POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Master of Science in Energy Engineering

GIS TOOL FOR ELECTRIC GRID MODELLING, OPERATION AND PLANNING

Supervisor: Prof. Marco Merlo
Co-supervisor: Ing. Matteo Moncecchi
Co-supervisor: Don Lino Pruneri

Candidates...................Alberto Gallizia
ID Number.................................898015

Academic Year 2018-2019

A padre Ugo,
a tutti meravigliosi volontari del Matogrosso

Abstract

"Ensure access to affordable, reliable, sustainable and modern energy for all" by the year 2030 represent the **Sustainable Development Goal 7**. According to United Nations in 2017 around 840 million of people are living without access to electricity, while approximately 1.0 billion people are connected to unreliable electric grid. Mini grid, due to recent improvement in technologies, falling costs and favorable enabling environments have become one of the best options to fill this gap. **The World Bank** has evaluated that mini grid could be the best option to reach, until 2030 at least 490 million people. Levelized Cost of Electricity (LCOE) of mini grids ranged from 0.55 to 0.85 [US$/kWh], by 2030 it is expected that mini grid **LCOE** decrease until 0.22 [US$/kWh]. One of the main challenge, developing countries rural mini grid system operators are facing, is the growing in operation complexity and need of planning with the increasing of grid size and number of costumers. A common tool actually used by system operators is a grid **GIS model**. Despite the simplicity and many advantages of this model, it is not possible to evaluate any grid electric behaviour. Simulating an electric grid model, in a **Power System Analysis Software**, it is instead possible to evaluate grid losses, line loading and other important parameters. In addition it is possible to evaluate grid weakness, simulating faults, abnormal operation and especially it is possible to arrange a robust long term planning. Since such grid electric model is not easy to be built by hand it is convenient to create it by the GIS one. The aim of this thesis is therefore to develop a semi-automatic procedure that generates an electric grid model from a GIS one. The procedure is implemented by a **Python** code, input are GIS table of elements, grid structure and load demand. Grid boundaries are finite to distribution grid, costumers load demand are delegated to MV-LV transformers load. To validate the procedure, it is tested to a case study, a rural grid in Chacas, Peru. Once the electric grid model is generated, some simulations regarding future grid improvements are performed.

Sommario

"Assicurare a tutti l'accesso a sistemi di energia economici, affidabili, sostenibili e moderni" entro il 2030 è **l'Obiettivo di Sviluppo Sostenibile** numero 7 ma secondo le Nazioni Unite nel 2017 ben 840 milioni di persone vivono ancora senza alcun accesso all'elettricità. La tecnologia delle Mini Grid, grazie a recenti sviluppi tecnologici, riduzioni dei costi e ad un ambiente di sviluppo più favorevole sta diventando una delle opzioni più realistiche per arrivare al raggiungimento di questo obiettivo. A tal proposito **The World Bank** ha identificato le Mini Grid come miglior opzione per raggiungere 490 milioni di persone entro il 2030. Il costo unitario equivalente dell'elettricità (LCOE), stimato per una minigrid si aggira fra i 0.55 e i 0.85 [US$/kWh], tale valore ci si aspetta che decresca, entro il 2030, intorno a 0.22 [US$/kWh]. Alcuni dei maggiori problemi che si trovano ad affrontare i gestori di reti elettriche rurali, in paesi in via di sviluppo, con il crescere delle dimensioni della rete e del numero di utenti, sono la sempre maggiore complessità della gestione giornaliera della rete e il bisogno di una sempre più accurata pianificazione. Spesso i gestori di rete usano dei **modelli GIS** per rappresentare la rete ma, nonostante la sua semplicità di utilizzo e i vari vantaggi che ne deriva, con questo strumento non è possibile valutare il comportamento elettrico della reta ma solo descriverne struttura e proprietà. Per simulare un modello elettrico di rete esistono degli appositi software coi quali è possibile ad esempio valutare perdite di carico, sovra caricamento delle linee ed altre importanti proprietà, inoltre è possibile valutare i punti deboli della rete, simulare guasti, individuare malfunzionamenti ma soprattutto tali software danno la possibilità di programmare delle pianificazioni di lungo periodo supportate da dati tecnici affidabili. Poiché non è un compito facile creare a mano tali modelli si è pensato alla possibilità di poterli generare da modelli GIS già esistenti; Lo scopo di questa tesi è appunto di creare una procedura semi automatica che crei un modello elettrico di una rete partendo da un modello GIS. Questa procedura, chiamata **HUGO**, viene eseguita con un apposito codice in Python, i dati in ingresso sono le tabelle attributi del file GIS, la struttura della rete e i carichi elettrici della rete. Il modello di rete ipotizzato si limita alla sola rete di distribuzione quindi i carichi sono rappresentati da coppie di valori (P, Q) attribuiti ai singoli trasformatori dalla media alla bassa tensione. Per poter validare la tesi è stata testata ad un caso di studio, la rete elettrica di Chacas, una valle fra le montagne del Perù. Una volta generato il modello della rete elettrica verranno eseguite alcune simulazioni, riguardanti delle reali modifiche programmate per la rete.

Contents

Chapter 1

Introduction

"Ensure access to affordable, reliable, sustainable and modern energy for all" by the year 2030 represent the **Sustainable Development Goal** number 7 (SDG7). According to The United Nations (UN) *SDG7 Tracking Report 2019* [1] in 2017 around 840 million of people are living without access to electricity, while according to [2] approximately 1.0 billion people are connected to unreliable electric grid. Access to affordable and reliable electricity is today considered essential to boost development and to fight poverty, it is also a basic condition to enable access to services, public goods and resources, therefore represent a solid basis for social progress and human empowerment.

In many emerging countries some basic social needs still have to be achieved, in this fight against poverty energy access plays a primary role in any fields such as for sanitation, water supply, in all the supply chain of agriculture, for communication technologies and education.

International organizations have created a huge amount of indicators to evaluate country's development, two of the most suitable in this context are the **Human Development Index** (HDI) and the **Energy Development Index** (EDI). HDI is a statistical tool used to measure a country's overall achievement in it's social and economic dimensions, EDI instead is a multidimensional indicator that measures the progress in transitioning to reliable, clean and efficient fuels and energy services. To confirm the key role of energy access in eradicating the poverty, the following figure 1.1 highlights the correlation between HDIs and EDIs indexes.

In the light of this situation many different actors have performed, are undertaking and are planning to do actions related to energy-related challenges. The world of Public and Private Donors is vast, from the big international or-

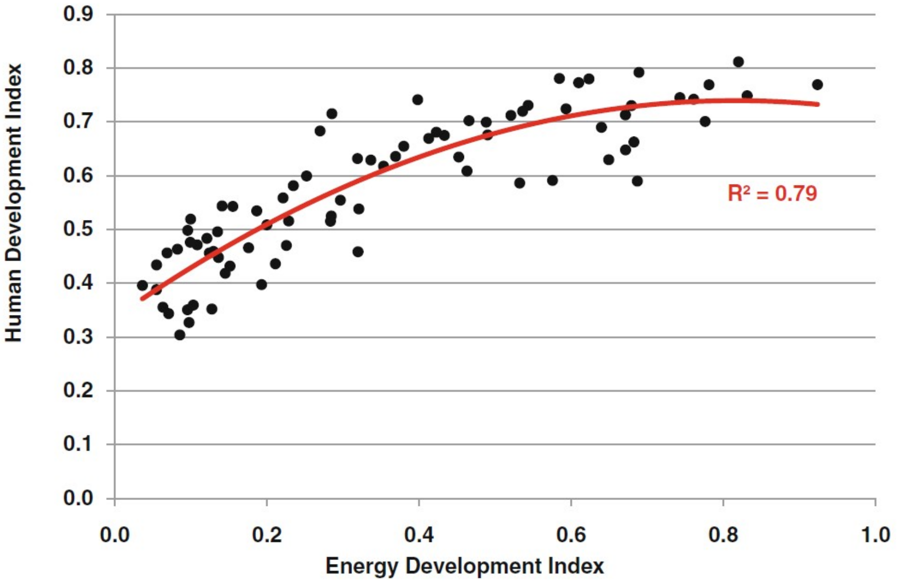


Figure 1.1: Comparison of the HDI to the EDI

ganizations, governmental agencies, development banks and funds to NGOs, power utilities associations, medium and small private funds.

1.1 Rural Electrification

The majority of people without access to electricity are living in developing countries and especially in rural areas of Africa, Developing Asia, Middle East and Latin America. According to UN [1] the rural access rate of 79% in 2017 was lower than the urban access rate of 97%. As indicated by the last updated **The World Bank** Database (2017) [3] and shown in figure 1.2, an impressive gap exist between the rural electrification rate of the most wealthy countries, sampled here by OECD members (about 100%), and Sub-Saharan Africa (22.6%).

