9. Distance

Let $G = (V, E)$ be a weighted graph. The distance $d_{u,v}$ between two vertices u and v of G, is the sum of weights of edges of the shortest path from u to v. If no path exists between them, then $d_{u,v} = +\infty$. If $u = v$, then $d_{u,v} = 0$.

The distances between each pair of nodes in a graph are often represented in a matrix nxn, where n is the total number of nodes, called distance matrix. Many algorithms, as the one that will be described next, use the distance matrix as a step to achieve the *shortest path problem* solution.

Dijkstra's algorithm

Aiming to solve the *shortest path problem*, Edsger W. Dijkstra proposed in 1956 an algorithm capable of find the shortest patch between two given nodes of a graph. First, consider an undirected connected weighted graph $G = (V, E)$, with at least two distinct vertices one defined as source node s and the other target node t, in order to find the shortest path between s and t the algorithm apply the following steps:

1. Create a Tree $T = (V', E')$ where initially $V' = \{s\}$ and $E' = \emptyset$.

2. While $t \notin V'$ do:

   (a) Compute all the distances between s and all nodes of T.

   (b) Find the edge $e = (u, v)$ from E where $u \in V'$ and $v \notin V'$ which minimizes $D_{s,v}$.

   (c) Add the edge e to E' and vertex v to V'.

The result of the algorithm is the shortest path between the source and the target, which can also be called the minimum path tree. Note that since Dijkstra does not require that all vertices of the input graph G must be connected, the only restriction is that the target node is connected to the source node, the minimum path tree is not necessarily a spanning tree of G. The total steps required for the algorithm to finish is $(n - 1)$ where n is the total number of vertices.

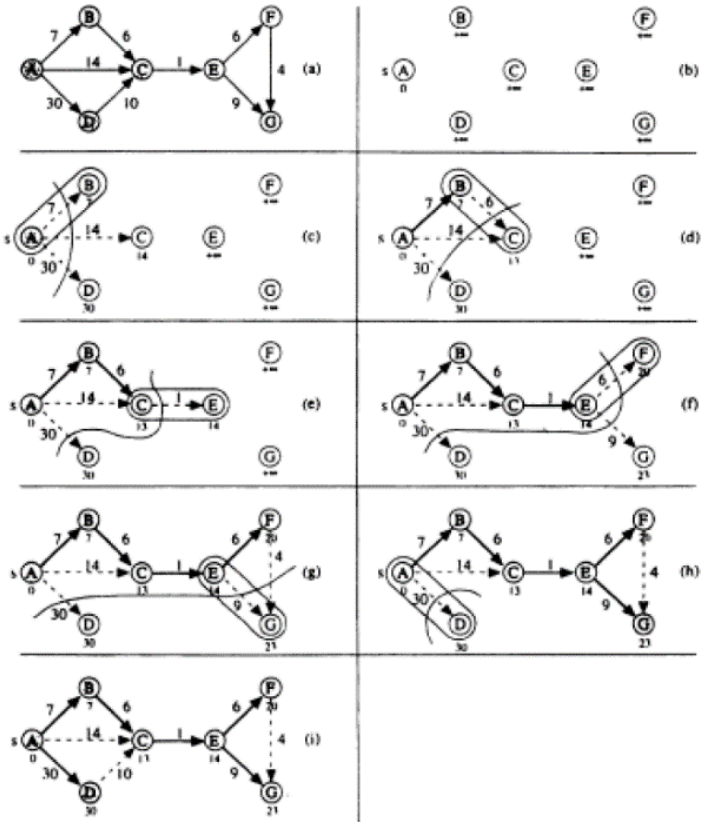

Figure 3.6: *Example of execution of the Dijkstra's algorithm*

Disjkstra's algorithms solves one of the problems of the MST, which is the necessity of connecting every node not allowing derivations. However it still follows a greedy strategy, choosing the best edge to incorporate assuming it will always end with the global optimum solution. Another limitation of the shortest path problem is that it only accounts for one source and one target node, but what if the problem requires to connect thirty distinct nodes from a graph composed by three thousands nodes? Then it becomes a more complex problem that requires a more complex solution.

Solutions for the Steiner tree problem

The *Steiner tree problem* can be summarized as:

> *Consider an undirected connected weighted graph $G = (V, E)$ and a subset of vertices $S \subset V$ called terminal nodes. Find a tree $T = (V', E')$ where $V' \subset S$ and the weight $w(T)$ is minimized.*

If we consider a *Steiner tree problem* where S is composed by only two elements, we simplify it back into the shortest path problem. Besides looking relatively simple, the Steiner tree problem is part of a specifically complex type of problems called NP-complete problems as proved by *Santuari* (2003). NP-complete problems cannot be solved by means of polynomial algorithms like Kruskal, Prim and Disjkstra, which require a polynomial amount of time to reach the solution. The time required to find the exact best solution is not possible to be evaluated and it could require months or even years. Heuristic approaches that seek to find approximated solutions have been developed and improved for decades. *Robins and Zelikovsky* (2008) presents a historical perspective of the solutions of the Steiner problem since 1966. A well-known method to approximate this problem is by making use of the MST and is shown in figure 3.7.

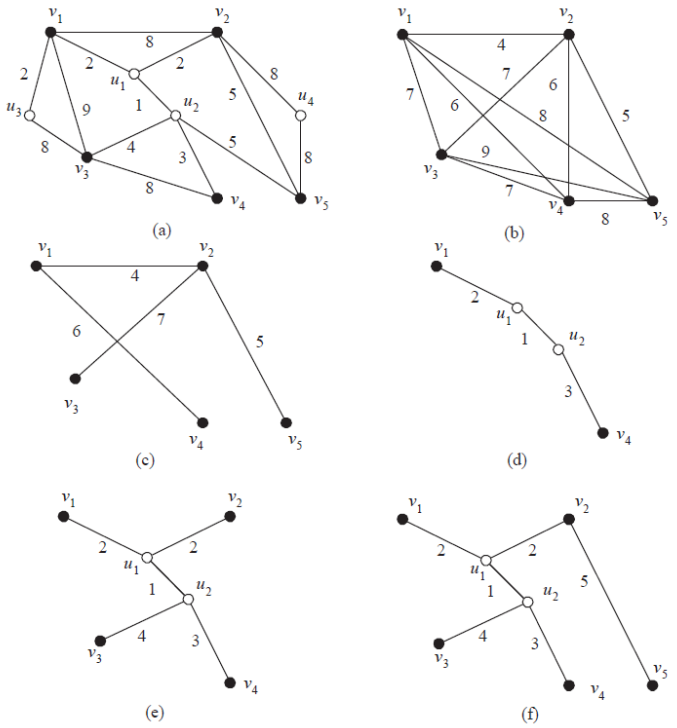

Figure 3.7: *Example of a solution of the Steiner tree problem by MST approximation (Ye Wu and Chao, 2004)*

Consider the initial graph $G = (V, E)$ presented in figure 3.7(a) with the subset S composed by the five terminal nodes (black dots); in order to find the Steiner tree we must follow these steps:

1. First compute all the shortest path weight between each terminal, this graph is called the metric closure (G_S), as shown in figure 3.7(b).

2. Compute the MST of the metric closure $MST(G_S) = (V', E')$, shown in figure 3.7(c).
3. Starting from an empty tree $T = \emptyset$, for each edge $e = (u, v) \in E'$ do:
   (a) Find the shortest path P from u to v on G.
   (b) If P contains only two vertices, then add P to T, shown by figure 3.7 (f)
   (c) If P contains more than two vertices, then add only the sub-paths from the terminals to the vertices already in T, shown in figure 3.7 (d) and (e).

The output of this procedure is called Steiner minimum tree (SMT). The MST-based approximation algorithm was proposed by several authors independently, the example here used to describe this approach was based on an implementation of one of those algorithms made by *Ye Wu and Chao* (2004). The approximated solution proposed is able to create a tree that includes several terminal nodes under a polynomial time which depends on the size of the metric closure. In order to evaluate the performance of a Steiner tree, it is common to use the Steiner ratio:

Definition 3.10. Steiner ratio

The Steiner ratio ρ is the ratio between the weight of the minimum spanning tree related to the terminal nodes $(MST(S))$ and the weight of the corresponding Steiner tree:

$$\rho = \frac{w(MST(G_S))}{w(SMT(G_S)} \tag{3.2}$$

This means that the Steiner ratio is always higher or equal to 1. Some different approaches such as the Euclidean Steiner tree problem, which represents the spatial distance in a simpler manner, is conjectured to achieve a Steiner ratio of approximately 1.2. Other approaches use the rectilinear distances, the most basic solutions, such as the one here described, can achieve a Steiner ratio equal to 2, while more complex solutions can decrease this value up to 1.35 as described by *Ye Wu and Chao* (2004).

3.2 Geographical Information System (GIS)

The procedure developed within this thesis work is settled on Geographic Information System (GIS). GIS is a system designed to manage and analyse spatial and geographic data. It has been used for several years in many areas, but had a boost with the improvement of digital technology. Modern systems use digital information generally acquired through ortho-rectified imagery, which revolves in taking aerial photograph from satellites and geometrically correct them in a way that the scale remains uniform, being able to correctly represent distances. This technique is widely used to create both private (Google Maps) and public (OpenStreetMap) databases.

These ortho-rectified satellite imagery are used as an accurate map background in a GIS software, in which several layers of information will be placed on top of this background to proper display and manage the spatial data in what is called Geoprocessing. The most powerful aspect of a GIS is to relate otherwise unrelated

information by using space and location as a common index. In this way GIS integrate different types of data by stating the exact position on Earth's surface to which they refer. In order to do so, all those information must be interpreted using the same Coordinate Reference System (CRS).

Map projection and reference systems

A CRS defines a specific coordinate-based projection, which represents the geographical position of an object. There are different ways of creating a map projection i.e. flatten the globe's surface into a plane, in which according to Gauss's *Theorema Egregium*, all of them have some sort of information loss due to distortion. Some distortions are acceptable and others not depending on the purpose of the projection, for this reason different map projections exist in order to preserve some properties at the expense of others. Whereas some are designed to proper represent distances (equidistant projections), some others are focused on the proper display of the areas (equal-area projections) or shapes (conformal projections).

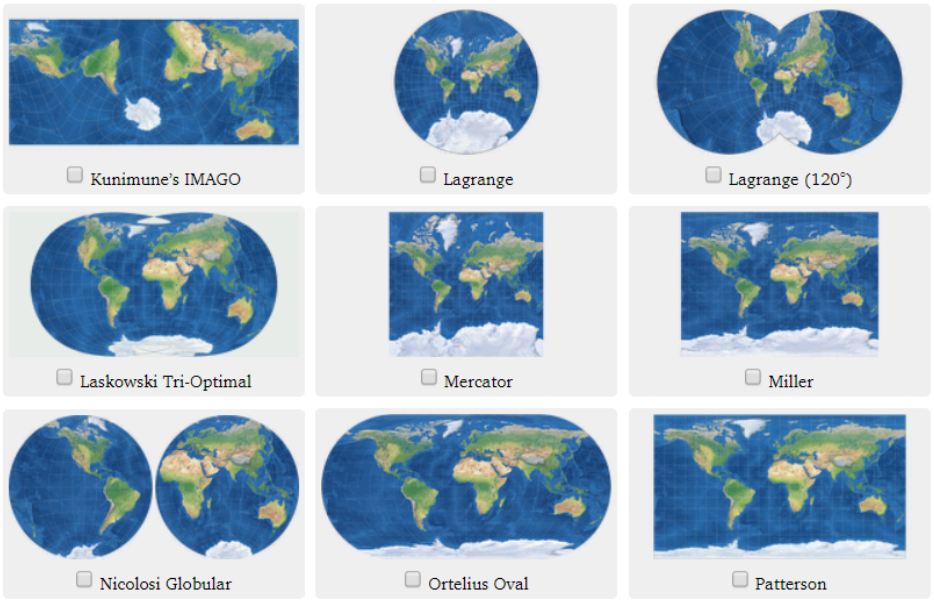

Figure 3.8: *Different map projections strategies of the Earth's surface*

The main goal of the projection is to create a two-dimensional system of coordinates that can be displayed on a flat surface. This is called a geographic coordinate system (GCS), which is a three-dimensional coordinate system of a sphere or spheroid (*Gardner*, 2020). The most widely GCS used is the WGS84 and have been adopted into many applications such as the well-known Global Positional System (GPS). Geographic systems use as coordinates lines of latitude, starting from the Equator line, and longitude, starting from the Greenwich meridian, which are measured in degrees unit. When being displayed in a graph, typically the longitude values are paired with the X axis while latitude values with the Y axis.

One way to mitigate the distortions caused by map projection is to reduce the area of interest from which the projection is made. The Universal Transverse Mer-

cator (UTM) coordinate system divides the world into 60 equal zones, 6 degrees of longitude wide. The idea is to keep the origin of latitude at the Equator but transpose the origin longitude to a specific coordinate based on which zone is selected. As one of the major advantages of using GIS is to relate different types of data using the same geographical reference, it is of extreme importance that all layers of information are projected with the same CRS. Choosing a non-optimal CRS or having layers projected into different systems will end up compromising the accuracy and consequently the legitimacy of the results found.

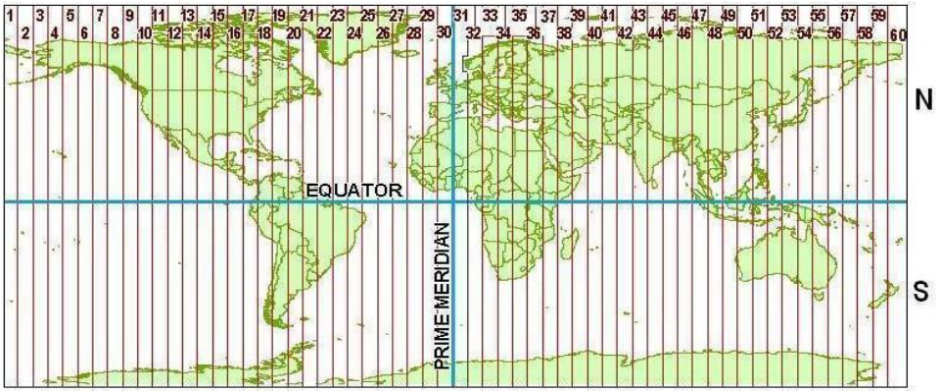

Figure 3.9: *UTM zones divided into north and south*

Type of data used in geoprocessing: vector and raster layers

There are two main types of data that are used as input in GIS software: vector and raster. Raster data are made of a matrix of pixels (also called cells) and each of them contains a value assigned (also called band). This type of data is well suited to display information that changes in a continuous way across an area. This is the case for satellite images and aerial photographs, this images are read as a raster layer. The higher the resolution of the image the higher is the amount of pixels it contains and therefore a better accuracy is obtained. Figure 3.10 shows the elevation of an area that will be later used in the case study.

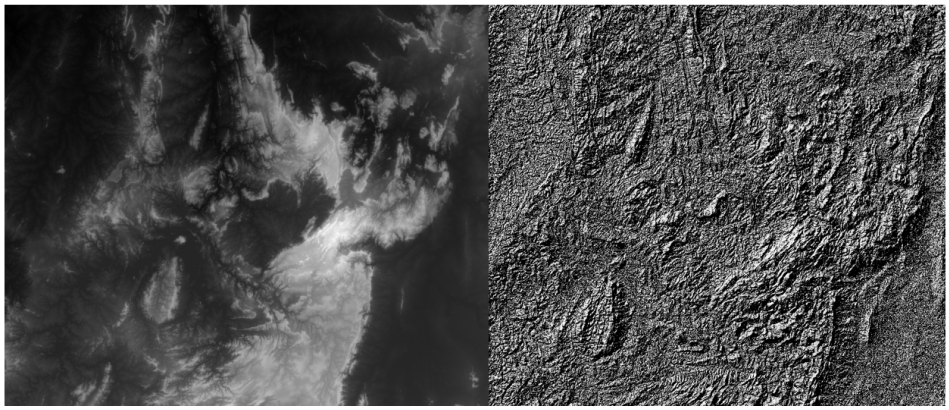

Figure 3.10: *Elevation in the municipality of Cavalcante, Brazil displayed as raster data in two different ways: single-band in grayscale (left) and hillshade(right)*

The data is public and is provided by the Brazilian National Institute of Spatial

Research (INPE), with a high resolution of approximately 30m (size of each pixel). Figure 3.11 demonstrates a zoomed image of the same raster layer where the pixel details can be seen.

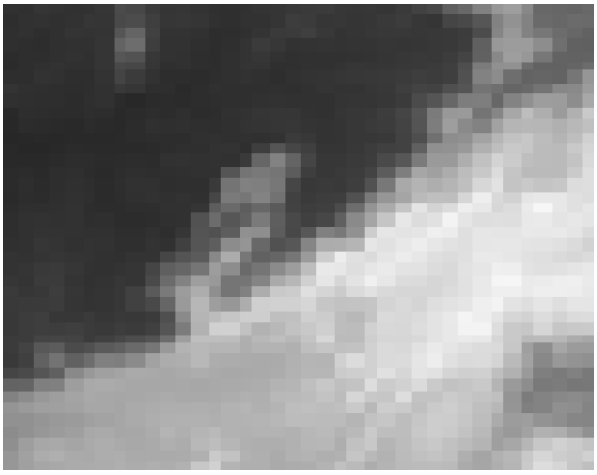

Figure 3.11: *Pixel details of the raster image of elevation of figure 3.10*

Vector data, instead of images, use specific types of geometry to represent the features present in the real world that should be displayed. These geometries are made up of one or more interconnected vertices. The most fundamental and basic geometry, which consist of only a single vertex, is a point. Points are used to describe features that the dimension is not an important aspect and simply describe a position in space using X, Y and in some cases a Z coordinates.

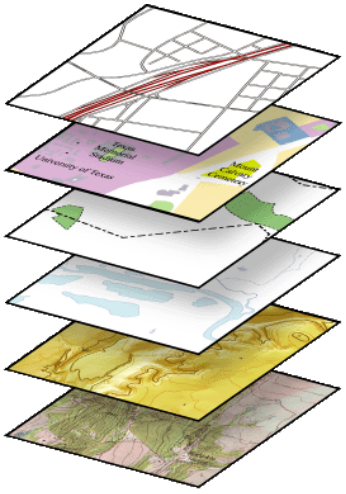

Figure 3.12: *Layer arrangement to create a GIS map*

Example of features that can be represented by a point geometry are generators location and the substations allocated along electrical lines. Two or more vertices, in which the first and last vertex are not equal, forms a line geometry. Lines are suited to represent linear features in GIS map such as roads, rivers or electrical lines. In the case where the first and last vertex coincide, a polygon geometry is formed. Polygons represent enclosed areas and are suited to display the shapes of federal boundaries,

cities and even buildings. For GIS, a layer is composed by either raster or vector data and in case of the latter it can have one of the three geometries described. Typically an arrangement of layers is used in order to create a GIS map (see figure 3.12), the higher the amount of information that is required to be displayed the higher the number of layers.

Uses of GIS mapping

The governmental institutions were the first to use geospatial data for territory analysis, such as mapping of natural resources and population census, and the public access to this type of data was quite limited until the end of 90's. The last decades were marked by a huge increase in the information available with data science, data processing and big data, becoming increasingly more popular. With the improvement of computational capabilities, the ability of processing all that information is quickly becoming one of the most powerful tools in the short future. Several fields of study were enhanced by this revolution, from financial analysis to political marking and GIS is no exception. With the widespread of satellites, internet, and mobile telecommunication, and the huge amount of public data that came as a result from this, GIS is becoming more and more famous across the globe while finding applications in a wide variety of fields.

A big amount of online applications were created, mostly related to governmental institutions, to facilitate the information access and reading for the public. A good example of this application is the interactive GIS map created by the Johns Hopkins University to display the outbreak of the virus COVID-19 that started in December 2019. In the map, shown in figure 3.13, different types of vector layers can be identified.

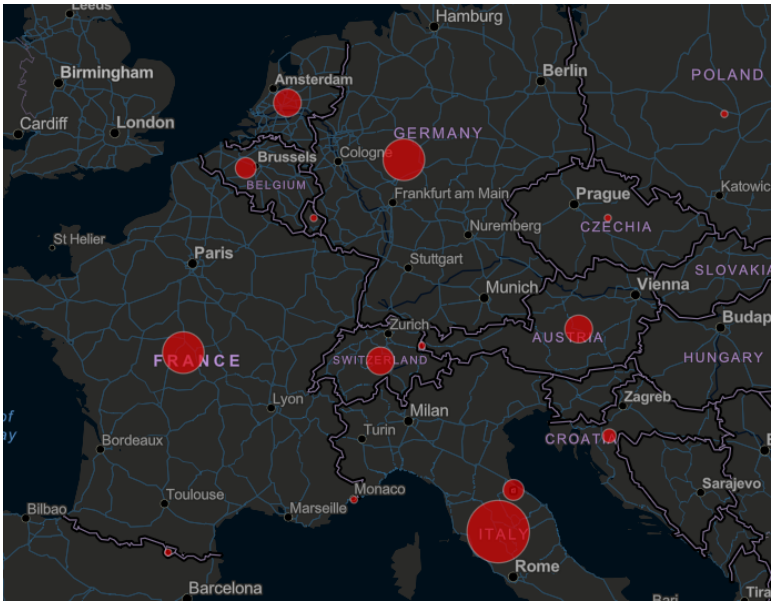

Figure 3.13: *GIS online application showing the increasing number of cases of the virus COVID-19 (Gardner, 2020)*

The main information is displayed as points that correlates its size according to

the number of cases in each country. In academic literature, GIS has been used for a long time. *Maleš-Sumić and Venkata* (1993) describes a way to automatically design an underground distribution system, using GIS as a base platform. The results show that the system proposed can achieve considerably cost savings and become a powerful tool to compare different types of designs, which were otherwise limited by the traditional manual design. *Cevallos-Sierra and Ramos-Martin* (2018) used GIS to identify locations in Ecuador, with the greatest potential for development of renewable energy power plants. It estimates the maximum energy of different renewable technologies and evaluates their possible contribution to the national power system. In rural electrification processes, GIS has been used to route lines and also to evaluate the best form of energy supply, as described by *Domínguez and Pinedo-Pascua* (2009) in the results of the study made by the research group gTIGER in Cuba.

Sources of GIS data

Nowadays there is an effort in creating more and more digital GIS information, as government and organizations have becoming aware of the big potential they have in displaying information and guiding policies. One of the biggest institutions that has been supporting researches and providing an open online platform for sharing GIS data sets is the World Bank. In their online database (available on *data.worldbank.org*, which is public and free, it is possible to find a vast diversity of information such as statistics about world development indicators, world education, gender and health nutrition. The institution also partners with many other organizations, such as the Energy Sector Management Assistance Program (ESMAP), to provide useful public information for academic research. For example the *energydata.info*, *globalsolaratlas.info* and *globalwindatlas.info* form together a platform containing updated information about solar irradiation, wind measurement, electrical network of several countries and many other information. As an example of the usefulness of these data sets, figure 3.14 displays the optimum tilt angle for PV modules across the world.

Figure 3.14: *Optimum tilt angle for PV panel modules*

Also many universities have open databases for public consultations such as the *OpenGridMap* from the Technical University of Munich (TUM) and *Enipedia* from Delft University of Technology (TU-Delft). All these platforms offer an overall worldly

view of these topics and can be used as a first input for analysis. If more detailed information is necessary, the governmental institutions could serve as a more accurate source of information. Most industrialized countries have precise data and statistics about their population and while a big part of this information is confidential, some governments offer public platforms in which the population can consult statistics such as crime rate, GDP, health indexes etc and population density.