Focusing our attention on rural electrification rate variation, during the last 25 years, of Latin America & Caribbean and in particular on Peru, it can be noted a slight increase of both values. In 2017 both Latin America & Caribbean (91.89%) and Peru (83.68%) have overtaken the rate respectively of 90% and 80%, this confirm that electrification was more rapid in rural areas than in cities, but also that a lot of work still need to be done, since the remaining population to be connected lives far away from the nearest electric grid, probably in very small villages of sparsely populated areas.

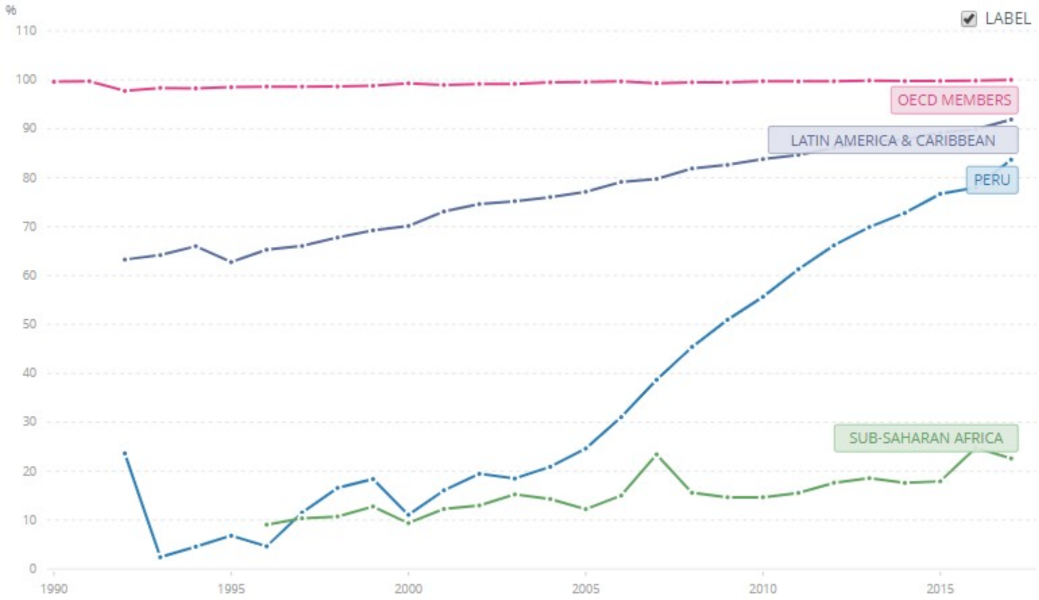


Figure 1.2: Access to electricity, rural (% of rural population) - Peru, Sub-Saharan Africa, Latin America and Caribbean, OECD members. The World Bank Database 1990-2017

As reported by the **Global Tracking Framework** 2015 [4] supplying electricity to rural areas is normally performed in two different approaches: extending an existing national grid or creating a brand new off-grid system. The convenience of a particular approach with respect to the other depends case by case, each county have different policies and tools for financing energy-related projects, it must also be considered the current state and coverage of national grid, the typology and state of existing power plants and the specific geography of the region to be electrified.

Despite that the first approach is considered the traditional one, and was applied by all developed countries during the last decades, in a contest such rural areas of Africa, Asia or Latin America grid extension can results to be economically unfeasible. These areas often involves large distances, are hardly accessible and have low values of load demand and load factor, installation costs and operating technical losses make it economically unfeasible. The second approach to supply electricity to rural areas is the creation of off-grid power systems. This approach involves the concept of **Distributed Generation** (DG), defined as small-scale electricity production near to the consumer, thus in these systems both generation and distribution are (logically) installed close to the load and sized in order to best fit with the local load demand. Off-grid power systems have also some social advantages because they normally use local resources, such as local work force, and are often integrated with renewable energy sources (RES).

1.2 Mini Grid

Both grid extension and off-grid approaches have advantages and disadvantages, an intermediate and worldwide appreciated technology is the so called mini grid power system, defined by [5] as an "electric power generation and distribution systems that provide electricity to just a few customers in a remote settlement or bring power to hundreds of thousands of customers in a town or city".

Mini grid received in the last years many attentions, and is actually considered an important candidate in the future power sector, thanks to the lack of investments in new transmission capacity, the recent advances in small-size generation technologies and the will of modern policies to increase RES penetration in the national energy mix. Other advantages of this technology, with respect to single DG, involves a wider exploitation of local energy resources, a better flexibility and economic benefits.

As previously introduced, in 2017 around 840 million people were living without access to electricity, due to recent improvement in technologies, falling costs and favorable enabling environments have made mini grid one of the best options to fill this gap, **The World Bank** in [5] has evaluated that mini grid could be the best option to reach, until 2030 at least 490 million people.

In term of size is difficult to define a mini grid, this category include a vast variety of projects of different scale, ranging from a few [kW] to several [MW] of installed capacity.

1.2.1 Current State of Mini Grid

Mini grids are generally the most economically viable option for servicing areas that are too expensive for the national grid but have high enough demand and population density to support commercial viability.

Generation technologies can be identified as conventional, non-conventional and hybrids. Conventional systems rely on classics technologies such as diesel generators and small hydro power plants, non-conventional systems are mainly Photovoltaic (PV), wind and storage. Hybrids technologies are system running with any mix of the previous sources, e.g. PV-diesel-storage or hydro-diesel-storage. Most of mini grids relies on conventional diesel powered generators, followed by hydro and solar-hybrid systems.

Globally, at least 19,000 mini grids are already installed in 134 countries and territories, representing a total investment of $28 billion, providing electricity to around 47 million people. Most of these projects was installed in South Asia (9.300), East Asia & Pacific (6.900), Africa (1.500) and OECD & Central Asia (1.100) [5].

Planned project of Mini Grid, according to [5] in 2019 are about 7.500, most of which in Africa (4.000) and South Asia (2.200). To support the development of these mini grids, in addition to governmental and private sector, many development partners contribute directly to financing projects with a current amount of 1.3 billion$ already committed for future projects.

The six organizations that are investing more than 100 million US$, in mini grid projects, are *The World Bank*, *Agence Française de Développement*, the *African Development Bank*, *Deutsche Gesellschaft für Internationale Zusammenarbeit*, the *Islamic Development Bank* and the *Department for International Development* of UK.

1.2.2 Prices and affordability

Levelized cost of electricity (LCOE), defined as the minimum average tariff at which electricity must be sold to cover project costs is a key indicator to evaluate economic performances of a power system. Another important indicator is **Load Factor**, defined as the ratio of average load demand and the total installed capacity, and especially its relationship with income-generating loads of a system. Indeed, residential and commercial loads have different peak hours, so the increase of income-generating activities will increase energy consumed without rising peak load, so load factor will increase too. Consequently increasing load factor of a power system will decrease LCOE.

ESMAP [5] calculated that LCOE of the actual mini grids ranged from 0.55 to 0.85 [US$/kWh] with a load factor of 22 percent. Further analysis, of the same source, indicate that the combination of increased income-generating uses and decreased component costs can bring mini grid LCOE down to 0.22 [US$/kWh] by 2030, with a power factor of 40 percent.

As previously mentioned, in 2017 still 21% of rural population rate had to be achieved by electrification, but what about the affordability of electricity service for the 79% that already had electricity access?

According to [1] and to the 2018 edition of **The World Bank's Regulatory Indicators for Sustainable Energy** (RISE) [6] the basic, subsistence-level electricity consumption of 30 [kWh/month] is unaffordable (i.e. it costs more than 5% of monthly household income) for the poorest 40% of households in half of the access-deficit countries. Basically it means that 285 million people in the world have access to electricity, but cannot afford the

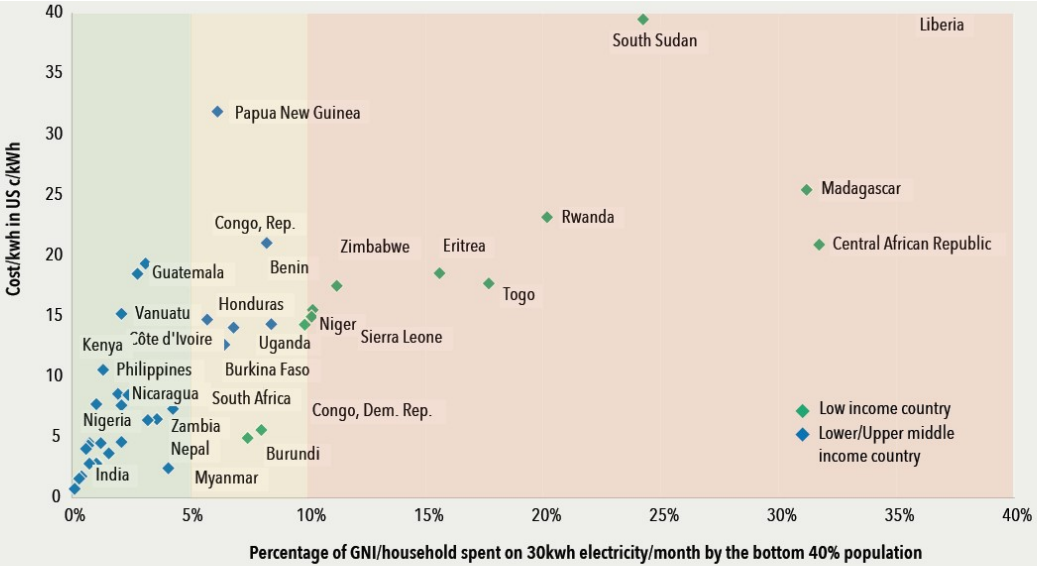


Figure 1.3: Electricity tariff as a share of GNI per household among the poorest 40% of household, by country, 2017. Source: RISE 2018, World Bank

corresponding bill. This data is evident in figure 1.3, the bottom 40% population of countries in yellow and red belts pay an electric bill higher than 5% of their own **Gross National Income** (GNI).

1.3 Importance of a Grid Model

The extension of national grid implies massive investments and projects, therefore are justified different studies and analysis. Supplying electricity with a mini grid need easier preliminary studies, lower investments and easier management, with the risk to incur in undervalued problems and a reduced capacity to future changes and grid expansion. One of the main issue of a easy mini grid management is the impossibility of an accurate acknowledgement of grid electrical status (such as voltage magnitude, phase displacement, line loading. . .).

This lack of information imply inefficiency in grid operation. In addition, bigger is the size of a grid, more is the negative impact of such inefficiencies. It is therefore necessary to fill the gap between simplicity of operation and an improved awareness of grid potentiality and its technical limits. Once recognized such limit of mini grid, it is thus important to estimate which are the main data we need to collect. Depending on the data available different typology of model can be built.

The main categories of data, necessary to generate an accurate grid model, are grid structure and load demand.

1.3.1 Load Demand

In the contest of Mini Grid, especially in cases of small and off-grid systems, it is a crucial issue the identification of load demand requested by local consumers and the creation of a grid model, in order to better evaluate grid operation, power flow analysis, problem that could occur and also possibilities for future extensions of the grid and changes in load behaviour.

Evaluation of load demand means the collection of data, from an existing database or from load estimation, about the electric use on consumers side in terms of energy consumed and power required by a single device, a single consumer, a group of consumers, several different groups within a specific area and of the entire grid. Obviously not all of these information are necessary for an ballpark estimate of a grid electrical status, more detailed information we have more accurate the model is. If load accurate information are not available it is possible to evaluate aggregation of data, normally easier to be estimate.

In a contest of rural electrification, the estimation of load demand have a greater importance since electricity demand must be met by almost all local resources and if well performed it allows an optimal sizing of plant, utilities, batteries which can signifying an important reduction in investments costs.

Load demand and energy needs can be performed through collection of existing historical data, when they are available or, in case of the implementation of a brand new power system, it is necessary a numeric estimation through sizing techniques.

1.3.2 GIS and CAD Grid Models

A useful tool of system grid operator are GIS and CAD programs. **Geographic Information Systems** (GIS) are programs that deal with models of geo-referenced objects, while **Computer Aided Design** (CAD) is a wider category of programs that generates graphical representations of physical object.

Thanks to these software it is possible to model a grid and all of its own utilities. A well modelled grid with geographic coordinates permits a useful grids representation in space and facilitate grid alteration, planning of future expansion and to save and visualize a huge amount of data regarding the grid.

A GIS, such as a CAD, model is normally composed of many layers. Each layer can represent a different set of utilities, such as MV lines or MV-LV transformers. Each element, or set of elements, is composed by many information stored in a special table called attribute table. An attribute table displays information on features of a selected layer. Each row in the table represents a feature (with or without geometry), and each column contains a particular piece of information about the feature. Features in the table can be searched, selected, moved or even edited. In these tables is possible also to collect data regarding costumers, their load, electric behaviour and bill report.

Despite attribute tables are useful tools for systems operators, they are nothing more than simple data bank. From a GIS or CAD it is not possible to perform any electric simulation or power flows, it is not easy to evaluate grid weakness such as overloaded lines, oversized ones or losses concentration. As previously indicated, a grid model is also important in a perspective of long term planning. Knowing both grid electric behaviour and an accurate prevision of future load estimation, will permit to perform a well suited planning of future investments and evaluations of expected problems.

1.3.3 Power System Analysis Software

To create grid models, many commercial power System Analysis Software (PSAS) were developed and are daily used all around the world. A PSAS is used to create models, analyze or calculate the design of power stations, overhead power lines, transmission towers, electrical grids, grounding and lightning systems and others. It is a type of application software which is used for power engineering problems, which are solved modelling them mathematically.

One of the most diffused PSAS is **PowerFactory**, developed by **DigSilent**. It is used to analysing generation, transmission, distribution and industrial systems. In the context of Rural electrification these tools are very useful especially for analyzing generation and distribution systems of unbalanced power systems.

Although power systems are normally designed to operate in balanced symmetrical three-phase sinusoidal conditions, there are many undesired situations called unbalanced conditions where it did not occur. In a such situation, one phase of three, or all of them, are not performing under the same conditions. An unbalanced condition can occur in many cases such as a huge different between phases load or phase impedance, an unexpected power outage in a specific phase or also when a non accurate planning was performed during the grid implementation.

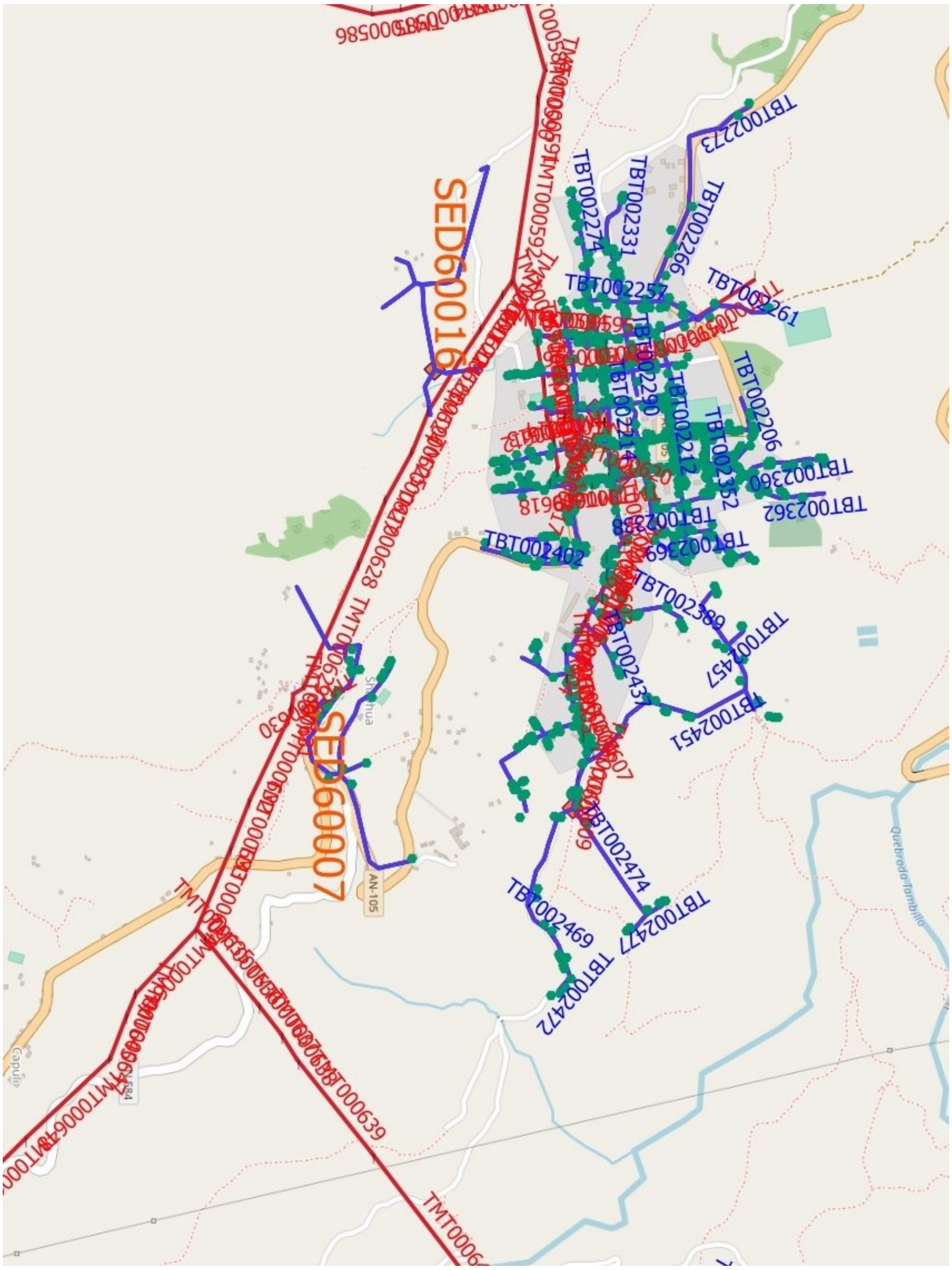


Figure 1.4: An example of distribution grid modelled in a GIS software (QGIS).