3.3 Terrain modeling

The combination of GIS technology with graph theory provides a powerful tool for spatial analysis and terrain modeling. Finding the least-cost path from a source point to a destination of a real environment using GIS as a modelling tool is not a recent phenomenon (*Lee and Stucky*, 1998). Modelling an area or terrain and applying a shortest path algorithm can be done using two different approaches: routing across a continuous surface and routing along a discrete network. These approaches differ in how the background of the graph used to compute the shortest-path will be modelled, using either raster or vector layers.

Raster-based modelling

As explained before, raster layers can be considered a superimposing square-grid of pixels on the plane. The sampling theories of Shannon and Nyquist state that the cell size, represented by the resolution, should be at least $2\sqrt{2}$ times smaller than the smallest detail, in order for the analysis to not have information loss. In the rasterization process a suitable weight or cost, which represents the cost per unit distance, is assigned to each grid-cell. Then a shortest-path algorithm is used to track the cells with lowest weight and create a path connecting terminal nodes.

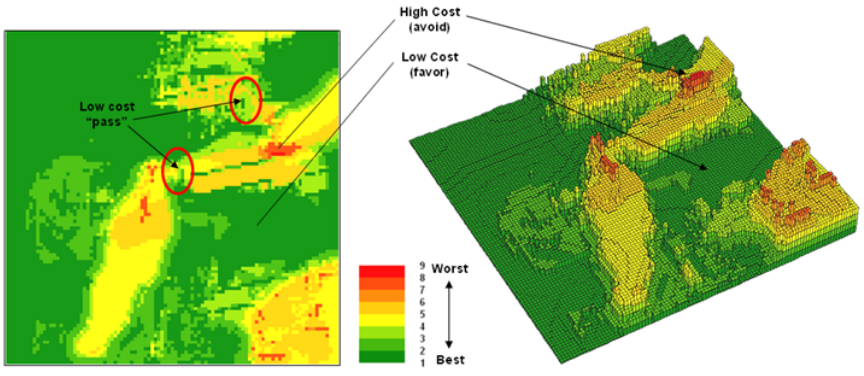

Figure 3.15: *Example of weighted raster model of a terrain*

These algorithms are much simpler and faster than the algorithms used to find the Steiner tree solution, making the raster-based terrain modeling very useful to deal with large and continuous surfaces. This simplicity however, comes at the expense of geometrical distortions. *Bemmelen et al.* (1993) describes two types of distortions that occurs in paths produced on a standard raster grid: *elongation* and *deviation*.

Figure 3.16 (a) shows the real distance of a theoretical path between a source node s and a target node t. The path which is as close as possible to the theoretical continuous-space path is shown in 3.16 (b). It has a stair-step route, which ends up increasing the length of the real distance. This length discrepancy, called *elongation* error, is defined as the ratio of the cost of the calculated shortest and the cost of the theoretical path. In general, on a uniform grid, the steps to find the shortest path make moves in two directions. When all moves in one direction are executed first, the max *deviation* error occurs as figure 3.16 (c) shows. Note that the *elongation* error of both paths presented in figure 3.16 (b) and (c) is the same, and has a value equal to $\sqrt{2}$, meaning that these two paths have a length 41% longer than the real distance.

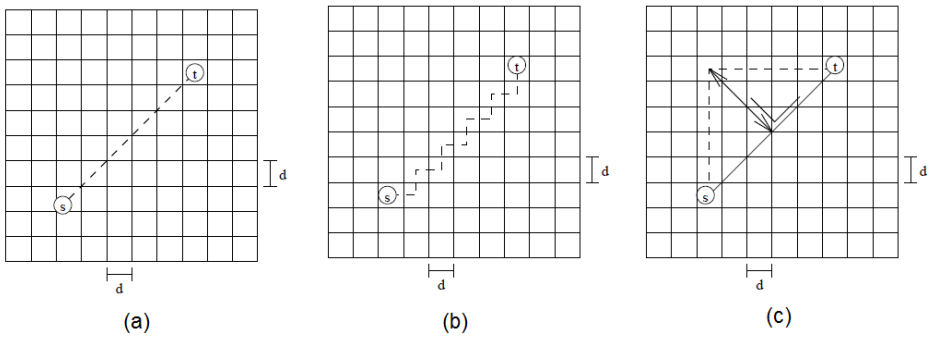

Figure 3.16: *Path between two raster cells. (a) shows the real distance, (b) shows the stair effect and (c) shows the max deviation of a path (Bemmelen et al., 1993)*

Vector-based modelling

The precision of the raster-based solutions, which translates into the quality of the resulted shortest paths, depends on the resolution of the grid. Another limitation of using raster-based terrain modeling for shortest path computation, is that the algorithms provide no natural way to distinguish a point where traffic routes cross from an overpass. Figure 3.17 shows an example of a small road network represented by vector and raster data models. Even considering a theoretical continuous path, represented in figure 3.17 (c), the final result is not able to trace the topological relationships among links in the discrete vector network. As a solution, *Choi et al.* (2014) developed an integrated technique that combines raster- and vector-based analysis in order to overcome the limitations of conventional raster-based algorithms in handling a discrete vector network.

The vector approach represents the terrain using polygons, homogeneous areas with a defined specific cost, and therefore are able to find an exact solution. Similar to the raster-based approach, to each node (now a vector point instead of raster cell) is assigned a weight which is translated to the cost of the line (edges of the graph) that connects these nodes. The main disadvantages of using this approach is the high complexity and time consumption of the algorithms, and the incapacity of represent continuous surfaces. When dealing with large data sets composed of several terminal nodes that must be connected, the complexity of finding the MST solution can become extremely high.

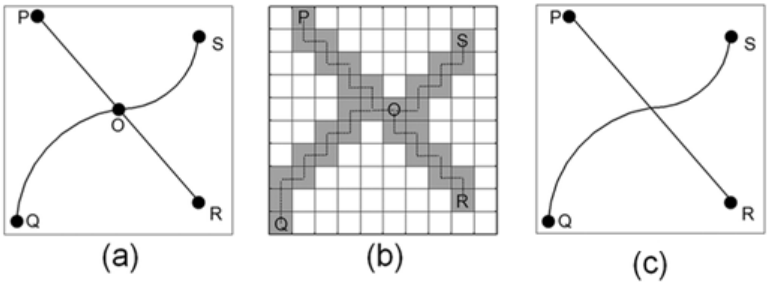

Figure 3.17: *Example of an imaginary road represented by (a) vector and (b) raster data models. (d) Overpass in the road network. Adapted from Choi et al. (2014)*

Weight assignment

After discussing the main algorithms and approaches for grid routing and how to optimally connect nodes in a weighted graph, it is necessary to establish the strategy by which these weights are evaluated. Weight assignment is a broad concept, in this work the focus is on how to evaluate the difficulty or cost to deploy an electric line between two points. When dealing with GIS there are two phases: data gathering and data processing. Weighting strategies are related to the latter, when based on all the information acquired by GIS databases, it becomes necessary to classify or evaluate them to develop conclusions. In literature there has been several applications of this field. *Choi et al.* (2009) presents a raster-based GIS model and adopts a weighting strategy of multi-criteria to evaluate the least-cost path analysis for haulage routing of dump trucks in large scale open-pit mines. In the field of rural electrification, weighting is also a fundamental part for evaluating energy resources and finding the most suitable energy source for a given area. In a similar strategy to the one developed for Ecuador by *Cevallos-Sierra and Ramos-Martin* (2018), *Amador and Domínguez* (2005) makes uses of GIS to divide an area in Spain into zones and uses weight strategies to evaluate technical and economical parameters to define what is the best energy source for each zone.

For electric grid routing, *Monteiro et al.* (2005) proposes that the weight assignment of edges and nodes, which reflect the difficulties of line deployment, is determined by many factors that eventually fall into three categories: Nongeographic cost components (NGC), terrain cross costs (TCC) and direction change costs (DCC). NGC reflects the costs that are independent on the geographic aspect and are related simply to the type of equipment of the line, which reflects a weight that is a function of the line length only. DCC considers the additional costs when a change of direction happens, and is associated with deviation towers that are necessary when a non-straight paths is followed. While NGC and DCC focus on the edges, TCC is associated with each single node and reflects costs related to the environment in which they are inserted. TCC can be classified as:

1. **Accessibility**: represents the costs related to equipment transportation, installation and maintenance. The farther a node is from cities, ports or roads, the lower its accessibility and thus higher its weight. This is especially important in rural areas that usually have poor accessibility

2. **Specific characteristics**: represents the costs related to the specificities of the geographic area such as soil type, land use, vegetation coverage, urban areas and corrosive areas near the shores.
3. **Terrain complexity**: represents the costs related to the terrain slope and orography. For example non-flat terrains requires not only higher towers but also more tower units.
4. **Wind speed**: represents the additional costs of tower reinforcements necessary to withstand the mechanical stress caused by high wind speed.
5. **Altitude**: represents costs associated with reinforcement due to icing and also a higher altitude requires a more complex surge protection due to increased probabilities of lighting.
6. **Obstacles**: represents the costs of crossing obstacles that can be either natural, such as lakes or dams, or artificial, such as roads, railways and other power lines.

The way all these components are evaluated will depend on each application and several different factors. The NGC, for example, can vary a lot depending on the country the electric grid will be deployed. The type of cable used, if it is produced nationally or not, what type of labour is necessary and how expensive is the labour. Therefore, creating a universally weighting strategy, that encompasses every possible scenario, would be not only impossible but also meaningless. What could be done is facilitate the procedure, so that an adaptation can be made creating a new weighting strategy for each application. One way of doing so is to change from a perspective of assigning additional values of cost to each of the elements to a more standardized penalty factor (PF). The approach used in the Open Source Spatial Electrification Tool (OnSSET), which is an energy modelling tool that estimates the most cost effective electrification strategy, goes in that direction. The information gathered through GIS, which fall into the categories previously described, are evaluated and combined in the following manner:

$$PF = 0.15 \times A + 0.2 \times B + 0.15 \times C + 0.3 \times D + 0.2 \times E \tag{3.3}$$

Where each of the variables represent one classification:

- A is related to the distance to the nearest road.
- B is related to the distance to the nearest substation.
- C is related to the elevation.
- D is related to the slope.
- E is related to the land cover.

Since OnSSET is a tool aimed to evaluate energy resources and not specifically grid routing, the coefficients related to the slopes (which influences solar power plants production) and distance to the nearest substation are more impactful. The idea of comparing relative weights to create a single penalty factor is definitely more

simplistic, but is also more suitable for open source tools such as OnSSET and GISEle since it requires from the user less information. In the end each developer must adopt its own weighting strategy and propose forms of evaluating the reliability of the obtained results.

3.4 Cluster analysis

Data clustering, also called cluster analysis, is a method of creating groups of objects, or clusters, in such a way that objects in one cluster share similar characteristics and these characteristics are different from objects in other clusters. Data clustering shall not be confused with classification, in which objects are assigned to predefined classes. In data clustering the classes do not exist a priori and therefore they must also be defined. Clustering is well consolidated in the fields of statistics and computer science and its applications have been of fundamental importance in other fields, such as machine learning, image processing, gene expressions and pattern recognition. Data clustering can be also defined as an indirect data mining, since it is not always explicit what kind of information or cluster results the user is looking for. The quality of a cluster analysis can be measured by its ability to discover some or all of the hidden patterns in a group of objects, resulting in clusters that have high intra-class similarity and low inter-class similarity.

The usage of data clustering and GIS for strategic planning of power systems has been adopted worldwide. In Europe, *Wiernes et al.* (2015) demonstrates how to achieve an optimal regional division of the European power system in terms of renewable energy sources. In USA *Rhodes et al.* (2014) employed cluster analysis to determine the shape of seasonal residential demand profiles based on a survey of over 100 homes, and describes how this information can be used to predict how changing demographics of neighborhoods could influence local distribution grid conditions. *Gan et al.* (2007) describes several approaches in creating clustering algorithms, in the following sections four of them, the ones considered relevant for the purpose of this thesis work, will be detailed: center-based, hierarchical, density-based and model-based.

Center-based clustering

The main goal of this clustering approach is to minimize its objective function, where each resulting cluster will be represented by its center also called centroid. In order to be represented by its center, the resulting clusters have a convex shape and therefore center-based algorithms are not good choices for finding arbitrary shapes. One of the most used center-based clustering algorithms is called k-means. It was first described in 1961 becoming very popular for being highly efficient in dealing with large and high dimensonal data sets. The k-means algorithm is divided into two main phases: initialization phase and iteration phase. In the initialization phase the algorithm randomly assign the objects to a natural number of k clusters. Then, in the iteration phase, the algorithm computes the distance between each object and each cluster center and proceeds to assign the object to the nearest cluster. Figure

3.18 presents a flowchart describing the basic steps of a k-means algorithm, which is simple and has a low computational complexity when compared to other methods.

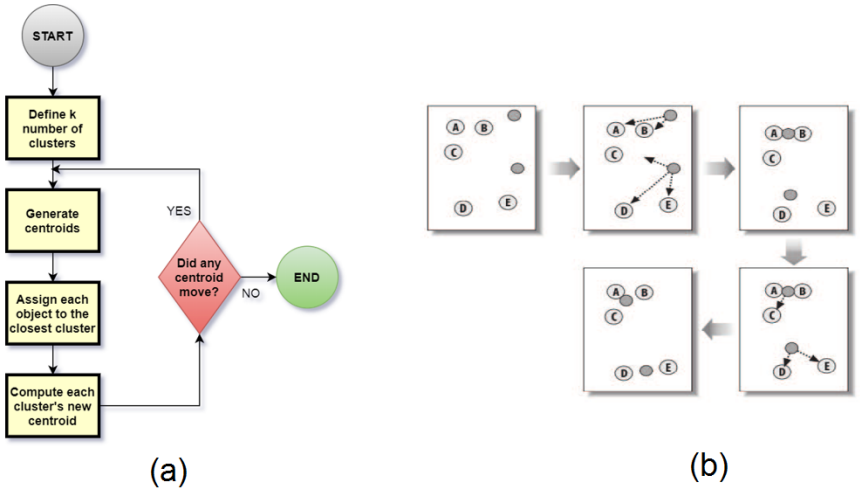

Figure 3.18: *Flowchart of basic k-means clustering algorithm (a). K-means algorithm step-by-step illustration for k equal to 2 (b)*

The computational complexity is linearly proportional to the size of the data set, making it fast and suitable for large amount of data. However with the simplicity comes two main drawbacks. The first is that the performance of the cluster analysis is dependent on the initialization parameters, which often are not known by the user. The second drawback is related to the limits of k-means when dealing with more complex spatial dimensions. In this scenario it could happen that the algorithm is unable to find non-convex clusters resulting in non-convergence. For this reason many methods of finding good initial parameters have been proposed in literature. As an example *Peña et al.* (1999) empirically compared four different methods and concluded that the one named Kaufman Approach provide a significant improvement of the results. Due to its simplicity and efficiency, center-based clustering is often used as a first evaluation strategy of large data sets and the basis of comparison when transitioning to other more complex and time-consuming algorithms.

Hierarchical clustering

The hierarchical clustering strategies inevitably fall into two categories: agglomerative or divisive. The agglomerative method, also called bottom-up, consists in building a binary merge tree, starting from the whole set of data elements, and proceeding by merging each pair of objects according to a proximity criterion. Therefore this process starts with a full set of clusters, one for each object, and ideally ends with a single clusters that agglomerates the whole data set. The divisive approach, on the other hand, does the opposite starting from a single cluster that contains all the object and successively divides itself into several clusters until a desired number of clusters is reached. For this reason the divisive method is also called top-down approach.

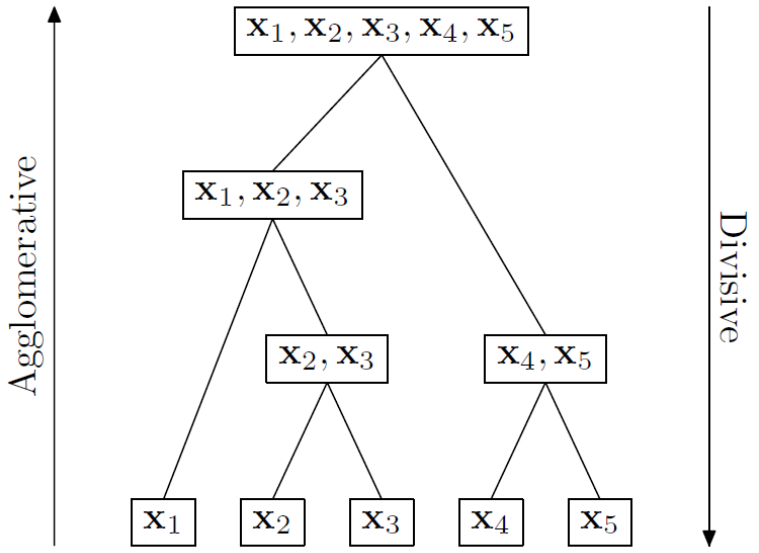

Figure 3.19: *Agglomerative hirarchical clustering and divisive hierarchical clustering (Gan et al., 2007)*

Hierarchical algorithms can be expressed either by a matrix of abstract symbols or through graph theory which is usually much easier to interpret. The graph is generally called a tree, from which the most used one is called Dendrogram. Dendograms are trees in which each merging node is associated with a specific height. A low value of height represents a high similarity between the two merged clusters or objects, whereas the maximum height will comprehend, ideally, all the clusters or objects and thus represents a lower similarity between them. Since each object is continuously compared to every other, the computational complexity of this method is high and therefore it is not recommended for large data sets.

The main drawback of this strategy is that if any object is incorrectly grouped at an early stage, it cannot be reallocated. Another limitation, similar to the k-means algorithm, is that in order to decide whether two objects must be agglomerated or divided, the algorithm must know a priori the parameters that defines those similarities. A variation in these similarity measures may lead to totally different results. In general, hierarchical cluster analysis is useful when a good amount of information regarding the objects is known. As an example *Divya and Vijayalakshmi* (2015)0 used agglomerative hierarchical clustering in order to detect wild fires by comparing the RGB values satellite images with known values. After plotting a Dendogram, the direction in which the fire propagates can be predicted from the clusters the algorithm creates.

Density-based clustering

In front of all limitations that the hierarchical and center-based cluster analysis have, *Ester et al.* (1996) created a new clustering algorithm, called DBSCAN, that manages to find arbitrary shapes with only one scan of the original dataset. DBSCAN stands for *Density-Based Spatial Clustering of Applications with Noise* and as its name suggests, forms clusters around high density areas. Objects that are sparse and do not belong

to any cluster are defined as noise. The notion of noise makes it more robust to anomaly detection, also called outliers, which are events that differ significantly from the majority of the data.

DBSCAN requires two parameters in order to perform the cluster analysis: neighbourhood of an object (point), named *Eps*, and the minimum points named *MinPts*. In each cluster there will be two kinds of points: points inside the cluster (core points) and those that stand on the border (border points). In general, the neighbourhood *Eps* of a border point contains significantly less points then a core point. For this reason, a simple approach that would require for each point in a cluster to have at least the minimum points *MinPts* in an *Eps* neighbourhood would fail. To deal with this situation, the requirement is that for every point p in a cluster C, there will be another point q so that p is inside the Eps_q and Eps_q has at least a $MinPts$ number of points. In this situation p and q are said to be directly density-reachable. If p and q are not directly neighbours but there is a chain of directly density-reachable points between them, then p and q are defined as density-reachable. At last, if two points p and q are both density reachable of a third point o, they are defined as density-connected.

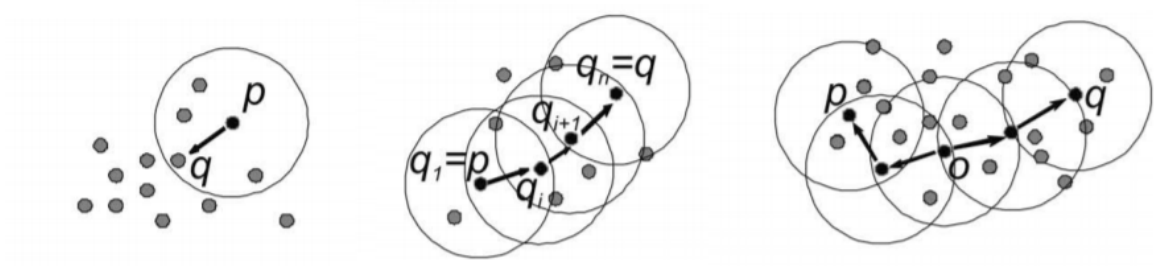

Figure 3.20: *Example of direcly density-reachable points (left), density-reachable points (center) and density-connected points (right). Adapted from Ester et al. (1996)*

Based on these definitions, *Han et al.* (2012) describes DBSCAN algorithm in a set of steps that can be summarized as:

1. User defines Eps and $MinPts$.
2. Starting from a random point, draw the corresponding Eps neighbourhood and check:
   (a) If there are at least $MinPts$ number of points inside Eps, classify this point as a core point.
   (b) If condition is not satisfied, check for each other points inside Eps:
      i. If there are points that satisfy the condition 2(a), then classify those as core points and the starting point as border point.
      ii. If no other point satisfies condition 2(a), classify the starting point as noise.
      iii. Continue until all points have been classified.
3. Group core points with all density-reachable points around them, progressively numbering the resulting groups. The total number of groups will be the number of clusters.

4. For each border point, assign it to a cluster based on the nearest core point.

As it can be seen, in opposition to k-means, DBSCAN does not require the user to know a priori the total number of cluster, as it will automatically detect and agglomerate the objects creating the clusters based on the density. The notion of density of the data set is more easily to be evaluated by the user, instead of the final number of clusters which is usually the answer that is demanded by the cluster analysis. Another advantage of the density-based algorithms is that they can work with high-dimension spatial analysis, since it can create arbitrary shaped clusters.

Since it was firstly created, several improvements were proposed to DBSCAN especially in optimizing the choice of the initial parameters Eps and $MinPts$. An example of this optimization could be making Eps an adaptive parameter that finds an appropriate value for each cluster. As proposed by *Zhu et al.* (2018) an estimation of this parameter could be achieved based on Gauss kernel density theory, creating a self-adapting algorithm at the expense of a higher computational complexity. This strategy could solve one of the drawbacks of the algorithm, which is that DBSCAN cannot cluster data sets with a large variability in density since the combination of initial parameters cannot be optimized for all clusters.

Ankerst et al. (1999) proposed an upgraded version of DBSCAN called OPTICS, which ended up being also very popular. There is also cases in which a border point can be reachable from more than one cluster and therefore it will be assigned based on the order that the data are processed. In most scenarios this event has little impact on the cluster results since core points and noises are not affected by it, but it demonstrate a flaw that makes the algorithm non deterministic.

Model-based clustering

Having a completely different approach, model-based clustering views the data set as a mixture of probability distributions, each of which represents a different cluster. This already shows one of the flaws of this clustering strategy, which is that it relies on the assumption that every data set can be accurately modelled mathematically. For this reason the performance of the model-based cluster analysis strongly depends on the conformity of the data set to the model created.