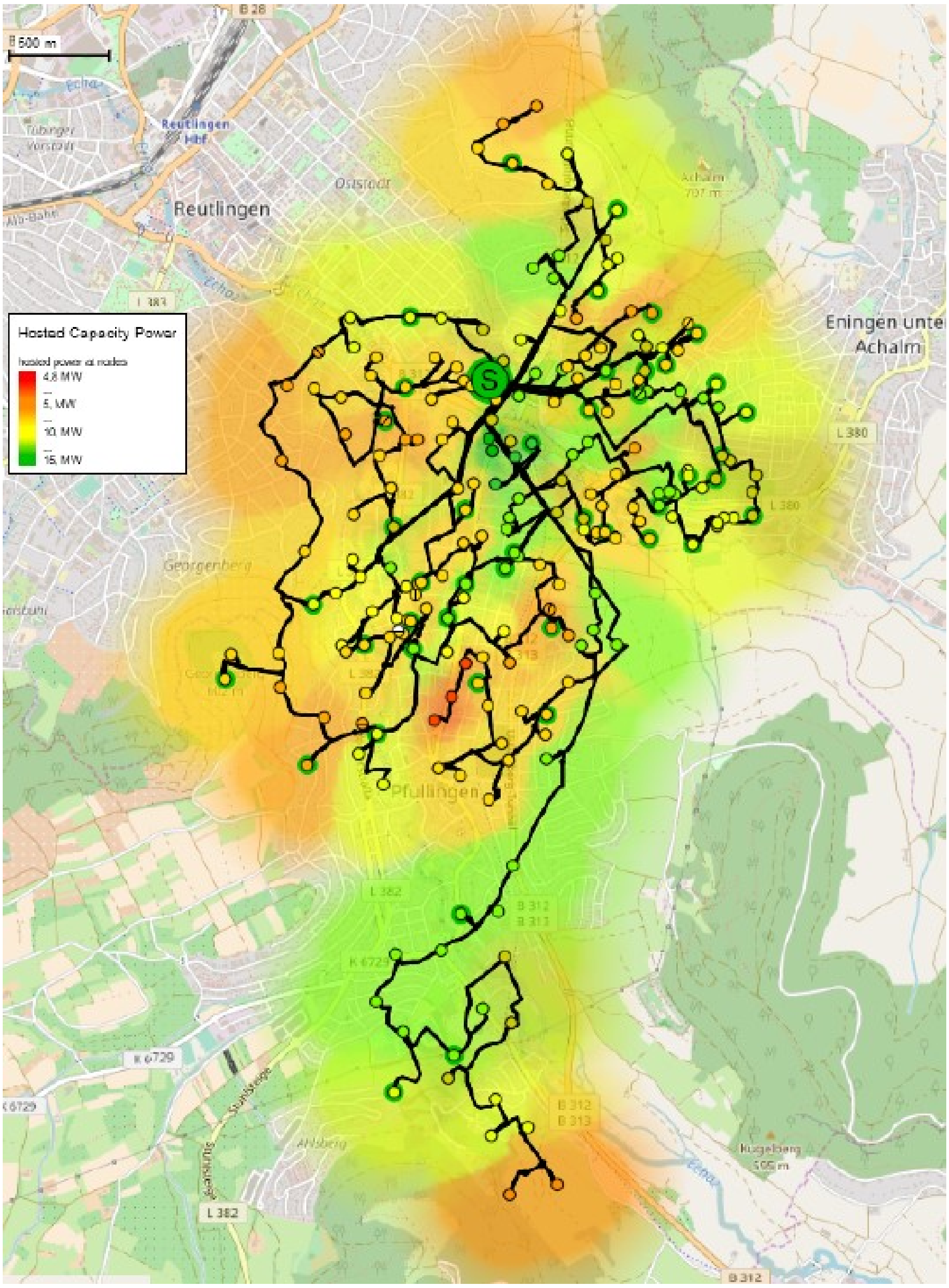


Figure 1.5: An example of distribution grid modelled in a power system analysis software (PowerFactory).

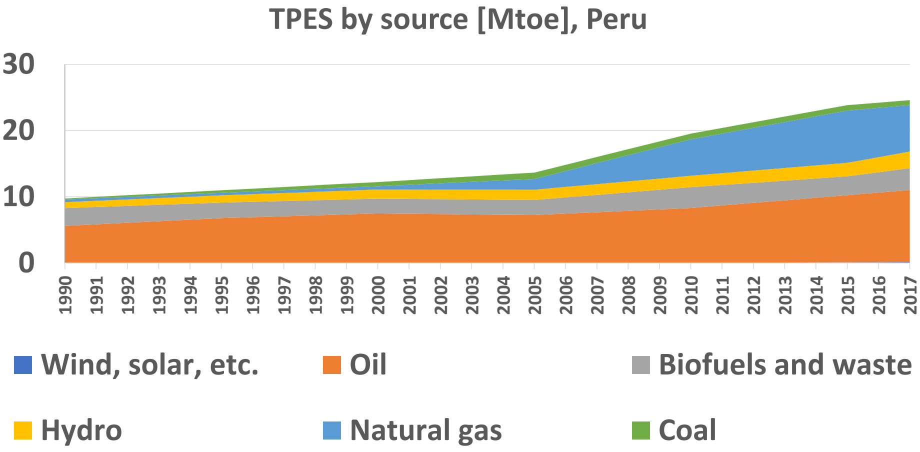


Figure 1.6: Electricity generation by source. IEA 2017 [8]

In order to evaluate the effect of unsymmetrical conditions it is necessary to perform a specific load flow calculation for a multiphase network representation. In this calculation each phase, with or without neutral return,is evaluated taking into account not symmetric working conditions on the other phases. Therefore many different parallel calculation are performed [7].

1.4 Peru, Chacas and Case Study

Peru is a developing country with a population of 31.99 million. Four people of five lives in urban areas while 7.07 million people lives in rural areas [3]. Electricity consumption is increasing year after year, in 2017 every Peruvian consumes 3.15 [MWh] of electricity, while the **Total Primary Energy supply** (TPES) per capita is 1.86 [toe/cap] [8].

The majority of electricity is generated with Coal (38.8%), Natural Gas (22.9%) and Hydro (16.3%), as it is possible to observe in figure 1.6. According to the **International Hydropower Association** [9] Peru in 2017 has 5.385 [MW] of installed capacity, with an hypothetical potential capacity around 70.000 [MW]. The Amazon basin region holds 97.7 per cent of Peru's water resources, where about a quarter of the population lives. The National Energy Plan 2014-2025 developed by the **Ministry of Energy and Min-**

ing (MINEM) [10] expects Peruvian energy demand to grow between 4.5 to 6.5 per cent a year by 2025, which in fact will be satisfied primarily by hydropower.

As soon as the constant increasing in electricity demand, in 2016 peak load capacity on Peruvian national grid reached only 6.560 [MW], considering an effective capacity of 12.078 [MW]. This electricity oversupply has positively affected electricity marginal prices. The marginal cost of energy has become the lowest in the South American countries, especially among the countries that have competitive power markets. According to [11] average liberalized electricity prices in 2017 was 46.93 [US$/kWh]. As it is possible to observe in figure 1.7 electricity marginal costs during the last ten years are constantly decreasing, while contract prices in the liberalized market is decreasing in a more soft pace.

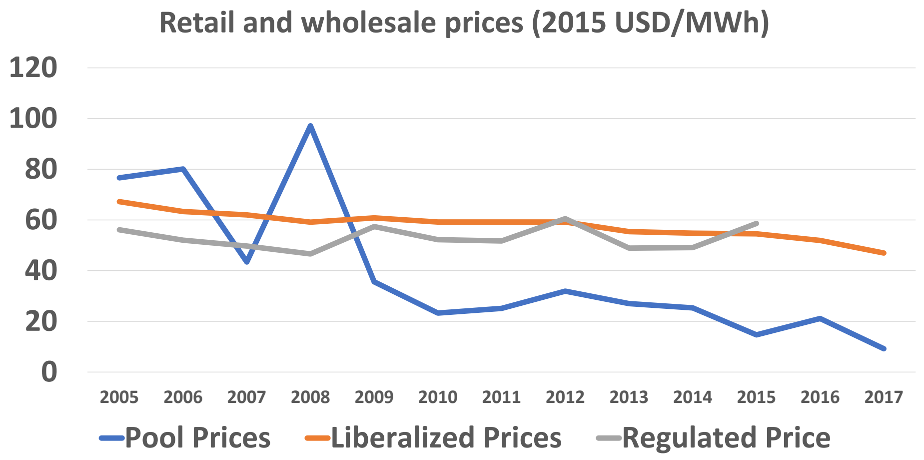


Figure 1.7: Electricity marginal costs and liberalized contracts prices. Peru [11]

1.4.1 Sistema Eléctrico Chacas

Chacas is a town in the district of Asunciòn, in Ancash region, located 300 km further north of Lima, the capital. It is located in a valley near Huascaran National Park in the Cordillera Blanca mountain range. Thanks to the proximity of a such mountain range there is a large availability of hydro power generation. The local electric power system, called **Sistema Eléc-**

trico Chacas, is widespread along all the valley. Electricity is generated by three hydropower plants thanks to three Francis and a Pelton turbine, all of them are fed by different run-off river water supply basins.

In 2019 the system was connected to 5.808 costumers. The great majority of costumers have a low load demand and energy request during the year, they were connected to the grid during the last few years and mainly relies on agriculture.

Turbine Name	Nominal Power [MW]	Voltage [kV]	Machine
Jambòn G1	0,784	0,44	Francis
Jambòn B	0,38	0,44	Francis
Collo	0,734	0,44	Francis
Huallin	4 (1,1)	6	Pelton

Table 1.1: Turbines available in Chacas-San Luis electric power system

In addition to the three hydropower plants electricity can be exchanged with the national grid through a small interconnection at the bottom of the valley, near the village of Pommalucay. At the moment, indeed, the Interconnection is mainly used to sell the electricity produced by Huallin power plant to the national grid.

1.4.2 Future of the grid

In the near future there are two important grid changes to be performed. The first one is the implementation by the MINEM of the following electrification step (V ETAPA) of the **National Plan of Electrification**, called *Plan Nacional de Electrification Rural* (PNER). The aim of V ETAPA is to connect 32 new villages, located in the bottom of the valley. V ETAPA is expected to be completed in 2022 and will deliver electricity to 489 new households, so to 1912 people.

The second important implementation is a new medium voltage (60 [kV]) connection to the national grid, for sake of simplicity we will call it *High Voltage Interconnection*, for better distinguish it from the 22.9 [kV] existing one. As previously explained, Huallin power plant is producing to sell whole the electricity to the national grid. This high voltage line will directly connect Huallin power plant to the grid, at the moment this power, in order to reach the actual interconnection, needs to cross all the power system along the valley. It is expected that removing this flows across the grid will lighten the maximum loading of the dorsal MV lines.

On the long period a new power plant is expected to be installed near the existing one in Collo. Information about installation year are not available, broadly it is expected to generate around 1 [MW].

Considering all of such information, and in addition the increasing in costumers and load demand (estimated by the local system operator as 5% per year), it is becoming crucial the development of an electric model. A simple costumer database and a GIS model are not anymore sufficient.

1.5 Purpose of the thesis

A grid model is necessary for evaluating the operation of a power system, identifying grid's strengths and weakness, it is a powerful tool for a robust planning of interventions and grid expansions. Besides all these things a three phase umbalanced grid model will also allow to identify differences between phases and the related problems. All of these benefits are expected to increase quality of the service, to ease daily operation for the local electricity distributor and, not least, decreasing electricity bill to final costumers.

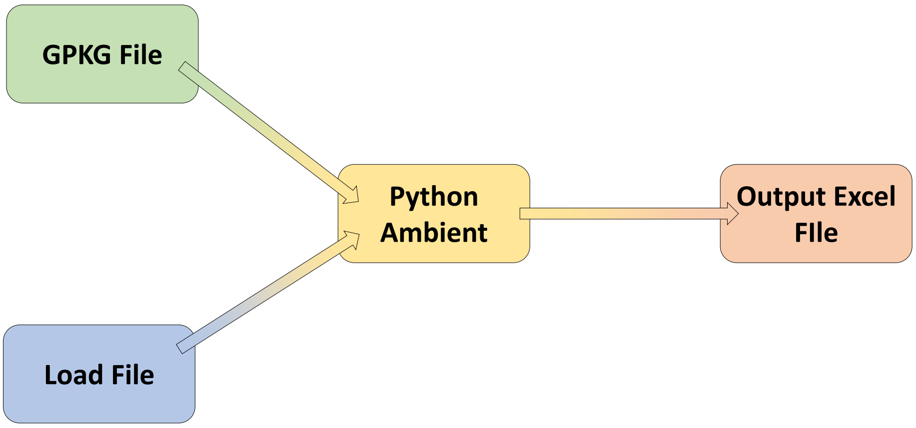


Figure 1.8: HUGO procedure

The aim of this thesis is the creation of a tool named (**HUGO**) for modelling a distribution grid in a power system analysis software starting from a GIS model of the same grid. The model created by **HUGO** is a three phases distribution grid, this will permits to perform three phases umbalanced power flows. GIS grid properties are collected in **GPKG** (GeoPackage is a standard for geospatial information file) file, the procedure developed collects these information about geometry and electric properties of a grid, integrated by other necessary information known a priori and a load demand estimation. Once collected all the information, **HUGO** creates a special output file that can be imported by power system analysis software. Procedure it is better explained in figure 1.8.

GIS information about the grid must be storage in a set of **GPKG** file that can be read by many **GIS** software. Load demand must be stored in another file (such as an Excel file) or a database.

Since **HUGO** works with distribution grids, model's boundary ends with MV-LV transformers so all loads of a certain area must be compressed into a couple of values: active and reactive power (P, Q). Therefore this (P-Q) couples of values, once assigned to the corresponding phase(s) will represent load demand of the entire village, or rural area, and will be modelled as a single three phase load connected to the transformer. Thus, on the power system analysis software, this (P-Q) couple will be substituted by three different couples, each representing a different phase. **HUGO** is written in a Python language, which is able to read and elaborate **GPKG** file, Excel file and databases.

HUGO output is a file, representing the grid model, that can be imported in a specific **PSAS**.

1.6 Fieldwork and a first implementation of HUGO

Since **HUGO** is not a completely automatic procedure, as we will better explain in chapter 2 the code must be tailored case by case. During October, November and December 2018 I was in Chacas to start thinking to the thesis, for collecting some data about Sistema Electrico Chacas and to do volunteer works. In this period, I was living in the parish administrated by **Operazione Matogrosso** (OMG) Italian community, a movement of missioners who have built many missions across the most poor region of South America, especially with Lino Pruneri, the system operator's chief executive, and his family.

OMG, thanks to free founding collected by volunteers in Italy, during the last 30 years have built almost all the distribution grid. Matogrosso's missioners also indirectly control **Eilhicha**, the system operator of Sistema Electrico Chacas, and Santa Lucia de la Parroquia de Chacas, a private firm owning Huallin power plant. **Eilhicha** is also the local retailer and controls the remaining three turbines of the power system.

1.6.1 Fieldwork

During this period I had the possibility to work with **Eilhicha**, and in a close contact with its 26 employees. Same of the many activities I have undertaken were useful for give an help to **Eilhicha** on it's daily operations with various activities, and also in preparation of the thesis. The mains are:

EILHICHA
Empresa de Interés Local
Hidroelétrica Chacas

Data Collection Eilhicha have a database collecting many information about costumers, load consumption, electricity generation, injection to and from the national grid and many others data about failures and issues of the grid. I learned how to extract data from the database and collected them for a subsequent evaluation.

Fuses Substitution One of the most recurrent issue in the power system are outages. In most cases outages incur when a fuse blow and need to be substituted by hand.

Damages Evaluation Some weeks before my arrival a landslide has pulled down some poles and the respective electric lines, we went to the damaged grid branch to evaluate where to rebuild them.

Woodcutter Life An ancient eucalyptus plant has fallen in the river next to Jambon power plant. To avoid the creation of a self-making wood dam, that could have created problems to the plant it was necessary to cut it.

Wi-Fi Substitution Power plants dialogues with the control station, located in Jambon power plant, through a Wi-Fi radio link. The one located in Huallin was out of service because of unknown entities, it was substituted with a more powerful one.

Costumers Services Eilhicha is also responsible to connect new costumers, to repair problems on household connection, to disconnect users who do not pay bills and other services. For all of these activities we went across the whole valley, passing from village to village by car, by truck or, preferably, by motocross.