Figure 3.21 shows a flowchart that summarizes this clustering technique. For each element in the data set a distribution model is created, a Gaussian mixture model for example, which is defined by standard statistical parameters such as means and co-variances. These statistical parameters are initially set using an agglomerative model-based, which uses the same general ideas as the agglomerative hierarchical clustering described in section 3.4. The results are the input of an Expectation-Maximization algorithm which estimates the final parameters. The number of clusters and the distribution of the component densities will produce different models for the data set, and the final model is determined by the Bayesian information criterion (BIC). The model with the highest BIC value is chosen as the best model for that data set.

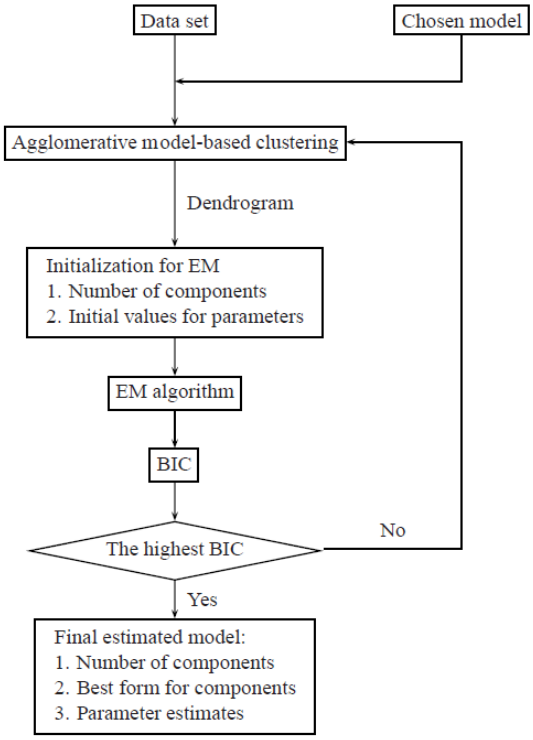

Figure 3.21: *Flowchart of the model-based clustering procedure (Gan et al., 2007)*

4

GISEle: GIS for Electrification

Making use of the all the theoretical background described in chapter 3, in 2019 the GISEle application was proposed as an effort to improve the planning of rural electrification in developing countries (*Carnovali and Edeme*, 2019). It is supported by and part of the initiative Energy4Growing (*e4g.polimi.it*) headed by the Energy Department of Politecnico di Milano, which promotes research to propel the access of electricity in developing countries. GISEle is an on-development python-based open source procedure that, starting from input GIS data, provides the least cost electrification solution to connect loads in rural areas, choosing among on- and off-grid strategies. Within this thesis work the procedure's algorithm have been improved and tested with a real case study with the goal of increasing the efficiency and output reliability .

4.1 Introduction to GISEle

As briefly described in chapter 1, the goal of GISEle is to propose a detailed topology of the electric grid connecting the majority of people inside a rural area and then, if necessary, connect it to an existing national grid. Also an off-grid solution using different types of energy sources would be evaluated and their costs compared to the previous solution. Figure 4.1 presents a flowchart that summarizes the whole procedure.

Starting from the left part of the flowchart, the first step is gathering GIS and social data about the area to be electrified. Example of information required are population distribution, terrain characteristics, roads, substations available and the existing grid if present. All this data is combined into a single file that embodies all the information and performs a weighting strategy. The result of this initial process is a grid of points that covers all the studied area in which each point's assigned weight represents the difficulty in building an electric line through it. To manage this information, and also to display the results in a more comprehensive way, GIS software are used: such as ArcGIS developed by Esri or the free and open-source QGIS. The weight as-

signment is performed inside GISEle's python platform, creating the penalty factors to the grid of points.

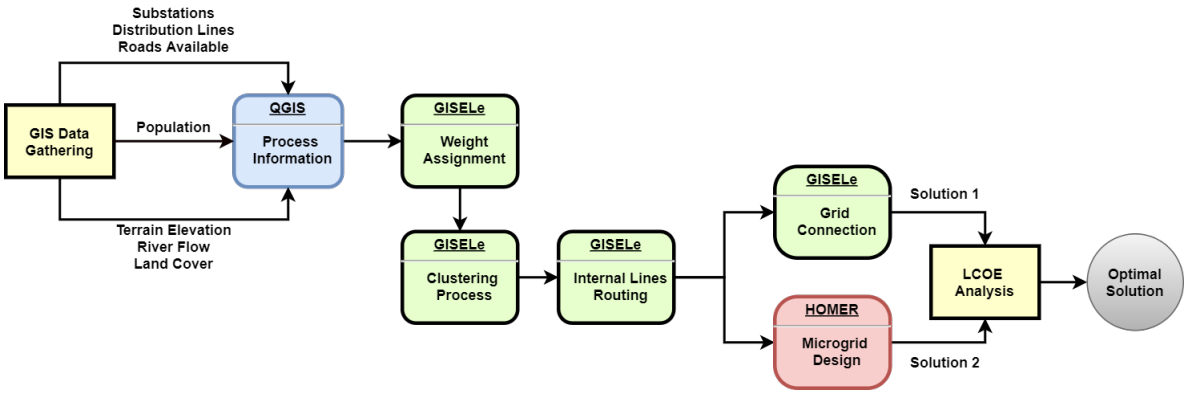

Figure 4.1: *Flowchart of GISEle's initial concept*

The next step is to run a cluster analysis to identify densely populated areas and divide the total studied area in smaller areas called *clusters* where each of them constitute a separated energy community.

After defining the clusters two different solution paths are created: the first will focus on connecting clusters to an existing grid and the second on designing isolated microgrid solutions. Both of them require that each individual cluster grid to be designed to connect the internal population. Then, following the upper path of the right part of figure 4.1, a connection is created between each cluster internal grid that supplies and the nearest substation of the existent distribution network. All that is made taking into consideration the least costly path using the strategies previously described in section 3.1. Following the bottom path GISEle provides an evaluation of the cluster's energy needs, and then design an optimal power system capable of supplying this demand. Because of the complexity of sizing generators and designing microgrids, as it was explained in section 2.4, a third-party software used in order to provide the second solution based on the load profile information.

The two solutions provided by GISEle are finally evaluated in terms of LCOE to propose the best solution for each cluster, and if it is economically feasible to connect it to the main grid or if an isolated microgrid solution is better suitable. Although this comparison is considered an important aspect of the whole procedure, and efforts have been made in order to incorporate generation sizing inside GISEle's platform, for the purpose of this thesis the off-grid solutions are not ideal. The reasoning behind this claim is that the main interest of the partner company Enel Global Infrastructure and Networks S.r.l (*Enel GI&N*), the ones that requested the rural electrification analysis, is to achieve a ratio of 100% of electrification by connecting all the population to the distribution grid. For this reason providing and evaluating off-grid solutions is outside of the scope of the proposed work.

Case study of Namanjavira

Within the thesis work of *Carnovali and Edeme* (2019) GISEle was tested in the province of Zambezia in Mozambique. This is an area of around 100 km^2 and a population of of five million inhabitants, of which 93% lives in rural areas characterized by a very low resilience in face of climate change and external shocks and where 70.5% of the population lives below the poverty line. The results of the analysis, which can be seen in figure 4.2, shows the region of Zambezia called Namanjavira in which the cluster analysis and grid routing was made.

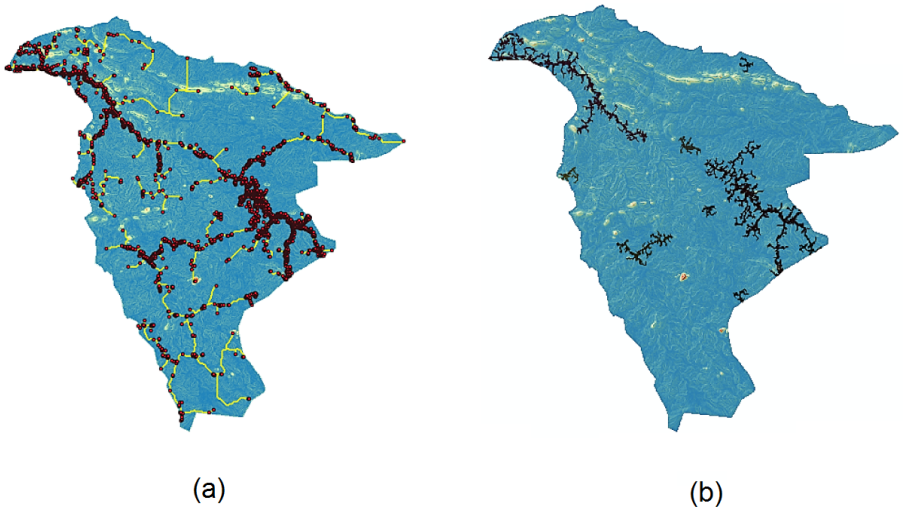

Figure 4.2: *Results obtained using GISEle in the area of Namanjavira in Mozambique (Carnovali and Edeme, 2019)*

At one side, shown in figure 4.2 (a), it demonstrates the results of the grid routing algorithm when applied in the whole area, while figure 4.2 (b) shows the internal grid of clusters created around the most densely areas only. Using the latter strategy GISEle was able to create internal grids capable of connect 60% of the population of Namanjavira, connect those grids where it is economically feasible while providing isolated microgrid designs for the less dense and sparse areas. By this case study it has been concluded that the approach proposed can improve the network expansion and the energy resources allocation analysis of the country, being a helpful tool for the *Renewables Energy Atlas* of Mozambique as an example.

4.2 Gathering and managing GIS data

In this section the first two blocks of the procedure will be discussed in detail. There are two main sources of information from which the majority of the data required to run GISEle come from: the country's government website and public data sets made available by international institutions. The quality of the information that is gathered in this first phase is crucial to ensure the validity of analysis and the reliability of the results.

Population density

The population density is arguably the most important input information for GISEle. It is through this information that the algorithm will perform its cluster analysis and select which areas should be electrified, and moreover through which points should the new electric grid pass. The University of Southampton together with several partners created the data bank *WorldPop* to help improve the spatial demographic evidence base for low and middle income countries. The platform provides information regarding population dynamics, poverty, births and many other aspects worldwide. For mapping settlements for example, it is used satellite imagery with a 30 meters spatial resolution and through interpretation of those images the population is estimated and mapped (*Gaughan et al.*, 2013). Figure 4.3 shows an example of the usage of satellite imagery analysis for population mapping.

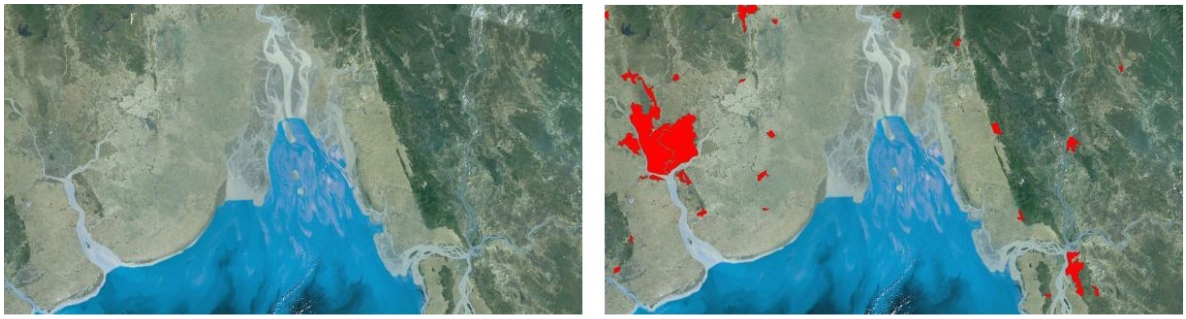

Figure 4.3: *(a) Landsat ETM image of Yangon and surrounds in Myanmar; (b) Settlement extents used in WorldPop mapping*

Another, and more traditional, way of mapping population density is through a census procedure. It is a highly accurate house to house process that tries to account for every person in absolute numbers. Since the scale of population nowadays is too high, governments worldwide utilize statistical analysis and information technology in a process that can be contrasted with sampling. In general, the quality and accuracy of data obtained through census made by governmental institutions are higher than the ones found in public data bases.

Other necessary data

Besides population density, GISEle requires other information in order to create the weighted grid of points that will be used in the cluster analysis and for grid routing. In order to create a representation of the difficulty in deploying the electric lines between the points, the weight assignment strategy evaluates the following topological data:

Elevation

One of the most fundamental and useful topographic measurement is the evaluation of the land elevation, also called hypsometry. In order to measure the elevation of a terrain a fixed reference point must be set, usually this being is the Earth's sea level,

and vertical distance between the point to be measured and this reference point is what defines the elevation. When above the sea level, the elevation can also be referred to Altitude. With technologies such as GIS and ortho-rectified imagery (see section 3.2) the ability of representing terrain elevation of areas became even faster and precise. It is common to see famous applications such as *Google Maps* able to provide a highly detailed three-dimensional Digital Elevation Model (DEM) of the Earth. These models can be represented as a raster layer, as depicted in figures 3.10 and 3.11, but also as vector points. There are several public data bases where elevation data can be found, for example the United States Geological Survey (USGS) provide a DEM of the whole world with a resolution of 30 arc-second, which corresponds to approximately 1 kilometer, called *GTOPO30* (*USGS*, 2020). According to the USGS, the vertical accuracy of the model is around 30 meters.

Slope

Intrinsically related to the land elevation, the slope (or grade) represents the inclination or the tangent of the angle that a surface forms with respect to the horizontal plane. The slope angle can be computed using the equation 4.1.

$$\alpha = \arctan \frac{\Delta h}{d} \tag{4.1}$$

Where,

- α is the slope angle;
- Δh is the variation of elevation;
- d is the distance along the surface.

The ratio $\frac{\Delta h}{d}$ can also be expressed as a percentage. The slope is extremely important for planning roads and railways. Land vehicles are usually rated based on their ability to ascend terrain, and trains in general have a much higher sensitivity in terms of inclination. For an increase of 1% in the slope, a locomotive can pull half of the load that it can pull on level track. The slope values can be acquired by using GIS analysis and working with elevation raster layers. Many GIS software, such as QGIS and ArcGIS, have functions capable of computing it through spatial analysis from a input raster layer of a DEM in a extremely fast manner.

Land cover

Land cover represents the physical material of a surface, which should not be mistaken with *land use* that represents how people utilised the land as a social-economical activity. There are several types of land covers such as: water, grass, asphalt etc. And the main issue is that different institutions define these types of land cover in a different way. This thesis uses the definition proposed by the *Global Land Cover 2000* (*JRC*, 2000) project that the European Joint Centre Research (JRC) developed. GLC2000 divides the land cover in 22 different types that can be seen together with the penalty factor assigned to each of them in the table 4.2.

Roads

The last input information required to run GISEle is the road map of the studied area. *OpenStreetMap* (OSM) is an open database that started in 2004 and is a collaborative project similar to *Wikipedia*, where volunteers around the world work to improve and keep the information updated. It has 2 million registered users who can collect data using manual survey, GPS devices, aerial photography, and other free sources. Throughout the years OSM became a reliable source of information, being often favourably compared with proprietary data bases such as *Google Maps*.

Generating the input files

With the support of a GIS application, such as QGIS and ArcGIS, all the input data are combined into a single input file. In order to do, first a grid of points that covers all the studied area must be created with a specific distance between each point. This distance defines the *resolution* of GISEle's analysis. The *resolution* is an important aspect of the methodology, it has to be selected according to all the GIS data that was gathered to find a suitable number. For raster layers, as described in section 3.2, the resolution can achieve high numbers due to the high technology of satellite imagery. A resolution between 100 meters and 1 kilometer is preferable. Below this range the number of points to be processed is too high an could become too complex to process, and above this range the accuracy of the analysis can be impaired.

For Low Voltage (LV) grids, a resolution of 1 kilometer can already be impairing as it require grid routing through small streets and each detail is important. High Voltage transmission grids (HV) however, does not requires high spatial detail as it usually covers long distances: a resolution of 200 meters could be unnecessarily complex. Figure 4.4 (a) shows an example of the grid of points with resolution of 1 kilometer around a small village. Note that only two points are inside the village, which indicates that a grid routing using this resolution will result in a straight line crossing the whole village. This information is not sufficient to represent a LV grid, which have to follow the village's roads to connect the houses that are cannot be seen due do the low resolution.

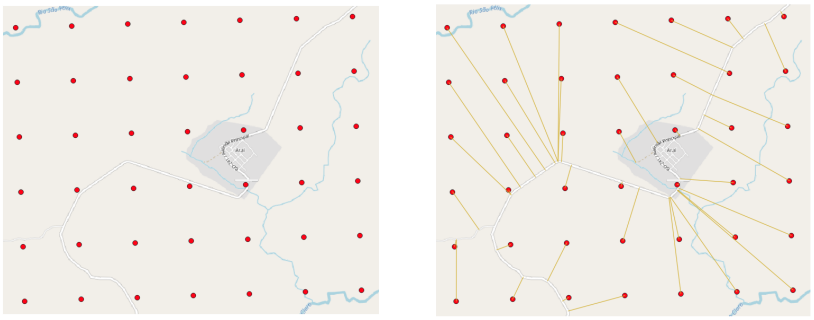

Figure 4.4: *(a) Grid of points with resolution of 1 km; (b) Example of distance to the nearest line function*

After the grid of points is created and the resolution of the grid is set, the information of all the input layers have to be associated to each point of the grid. For

raster layers such as the elevation layer shown in figure 3.10, a sampling function can be used. The *sampling raster values* function allows a point vector layer, such as the grid of points created, to use points to sample values from the raster layer and add its values (also called attributes) to it. Population density, elevation, slope and land cover in general are represented as raster layers. The population density can be also represented with polygons or points, in which case other functions such as *join attributes* can be used to add its values to the grid of points. The last element that is accounted for computing the weight is the distance from each point of the grid to the nearest road. Again, with the support of GIS software using a function such as *distance to the nearest hub* it is possible to compute from a point vector layer the distance to a nearest line vector layer. This procedure is shown in figure 4.4 (b).

After combining all the data into a single layer file, called *shapefile* (extension .shp), it is then exported as a table of attributes of every point in the grid. Each row of the table represents a point of the grid and each column represents an associated information relating the point to its population, elevation, and other gathered data . Besides the grid information, another file must be created containing the information regarding the substations available for connection. This file is simpler and is also a point vector layer where each point represents a substation and its type, that could be high voltage, medium voltage or low voltage. To each of these types it is assigned a power limit for connection and an extra cost if the connection is to a HV substation. The usage of this information and why it is necessary will be discussed in section 4.5. This is also combined in a single *shapefile* that is exported as table to form the two input files required by GISEle. Five additional input parameters are required by the routine. The first two parameters are the resolution used by the user to create the grid of points and the CRS of the GIS information gathered. The third parameter is the line base cost that is used to compute the cost of the electric line deployed. The fourth one is the population threshold that is used to select the target nodes that should be electrified. A population threshold of 5 means that every node, that is inside a cluster, containing at least 5 people will be a target node used by the routing algorithm for creating the cluster grid. This parameter is particularly important for the main branch approach proposed in this thesis, which will be described in section 5.3. In order to compute each cluster's load and to size the electric cables, an estimation of the last parameter (load per capita) is necessary. This number may vary depending on the country and on the area that is being analysed. Usually the local DSO has enough information to provide an accurate estimation or an average value based on the household consumption. A summary of all the input information is listed in the table 4.1.

Grid File	Substation File	Parameters
Point Identification (ID)		
Point Coordinates (X, Y)	Substation Identification (ID)	Resolution [m]
Population	Substation Coordinates (X, Y)	Coordinate System (CRS)
Elevation	Substation Type	Line base cost [€/km]
Slope	Power limit for connection	Population threshold
Land cover	Extra cost for HV connection	Load per capita [W]
Distance to the nearest road		

Table 4.1: *Summary of input information given to GISEle*

4.3 Weighting strategy

The first process GISEle performs is the weighting strategy of the input grid of points. Each point is embedded with spatial characteristics that impact the cost of deployment of a line. As discussed in section 3.3 there are many forms of representing these impacts. A combination of some elements of the methodology of *Monteiro et al.* (2005), and the penalty factor strategy used in the OnSSET tool, was considered the best approach. A unitary penalty factor, which represents the base cost (i.e. the optimal conditions for building electrical lines) is increased according to the topological characteristics of the weighted point. The total penalty factor is, therefore , a summation of all penalty factors assigned to each of the following characteristics of the grid points given by table 4.1: slope, land cover and distance to the nearest road. The proposed penalty factor is defined by the equation 4.2.

$$PF = 1 + \sum_{i=0}^{n} P_i \tag{4.2}$$

Where,

- PF is the total penalty factor a point;
- n is number of topological aspects that will be evaluated (in this case $n = 3$);
- P_i is the penalty factor associated with the topological aspect i.

Population density and terrain elevation do not impact the difficulty of line deployment, but can be indirectly related to costs. Elevation is used in GISEle as a third coordinate to compute the three-dimensional distance between two points, and even though the distance of a line is inherently connected to its costs, the distances and weights are computed in distinct ways. The distance to the nearest road weight factor is modelled as a linear curve with a max limit, as shown in figure 4.5. The availability of a nearby road to allow fundamental logistics, such as transport of material and personnel, is of extreme importance for a project of line deployment. The further the line is from a road, the higher its costs as it is assumed a limit of the penalty factor of 6 times the basic cost for points farther than 1 kilometer of the nearest road.

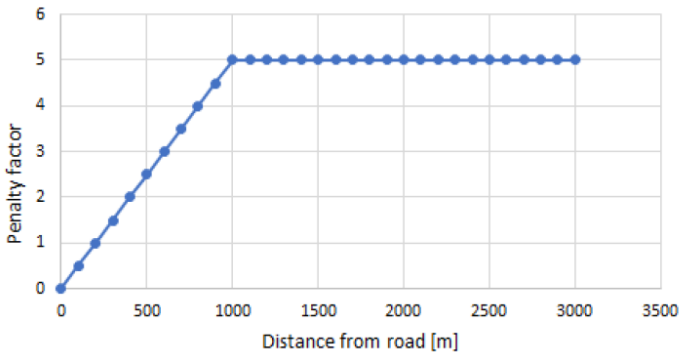

Figure 4.5: *Modelling of the penalty factor associated with the distance to the nearest road (Carnovali and Edeme, 2019)*

Instead of a linear model, the slope penalty factor, which is shown in figure 4.6, has an exponential profile which is designed to have a return value of 1 at 35 degrees. It is quite intuitive that the steeper a terrain is, the harder it is to reach and build an a line on it.

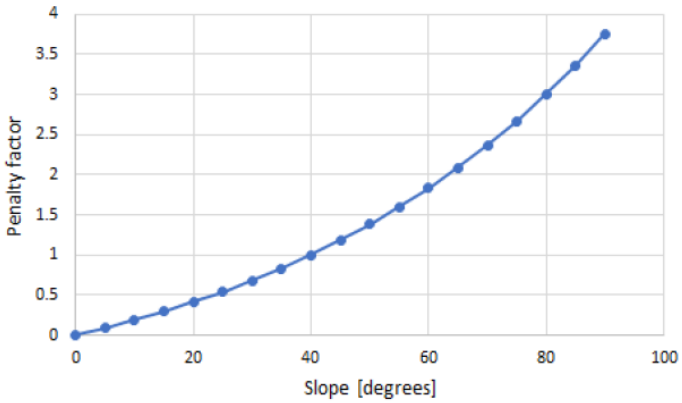

Figure 4.6: *Modelling of the penalty factor associated with the terrain slope (Carnovali and Edeme, 2019)*

Lastly, the penalty factor associated with the land cover is not defined by a function, instead a value is directly assigned depending on the type of terrain (types are defined based on *GLC*2000, see section 4.2), as it can be seen in table 4.2.