1.6.2 Adaptation of Hugo for the Case Study

Once collected useful data and evaluated how to proceed with the thesis we tried to implement the developed procedure for Sistema Electrico Chacas. Since **HUGO** is not a completely automatic procedure, the code must be tailored case by case. In this second part of work these steps have been completed:

Input Data Evaluation A detailed evaluation of input data was performed. Some data were stored in GPKG file, some were available in Excel files and other did not appear in any file and were collected by hand.

Output File Choice As a Power System Analysis software we have chosen DigSilent. Importing an existing project in DigSilent is made through a standard interface named DGS that allows importing of complete network models as well as updating existing ones. DGS can be imported by many different databases or files typology. The one we chose is Excel file so we analyzed what exactly a DGS Excel file need to be done and how to create it.

Python Coding Creating the code for DGS Excel file generation was the most difficult step. It is designed to work only for the case study but not only to the actual grid. Any small variation on GIS model of the grid will be easily reproduced in the new model.

Simulations The final step was to test the new grid model in DigSilent. Simulations performed are a power flow of Chacas grid as it is, the future implementation of V ETAPA, the new 60 [kV] interconnection and a simulation of load demand increasing in the next 20 years.

1.7 Description of the following chapters

Thesis work is subdivided in the following chapters:

Chapter 2 We introduce to the context of electric grid, of grid models and the theory of power flow analysis. It is also explained the proposed procedure.

Chapter 3 The core of the procedure is a code written in Python. Code input, operation and output are explained.

Chapter 4 Case study in introduced. Chacas electric grid, costumers peculiarity and the local system operator are described.

Chapter 5 In this chapter we use the previously described procedure in case study. Since the Python code must be tailored case by case its operation is illustrated in detail.

Chapter 6 Once obtained the electric grid model it is used to perform a base case analysis and three other simulations regarding the future grid operation.

Chapter 7 Conclusions of the work are taken.

Chapter 2

Implemented Procedure

In this chapter we are going to make an overview of electric grid environment. We are going to describe the structure of electric grid, its utilities and how is possible to evaluate main parameters. Than we are going to see properties of static GIS model and the ones of a power system analysis software. The last section of this chapter describe the procedure developed in the context of this thesis.

2.1 Electric Grid

An electric grid is an interconnected network for delivering electricity from producers to consumers. It consists of: one or many generating stations that produce electric power, electrical substations for stepping electrical voltage up for transmission or down for distribution, high voltage transmission lines that carry power from distant sources to demand centers and distribution lines that connect each individual customers to the rest of the network.

In the context of mini grid it is not normally present high voltage transmission and power is produced by small power plant that often rely on renewable sources.

2.1.1 Grid Structure

Generic structure of an electric grid is composed by many utilities that can be summarized in the following list.

Generators The power plant produce electric energy to met the load demand, while frequency and voltage across the network are maintained at a constant level. Many different generators units exist, based on fuel usage. Those that use fossil fuels and nuclear fuels are categorized as thermal power plants where the generators are connected to steam or gas turbines to produce electric energy. All of the other different type of resources are categorized as renewable energy resources. These mainly include hydro, wind, solar, biomass and geothermal.

Substation Since the parts of an electric grid are at a different voltage levels is necessary to increase or decrease the electricity transported by the grid. Terminal voltage of a synchronous generator usually not exceed tents of [kV], so in a power plants is always present an initial step-up transformer. To step down the voltage from transmission to distribution grid or to industrial costumers primary substation are used, than voltage is again stepped-down with secondary substations while for reaching low voltage costumers (such as residential or small industries) low voltage transformer are used.

Transmission Transmission lines are used to transport electricity to long distances, the high voltage (HV) is necessary in order to reduce losses. Normally to be considered high, voltage need to be above 69 [kV] [12]. The standard HV are 115, 138, 161, 230, 345, 500 and 765 but some research are being connected on lines levels of 1.110 to 1.500 [kV] [12].

Distribution This is the primary distribution system. Secondary substations step-down electricity until a voltage level between 4 and 34.5 [kV], this part of electric network have normally short distances and is connected to a huge number of costumers. Many industrial loads are connected directly to this voltage level and did not need an ulterior stepping-down.

Load Demand Load demand is normally connected to low voltage (LV). The last step of voltage decreasing reduce voltage to a range between 100 and 240 [V] [13].

2.1.2 Distribution Grid Parameters

To aim of a grid model is to calculate the so called **exact solution** of a line, so voltage and current of each line. To do it, is necessary to introduce the main parameters of a distribution line. An electric line has four parameters affecting its ability to fulfill its function as part of a power system: resistance, inductance, capacitance and conductance. In the last list we will describe these parameters, conductance will be neglected since its effects are negligible in a distribution system.

Resistance The conductor resistance R of a line is the most important cause of power losses in transmission and distribution lines. It is calculated as in equation 2.1, where ρ is conductor resistivity, l is line length and A cross-sectional area:

$$R_0 = \frac{\rho l}{A} \qquad [\Omega] \tag{2.1}$$

Inductance The inductance L of a distribution line is calculated as flux linkages λ per ampere I. If permeability μ is constant, sinusoidal current produces sinusoidally varying flux in phase with the current.

$$L = \frac{\lambda}{I} \qquad [H/m] \tag{2.2}$$

Capacitance Capacitance of a line is defined as the charge of the conductor per unit of potential difference with respect to another conductor. Each line have a capacitance with respect to others conductors line and with the earth. We will only introduce this last parameter.
The earth affects the capacitance C of a distribution line because its presence alters the electric field of the line. In order to evaluate this parameter it can be used a geometric mean distance method (GMD) that can be adopted for single-circuit three phase line. In equation 2.3 k is the permittivity of the material surrounding the cable surface, D_{eq} and D_{sC} are geometric parameters.

$$C_n = \frac{2\pi k}{\ln \frac{D_{eq}}{D_{sC}}} \qquad [F/m] \tag{2.3}$$

Conductance This parameter account for the leakage currents, dissipated by cables and insulators, is not taken in consideration.

2.1.3 Electric Grid Model

Once assumed that the three previously introduced parameters, are uniformly distributed through the length of the line, it is possible to found the exact solution of a line. We will describe here the generic solution, useful for long lines, whether for small lines it could be possible to use a simplifying method

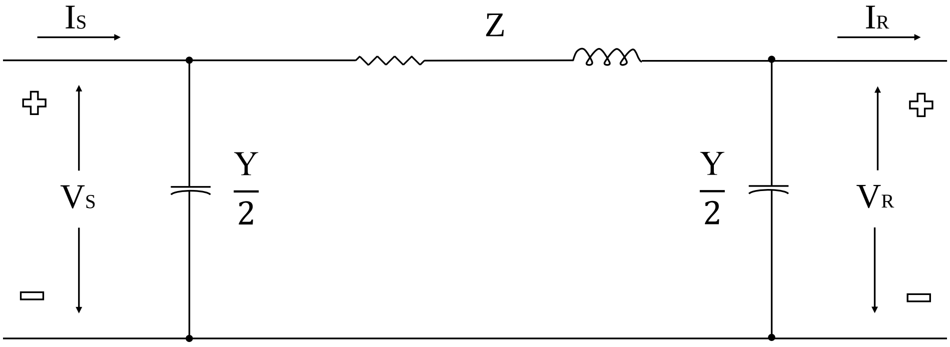


Figure 2.1: Π model of a distribution grid.

called **Lumped Parameter**. The model we are going to propose is the so called **π model** (see figure 2.1).

Solving an electric **π model** means calculate current V_R and voltage I_R at any receiving node, knowing line parameters (R_S, L_S, Y_S) and voltage V_S and current I_S at the sending node. The solution is:

$$V_S = AV_R + BI_R \tag{2.4}$$
$$I_S = CV_R + AI_R \tag{2.5}$$

Where

$$A = \frac{ZY}{2} + 1$$
$$B = Z$$
$$C = Y\left(1 + \frac{ZY}{4}\right)$$

This solution is valid for a single phase line, if more phases are used a similar solution is possible to found. The **π model** became as in figure 2.2, the solution can be obtained in a matrix format.