Type	Penalty Factor
Tree Cover, broadleaved, evergreen (>15% tree cover)	1
Tree Cover, broadleaved, deciduous, closed (>50% tree cover)	5
Tree Cover, broadleaved, deciduous, open (15-40% tree cover)	1
Tree Cover, needle-leaved, evergreen	3
Tree Cover, needle-leaved, deciduous	3
Tree Cover, mixed leaf type	3
Tree Cover, regularly flooded, fresh water	8
Tree Cover, regularly flooded, saline water	8
Mosaic: Tree cover/Other vegetation	2
Tree Cover, burnt	2
Shrub Cover, closed-open, evergreen	1
Shrub Cover, closed-open, deciduous	1
Herbaceous Cover, closed-open	1
Sparse Herbaceous or sparse Shrub Cover	1
Regularly flooded Shrub and/or Herbaceous Cover	6
Cultivated and managed area	1
Mosaic: Cropland/Tree Cover/Other natural vegetation	1
Mosaic: Cropland/ Shrub or Grass Cover	1
Bare Areas	1
Water Bodies	9
Snow and Ice	7
Artifcial surfaces and associated areas	1

Table 4.2: *Penalty factors for different types of land cover. Adapted from Carnovali and Edeme (2019)*

Harsh terrains such as water bodies, ice and dense forests are assigned with high penalty factors that can increase up to 9 times the base cost of a line. In general, grid points that are nearby roads tend to have a much lower penalty factor since roads are

usually made within a low limit of slope degree and on good terrain or land cover. Consequently all three penalty indices have a low number, resulting in a overall low penalty factor. This effect can be seen in the figure **??** in chapter **??**.

4.4 Clustering

After collection, processing the input data and creating the weighted grid of points, the cluster analysis is performed. This process is made by making use of the population attribute that each of the points of the grid has. Clustering will allow GISEle to take advantage of the points position by grouping the ones that are populated and are nearby, thus creating the clusters. Using cluster analysis is useful because of the following benefits:

- **Reduction of the overall cost per capita for connection.** Focusing on densely populated areas allows for electric lines to connect more people per kilometer of line deployed and therefore reducing the overall cost per capita for person connected. Even if the goal is to connect every single person in an area, simply creating a grid that connects every populated point is often not financially feasible. Therefore, clustering allows to focus on densely populated areas and increases the amount of people that can be connected by a set amount of lines or costs.
- **Reduction of the computational burden.** Another fundamental benefit of the cluster analysis is to allow the grid routing algorithm to focus on a smaller portion of points each time. By transitioning from cluster to cluster, GISEle is able to perform faster than it would if the whole studied area were used. In fact, in this way the procedure is able to analyze vast areas requiring a reasonable amount of computational time and not exceeding RAM limits. By clustering, the amount of data involved in each routine process, and so the memory required to manage them, is limited.

From all cluster strategies described in section 3.4, the one that better suits GISEle's approach is the density-based cluster analysis. Since there is no hierarchy relations between points, hierarchical clustering is the less suitable solution. Center-based algorithm characteristic of using only convex shapes, and the requirement that the users must know *a priori* the final number of clusters, makes it not particular appropriate. Lastly the model-based clustering algorithms would require a huge effort into modeling and finding probability functions that can fit the population distribution of the studied area. This is particularly harder when rural areas are considered since the spatial distribution of houses generally does not follow any specific planning, as opposed to urban areas.

Therefore, GISEle's clustering algorithm is density-based and is based on $DBSCAN$. However a slight adaptation is made in order to better fit GISEle's goals: the points are weighted according to the population. Consequently the variable $MinPts$ will not evaluate the minimum number of points in a neighborhood to determine if it is a terminal node, but instead it will evaluate the minimum number of people. Moreover, the main drawbacks of a density-based cluster algorithm (see section 3.4) does

not affect the suitability of this algorithm for GISEle's purposes. For example, the non deterministic characteristic of neighbour points will not affect the final electrification status of the points, but rather only the cluster assignment.

In order to help define the two initial parameters *MinPts* and *eps*, GISEle uses the high process speed of *DBSCAN* to compute several results at time, returning four tables containing additional information of the cluster results. These tables have *MinPts* as columns and *eps* as rows, giving the additional information result for each combination of them. The first table presents the number of clusters; the second presents the ratio between the clustered area and the total area in percentage; the third presents the ratio between the clustered people and the total population in percentage; lastly the fourth table presents the ratio of clustered people and clustered area. An example of this output is presented in tables 4.3 to 4.6. There is a trade-off between these four indicators, and the ideal value of *MinPts* and *eps* will depend on which of them are more valuable for the analysis. For example, in cases which the economical aspect is the most relevant: the reduction of the clustered area and a high ratio of people per area is preferable, since these reflect the length of cables deployed and therefore the overall cost. For cases in which the microgrid solutions are cheap and available, the total number of clusters is not an important indicator since each tiny cluster can have a small off-grid solution.

	10	40	70	100	130	160
500	133	17	13	10	8	7
1000	108	19	8	7	6	5
1500	43	20	9	9	5	4
2000	32	19	10	8	6	5
2500	22	16	15	9	6	6
3000	17	16	14	8	6	6
3500	11	14	11	11	7	6
4000	6	11	9	8	8	7
4500	1	10	11	7	7	7
5000	1	9	10	9	8	7

Table 4.3: *Number of clusters for MinPts range of 10-160 and eps range of 500-5000*

	10	40	70	100	130	160
500	1	0	0	0	0	0
1000	8	1	0	0	0	0
1500	29	8	3	2	1	1
2000	35	11	5	3	2	2
2500	55	22	13	9	6	5
3000	66	30	19	12	9	7
3500	77	40	26	19	13	11
4000	85	49	32	25	19	16
4500	91	61	45	35	28	23
5000	94	68	52	43	34	27

Table 4.4: *% of clustered area over the total area for MinPts range of 10-160 and eps range of 500-5000*

	10	40	70	100	130	160
500	77	58	56	53	51	49
1000	87	65	59	57	56	54
1500	95	79	67	65	60	58
2000	96	82	71	66	62	60
2500	98	88	81	72	67	66
3000	99	91	85	75	70	69
3500	99	94	87	82	75	70
4000	99	95	89	84	78	75
4500	99	97	93	87	84	79
5000	99	98	95	91	86	81

Table 4.5: *% of clustered people over the total population for MinPts range of 10-160 and eps range of 500-5000*

	10	40	70	100	130	160
500	54.14	318.94	401.69	497.6	595	659.43
1000	12.86	64.99	114.27	141	164.28	196.35
1500	4.32	12.89	23.65	30.33	42.90	52.83
2000	3.59	9.72	17.24	21.97	30.82	36.26
2500	2.34	5.25	7.79	10.35	13.62	16.11
3000	1.97	3.98	5.78	7.82	10.13	11.45
3500	1.68	3.09	4.36	5.47	7.06	8.16
4000	1.54	2.56	3.56	4.31	5.41	6.10
4500	1.43	2.06	2.69	3.25	3.86	4.47
5000	1.39	1.88	2.36	2.75	3.26	3.84

Table 4.6: *Ratio between number of people and total area for MinPts range of 10-160 and eps range of 500-5000*

In general, a good cluster analysis will try to minimize the number of clusters and total clustered area while maximizing the amount of clustered people. The whole point of using density-based clustering is to create groups of individuals based on the density, this goal translated to the analysis made by GISEle is reflected on increasing the population density. This population density is reflected in the ratio of clustered people and clustered area of table 4.6, the higher this ratio the more successful the cluster analysis is. It is also possible as an after-process, to merge clusters if the clustering result is not sufficient and end up creating clusters too near. At the end of the cluster analysis, all points have two additional attributes assigned to each of them: weight and cluster identification.

4.5 Grid routing

Creating a graph

The following step is to create an electric grid connecting the populated points inside each cluster. The input weighted grid of points is first transformed in a graph $G = (V, E)$, where each edge weight connecting two vertices u and v is calculated by equation 4.3. In the input grid, each point has 8 other neighbours and therefore 8 edges connecting each pair of point as it can be seen in figure 4.7.

$$C_{u,v} = L_{u,v} \times BC \times \frac{p_u + p_v}{2} \tag{4.3}$$

Where,

- $C_{u,v}$ is the cost of connecting vertices u and v in Euros (€);
- $L_{u,v}$ is the distance between vertices u and v in km;
- BC is the base cost of an electric line in [€/km];
- p_u and p_v are the penalty factors assigned to the vertices u and v.

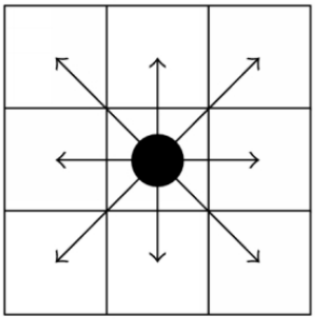

Figure 4.7: *Edges between one point and its eight neighbours (Carnovali and Edeme, 2019)*

The weight assigned to an edge is therefore the arithmetic mean of the weights of the two vertices that edge connects. The base cost of an electric line is one of the input parameters, and must be estimated and given by the user depending on the type of line that is being deployed. The 8 neighbouring points of a point u are all the points at a distance smaller than the diagonal length of the grid, given by equation 4.4. The $\sqrt{2}$ factor is necessary to include the points that are diagonally neighbours.

$$d_{u,v} \leq \sqrt{2} \times resolution \tag{4.4}$$

Steiner tree creation

The Python package *NetworkX* allows for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. One of the many functions it provides is the ability to create an approximated Steiner tree by computing the minimum spanning tree of the sub-graph of the metric closure (G_S) of the graph induced by the terminal nodes T (described in section 3.1). The function asks for a weighted graph $G = (V, E, w)$, which is the *Edge cost matrix* previously computed, and also the terminal nodes that must be connected. The terminal nodes are all the populated points inside the cluster that is being analysed (this is made one cluster at time), or the points in which the population is higher than a set threshold. The Steiner ratio (ρ) achieved by this approximated solution is given by the equation 4.5 and varies from 1 (in a really simple tree with only two terminal nodes) to 2 (high amount of terminal nodes)

$$\rho = 2 - \left(\frac{2}{T}\right) \tag{4.5}$$

There are two data sets being processed: the weighted graph G that contains information about all the points of the initial input grid and the terminal nodes T that contains information only of the populated points of a given cluster. The terminal nodes considered for creating the cluster grid are all nodes in which the population is higher or equal than the *population threshold* parameter given by the user. The computational complexity of the Steiner tree problem is equal to $O(G \times T^2)$. Running the function with a big data set as G (as it contains the whole set of points) would require a huge amount of time especially at high resolution. For this reason, similar to what is done with T, only the points that are assigned to the cluster being analysed are selected. This way both G and T are reduced to evaluate only the points inside each cluster per time.

Figure 4.8 shows the result of GISEle's routing algorithm, in which:

- Green lines represent the computed Steiner tree;
- Red points represent the terminal nodes;
- Blue and yellow points are points which belong to two different clusters;
- Black points represents populated points too sparse to be clustered;
- Brown points represent the whole input weighted grid of points;
- The darker area is outside the region studied for electrification.

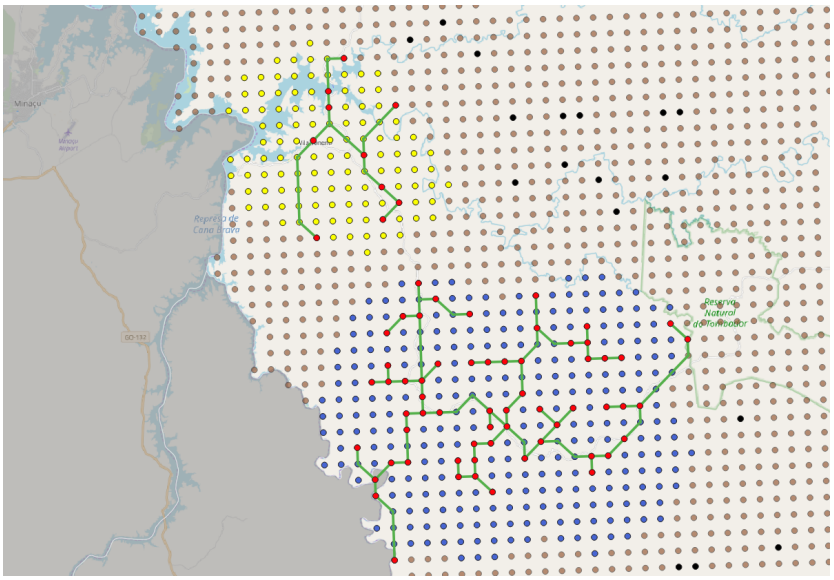

Figure 4.8: *GISEle's Steiner tree approximation for two different clusters for a resolution of 1 km*

Depending on the resolution adopted and the size of the area to be electrified, computing the Steiner tree can be complex. If the studied area is densely populated

and the cluster analysis results in big clusters with a high amount of terminal nodes T to be electrified, computing the Steiner tree could be unfeasible. To deal with this situation, the user could increase the *population threshold* and reduce the amount of terminal nodes that require electric connection. In cases where increasing the *population threshold* is not ideal, other solutions can be utilized, such as the alternative routing algorithm proposed by *Carnovali and Edeme* (2019).

The output of this step is the design of the grid topology at minimum cost for each cluster as shown in figure 4.8. These graphs represent the rural electric grids created for each non-electrified small community, considering the cheapest path possible. After the creation of the cluster internal grids, these rural communities can be connected to the existing electric grid at the nearest substation available, or a local microgrid generation can be sized offering an off-grid solution.

HV/MV Substation connection

In order to create a substation connection between the electric grid within each separate energy community to the already present national grid, all the available substations are evaluated considering two factors:

- Distance to the substation

This factor is straight forward: since the cost of an electric line is proportional to its length, it is evident that between two substations of the same type the nearest one should be chosen for connection in order to reduce its cost. The distance is computed among each cluster point and each available substation, and the substation that has the lowest distance to the cluster is assigned for connection. The nearest substation is always assigned unless other factors are taken into consideration, such as the type of substation.

- Type of substation

Depending on the amount of population, which can be translated into the amount of load, a suitable substation must be chosen for connection. It would not be realistic and rather unfeasible to connect a small community accounting for a nominal power of 10 kW to a HV/MV substation. As discussed in chapter 2, the costs of deploying a new MV line, protections and a new HV/MV transformer does not offset the gain in electrification of such small community. Moreover, the technical requirements for connecting to HV substations are stricter, for example in terms of power factor, and could be not achievable for a long rural MV. On the other hand, a big cluster with dense population translated into a heavy load, could not be supplied by a nearby low power MV substation. In this situation the connection to a farther substation is necessary since the connection to the nearest one is technically unfeasible. In terms of rural areas the first scenario is more likely to happen, but in any case these examples endorse why choosing a substation solely based on the distance is not sufficient.

An evaluation of these two factors is recommended, and in order to do so GISEle uses the other information present in the substation table file and the parameter load

per capita (see table 4.1). The load per capita is used to compute the total amount of load required by each cluster grid, which is defined by equation 4.6.

$$C_{Load,k} = \sum_{T} P_T \times LpC \tag{4.6}$$

Where,

- $C_{Load,k}$ is the total load of the cluster k;
- P_T is the population of the terminal node T;
- LpC is the load per capita parameter.

Based on the total amount of load (C_{Load}) and the substation input file that defines their power limits for connection, GISEle assigns the most suitable substation to each cluster k using the procedure shown in the flowchart of figure 4.9, which can be summarized in the following steps:

1. From all the available substations, remove the ones in which $C_{Load,k}$ is outside the substation's power limitation range.
2. For each of the remaining substations, do:
   (a) Find the nearest terminal node of cluster k.
   (b) Compute the 3D distance between the substation and the nearest terminal node.
3. Assign to cluster k the substation that has the shortest 3D distance.

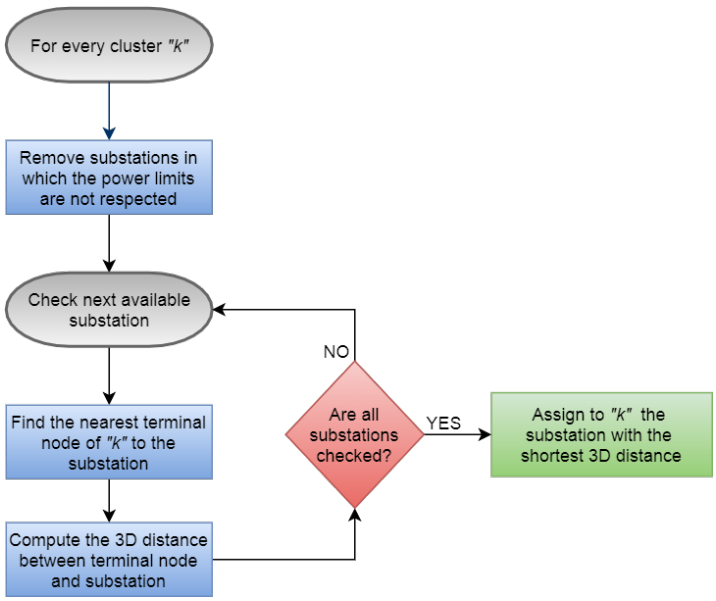

Figure 4.9: *Flowchart of GISEle's substation assignment routine*

This procedure allows to identify for each cluster a terminal node and a point of connection to the national grid (substation). In order to connect only these two

points the Steiner tree algorithm is not necessary: as it becomes a *shortest path problem* the Dijkstra algorithm is faster and more suitable. Using the same Python package *NetworkX*, another function allows for computing the shortest path between a source point (terminal node) and a target point (substation assigned) through a weighted graph $G = (V, E, w)$. Similar to the Steiner tree case, the first step is to create the *Edge cost matrix* based on the input grid of points file. The difference this time is that it is not possible to use only the points inside the cluster, since often the substation will be outside of the clustered area. Another way of reducing the size of G and reducing computational burden is to reduce the analysis by creating a selective box that encompasses all the points between source and target with a margin, as depicted in figure 4.10. In it is possible to see:

- Blue lines which represent the the selective box;
- Green lines which represent the least cost grid designed inside the cluster;
- Black lines which represent the substation connection;
- Red points which represent the terminal nodes with the source node highlighted;
- Black squares which represent the available substations;
- Brown points which represent the whole input weighted grid of points with the ones inside the box highlighted;
- The darker area is outside the region studied for electrification.

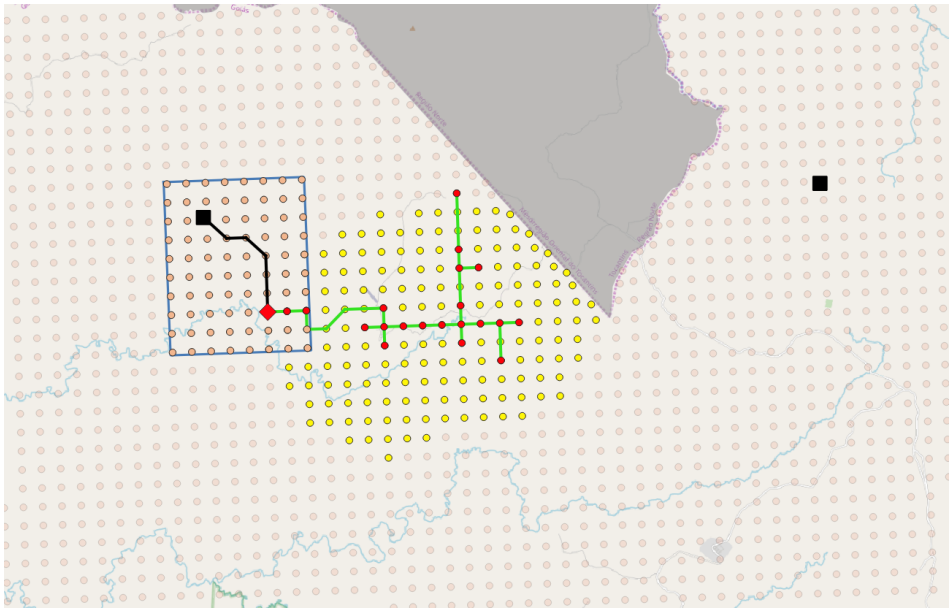

Figure 4.10: *Box selecting only points between source and target nodes*

The reduction of the amount of points of G by considering only the points inside the box, relieves the computational burden and allows for a faster shortest path computation. The way the box is computed is by using the coordinates of the source and target node and adding an extension (ext) based on the distance between them. The

box coordinates can be expressed as $X_{b,max}$, $Y_{b,max}$, $X_{b,min}$, $Y_{b,min}$, and are computed by the equations 4.7 to 4.10:

$$X_{b,max} = \max(X_{source}, X_{target}) + ext \tag{4.7}$$

$$Y_{b,max} = \max(Y_{source}, Y_{target}) + ext \tag{4.8}$$

$$X_{b,min} = \min(X_{source}, X_{target}) + ext \tag{4.9}$$

$$Y_{b,min} = \min(Y_{source}, Y_{target}) + ext \tag{4.10}$$

The selective box can also be used in the Steiner tree computation, instead of using the clustered points, for clusters that have a non-convex shape that could possibly require a path that would go outside of the cluster. Figure 4.11 shows an example of the substation connection algorithm for two different clusters and two different types of substation HV (black squares) and MV (red squares). In this scenario both clusters fall into the power limitations categories: the bigger cluster (blue) is too heavy loaded to be connected to a MV substation and even though there is one available inside the cluster (lowest possible distance), the algorithm decides to connect (black line) to the farther HV substation to respect the connection power limitations. The green cluster on the other hand is not too dense, and the amount of load does not offset the extra costs of connecting to a HV substation, therefore the connection (red line) is created to the nearest MV substation.