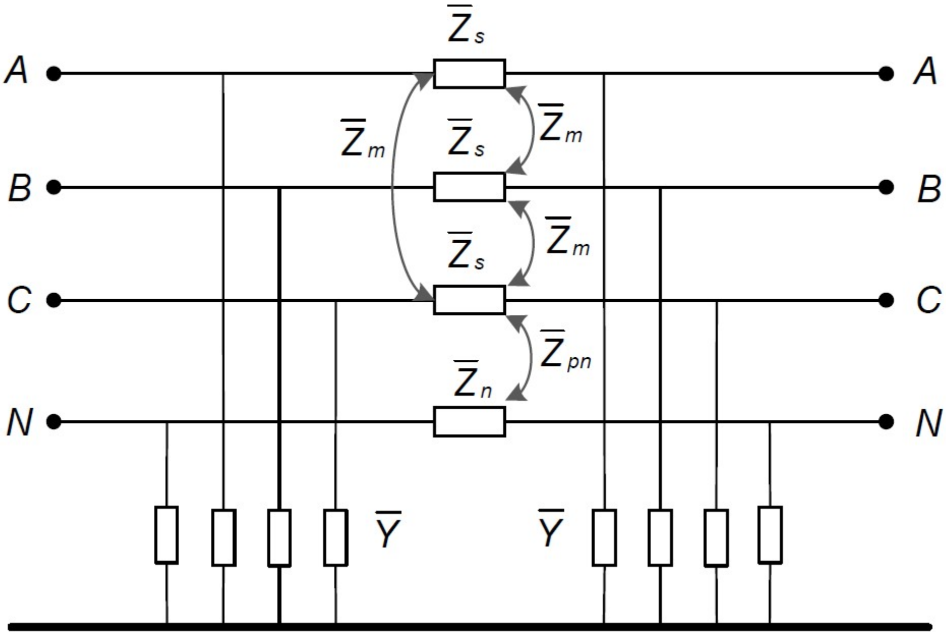


Figure 2.2: Π model of a three phase and neutral distribution grid.

2.1.4 Power Flow Calculation

To solve a whole power system, so to evaluate bus voltages and currents, it is used a bus impedance matrix Z_{bus}. This matrix is calculated inverting the bus admittance matrix. The solution of each line can be obtained with a matrix solution.

$$V = Z_{bus} I \tag{2.6}$$

where

$$Z_{bus} = Y_{bus}^{-1}$$

The main information obtained from a power-flow study is the magnitude and phase angle of the voltage at each bus, the real and reactive power flowing in each line, as well as the system losses.

2.2 Grid Models

As we said, two type of models are used to represent a distribution power system: Geographic Information System (GIS) static models and electric models developed inside a Power System Analysis Software (PSAS). GIS models on their side have the advantage of usage simplicity and software low cost, while a PSAS allow to perform electric analysis and a variety of other tasks.

2.2.1 GIS Model

A GIS model of an electric grid is a static tool that can be useful for many usages. The most evident are the georeference visualization of grid utilities and the creation of a database.

Georeference View Each element is saved as a graphical object that is stored in layers. Geographic coordinates are assigned to elements, they can be visualized or hidden, the system operator have therefore the possibility of an overall view on the whole grid, or of parts of it.

Database Grid elements are stored with many properties as desired in a specific table for each type of elements. These tables are not only useful to the geographic localization of the object but also include commercial or physical data.
For example an hypothetical **Costumers table** can contain information about:

- Longitude and Latitude, of the connection point.
- Name of the household or commercial activity.
- Locality of costumers, or maybe the LV transformer he is connected to.
- Contract code.
- Model and brand of the meter.
- Year of connection.
- The phase(s) to which the costumer is connected to.
- Others, such as if it is disconnected or of troubles can be encountered in approaching the costumer or if he have the possibility to receive national incentives.

Editing Grid elements can be easily edited, added or deleted, it is possible to do it both manually from tables from the graphical view. These changes can be for example substitution of a line or connection of a new household. This tool give the possibility to implement changes on the real grid without loosing information in the GIS model.

Planning Graphical grid representation can be useful to have an overview to the power system. The system operator can evaluate whether, and where, to implement some grid extension, for sake of security of supply. These evaluations are not based on technical evaluations, but also on operators experience.

Despite all advantages of a grid GIS model, first of all the simplicity of usage, it is not possible to perform an accurate analysis of grid performances and a robust planning. From a GIS model it is not possible to perform any electric simulation, load flow analysis, short-circuits analysis or contingency analysis, it is not easy to evaluate grid weakness such as overloaded lines, oversized ones or losses concentration. More grid size increase more it is necessary to better evaluate its electric behaviour, the cost of accurate analysis became more and more justified.

2.2.2 Power System Analysis Software

As we previously introduced, the solution of a grid electric model means the calculation of voltage magnitude and phase angle at each bus of the power network. To do this the main method developed is called power-flow methods, the problem formulation is a set of multiple nonlinear equations that represent the real and reactive power balances at each bus. The two most used procedure are **Gauss-Seidel** and **Newton-Raphson**, both are iterative methods. The main task of a PSAS is to solve these equation and evaluate the state of a grid, in a certain condition, in order to identify the status of a grid and its weakness or strengths.

Figure 2.3: PowerFactory, developed by DigSilent

Many commercial power system analysis software were developed and are daily used all around the world. These PSAS, such as PowerFactory have a huge number of functionality from standard features to highly sophisticated and advanced applications including wind-power, distributed generation, real-time simulation and performance monitoring for system testing and supervision.

In the context of distribution grid, PSAS provides comprehensive modelling features for studying all kinds of electrical networks with different phasing technologies, meshed or radial topologies and railway supply systems connected to public distribution systems. They also can give suggestion to the system operator, in order to reduce network unbalance, improve quality of supply and optimise distribution networks with different functions such as multi-phase load flow analysis, short-circuit analysis harmonic analysis, quasi dynamic simulation, optimal power restoration and reliability assessment. Other standard features include the modelling of distributed generation and virtual power plants, voltage drop analysis, consideration of LV load diversity, daily load and generation profiles and easy-to-use protection coordination functionality.

All of these application can be useful to the system operators for daily grid operation, but especially for planning.

2.3 Proposed Procedure

In the previous sections we explained the need of an electric grid model. But, how is it possible to generate an electric model starting from a GIS one? Obviously it is not simply possible to import GIS model into a PSAS one. Grid electric information are stored in GIS software as trivial data, in addition each PSAS and GIS software has its own characteristics. Any possible solution to perform this work need necessary to be tailored case by case, both GIS and PSAS software must be known, such as tables specific name of GIS data. An help to us is given by the fact that GIS software frequently works with GPKG file extension, this extension can be edited and visualized by any GIS software. The procedure proposed in this thesis is independent by the software used for visualizing GIS grid and from the PSAS one.

The procedure structure is shown in figure 2.4 and more in detail in figure 2.6. As we will extensively explain in the next chapter, and briefly explain in the next list, input are GPKG file and a separate load demand file while the main procedure in developed in a Python ambient and the output is a DGS output file, representing the electric model of the grid ready to be imported in a PSAS software.

GPKG File As we said before, input data for grid structure and electric information are stored in GPKG file. Structure information regards grid geographic coordinates, utilities dimensions (such as line branches length) and information about chain assemblages of line branches.

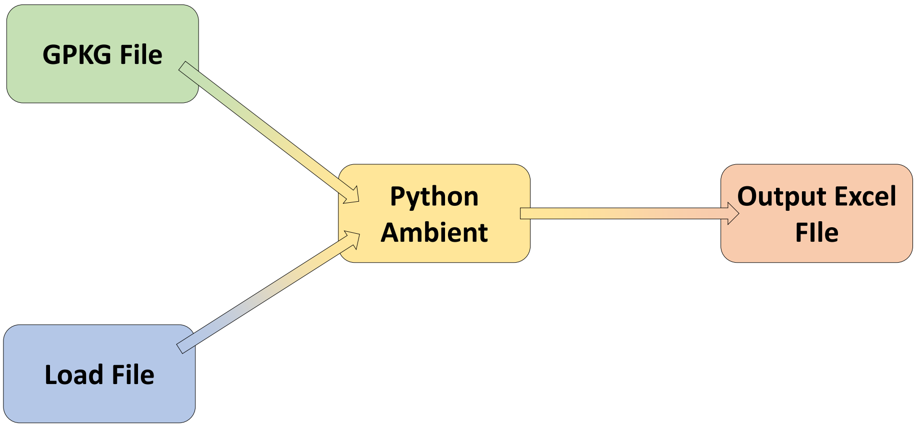


Figure 2.4: HUGO procedure

Load File Load information are expected to be stored in a different file. The necessity of such information split is that normally information about load demand do not exist, or are collected by the system operator in a different file than grid structure. Since the procedure is expected to work with distribution grid, load demand are assigned to MV-LV transformers. Therefore the load of all costumers, supplied by a certain MV-LV transformers, are represented by a unique couple of (P, Q) values, so active and reactive power. An important potentiality of **HUGO** is that it can works also if load demand is not defined. In this case it is possible a load estimation through any of the existing techniques, such as intuitive sizing techniques, based on functioning duration or load factor, or through advanced approaches bases on stochastic methods.

Python Ambient Once all information are imported inside the Python Ambient, they are used to generate the output file. The code developed will be strictly dependent by the case study, but a general description of the necessary steps is described in the next chapter. Since the data we expected to import by **GPKG** file and by load one, regard only information about the distribution grid we are expected to know a priori information about generators, external grids and other utilities. These information must be directly inserted by the code in specific pre-imposted sheets. **HUGO** is a tool that can be ran every time it is requested. Thus, when any alteration implemented in the real grid is included in the **GIS** model it is possible to run **HUGO** code and generate a brand new electric grid model.