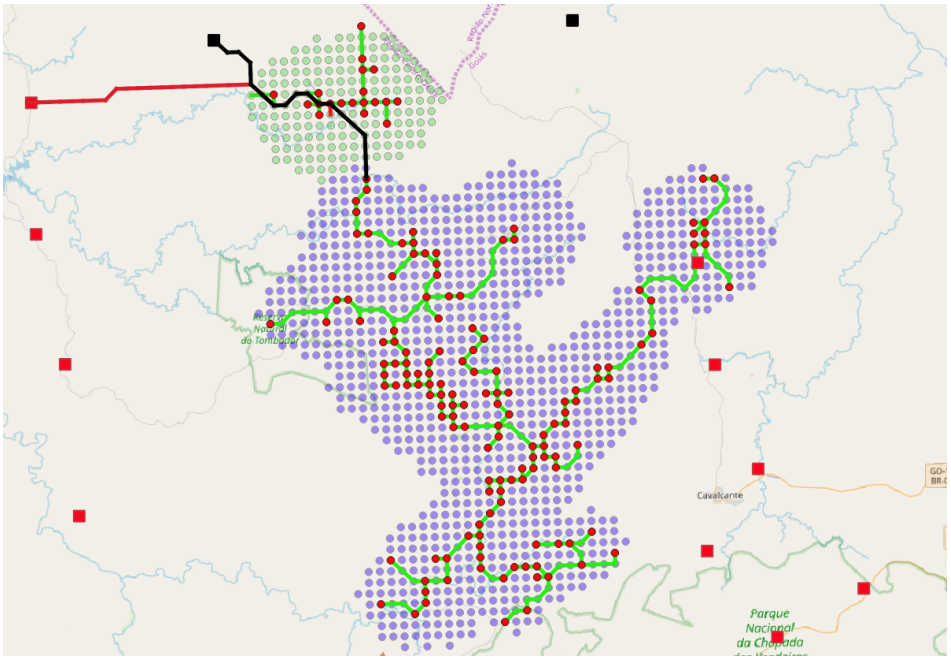

Figure 4.11: *Substation connection algorithm*

4.6 Microgrid Sizing

As an alternative to the connection to the distribution grid, GISEle can also provide an evaluation of off-grid solutions based on the available resources. In this case, if too far from an available substation, a cluster can have its own isolated microgrid based

on three main sources: wind, solar photovoltaic and hydro. To evaluate the economical feasibility of deploying a microgrid, the amount of natural resources available in each cluster area must be measured. For wind and solar power plants the resource availability is measured by the wind velocity and solar irradiation, while hydro is evaluated based on the presence of nearby rivers and its flow-rate. There is a vast online database of measurements of solar irradiation and wind velocity, GISEle uses the one called RenewablesNinja (*Pfenninger and Staffell*, 2016) and developed by the Imperial College London and ETH Zurich. This database has been chosen as the main source because it gives open API access allowing GISEle to, after exporting the area and location of the cluster being analysed, automatically contact and download the necessary data directly from their website (*renewables.ninja*). For evaluating the availability of hydro resources *Carnovali and Edeme* (2019) developed a methodology to estimate river flow rate based on precipitation and other GIS information.

With the information gathered about renewable resources availability, the microgrid sizing is outsourced and made based on two possible softwares: *Homer Energy Pro* and *Calliope*. *Homer Energy Pro* is a well established and highly consolidated tool designed specifically for solving microgrid optimization problems based on available resources and loads. The main disadvantage is that it has a proprietary license which is not compatible with the open-source toll that GISEle aims to be. Moreover, proprietary tools do not share their source code making it so the compatibility between the two software compromised. *Calliope* instead is open code written in Python language, therefore compatible with GISEle, which can be customized and embedded. However it does not offer yet the many options or the reliability that can be achieved using *Homer Energy Pro*.

Whichever software is considered, the result of the generation sizing will be a set of different possible energy solutions that are able to fulfill the energy demand of the cluster grid, while considering a lifespan of 25 years. As described in section 2.4, there is a vast range of possibilities in choosing the best off-grid solution for a specific grid. GISEle selects three different setups for each cluster, which will be subjected to the final evaluation:

- The cheapest out of all possibilities;
- The one still exploiting fossil fuels such as diesel generators, but with the highest fraction of renewable sources;
- The cheapest one that uses 100% renewable energy.

Lastly, a LCOE analysis is made to decide whether the off-grid solution is financially better in respect to creating a connection to the existent distribution grid. The LCOE of the off-grid solution using microgrids and the on-grid solution connecting cluster and distribution grid, are obtained using equations 4.11 and 4.12 respectively.

$$LCOE_{off-grid} = \frac{C_{grid}}{E_{Total}} + LCOE_{gen} \tag{4.11}$$

$$LCOE_{on-grid} = \frac{C_{grid} + C_{con}}{E_{Total}} + COE \tag{4.12}$$

Where,

- C_{grid} is the capital cost for the cluster internal electric grid.
- C_{con} is the capital cost for the electric connection between cluster and the existing grid.
- E_{Total} is the total energy forecast of total amount of energy produced and sold by the energy system in 25 years of operation.
- $LCOE_{gen}$ is the levelized cost of electricity of the generator set.
- COE cost of electricity of the local DSO.

GISEle strengths and flaws

The combination of all the procedures described in this chapter represents the main idea behind GISEle in its first conception. In a global framework where the access to electricity is one of the biggest challenges faced by the international community in the near future, GISELe places itself among the available tools for electrification planning. Being a free and open-source application written in Python language, allows for a possibility of constant development of new functionalities and overall improvement. This is one of the main accomplishments of this thesis work: to work together with the company *Enel GI&N* in order to improve the application and achieve a more realistic result.

The main limitation of GISEle is that, for now, it works only on a topological level of the electric grid. It does not consider anything related to power flow analysis, system reliability, voltage regulation or many other aspects that are essential in real electrification planning. Moreover, even the topological result achieved by GISEle presents some flaws. As an example, looking at the result of substation connection presented in figure 4.11, it is possible to see that each cluster has its own substation connection, however the possibility of two cluster sharing the same substation is not considered. Moreover, the Steiner tree solution, even though it is the least-costly, might not be a good representation of the hierarchical structure of a real power system. In the next chapter, which describes the work realized within this thesis project, some of these flaws will be addressed.

5

New Approach: Main Branch and Collaterals

In this chapter, the main contributions and the work realized in the last months will be described. Through a collaboration with *Enel GI&N*, a new approach for GISEle's grid routing routine is proposed in order to achieve a more a realistic topology of a MV distribution grid. Besides that, other improvements such as the optimization of the substation connection, allowing for shared connections between clusters, were developed and are here reported.

5.1 Enel GI&N collaboration

This thesis work is part of the master's degree program for electrical engineering in Politecnico di Milano, more specifically the smartgrid track. The master's program in smartgrid was recently created within a context of an increase amount of distribution generation and the boom of renewable energy, which changes the traditional structure of power system where loads as seen as captive users. Nowadays many people have their own micro generation that can supply their own energy demand, or even inject energy in the existing grid. The changes that have been happening in electric networks worldwide requires actions from system operators, which now have to adapt to deal with the challenges that came together with this process. Challenges such as bi-directional power flow, duck curve and voltage regulation have been widely discussed in academia and in the private sector for new solutions to be found.

To study these problems the Italian company Enel Global Infrastructure and Networks S.r.l (*Enel GI&N*) partnered with Politecnico di Milano to propose a set of research topics which are of great interest inside the company. One of the countries in which the company collaborates with DSO's is Brazil, which has been expanding its electric network in the last decades, especially in rural areas as it will be better discussed in the next chapter **??**. Wishing for improving their rural electrification

strategies in a case study of a rural area in Brazil, Enel GI&N proposed the project that ended up becoming this thesis.

The developed GISEle procedure produced important results in the case study of Namanjavira in Mozambique, however after the first preliminary results from the Brazilian case study, Enel GI&N proposed several improvements that could enhance GISEle's analysis. The main idea is to go beyond the simple topological aspect of an electric network, to a more realistic approach that considers many other aspects such as: how substations are evaluated and allocated, how to find a topology that better represents a real distribution grid and how this can be used to proper size cables based on the load they supply. Therefore, the main goals of the analysis proposed in this thesis work, which were decided together with the partner company Enel GI&N, can be summarized as:

- Create electric grids connecting everyone in a municipality of Brazil;
- Connect those grids to the existing distribution network;
- Generate a topological representation based on GIS of the items mentioned above;
- Size the cables and report the costs.

In order to achieve the goals above mentioned, many improvements had to be made to GISEle and this involves the major part of the work that has been done within this thesis. Some minor improvements have already been incorporated and explained in chapter 4. The evaluation of substations considering connection limits and different types, the evaluation of the ratio between number of people and total area, and the possibility of an easy cluster merging, are few of the improved aspects of GISEle that were not present in its first version developed and released by *Carnovali and Edeme* (2019). In the next sections, the major contributions that were developed will be described.

5.2 Substation connection optimization

Although the substation connection routine performed well in the Mozambique case study, the results show an important flaw: the possibility of two clusters sharing the same substation is not considered. Considering the example given by figure 4.11, an alternative connection for the green cluster, and more realistic, would be to connect it to the MV grid of the blue cluster and share the same HV/MV substation connection. To achieve this result an optimization algorithm is proposed, which evaluates the possibility of connecting internal cluster grids, comparing to the already in place substation connection.

In order to deal with the high amounts of connections (one for each cluster) and allow clusters to share the same substation, a python routine was created. The substation optimization routine is a post-process, i.e. it requires that all the cluster internal grids and their respective substation connection have already been created. It can be summarized by the following steps:

1. Starting by the cluster with the cheapest connection: create a connection to every other cluster.
2. Check if there is a cluster connection cheaper than the substation connection already in place.
3. If yes, check if the power limits for connection are respected for the sum of the two cluster loads.
4. If positive, maintain this cluster connection instead of the initial substation connection. Otherwise discard it and maintain the previous connection.

The flowchart of figure 5.1 describes the general idea behind the routine.

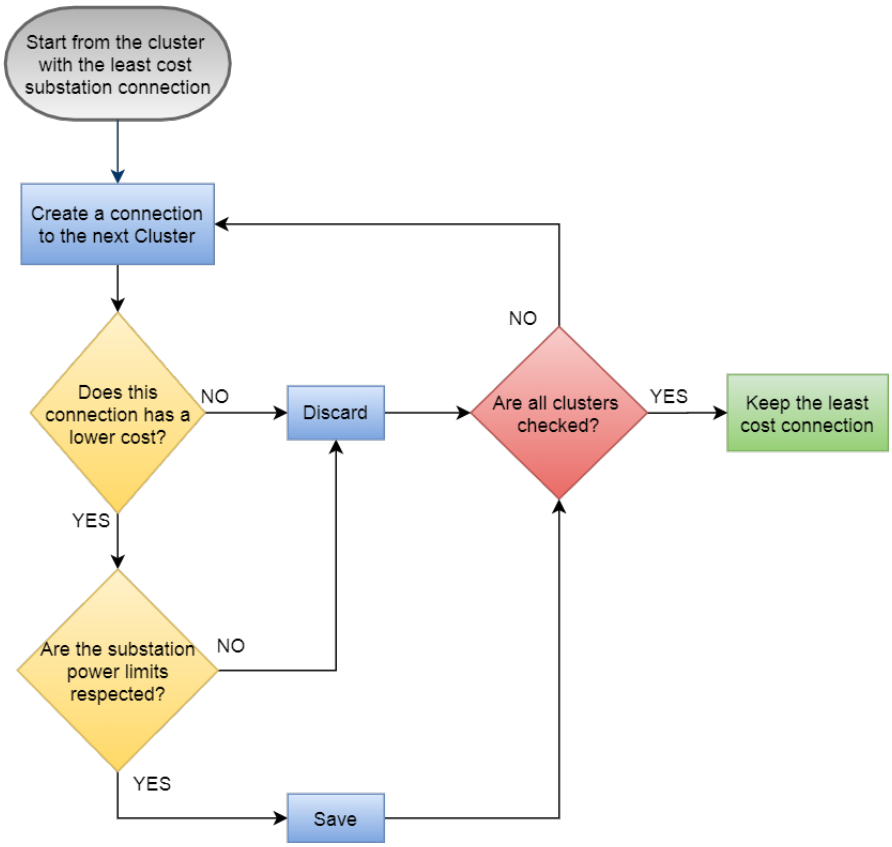

Figure 5.1: *GISEle's optimization routine for substation and cluster connections*

In the end of the process, the substation connection will be shared when it is technically feasible and less costly. To validate the results, the proposed routine was applied to the clusters of the case study in Mozambique. Figure 5.2 demonstrates the GISEle output before running the optimization algorithm proposed. It has 4 different clusters (categorized by different colors) with populated terminal nodes (selected for electrification) as red points, cluster grids as black lines and substation connections as blue lines.

In this case every cluster has a separated connection to a substation which is not efficient. Since all clusters are connected to the same type of substation (HV substations depicted as black squares), a better solution would be for them to share the

same connection, which can be seen in figure 5.3 after running the optimization algorithm in this example. Since a high power limit for the HV substation was assumed, all the clusters were able to share the same substation consequently reducing the costs of connections.

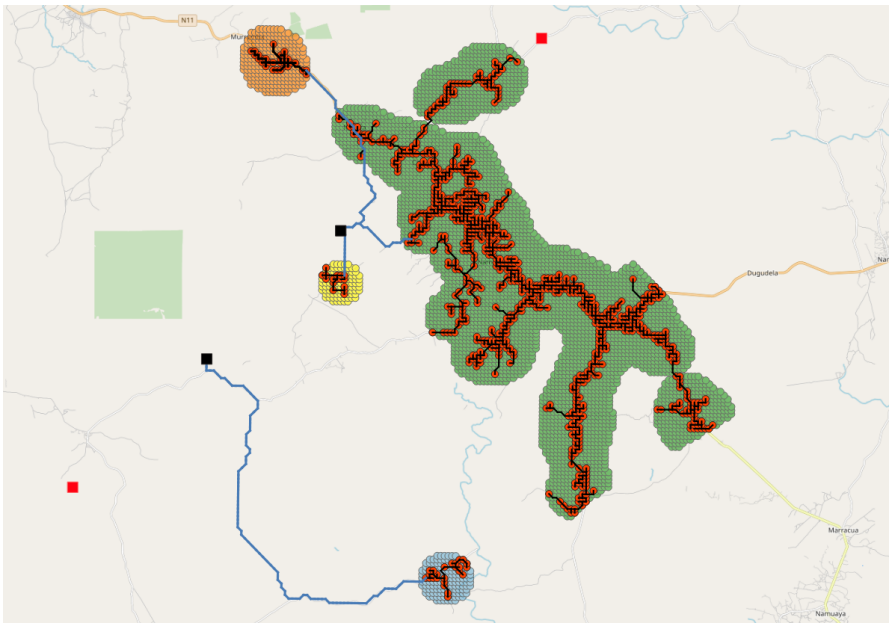

Figure 5.2: *Example of cluster grid and substation connections before the optimization algorithm*

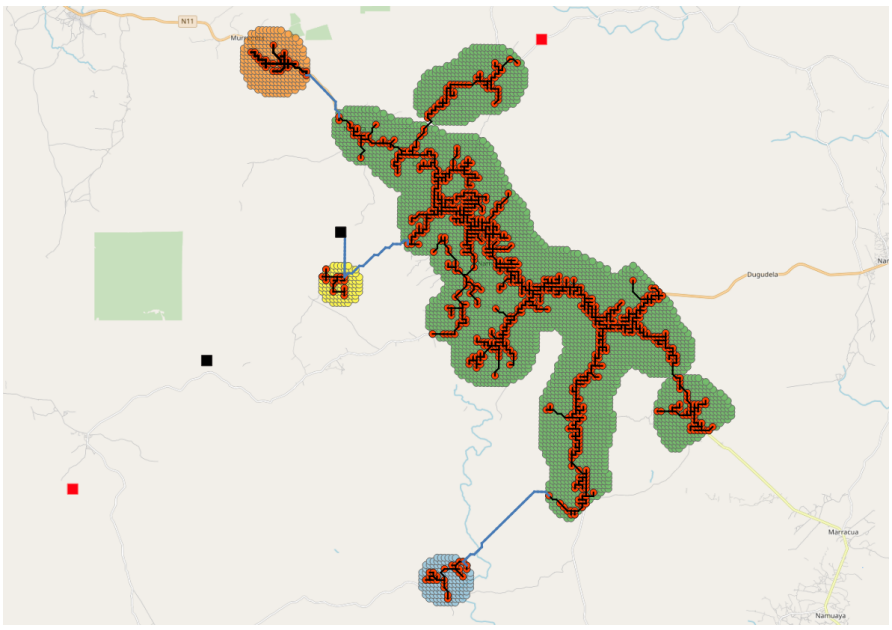

Figure 5.3: *Example of cluster grid and substation connections after the optimization algorithm*

It is necessary to emphasize that, for now, GISEle considers only the topological aspect of the electrical grid routing problem, and thus does not evaluate other important aspects such as power flow and system reliability. For this reason it always assumes that single substation connection is able to supply many clusters. A more

detailed electrical analysis is planned to be incorporated into GISEle in future works, and will be better discussed in chapter 5.4.

5.3 Main branch and collaterals: a more realistic approach

Despite GISEle's ability to properly perform weighting, clustering, grid routing and substation connections in the most efficient way, when the results are compared to a real distribution grid, a difference is quite evident. Looking at a real case example, the existent MV distribution network of a state in Brazil, shown in figure 5.4, it can be concluded that real electric grids do not necessarily follow a simple shortest path solution.

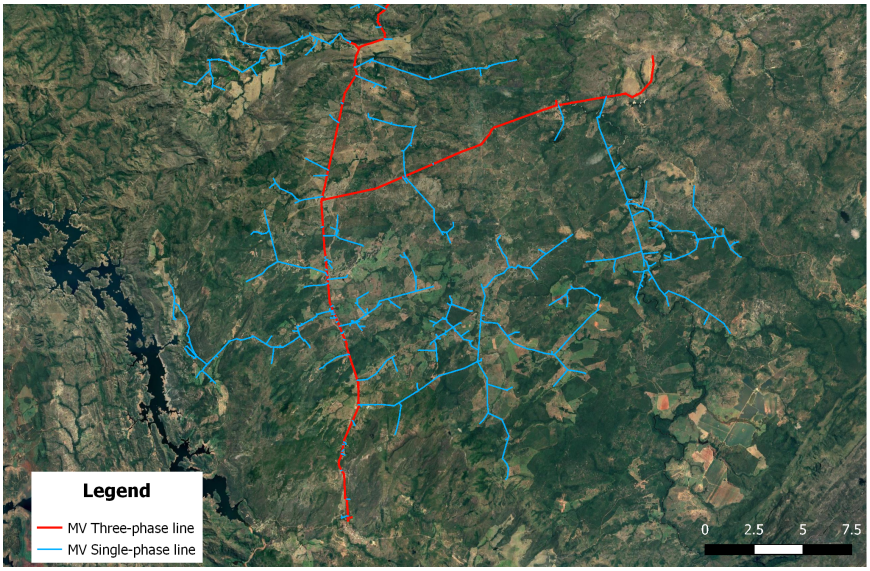

Figure 5.4: *Part of the MV distribution grid of the state of Goiás in Brazil. (Enel GI&N)*

Electric power networks have a hierarchy between transmission, MV distribution and LV distribution that is reflected in the topology. In general, high voltage systems connect densely populated areas with heavy load, and from it derivations are created to connect lighter loads. This is made so that cables are sized according to the load intended to supply. In GISEle's standard approach, there is no distinction between the cables used in the grid routing algorithm, and all of them share the same line base cost parameter. This creates a really efficient but unrealistic uniform type of electric network.

After discussing the first preliminary results with Enel GI&N, it was decided that this topological difference had to be addressed. The idea is to go from a simple minimum spanning tree, shown in figure 5.5 (a), to one that better represent this hierarchical behavior present in electric power systems, similar to what is shown in figure 5.5 (b). In the later, a main feeder (highlighted in red) is created aiming to provide a high power backbone for the electric network, from which low power cables derive to connect distant terminal nodes. This structure is closer to the real grid present in figure 5.4, where the main feeder can be correlated to the HV transmission network

(dark blue) and the derivations to the LV and MV networks (light blue).

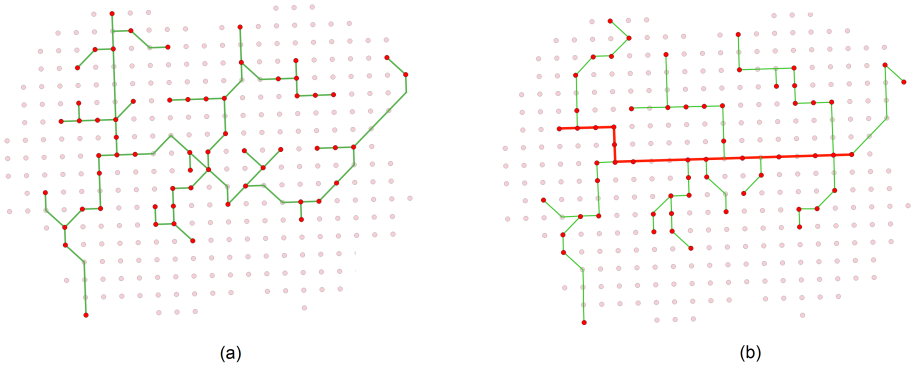

Figure 5.5: *Example of (a) a simple MST and (b) a hierarchical structure composed of a main feeder and its derivations*

To achieve this topology a two-step procedure using the previously described parameters is proposed. By using two different resolution grids, it is possible to achieve a more realistic grid composed by: a more expensive high power line that connects densely populated areas (*main branch*) and cheaper low power derivations that will connect sparse population (*collaterals*). Figure 5.6 summarizes the approach proposed by showing a variation of the GISEle's flowchart presented in figure 4.1.

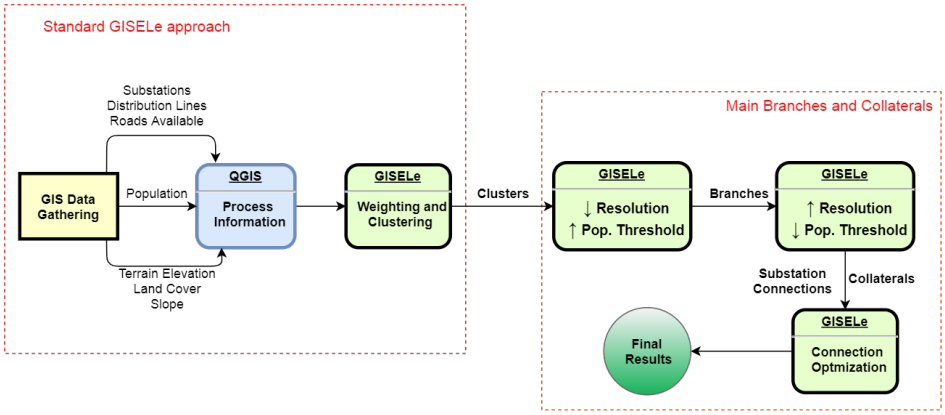

Figure 5.6: *GISEle's flowchart variation for the main branch and collaterals approach proposed*

The first steps remain the same until the cluster analysis finishes. Now, the grid routing algorithm will be executed two times: first with a lower resolution and higher population threshold, for computing the main branches, and then with the initial resolution and population thresholds. Finally the substation connections and optimization also remain mostly the same. The process of lowering the resolution can be understood as grouping the points of the initial grid data set, by doing so all the population of the grouped points are summed. All the other information (elevation, slope, land cover) are evaluated the same way, by sampling the raster layer values using the grid of points (now at a lower resolution).

Figure 5.7 exemplifies the idea behind lowering the resolution. It depicts a grid of squares of 1km^2 (for 1km resolution) or 16 km^2 (4 km resolution) each, and each

square centroid is a point of the input grid file. Each of these points will have a population value assigned to it that represents all the people inside the square area.

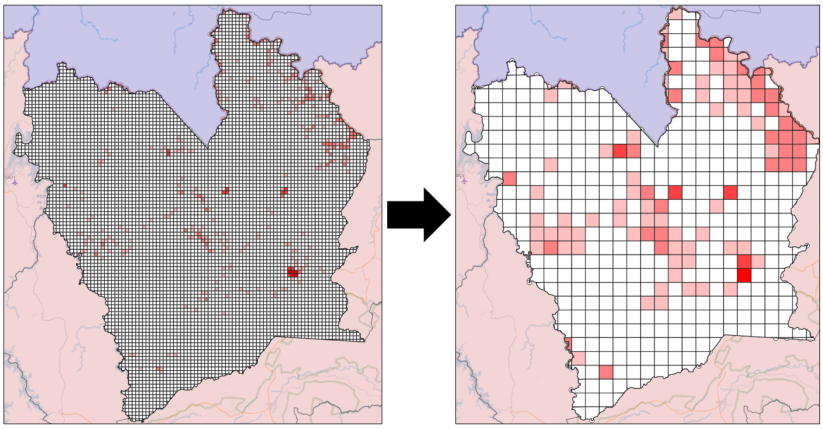

Figure 5.7: *Example of lowering resolution from 1km to 4km*

By lowering the resolution, a higher population threshold parameter can be set, forcing the grid routing algorithm to generate a Steiner tree whose terminal nodes will be only the ones with a population higher than the threshold. This process must be carefully planned since an over-reduction of the resolution would create a too-simplistic grid, composed by unrealistic straight electric lines. On the other hand if the resolution is not low enough, the main branch grid would end up connecting most of the terminal nodes making the creation of collaterals irrelevant. The result of this process is a main branch that connects densely populated areas and from which the collaterals will derive. It could happen that clusters localized in sparse areas, even lowering the resolution, might not have sufficient people to require a high power line.

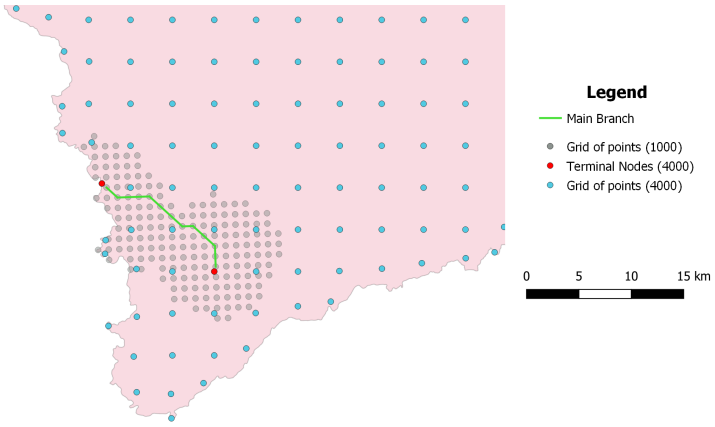

Figure 5.8: *Example of main branch creation using resolutions of 1000 and 4000 meters*

Figure 5.8 shows an example of the routing algorithm for creating the main branches. There two grid of points of different resolutions, one lower (light blue points) and one higher delimiting the clustered area (gray points). In order to define the terminal nodes (red points), the lower resolution is used so to find in which areas the

population is higher than the threshold set. Then, the algorithm classifies as terminal nodes the correspondent points on the high resolution grid. After this definition the routing algorithm, that defines the least cost path, uses the standard grid of points of high resolution to connect these terminal nodes.

After this process the grid routing algorithm is executed again, but this time using the initial standard parameters. Initially, two methods were considered: create for each single terminal point a direct connection to the main branch using Dijkstra's algorithm, or run the Steiner tree routing algorithm considering all the points. The first approach is relatively simple, it revolves around using the same technique used for routing substations connections to connect each terminal node to the main branch. The results of this process can be seen in the left topology of figure 5.9.

The Steiner tree method requires a more complex solution. Simply running the algorithm as it is would give the same results of the standard approach since the initial weighted grid of points that generates the Steiner tree is the same. Consider $G = (V, E, w)$ as the initial grid of points V with $E = \emptyset$ and another graph $G' = (V', E', w')$ representing the main branch. Note that V' and w' are both subsets of V and w respectively. In order to generate the desirable topology, the weights of G are reassigned based on equation 5.1.

$$\forall V \in G \wedge V = V' \Rightarrow w(V) = 0 \tag{5.1}$$

The idea is that for every vertex V of G that also belongs to a main branch, its weight $w(V)$ is reduced to a value close to zero. This way, when computing the *edges cost matrix*, it will assign a reduced weight to those edges that form the main branches. The grid routing algorithm will then create a Steiner tree of the terminal nodes exploiting the already in place main branch, creating this ways the collaterals as depicted in the right topology in figure 5.9.

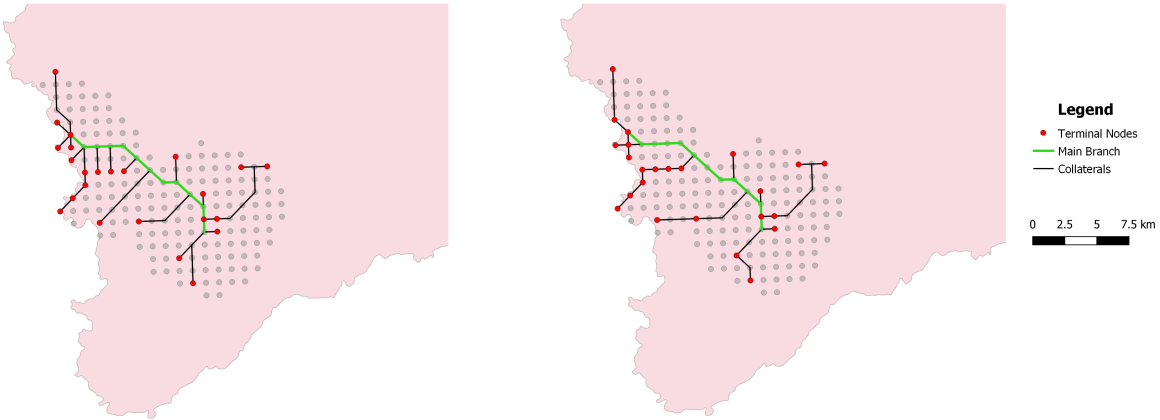

Figure 5.9: *Example of collaterals creation using two different methods: Dijkstra's algorithm (left) and Steiner MST with weight reassignment (right)*

When both methods were compared, the Steiner tree algorithm outperformed having an overall cost lower in every cluster examined. While still considering the least-cost path between each terminal node and the main branch, Dijkstra's limits

the analysis to each connection separately and fails to find a global optimized solution. For this reason, the direct connection method was discarded, and the method of using Steiner tree routing algorithm using weight reassignment was adopted.

For clusters that do not have a main branch, the grid routing algorithm will not have changes and will compute the standard Steiner tree connecting all the populated points. In this cases, it can be understood as if a single collateral line is created connecting all the terminal nodes. Finally, for creating the substation and cluster connections, only the points along the main branch, where it is available, will be considered as candidates for point of connection. It would not be reasonable to allow collaterals, designed to be low power cables, to be the connection between all the loads of a cluster and the substation.

Cable sizing

To achieve the last goal set in section 5.1 each derivation must be first identified. GISEle's output returns the main branches, collaterals and substation/cluster connections that are all multi-line vector layers. These lines have no identification other than the cluster to which they belong. To identify each individual collateral another function of the Python package *Networkx* is used. The *connected components* function returns the connected components of a given graph, which after some data processing results a collaterals identification as the one shown in figure 5.10

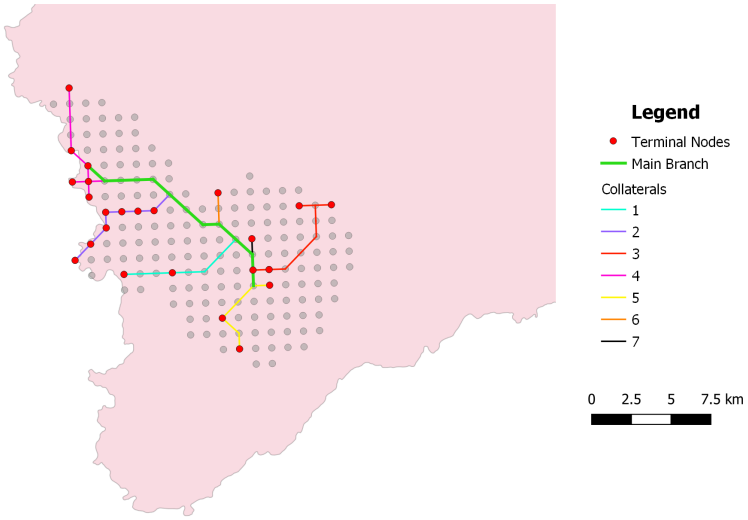

Figure 5.10: *Example of collaterals identification for cable sizing*

After the proper identification of each collateral, similar to what was made to compute the clusters' load using equation 4.6, all the population that is being supplied by a line can be summed in order to estimate its power capacity. For clusters that do not have collaterals, this task is simple since there is a single line connecting all the terminal nodes and thus the line capacity should be sized to supply the whole cluster load. In a similar fashion, main branches are also designed to supply the whole cluster loads. Collaterals however, and this is the main advantage of this

approach, will be sized according to the population they supply. To each collateral CO_i of a cluster k, a power capacity is assigned based on equation 5.2.

$$CO_i = \sum_{T_i} P_{T_i} \times LpC \tag{5.2}$$

Where,

- CO_i is the total load that the collateral i must supply;
- T_i are all the terminal nodes T of the collateral i.
- P_{T_i} is the population of the terminal node T_i;
- LpC is the load per capita parameter.

Lastly, the cost of main branch and collaterals will differ based on the two different line base costs parameters initially set. Chapter 2 already described an overview of all the techniques that have been used for rural electrification and their estimated costs. By choosing two different line base costs, a higher one for main branches and a lower one for collaterals, the total investment cost of the line deployment (not considering MV/LV transformer and other equipment) can be reduced, as verified by the results that will be reported in chapter **??**. The results are based on the case study realized in Brazil. The country's electrification overview and the context within the case study was mas made will be discussed in the next chapter.

5.4 Coding and computational effort

All the procedures, described in chapters 4 and 5, realized by GISEle are made using the programming language Python 3.7. Python is a well-established open source programming language, that today is used in a variety of applications and in many fields such as machine learning, web development, artificial intelligence, and data science. This programming language was selected due to the three main characteristics that make it well suitable to the goals GISEle longs to achieve:

- Python is open-source and flexible, being able to run on any platform or operating systems available, while also being easily integrated with other third-party software such as QGIS.
- It has a robust standard library allowing the selection of a wide range of packages according to the application.
- Python offers useful tools to deal with graphs, big databases and GIS information, which are the core of GISEle's procedures.

In the following sections, a brief description of the most important package that are used in GISEle's development will be made.

Pandas

Pandas is a well-known software library written for Python specifically designed for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series, also called data-frames. Pandas allows for importing data from various file formats such as csv (the input file type used by GISEle), and for many data manipulation operations such as merge, concatenation as well as data cleaning features such as filling, replacing.

Shapely

This is a package used for manipulation and analysis of planar geometric objects. Shapely is very useful in dealing with GIS vector layers such as multi-points, multi-lines and polygon shapefiles, which are extensively used in GISEle's routines and outputs. However it does not consider complex data formats, such as three-dimensional shapes or coordinate systems.

Geopandas

Geopandas combines the capabilities of the two packages above mentioned to facilitate the working with geospatial data. It extends the data-frames used by Pandas to allow spatial operations on geometric types, for example assigning coordinate systems, while achieving a high-level interface to multiple geometries to shapely. It is, therefore, the most suitable tool to manage GIS data.

NetworkX

Finally, NetworkX is used for creation, manipulation, and study of the structure, dynamics, and functions of complex networks. It embodies the main aspects of graph theory, allowing for the computation of Steiner tree, Dijkstra's algorithm, and graph manipulation, being used in most of the grid routing algorithms performed by GISEle.

The combination of all these tools allowed for the development of GISEle as a complex algorithm divided in many sub-parts such as weight assignment, cluster analysis and sensitivity, load profiling and grid routing. The amount of information contained in all these functionalities can reach the number of 3000 lines of Python coding, created throughout the GISEle development. A part of these codes, particularly the ones created within this thesis work, are available in the annexes A.1 to A.3. Annex A.1 presents the grid routing algorithm used for generating all the cluster grids through the computing of the Steiner spanning tree. Annex A.2 presents the shortest path algorithm that is used to create connection between two nodes, for example for HV/MV substation connections. Moreover it presents the substation connection optimization procedure described in section 5.2. Lastly, A.3 presents the algorithm developed for the main branch and collaterals procedure described in section 5.3.

Computational effort

Besides the complexity and size of GISEle's algorithm, the rural electrification procedure can be considered fast, especially for MV and HV grid routing that does not require high resolution definition. For the case study performed in this thesis work the area to be electrified is 6953 km^2, and with a resolution of 1km the total number of points necessary to cover the whole area is 7085. For the case study in Mozambique realized by *Carnovali and Edeme* (2019), the total area is smaller (approximately 3150 km^2) , but a higher resolution of 200m is used, resulting in a total amount of points of 69922 to be considered in the analysis. These values demonstrate the importance of the resolution for the analysis, and why it should be carefully evaluated.

The computational complexity is directly proportional to this number, which represents the amount of points in the input shapefile given to GISEle. It also depends on the cluster analysis performed, specifically each cluster's area. Since the grid routing algorithm create each cluster internal grid separately, the total clustered area is not important but the size of each cluster. The cluster size will determine the total amount of terminal nodes considered for the Steiner tree creation, which represents the majority of the computational complexity of the algorithm. For this reason, in order to deal with big databases with high resolution another algorithm solution was proposed as mentioned in section 4.5. Depending on which algorithm is used to perform the grid routing of large clusters, this procedures can take few minutes or even hours. Another aspect that has a big impact in the total computer memory (RAM) required to run GISEle is the distance of the HV/MV substation connections. If the cluster is too far from the substation, the total amount of combinations performed to find the shortest path can be unfeasible. In fact, for low resolution analysis, such as the one in Mozambique, a distance limit (between 15 to 30 km) for HV/MV substation connections is set.

Table 5.1 present the average time required by the main steps of GISEle's procedures here described. These values were obtained by running it in a personal computer with the following configuration: CPU Intel Core i7-4700HQ 2.4 GHZ and 16GB RAM. For applications with less than 10000 number of points (such as the case study performed within this thesis) the time required to run the algorithm is about 20 minutes. If the main branch and collaterals approach is considered, in which case the algorithm is executed twice, this value can increase up to 30 minutes.

Procedure	Time required
Weighting process	1-5 minutes
Cluster analysis and sensitivity	1-5 minutes
Grid routing	5 minutes to 2 hours
Substation connection optimization	1 - 10 minutes

Table 5.1: *Average time required by each procedure made by GISEle*

Conclusions and future research

In the context of global electrification, it is seen that access to electricity is expected to increase, more so in sub-Saharan Africa and in countries under development. Considered by UN as one of the Sustainable Development Goals (SDG's), access to electricity should be a basic human right and a necessity to human dignity. Moreover, having an carefully planned electrification process is of utmost importance for many countries to be able develop their economies, which could help to decrease the blatant inequality that today's world faces. Following a decade of steady progress, the global electrification rate reached 89 percent and 153 million people gained access to electricity each year. However, to achieve the ambition of a world where everyone has access to electricity, more than 800 million people have yet to be reached, most of which are located in remote rural areas. It is of primordial importance, therefore, to develop tools that can support distribution grid planning for rural electrification.

It was under this circumstances that GISEle was created, with the goal of creating a methodology capable of enhance rural electrification projects by combining Geographic Information System (GIS), with terrain analysis and graph theory, to provide the best topology for an electric network. As a fundamental characteristic, GISEle is a free open-source technology that aims to be suitable for anyone interested: international organizations, public sector and the private market. The methodology consists in three main steps: data analysis, clustering and grid routing.

During the initial phase of data gathering and analysis, GIS data sets are created in order to properly map several information about the area to be electrified. Some of these information are: population density, elevation, slope and roads. These data are downloaded from online public databases from international organizations such as World Bank, and managed using the free GIS software QGIS. They are all processed using a weighting strategy that translates the topological aspect of the terrain to the difficulty of line deployment. This strategy is based on penalty factors that represent the relative additional costs, that are added on a baseline cost per electric line kilometer. The output of the data analysis is a weighted grid of points, each carrying all the information above mentioned, that correlates them to its geographical position. Then, a density-based clustering technique is used to strategically aggregates groups of people in small areas called clusters. The density-based algorithm DBSCAN is executed several times, performing a cluster sensitivity analysis whose goal is to find a good set of initial parameters eps and $MinPts$. The output is a set number of clusters, partially covering the initial area considered, in which the grid routing algorithm is performed. The last and most important step is the creation of the electric network topology. To do so GISEle uses the concept of Steiner tree

to create a network topology connecting all the aggregated people in each cluster, and then, if necessary, it makes use of Dijkstra's algorithm to connect each cluster grid into an existing distribution network. As an alternative it can use a third-party software to design a generator set based on renewable energy sources, evaluation through LCOE analysis the best solution.

This thesis contribution involves improving the initial concept of GISEle, applying it in a real case of rural electrification planning with the support of the partner company Enel Global Infrastructure and Networks S.r.l. The initial thesis work consisted in strategically evaluated the rural area under study performing a preliminary sensitivity analysis considering different amounts of people to be connected. Then, based on the feedback from Enel GIN engineers, many improvements were incorporated in order to enhance GISEle's performance. Substation typification and assignment, clustering sensitivity, cluster merging and human-machine interface are among the elements that were improved. Furthermore, a new routine for improving the substation connection and a new approach on the grid routing algorithm were developed. The substation connection optimization routine evaluates the cost of each connection between the cluster internal grids and the existent distribution grid, choosing the least-cost connections and allowing for clusters to share the same MV substation. The new approach of electrification developed, called main branch and collaterals, allows GISEle to create a more realistic distribution network topology. By using data analysis to create two grid of points of different resolution, densely populated areas can be connected using a high power main feeder while sparse population are connected from its derivations. Using this approach, each of these derivations (collaterals) can be accordingly sized based on the peak power of the loads they supply.

The results of this rural electrification analysis, which focus only on the topological aspect and does not consider other important factors such as system reliability and quality of service, suggests that a distribution grid expansion to reaches local population in wide and dispersed areas can be optimized through better routing. The new main branch and collaterals approach manage to reduce up to 47% the total investment cost in line deployment, in respect with the initial GISEle approach. The cost per person connected went from 5212 euros in the standard approach to 2785 euros using the main branch and collaterals. If households are considered, the standard approach resulted in 17199 euros per household connected, while the main branch and collaterals approach resulted in 9190 euros. The total length of lines necessary to achieve 100% electrification was 1635 km, with a total investment cost of 33.93 million euros. This reduction in cost suggests that an optimized electrification strategy through better routing could shift the balance point between on-grid and off-grid solutions such as the use of microgrids and PV generation. These costs are related to the line deployment only, other costs such as the MV/LV transformers and protection equipment were not considered. This result was achieved also due to the proper evaluation of many rural electrification strategies proposed in literature, choosing among them the one that best suits the interests of *Enel GI&N*.

Lastly, many possibilities are still open for GISEle's growth. In the near future, as a

continuation of this thesis work and the collaboration with Enel GI&N, it is expected to go further into the electrical elements of the procedure. The topological analysis here described will then comprehend other important aspects for electric power systems planning such as: power flow, voltage regulation and system reliability. With all these aspects integrated, the project goal is to establish GISEle as a consolidated tool, and make freely available to support electrification planing worldwide.

Appendix A

Python Codes

A.1 GISEle's Steiner tree routing algorithm

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 18 11:27:23 2019

@author: Silvia Corigliano , Tommaso Carnovali, Darlain Edeme
"""
import pandas as pd
import geopandas as gpd
import numpy as np
import networkx as nx
import time
import math
from scipy import sparse
from fiona.crs import from_epsg
from scipy.spatial import distance_matrix
from scipy.sparse import csr_matrix
from scipy.sparse.csgraph import minimum_spanning_tree
from Codes.Weight_matrix import*
from Codes.Steiner_tree_code import*
from shapely.geometry import Point, LineString, Polygon, box
from scipy.spatial.distance import cdist


def Steinerman(mesh, cluster_points, Proj_coords, paycheck, resolution):

    start_time = time.time()
    # remove repetitions in the dataframe
    mesh.drop_duplicates(['ID'], keep='last', inplace=True)
    cluster_points.drop_duplicates(['ID'], keep='last', inplace=True)
    # Creation of a pandas DataFrame containing only coordinates of populated
        points
    d = {'x': cluster_points['X'], 'y': cluster_points['Y'],
         'z': cluster_points['Elevation']}
    pop_points = pd.DataFrame(data=d)
    pop_points.index = cluster_points['ID']
```

```
    print('pop_points has been created')

    # Building a box
    xmin = cluster_points['X'].min()
    xmax = cluster_points['X'].max()
    ymin = cluster_points['Y'].min()
    ymax = cluster_points['Y'].max()
    wide = Point(xmin, ymin).distance(Point(xmax, ymax))
    extension = wide / 5
    bubble = box(minx=xmin - extension, maxx=xmax + extension, miny=ymin -
        extension,
                 maxy=ymax + extension)
    df_box = mesh[mesh.within(bubble)]
    df_box.index = pd.Series(range(0, len(df_box['ID'])))

    solocoord2 = {'x': df_box['X'], 'y': df_box['Y']}
    solocoord3 = {'x': df_box['X'], 'y': df_box['Y'], 'z': df_box['Elevation']}
    Battlefield2 = pd.DataFrame(data=solocoord2)
    Battlefield2.index = df_box['ID']
    Battlefield3 = pd.DataFrame(data=solocoord3)
    Battlefield3.index = df_box['ID']
    print('Checkpoint 2')

    Distance_2D_box = distance_matrix(Battlefield2, Battlefield2)
    Distance_3D_box = pd.DataFrame(cdist(Battlefield3.values, Battlefield3.
        values, 'euclidean'),
                                   index=Battlefield3.index, columns=
                                       Battlefield3.index)
    Weight_matrix = weight_matrix(df_box, Distance_3D_box, paycheck)
    print('3D distance matrix has been created')
    diag_length = resolution * 1.5

    # Definition of weighted connections matrix
    Edges_matrix = Weight_matrix
    Edges_matrix[Distance_2D_box > math.ceil(diag_length)] = 0
    Edges_matrix_sparse = sparse.csr_matrix(Edges_matrix)
    print('Checkpoint 4: Edges matrix completed')
    Graph = nx.from_scipy_sparse_matrix(Edges_matrix_sparse)

    # Lavoro per reperire i giusti indici di cluster_points
    a = list(cluster_points['ID'].to_numpy())
    cluster_points_box = gpd.GeoDataFrame(crs=from_epsg(Proj_coords))
    for i in a:
        p = df_box.loc[df_box['ID'] == i]
        cluster_points_box = pd.concat([cluster_points_box, p], sort=True)

    # Steiner tree connecting point with population over a defined threshold
    terminal_nodes = list(cluster_points_box.index)
    T = steiner_tree(Graph, terminal_nodes, weight='weight')
    # create output shapefile
    Adj_matrix= nx.to_numpy_matrix(T)
    N = (Adj_matrix > 0).sum()
    nodes = list(T.nodes)
    Adj_matrix = pd.DataFrame(Adj_matrix)
    Adj_matrix.columns = nodes
    Adj_matrix.index = nodes

    connections = []
    k = 0
    grid = gpd.GeoDataFrame(crs=from_epsg(Proj_coords))
```

```
92      print('Checkpoint 5')
93      #grid['T']=()
94      grid['id'] = pd.Series(range(1, N + 1))
95      grid['Weight'] = pd.Series(range(1, N + 1))
96      grid['ID1'] = pd.Series(range(1, N + 1))
97      grid['ID2'] = pd.Series(range(1, N + 1))
98      grid.loc[:, 'geometry'] = None
99      grid['geometry'].astype = gpd.geoseries.GeoSeries
100     for i, row in Adj_matrix.iterrows():
101         for j, column in row.iteritems():
102             con = [min(df_box.loc[i, 'ID'], df_box.loc[j, 'ID']),
103                    max(df_box.loc[i, 'ID'], df_box.loc[j, 'ID'])]
104             if column > 0 and con not in connections:
105                 grid.at[k, ['geometry']] = LineString([(df_box.loc[i, ['X']],
106                                                         df_box.loc[i, ['Y']]),
107                                                        (df_box.loc[j, ['X']],
108                                                         df_box.loc[j, ['Y']])])
109                 grid.at[k, ['id']] = 0
110                 grid.at[k, ['Weight']] = Edges_matrix.loc[df_box.loc[i, 'ID'],
111                                                           df_box.loc[j, 'ID']]
112                 grid.at[k, 'ID1'] = df_box.loc[i, 'ID']
113                 grid.at[k, 'ID2'] = df_box.loc[j, 'ID']
114                 connections.append(con)
115                 k = k+1
116     grid.head()
117     indexNames = grid[(grid['id'] > 0)].index
118     grid.drop(indexNames, inplace=True)
119     grid.dropna(inplace=True)
120     grid.crs = cluster_points.crs
121     total_cost = grid['Weight'].sum(axis=0)
122     grid['line_length'] = grid['geometry'].length
123     total_length = grid['line_length'].sum(axis=0)
124
125     final_time = time.time()
126     total_time = final_time - start_time
127     print("Total time for tree's evolution algorithm was: " + str(total_time) +
            "seconds")
128
129     return grid, total_cost, total_length, connections
```

A.2 Substation connection optimization algorithm

```
# -*- coding: utf-8 -*-
"""
Substation connection optimization algorithm
Created on Wed Jan 08 2020

@author: Vinicius Gadelha
"""
import glob
import os

import pandas as pd
import geopandas as gpd
import numpy as np
import networkx as nx
import time
import math
from fiona.crs import from_epsg
from scipy.spatial import distance_matrix
from scipy.sparse import *
from scipy.sparse.csgraph import minimum_spanning_tree
from Codes.Weight_matrix import *
from Codes.Steiner_tree_code import *
from shapely.geometry import Point, LineString, Polygon, box
from scipy.spatial.distance import cdist
from shapely.ops import nearest_points

def grid_direct_connection(mesh, gdf_cluster_pop, substation_designata,
    Proj_coords,
                           paycheck, resolution):
    Point1 = Point(float(substation_designata['X']), float(substation_designata
        ['Y']))
    id1 = int(substation_designata['ID'].values)
    Point2 = Point(float(gdf_cluster_pop['X']), float(gdf_cluster_pop['Y']))
    id2 = int(gdf_cluster_pop['ID'].values)
    dist = Point1.distance(Point2)
    if dist < 1000:
        extension = dist
    elif dist < 2000:
        extension = dist / 2
    else:
        extension = dist / 4

    xmin = min(Point1.x, Point2.x)
    xmax = max(Point1.x, Point2.x)
    ymin = min(Point1.y, Point2.y)
    ymax = max(Point1.y, Point2.y)
    bubble = box(minx=xmin - extension, maxx=xmax + extension, miny=ymin -
        extension,
                 maxy=ymax + extension)
    df_box = mesh[mesh.within(bubble)]
    df_box.index = pd.Series(range(0, len(df_box['ID'])))

    solocoord2 = {'x': df_box['X'], 'y': df_box['Y']}
    solocoord3 = {'x': df_box['X'], 'y': df_box['Y'], 'z': df_box['Elevation']}
    Battlefield2 = pd.DataFrame(data=solocoord2)
    Battlefield2.index = df_box['ID']
    Battlefield3 = pd.DataFrame(data=solocoord3)
```

```
    Battlefield3.index = df_box['ID']
    print('Checkpoint 2')

    Distance_2D_box = distance_matrix(Battlefield2, Battlefield2)
    Distance_3D_box = pd.DataFrame(cdist(Battlefield3.values,
                      Battlefield3.values, 'euclidean'),
                      index=Battlefield3.index, columns=Battlefield3.index)
    Weight_matrix = weight_matrix(df_box, Distance_3D_box, paycheck)
    print('3D distance matrix has been created')
    diag_length = resolution * 1.7

    # Definition of weighted connections matrix
    Edges_matrix = Weight_matrix
    Edges_matrix[Distance_2D_box > math.ceil(diag_length)] = 0
    Edges_matrix_sparse = sparse.csr_matrix(Edges_matrix)
    print('Checkpoint 4: Edges matrix completed')
    Graph = nx.from_scipy_sparse_matrix(Edges_matrix_sparse)
    source = df_box.loc[df_box['ID'].values == id1, :]
    target = df_box.loc[df_box['ID'].values == id2, :]
    source = source.index[0]
    target = target.index[0]
    print('Checkpoint 5')
    # Lancio Dijkstra e ottengo il percorso ottimale per connettere 'source' a '
        target'
    path = nx.dijkstra_path(Graph, source, target, weight='weight')
    Steps = len(path)
    # Analisi del path
    r = 0  # Giusto un contatore
    PathID = []
    Digievoluzione = gpd.GeoDataFrame(crs=from_epsg(Proj_coords))
    Digievoluzione['id'] = pd.Series(range(1, Steps))
    Digievoluzione['x1'] = pd.Series(range(1, Steps))
    Digievoluzione['x2'] = pd.Series(range(1, Steps))
    Digievoluzione['y1'] = pd.Series(range(1, Steps))
    Digievoluzione['y2'] = pd.Series(range(1, Steps))
    Digievoluzione['ID1'] = pd.Series(range(1, Steps))
    Digievoluzione['ID2'] = pd.Series(range(1, Steps))
    Digievoluzione['Weight'] = pd.Series(range(1, Steps))
    Digievoluzione['geometry'] = None
    Digievoluzione['geometry'].stype = gpd.geoseries.GeoSeries
    print('Checkpoint 6:')

    for h in range(0, Steps - 1):
        con = [min(df_box.loc[path[h], 'ID'], df_box.loc[path[h + 1], 'ID']),
               max(df_box.loc[path[h], 'ID'], df_box.loc[path[h + 1], 'ID'])]
        # if con not in connections:
        PathID.append(df_box.loc[path[h], 'ID'])
        PathID.append(df_box.loc[path[h + 1], 'ID'])
        Digievoluzione.at[r, 'geometry'] = LineString(
            [(df_box.loc[path[h], 'X'], df_box.loc[path[h], 'Y'],
              df_box.loc[path[h], 'Elevation']),
             (df_box.loc[path[h + 1], 'X'], df_box.loc[path[h + 1], 'Y'],
                 df_box.loc[path[h + 1], 'Elevation'])])
        Digievoluzione.at[r, ['id']] = 0
        Digievoluzione.at[r, ['x1']] = df_box.loc[path[h], 'X']
        Digievoluzione.at[r, ['x2']] = df_box.loc[path[h + 1], 'X']
        Digievoluzione.at[r, ['y1']] = df_box.loc[path[h], ['Y']]
        Digievoluzione.at[r, ['y2']] = df_box.loc[path[h + 1], ['Y']]
        Digievoluzione.at[r, 'ID1'] = df_box.loc[path[h], 'ID']
        Digievoluzione.at[r, 'ID2'] = df_box.loc[path[h + 1], 'ID']
```

```
114             Digievoluzione.at[r, 'Weight'] = Edges_matrix.loc[
115                 df_box.loc[path[h], 'ID'], df_box.loc[path[h + 1], 'ID']]
116             # connections.append(con)
117             r += 1
118
119         print('Checkpoint 7: Genomica di digievoluzione identificata')
120         # Elimino le righe inutili di Digievoluzione e i doppioni da PathID
121         indexNames = Digievoluzione[Digievoluzione['id'] > 0].index
122         Digievoluzione.drop(indexNames, inplace=True)
123         PathID = list(dict.fromkeys(PathID))
124         cordone_ombelicale = Digievoluzione
125         cordone_ombelicale = cordone_ombelicale.drop(['id', 'x1', 'x2', 'y1', 'y2'],
                axis=1)
126         cordone_ombelicale['line_length'] = cordone_ombelicale['geometry'].length
127         total_cost = int(cordone_ombelicale['Weight'].sum(axis=0))
128         total_length = cordone_ombelicale['line_length'].sum(axis=0)
129
130         return cordone_ombelicale, total_cost, total_length
131
132
133 def grid_optimization(gdf_clusters, geodf_in, grid_resume, proj_coords,
        resolution,
134                       paycheck):
135         os.chdir('D:\Vinicius\Mestrado\Thesis\GISELE\Output\Chunks')
136         new_connection_merged = pd.DataFrame()
137         grid_resume = pd.read_csv('grid_resume.csv')
138         grid_resume.reset_index(drop=True, inplace=True)
139         grid_resume = grid_resume.sort_values(by='Connection_Cost')
140         clusters_list_2 = grid_resume['Cluster']
141         check = np.zeros(len(clusters_list_2), dtype=bool)
142         grid_resume = grid_resume.sort_values(by='Cluster')
143
144         count = 0
145
146         while not any(check):  # checks if all variables in check are True
147             for i in clusters_list_2:
148                 betterconnection = False
149                 print('Evaluating if there is a better connection for cluster ' +
150                       str(i))
151                 k = np.where(grid_resume['Cluster'] == i)  # find the index of the
152                 # cluster we are analysing
153                 dist = pd.DataFrame(index=clusters_list_2, columns=['Distance',
154                                                     'NearestPoint', 'Cost'])
155                 for j in clusters_list_2:
156                     k2 = np.where(grid_resume['Cluster'] == j)  # index of the
157                     # cluster we trying to connect
158                     if grid_resume.Connection_Length.values[
159                         k] == 0:  # k is necessary due to cluster merging causing
160                         # indexes and cluster numbers to not match
161                         check[count] = True
162                         break
163                     if i == j:
164                         dist.Distance[j] = 9999999
165                         dist.Cost[j] = 99999999
166                         continue
167
168                     grid1 = gpd.read_file("Chunk_" + str(j) + ".shp")
169                     chunk_points = pd.DataFrame()
170                     for id in grid1.ID1:  # getting all the points inside
171                         # the chunk to compute the nearest_points
```

```
172                     chunk_points = gpd.GeoDataFrame(
173                         pd.concat([chunk_points, gdf_clusters[gdf_clusters['ID']
                                == id]],
174                                   sort=True))
175                 last_id = grid1.loc[
176                     grid1.ID2.size - 1, 'ID2']  # this lines are necessary
                        because the
177                 # last point was not taken
178                 chunk_points = gpd.GeoDataFrame(
179                     pd.concat([chunk_points, gdf_clusters[gdf_clusters['ID'] ==
                        last_id]],
180                               sort=True))
181                 uu_grid1 = chunk_points.unary_union
182
183                 grid2 = gpd.read_file("Chunk_" + str(i) + ".shp")
184                 chunk_points = pd.DataFrame()
185                 for id in grid2.ID1:  # getting all the points inside the chunk
186                     # to compute the nearest_points
187                     chunk_points = gpd.GeoDataFrame(
188                         pd.concat([chunk_points, gdf_clusters[gdf_clusters['ID']
                                == id]],
189                                   sort=True))
190                 last_id = grid2.loc[
191                     grid2.ID2.size - 1, 'ID2']  # this lines are necessary
                        because
192                 # the last point was not taken
193                 chunk_points = gpd.GeoDataFrame(
194                     pd.concat([chunk_points, gdf_clusters[gdf_clusters['ID'] ==
                        last_id]],
195                               sort=True))
196                 uu_grid2 = chunk_points.unary_union
197                 del chunk_points
198
199                 dist.NearestPoint[j] = nearest_points(uu_grid1, uu_grid2)
200                 p1 = gdf_clusters[gdf_clusters['geometry'] == dist.NearestPoint[
                    j][0]]
201                 p2 = gdf_clusters[gdf_clusters['geometry'] == dist.NearestPoint[
                    j][1]]
202                 # if the distance between grids is higher than 1.2 times the
                    length
203                 # former connection, skip
204                 if (dist.NearestPoint[j][0].distance(dist.NearestPoint[j][1])) /
                     1000 > \
205                         1.2 * (grid_resume.at[k[0][0], 'Connection_Length']):
206                     dist.Distance[j] = 9999999
207                     dist.Cost[j] = 99999999
208                     continue
209
210                 direct_connection, direct_cost, direct_length = \
211                     grid_direct_connection(geodf_in, p1, p2, proj_coords,
                        paycheck,
212                                            resolution)
213
214                 dist.Distance[j] = direct_length
215                 if grid_resume.ConnectionType.values[np.where(grid_resume['
                    Cluster']
216                                                               == j)[0] == 'HV':
217                     direct_cost = direct_cost + 1000000
218                 dist.Cost[j] = direct_cost
219                 if grid_resume.ConnectionType.values[np.where(grid_resume['
```

```
                    Cluster']
220                                                          == j)][0] == '
                                                                MV_3P':
                    if grid_resume.ClusterLoad.at[k[0][0]] + \
                            grid_resume.ClusterLoad.at[k2[0][0]] > 3000:
                        continue
                if grid_resume.ConnectionType.values[np.where(grid_resume['
                    Cluster']
225                                                          == j)][0] == '
                                                                MV_1P':
                    if grid_resume.ClusterLoad.at[k[0][0]] +\
                            grid_resume.ClusterLoad.at[k2[0][0]] > 300:
                        continue
                if min(dist.Cost) == direct_cost:
                    if direct_cost / 1000 < grid_resume.Connection_Cost.values[k
                        ]:
                        if check[np.where(clusters_list_2 == j)]:
                            direct = 'NewConnection_' + str(i) + '.shp'
                            direct_connection.to_file(direct)
                            print('A new connection for Cluster ' + str(i) +
                                  ' was successfully created')
                            grid_resume.at[k[0][0], 'Connection_Length'] \
                                = direct_length / 1000
                            grid_resume.at[k[0][0], 'Connection_Cost'] = \
                                direct_cost / 1000
                            grid_resume.at[k[0][0], 'ConnectionType'] = \
                                grid_resume.ConnectionType.values[np.
                                    where(grid_resume['Cluster'] == j)][0]
                            betterconnection = True
                            best_connection = direct_connection
            check[count] = True
            count = count + 1
            if betterconnection:
                new_connection_merged = \
                    gpd.GeoDataFrame(pd.concat([new_connection_merged,
                                                best_connection], sort=True))
            elif not betterconnection:
                if grid_resume.Connection_Length.values[k] > 0:
                    best_connection = gpd.read_file('Connection_Grid' + str(i) +
                         '.shp')
                    new_connection_merged = gpd.GeoDataFrame(
                        pd.concat([new_connection_merged, best_connection], sort
                            =True))

    grid_resume.to_csv('grid_resume_opt.csv')
    new_connection_merged.crs = geodf_in.crs
    new_connection_merged.to_file('total_connections_opt')
    return grid_resume
```

A.3 Main branch and collaterals

```
"""
Main branches and collaterals algorithm

Created on Mon Jan 27 2020

@author: Vinicius Gadelha
"""

import os
import geopandas as gpd
import pandas as pd
import math
import numpy as np
import sys
from shapely.geometry import MultiPoint
from shapely.ops import nearest_points
from Codes.remove_cycles_weight_final import *
from scipy.spatial import distance_matrix
from scipy.spatial.distance import cdist
from scipy.sparse import csr_matrix
from scipy.sparse.csgraph import minimum_spanning_tree
from shapely.geometry import Point, LineString, box
from fiona.crs import from_epsg
import matplotlib.pyplot as plt
import networkx as nx
import time
from Codes.Weight_matrix import *
from Codes.Steiner_tree_code import *
from Codes.Steinerman import *
from Codes.Spiderman import *
import warnings

warnings.filterwarnings("ignore")

def nearest(row, geom_union, df1, df2, geom1_col='geometry',
            geom2_col='geometry',src_column=None):
    """Find the nearest point and return the corresponding value from specified
        column."""
    # Find the geometry that is closest
    nearest = df2[geom2_col] == nearest_points(row[geom1_col], geom_union)[1]
    # Get the corresponding value from df2 (matching is based on the geometry)
    value = df2[nearest][src_column].get_values()[0]
    return value

#   IMPORTING THE WEIGHTED CSV

os.chdir(r'\Input/2_weighted_datasets')
file_name = 'enel_studyarea3_weighted'
geo_csv = pd.read_csv(file_name + ".csv")
print("Layer file successfully loaded.")

geometry = [Point(xy) for xy in zip(geo_csv.X, geo_csv.Y)]
Proj_coords = 32723  #  CRS adopted
geodf_in = gpd.GeoDataFrame(geo_csv, crs=from_epsg(Proj_coords),
                            geometry=geometry)
unit = 1  # Unit of the CRS if in meters (1) or degrees (0)
resolution = 1000  # Resolution in meters
```

```
paycheck = 10000  #  Line base cost
pop_threshold = 20
limitHV = 3000  #  Power limits for connection
limitMV = 300
extracostHV = 5000000

#  IMPORTING CLUSTERS

os.chdir(r'\Output\Clusters')
gdf_clusters = gpd.read_file("gdf_clusters.shp")
clusters_list_2 = np.unique(gdf_clusters['clusters'])  # takes all
# unique values inside the dataframe
clusters_list_2 = np.delete(clusters_list_2, np.where(clusters_list_2 == -1))
# removes the noises

os.chdir(r'\Output\Grids')
grid_resume_old = pd.read_csv('grid_resume.csv')
clusters_load = grid_resume_old['ClusterLoad']
grid_resume = grid_resume_old.sort_values(by='ClusterLoad')
# clusters_list_2 = grid_resume['Cluster']

print("Clusters successfully imported")

#  IMPORTING THE LOWER RESOLUTION GRID

os.chdir(r'\Resolution')
gdf_zoomed = gpd.read_file("enel_studyarea3_4000.shp")


#  IMPORTING SUBSTATIONS
os.chdir(r'/Input//0_Dataset//Substations')
substations = gpd.read_file("substations.shp")
substations['ID'] = range(0, len(substations))

os.chdir(r'\Output\Chunks')

grid_resume = pd.DataFrame(index=clusters_list_2,
                           columns=['Cluster', 'Chunk_Length', 'Chunk_Cost',
                                    'Grid_Length','Grid_Cost', '
                                        Connection_Length',
                                    'Connection_Cost','ConnectionType',
                                    'Link_Length', 'Link_Cost'])
grid_resume['Cluster'] = clusters_list_2
grid_resume['ClusterLoad'] = grid_resume_old.ClusterLoad.values
total_chunk = total_derivations = pd.DataFrame()
total_branch_steiner = total_branch_spider = chunk_points = \
    all_branch_direct = total_connections = pd.DataFrame()


# START OF THE MAIN CHUNK CREATION

for i in clusters_list_2:
    gdf_cluster_only = gdf_clusters[gdf_clusters['clusters'] == i]
    gdf_zoomed_pop = gdf_zoomed[gdf_zoomed['Cluster'] == i]

    gdf_zoomed_pop = gdf_zoomed_pop[gdf_zoomed_pop['Population']
                                    >= pop_threshold]

    points_to_electrify = int(len(gdf_zoomed_pop))
    if points_to_electrify > 2:
```

```
        print('-' * 100)
        print('CREATING CHUNK FOR CLUSTER ' + str(i))
        print('-' * 100)

        chunk_grid, chunk_cost, chunk_length, connections2 \
            = Steinerman(gdf_cluster_only, gdf_zoomed_pop, Proj_coords,
                         paycheck, resolution)
        fileout = "Chunk_" + str(i) + '.shp'
        grid_resume.at[i, 'Chunk_Length'] = chunk_length / 1000
        grid_resume.at[i, 'Chunk_Cost'] = chunk_cost / 1000
        chunk_grid.to_file(fileout)
        total_chunk = gpd.GeoDataFrame(pd.concat([total_chunk, chunk_grid],
                                                 sort=True))

    elif points_to_electrify == 2:
        print('-' * 100)
        print('CREATING CHUNK FOR CLUSTER ' + str(i))
        print('-' * 100)
        p1 = gdf_zoomed_pop[gdf_zoomed_pop['ID'] == gdf_zoomed_pop.ID.values[0]]
        p2 = gdf_zoomed_pop[gdf_zoomed_pop['ID'] == gdf_zoomed_pop.ID.values[1]]
        chunk_grid, chunk_cost, chunk_length = \
            grid_direct_connection(geodf_in, p1, p2, Proj_coords, paycheck,
                                                                  resolution
                                                                      )
        fileout = 'Chunk_' + str(i) + '.shp'
        grid_resume.at[i, 'Chunk_Length'] = chunk_length / 1000
        grid_resume.at[i, 'Chunk_Cost'] = chunk_cost / 1000
        chunk_grid.to_file(fileout)
        total_chunk = gpd.GeoDataFrame(pd.concat([total_chunk, chunk_grid],
                                                 sort=True))

    elif points_to_electrify < 2:
        print('-' * 100)
        print('NO CHUNK NECESSARY FOR CLUSTER ' + str(i))
        print('-' * 100)
        fileout = 'Chunk_' + str(i) + '.shp'
        grid_resume.at[i, 'Chunk_Length'] = 0
        grid_resume.at[i, 'Chunk_Cost'] = 0
total_chunk.crs = geodf_in.crs
total_chunk.to_file('total_chunk')

#  START OF THE BRANCH TECHNIQUES

pop_threshold = 1  #  change of parameters to collaterals
paycheck = 3333
for i in clusters_list_2:

    gdf_clusters_pop = gdf_clusters[gdf_clusters['clusters'] == i]
    gdf_clusters_pop = gdf_clusters_pop[gdf_clusters_pop['Population']
                                        >= pop_threshold]

    # ASSIGNING SUBSTATIONS
    substations_new = substations
    if clusters_load[np.where(clusters_list_2 == i)[0][0]] > limitHV:
        substations_new = substations[substations['Type'] == 'HV']
    elif clusters_load[np.where(clusters_list_2 == i)[0][0]] > \
            limitMV and clusters_load[
        np.where(clusters_list_2 == i)[0][0]] < limitHV:
        substations_new = substations[substations['Type']
                                      == 'MV_3P']
```

```
    elif clusters_load[np.where(clusters_list_2 == i)[0][0]] \
            < limitMV:
        substations_new = substations[substations['Type'] != 'HV']

    unary_union = geodf_in.unary_union
    substations_new['nearest_id'] = substations_new.apply\
        (nearest, geom_union=unary_union, df1=substations_new,
         df2=geodf_in, geom1_col='geometry', src_column='ID', axis=1)
    subinmesh = gpd.GeoDataFrame(crs=from_epsg(Proj_coords))
    for k, row in substations_new.iterrows():
        subinmesh = subinmesh.append(geodf_in[geodf_in['ID'] ==
                row['nearest_id']], sort=False)
    subinmesh.reset_index(drop=True, inplace=True)

    d1 = {'x': gdf_clusters_pop['X'], 'y': gdf_clusters_pop['Y'],
          'z': gdf_clusters_pop['Elevation']}
    cluster_loc = pd.DataFrame(data=d1)
    cluster_loc.index = gdf_clusters_pop['ID']

    d2 = {'x': subinmesh['X'], 'y': subinmesh['Y'], 'z': subinmesh['Elevation']}
    sub_loc = pd.DataFrame(data=d2)
    sub_loc.index = subinmesh['ID']
    Distance_3D = pd.DataFrame(cdist(cluster_loc.values,
                                     sub_loc.values, 'euclidean'),
                               index=cluster_loc.index,
                               columns=sub_loc.index)
    mm = Distance_3D.min().idxmin()
    substation_designata = subinmesh[subinmesh['ID'] == mm]
    uu_substation = substation_designata.unary_union
    connectiontype = substations_new[substations_new['nearest_id'] == mm]
    grid_resume.at[i, 'ConnectionType'] = connectiontype.Type.values[0]

    if grid_resume.at[i, 'Chunk_Length'] == 0:
        print('-' * 100)
        print('CREATING GRID FOR CLUSTER ' + str(i))
        print('-' * 100)
        chunk_grid, chunk_cost, chunk_length, connections2 = \
            Steinerman(geodf_in, gdf_clusters_pop, Proj_coords,
                       paycheck, resolution)
        fileout = "Chunk_" + str(i) + '.shp'
        chunk_grid.to_file(fileout)
        grid_resume.at[i, 'Grid_Length'] = chunk_length / 1000
        grid_resume.at[i, 'Grid_Cost'] = chunk_cost / 1000
        total_branch_steiner = \
            gpd.GeoDataFrame(pd.concat([total_branch_steiner,
                                        chunk_grid], sort=True))

        uu_chunk = chunk_grid.unary_union
        nearpoints = nearest_points(uu_chunk, uu_substation)
        p1 = gdf_clusters[gdf_clusters['geometry'] == nearpoints[0]]
        if p1.ID.values[0] == substation_designata.ID.values[0]:
            grid_resume.at[i, 'Connection_Length'] = 0
            grid_resume.at[i, 'Connection_Cost'] = 0
            continue
        connection_grid, connection_cost, connection_length \
            = grid_direct_connection(geodf_in, p1, substation_designata,
                                     Proj_coords, paycheck, resolution)

        total_connections = gpd.GeoDataFrame(pd.concat([total_connections,
                                               connection_grid], sort=True))
```

```
235         fileout = "Connection_Grid" + str(i) + '.shp'
236         connection_grid.to_file(fileout)
237         grid_resume.at[i, 'Connection_Length'] = connection_length / 1000
238         if connectiontype.Type.values[0] == 'HV':
239             grid_resume.at[i, 'Connection_Cost'] = connection_cost / 1000 + \
240                                                    extracostHV / 1000
241         else:
242             grid_resume.at[i, 'Connection_Cost'] = connection_cost / 1000
243         continue
244
245     # ASSIGNING WEIGHT EQUAL TO 0 TO THE MAIN CHUNK AND PERFORMING STEINERMAN
246
247     chunk = gpd.read_file("Chunk_" + str(i) + ".shp")
248     print('-' * 100)
249     print('CREATING DERIVATION FOR CLUSTER ' + str(i))
250     print('-' * 100)
251     geodf_noweight = geodf_in
252     for id in chunk.ID1:
253         geodf_in_id = geodf_in[geodf_in['ID'] == id]
254         geodf_in_id.loc[:, 'Weight'] = 0.001
255         geodf_noweight.loc[np.where(geodf_in.ID == id)[0][0], 'Weight'] = 0.001
256     last_id = chunk.loc[chunk.ID2.size - 1, 'ID2']  # this lines are
257     # necessary because the last point was not taken
258     geodf_in_id = geodf_in[geodf_in['ID'] == last_id]
259     geodf_in_id.loc[:, 'Weight'] = 0.001
260     geodf_noweight[np.where(geodf_in.ID == last_id)[0][0], 'Weight'] = 0.001
261
262     chunk_grid, chunk_cost, chunk_length, connections2 = \
263         Steinerman(geodf_noweight, gdf_clusters_pop, Proj_coords, paycheck,
264                    resolution)
265     count = 0
266
267     for value in chunk_grid.values:
268         if value[2] in chunk.values and value[3] in chunk.values:
269             chunk_grid = chunk_grid.drop((np.where(chunk_grid.values[:, 1]
270                                                    == value[1])[0][0]))
271             chunk_grid = chunk_grid.reset_index(drop=True)
272     chunk_cost = sum(chunk_grid['Weight'])
273     grid_resume.at[i, 'Grid_Cost'] = chunk_cost / 1000
274     grid_resume.at[i, 'Grid_Length'] = chunk_length / 1000
275     fileout = "Steiner_Grid" + str(i) + '.shp'
276     chunk_grid.to_file(fileout)
277     total_branch_steiner = gpd.GeoDataFrame(pd.concat([total_branch_steiner,
278                                                        chunk_grid], sort=True))
279     chunk_inmesh = pd.DataFrame()
280     for id in chunk.ID1:  # getting all the points inside
281         # the chunk to compute the nearest_points
282         chunk_inmesh = gpd.GeoDataFrame(pd.concat([chunk_inmesh,
283             gdf_clusters[gdf_clusters['ID'] == id]], sort=True))
284     last_id = chunk.loc[chunk.ID2.size - 1, 'ID2']  # this lines are
285     # necessary because the last point was not taken
286     chunk_inmesh = gpd.GeoDataFrame(pd.concat([chunk_inmesh,
287                 gdf_clusters[gdf_clusters['ID'] == last_id]], sort=True))
288     uu_chunk = chunk_inmesh.unary_union
289     nearpoints = nearest_points(uu_chunk, uu_substation)
290     p1 = gdf_clusters[gdf_clusters['geometry'] == nearpoints[0]]
291     if p1.ID.values[0] == substation_designata.ID.values[0]:
292         grid_resume.at[i, 'Connection_Length'] = 0
293         grid_resume.at[i, 'Connection_Cost'] = 0
294         continue
```

```
295
296     connection_grid, connection_cost, connection_length = \
297         grid_direct_connection(geodf_in, p1, substation_designata,
298                                Proj_coords, paycheck, resolution)
299
300     total_connections = gpd.GeoDataFrame(pd.concat([total_connections,
301                                             connection_grid], sort=True))
302     fileout = "Connection_Grid" + str(i) + '.shp'
303     connection_grid.to_file(fileout)
304     grid_resume.at[i, 'Connection_Length'] = connection_length / 1000
305     if connectiontype.Type.values[0] == 'HV':
306         grid_resume.at[i, 'Connection_Cost'] = connection_cost / 1000 \
307                                                + extracostHV / 1000
308     else:
309         grid_resume.at[i, 'Connection_Cost'] = connection_cost / 1000
310
311
312 #   CONNECTION OF POINTS OUTSIDE THE CLUSTERS
313
314 total_grid = total_branch_steiner
315 total_grid.reset_index(inplace=True, drop=True)
316 total_link = grid_points = total_link_nodups = pd.DataFrame()
317 gdf_clusters_out = gdf_clusters[gdf_clusters['clusters'] == -1]
318 gdf_clusters_out = gdf_clusters_out[gdf_clusters_out['Population'] >= 1]
319
320 for id in total_grid.ID1:  #getting all the points inside the
321     # chunk to compute the nearest_points
322     grid_points = gpd.GeoDataFrame(pd.concat([grid_points,
323                         gdf_clusters[gdf_clusters['ID'] == id]], sort=True))
324 last_id = total_grid.loc[total_grid.ID2.size - 1, 'ID2']  # this
325 # lines are necessary because the last point was not taken
326 grid_points = gpd.GeoDataFrame(pd.concat([grid_points,
327         gdf_clusters[gdf_clusters['ID'] == last_id]], sort=True))
328 uu_grid = grid_points.unary_union
329
330
331 for i in gdf_clusters_out.geometry:
332     nearpoints = nearest_points(uu_grid, i)
333     p1 = gdf_clusters[gdf_clusters['geometry'] == nearpoints[0]]
334     p2 = gdf_clusters[gdf_clusters['geometry'] == nearpoints[1]]
335
336     link, link_cost, link_length = grid_direct_connection(geodf_in,
337                                p1, p2, Proj_coords, paycheck, resolution)
338
339     total_link = gpd.GeoDataFrame(pd.concat([total_link, link], sort=True))
340
341     #  REMOVING DUPLICATIONS
342 uniqueID = np.unique(total_link.ID1, return_index=True)
343 total_link['Index'] = np.arange(total_link.values.__len__()).\
344     reshape(total_link.values.__len__(),1)
345 total_link_nodups = pd.DataFrame()
346 for id in uniqueID[1]:
347     total_link_nodups = gpd.GeoDataFrame(pd.concat([total_link_nodups,
348                         total_link[total_link['Index'] == id]], sort=True))
349 link_cost = total_link_nodups.Weight.sum()
350 link_length = total_link_nodups.length.sum()
351 grid_resume.at[0, 'Link_Length'] = link_length/1000
352 grid_resume.at[0, 'Link_Cost'] = link_cost/1000
353
354 total_branch_steiner.crs = total_branch_spider.crs =\
```

```
    all_branch_direct.crs = total_connections.crs = \
    total_link_nodups.crs = geodf_in.crs
total_branch_steiner.to_file('total_branch_steiner')
total_connections.to_file('total_connections')
total_link_nodups.to_file('total_link')
grid_resume.to_csv('grid_resume.csv')
grid_optimized = grid_optimization(gdf_clusters, geodf_in,
                grid_resume, Proj_coords, resolution, paycheck)


# SIZING EACH DERIVATION -
for cluster in clusters_list_2:
    if grid_resume.at[cluster, 'Chunk_Length'] == 0:
        continue
    derivations = gpd.read_file('Steiner_Grid' + str(cluster) + '.shp')
    chunk = gpd.read_file('Chunk_' + str(cluster) + '.shp')
    for i in derivations.values:
        if i[2] in chunk.values and i[3] in chunk.values:
            derivations = derivations.\
                drop((np.where(derivations.values[:, 1] == i[1])[0][0]))
            derivations = derivations.reset_index(drop=True)
    count = 1
    G = nx.Graph()
    G.add_edges_from([*zip(list(derivations.ID1),list(derivations.ID2))])
    components = list(nx.algorithms.components.connected_components(G))
    for i in components:
        points = np.array(list(i))
        pop_derivation = 0
        for j in points:
            for k in range(derivations.values.__len__()):
                if derivations.at[k,'ID1'] == j or \
                        derivations.at[k,'ID2'] == j:
                    derivations.at[k, 'id'] = count
            pop_derivation = pop_derivation + \
             gdf_clusters[gdf_clusters['ID'] == j].Population.values
        derivations.loc[derivations['id'] == count,
                        'Power'] = int(pop_derivation)*0.4
        count = count+1
    total_derivations = gpd.GeoDataFrame(pd.concat([total_derivations,
                                                    derivations], sort=True))
total_derivations.crs = geodf_in.crs
total_derivations.to_file('derivations')
```

Bibliography

Amador, J., and J. Domínguez (2005), Application of geographical information systems to rural electrification with renewable energy sources, *Renewable Energy*, *30*(12), 1897–1912, doi: 10.1016/j.renene.2004.12.007.

Ankerst, M., M. M. Breunig, H.-p. Kriegel, and J. Sander (1999), OPTICS: Ordering Points To Identify the Clustering Structure, *ACM SIGMOD Record*, *28*(2), 49–60.

Bakkabulindi, G., M. R. Hesamzadeh, M. Amelin, and I. P. Da Silva (2012), Planning algorithm for Single Wire Earth Return distribution networks, *IEEE Power and Energy Society General Meeting*, pp. 1–7, doi: 10.1109/PESGM.2012.6344816.

Bemmelen, J., W. Quak, M. Hekken, and P. Oosterom (1993), Vector vs . raster-based algorithms for cross country movement planning, *Proceedings Auto-Carto*, *11*.

Bertollo, H. C. (2008), Contribuições ao Estudo dos Aterramentos de Sistemas Monofilares com Retorno pelo Terra Contribuições ao Estudo dos Aterramentos de Sistemas Monofilares com Retorno pelo Terra, Master's thesis, Universidade Federal di Viçosa.

Bongiorni, and Civardi (2010), Innovative solution for rural areas electrification: T-pass, Master's thesis, Politecnico di Milano.

Brooking, T., and J. Van Rensburg (1992), The Improved Utilisation of Existing Rural Networks With The Use of Intermediate Voltage And Single Wire Earth Return Systems, *IEEE*, pp. 228–234.

Carnovali, T., and D. Edeme (2019), GISEle: an innovative GIS-based approach for electric networks routing, Ph.D. thesis, Politecnico di Milano.

CEPEL (2002), Seleção de Sistemas - MRT, *Tech. rep.*, Centro de Pesquisas de Energia Eletrica.

Cevallos-Sierra, J., and J. Ramos-Martin (2018), Spatial assessment of the potential of renewable energy: The case of Ecuador, *Renewable and Sustainable Energy Reviews*, *81*(July 2016), 1154–1165, doi: 10.1016/j.rser.2017.08.015.

Choi, Y., H. D. Park, C. Sunwoo, and K. C. Clarke (2009), Multi-criteria evaluation and least-cost path analysis for optimal haulage routing of dump trucks in large scale open-pit mines, *International Journal of Geographical Information Science*, *23*(12), 1541–1567, doi: 10.1080/13658810802385245.

Choi, Y., J.-G. Um, and M.-H. Park (2014), Finding least-cost paths across a continuous raster surface with discrete vector networks, *Cartography and Geographic Information Science*, *41*(1), 75–85, doi: 10.1080/15230406.2013.850837.

Divya, T. L., and M. N. Vijayalakshmi (2015), Analysis of wild fire behaviour in wild conservation area using image data mining, *Proceedings of 2015 IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2015*, pp. 1–3, doi: 10.1109/ICECCT.2015.7226082.

Domínguez, J., and I. Pinedo-Pascua (2009), GIS tool for rural electrification with renewable energies in Latin America, *Proceedings of the International Conference on Advanced Geographic Information Systems and Web Services, GEOWS 2009*, (March 2009), 171–176, doi: 10.1109/GEOWS.2009.25.

Enel GI&N (), Enel webpage - connect with us, Contact on https://www.enel.it/en/contattaci, accessed: 2020.03.23.

Ester, M., H. Kriegel, J. Sander, and X. Xiaowei (1996), A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *KDD-96*, *2*, doi: 10.1016/B978-044452701-1.00067-3.

Euler, L. (1736), Solutio problematis ad geometriam situs pertinentis, doi: 10.1017/cbo9781139058292.006.

Fahmi, M. I., R. Rajkumar, R. Arelhi, R. Rajkumar, and D. Isa (2014), Solar PV system for off-grid electrification in rural area, *IET Seminar Digest*, *2014*(CP659), 1–6, doi: 10.1049/cp.2014.1496.

Fandi, J. (2013), Sistema de Distribuição de Energia Elétrica a Dois Condutores para Atendimento a Cargas Rurais Trifásicas, Ph.D. thesis, Universidade Federal de Uberlandia.

Farret, F. A., and M. G. Simoes (2006), *Micropower System Modeling with Homer*, Integration of Alternative Sources of Energy, IEEE.

Gan, G., C. Ma, and J. Wu (2007), Data Clustering: Theory, Algorithms, and Applications, *Data Clustering: Theory, Algorithms, and Applications*, (January 2007), doi: 10.1137/1.9780898718348.

Gardner, L. (2020), An interactive web-based dashboard to track covid-19 in real time, Available on https://gisanddata.maps.arcgis.com/apps/opsdashboard/index.html#/bda7594740fd40299423467b48e9ecf6, accessed: 2020.03.04.

Gaughan, A. E., F. R. Stevens, C. Linard, P. Jia, and A. J. Tatem (2013), High resolution population distribution maps for southeast asia in 2010 and 2015, *PLOS ONE*, *8*, 1–11, doi: 10.1371/journal.pone.0055882.

Ghiani, E., C. Vertuccio, and F. Pilo (2016), Optimal sizing of multi-generation set for off-grid rural electrification, *IEEE Power and Energy Society General Meeting*, *2016-Novem*, 1–5, doi: 10.1109/PESGM.2016.7741718.

Han, J., M. Kamber, and J. Pei (2012), *Data mining concepts and techniques, third edition*, Morgan Kaufmann Publishers, Waltham, Mass.

Hosseinzadeh, N., J. E. Mayer, and P. J. Wolfs (2011), Rural single wire earth return distribution networks - Associated problems and cost-effective solutions, *International Journal of Electrical Power and Energy Systems*, *33*(2), 159–170, doi: 10.1016/j.ijepes.2010.08.009.

Huertas, J. S. C., and M. C. Tavares (2016), Rural electrification using capacitive induced voltage on transmission lines' shield wires, *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, *2016-Decem*(2014), 89–93, doi: 10.1109/APPEEC.2016.7779476.

Huertas, J. S. C., and M. C. Tavares (2019), Analyzing Rural Electrification Topologies Based on Induced Voltage at Insulated Shielding Wires, *IEEE Transactions on Power Delivery*, *34*(1), 53–62, doi: 10.1109/TPWRD.2018.2880427.

IEA (2019a), Sdg7: Data and projections, Available on https://www.iea.org/reports/sdg7-data-and-projections, accessed: 2020.03.13.

IEA (2019b), World energy outlook 2019, Available on https://www.iea.org/reports/world-energy-outlook-2019, accessed: 2020.03.13.

Iliceto, F. (2016), Rural Electrification with the Shield Wire Scheme in Low-Income Countries, *Tech. rep.*, doi: 10.1596/26647.

JRC (2000), Global land cover 2000 - products, Available on https://forobs.jrc.ec.europa.eu/products/glc2000/products.php, accessed: 2020.03.04.

Kaijuka, E. (2007), GIS and rural electricity planning in Uganda, *Journal of Cleaner Production*, *15*(2), 203–217, doi: 10.1016/j.jclepro.2005.11.057.

Karhammer, R., A. Sanghvi, E. Fernstrom, M. Aissa, J. Arthur, J. Tullock, I. Davies, S. Bergman, and S. Mathur (2006), Sub-Saharan Africa: Introducing Low Cost Methods in Electricity Distribution Networks, *ESMAP Technical Paper 104/06*, (October), 427.

Kashem, M. A., and G. Ledwich (2004), Distributed generation as voltage support for single wire earth return systems, *IEEE Transactions on Power Delivery*, *19*(3), 1002–1011, doi: 10.1109/TPWRD.2003.822977.

Kolhe, M., K. M. Ranaweera, and A. G. Gunawardana (2014), Techno-economic analysis of off-grid hybrid renewable energy system for Sri Lanka, *2014 7th International Conference on Information and Automation for Sustainability: "Sharpening the Future with Sustainable Technology", ICIAfS 2014*, pp. 1–5, doi: 10.1109/ICIAFS.2014.7069572.

Lee, J., and D. Stucky (1998), On applying viewshed analysis for determining least-cost paths on digital elevation models, *International Journal of Geographical Information Science*, *12*(8), 891–905, doi: 10.1080/136588198241554.

Maleš-Sumić, H., and S. S. Venkata (1993), Automated underground residential distribution design part 2: Prototype implementation and results, *IEEE Transactions on Power Delivery*, *8*(2), 644–650, doi: 10.1109/61.216871.

Monteiro, C., I. J. Ramírez-Rosado, V. Miranda, P. J. Zorzano-Santamaría, E. García-Garrido, and L. A. Fernández-Jiménez (2005), GIS spatial analysis applied to electric line routing optimization, *IEEE Transactions on Power Delivery*, *20*(2 I), 934–942, doi: 10.1109/TPWRD.2004.839724.

Moss, T. (2019), Global energy inequality goes deeper than bitcoin, Available on https://onezero.medium.com/global-energy-inequality-goes-deeper-than-bitcoin-dfd058c31330, accessed: 2020.03.13.

Orsini, N. (2013), Innovative Solution for Power Supply of Auxiliary Services of High Voltage Substations: TIP, Master's thesis, Politecnico di Milano.

Peña, J. M., J. A. Lozano, and P. Larrañaga (1999), An empirical comparison of four initialization methods for the K-Means algorithm, *Pattern Recognition Letters*, *20*(10), 1027–1040, doi: 10.1016/S0167-8655(99)00069-0.

Pfenninger, S., and I. Staffell (2016), Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data, *Energy*, *114*, 1251 – 1265.

Ramos, J. E., A. Piantini, V. A. Pires, and A. D'Ajuz (2009), The Brazilian experience with the use of the Shield Wire Line technology (SWL) for energy distribution, *IEEE Latin America Transactions*, *7*(6), 650–656, doi: 10.1109/TLA.2009.5419362.

Ramos, J. E., A. Piantini, V. A. Pires, and A. D'Ajuz (2018), Desempenho De Linhas de Transmissão Com Cabos Para-raios, *Congresso Tecnico Cientifico da Engenharia e da Agronomia, CONTECC2018*.

Rhodes, J. D., W. J. Cole, C. R. Upshaw, T. F. Edgar, and M. E. Webber (2014), Clustering analysis of residential electricity demand profiles, *Applied Energy*, *135*, 461 – 471, doi: https://doi.org/10.1016/j.apenergy.2014.08.111.

Robins, G., and A. Zelikovsky (2008), Minimum Steiner Tree Construction*, *Handbook of Algorithms for Physical Design Automation*, doi: 10.1201/9781420013481.ch24.

Santuari, A. (2003), Steiner Tree NP-completeness Proof, *Transform*, pp. 1–5.

Song, R., S. Lu, T. Sirojan, B. T. Phung, and E. Ambikairajah (2017), Power quality monitoring of single-wire-earth-return distribution feeders, *International Conference on High Voltage Engineering and Power Systems, ICHVEPS 2017 - Proceeding*, *2017-Janua*, 404–409, doi: 10.1109/ICHVEPS.2017.8225879.

Taufik, T. (2014), The DC House project: An alternate solution for rural electrification, *Proceedings of the 4th IEEE Global Humanitarian Technology Conference, GHTC 2014*, pp. 174–179, doi: 10.1109/GHTC.2014.6970278.

Tully, S. (2006), The human right to access electricity, *The Electricity Journal*, *19*, 30–39, doi: 10.1016/j.tej.2006.02.003.

USGS (2020), Usgs eros archive - digital elevation - global 30 arc-second elevation (gtopo30), Available on https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-global-30-arc-second-elevation-gtopo30?qt-science$_c enter_o bjects = 0 qt - science_c enter_o bjects, accessed$: 2020.03.15.

Wiernes, P. E., N. Van Bracht, A. Moser, and S. Bohlen (2015), A novel geo-spatial clustering tool applied to power system strategic planning, *Proceedings - International Conference on Modern Electric Power Systems, MEPS 2015*, pp. 1–6, doi: 10.1109/MEPS.2015.7477156.

Williamson, S. G. (2010), Lists , Decisions and Graphs With an Introduction to Probability Unit GT : Basic Concepts in Graph Theory Edward A . Bender.

WorldBank (2018), Access to energy is at the heart of development, Available on https://www.worldbank.org/en/news/feature/2018/04/18/access-energy-sustainable-development-goal-7, accessed: 2020.03.13.

Ye Wu, B., and K. Chao (2004), Steiner Minimal Trees, in *Spanning Trees and Optimization Problems*, 1, pp. 1–6, doi: 10.1137/0116001.

Zhu, L., J. Zhu, C. Bao, L. Zhou, C. Wang, and B. Kong (2018), Improvement of DBSCAN algorithm based on adaptive EPS parameter estimation, *ACM International Conference Proceeding Series*, doi: 10.1145/3302425.3302493.