Output File PSAS software can use many different file to import a complete project. For example, **DGS** import procedure used by the software **PowerFactory**, can both import Excel file or many different type of databases. The method chosen in this thesis is **DGS** import Excel file, otherwise a similar approach can be used for different methods.

Summarizing, **HUGO** is a semi automatic procedure that generates an electric grid model starting from a **GIS** model of a grid and from load information. Once the procedure is completed it became a useful tool for quickly update the grid model, every time grid alteration are implemented and analyzing its impact to the grid. Having an electric grid model, an therefore robust technical simulations, will also permits the system operator to make more reliable long term choices.

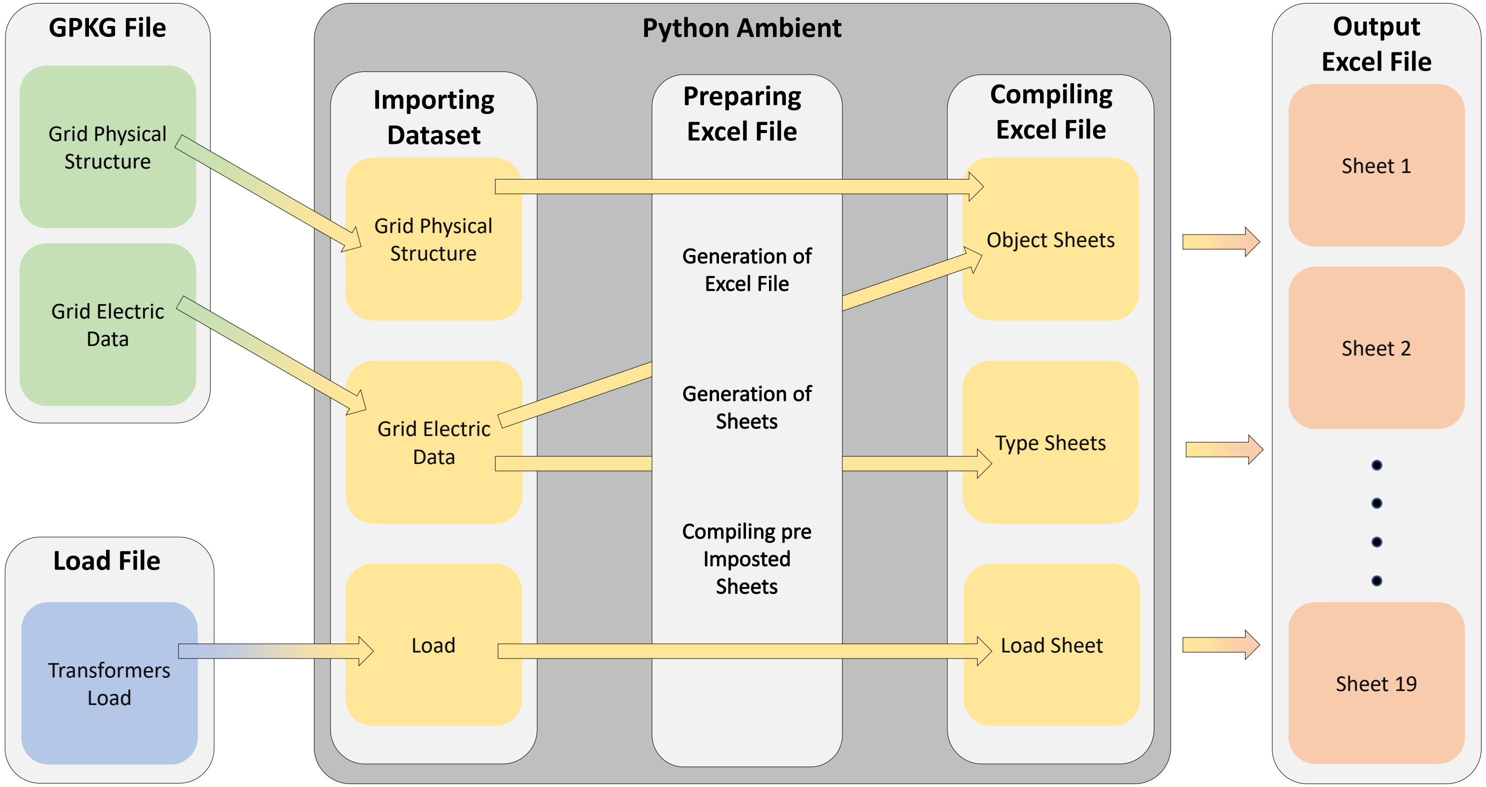


Figure 2.5: Detailed HUGO procedure

Chapter 3

Proposed Procedure

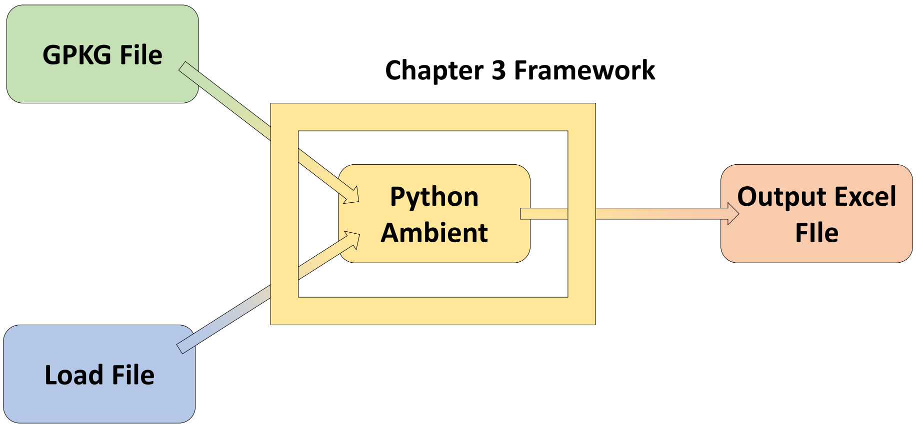


Figure 3.1: HUGO procedure

In this chapter it is explained and described **HUGO** code, this step is the central part, named **Python Ambient**, of the procedure shown in figure 3.1. As we better explained in chapter 2, inside Python environment **HUGO** imports the input data and generates a DGS Excel file. This DGS file once imported in a PSAS generates an electric model of the grid.

The code is written in **PyCharm** for the Python language. **PyCharm** is an integrated development environment used in computer programming, developed by the Czech company JetBrains. In this chapter we will describe how **HUGO** works, starting from how the code will import GPKG layers

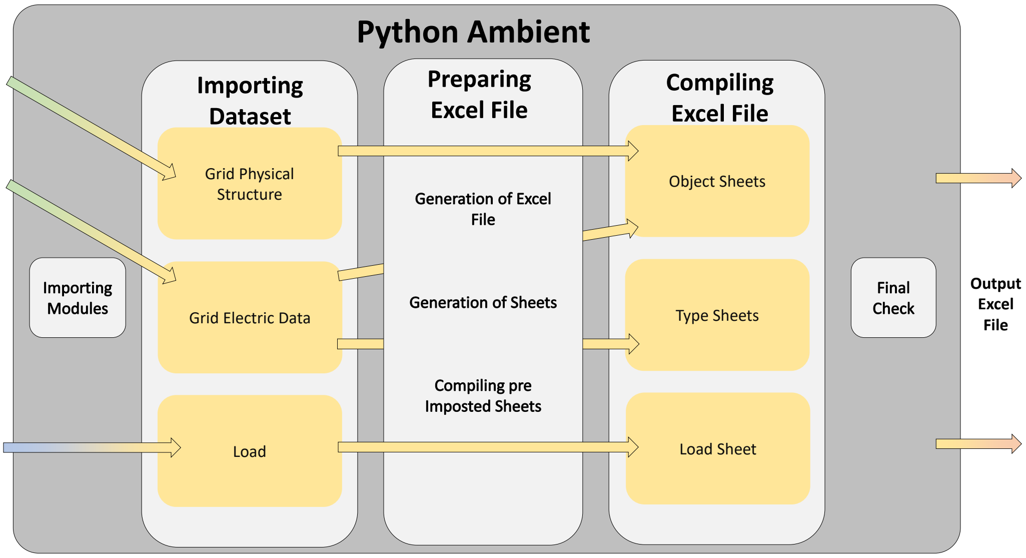


Figure 3.2: Overview of HUGO procedure inside Python ambient

and other input information, until the output Excel file will be created and automatically complied. As we have introduced in chapter 1, **HUGO** must be tailored case by case. As a whole the procedure is applicable to different case study, but the pre-existing differences between data bases necessitate to adapt the code case by case. In chapter 4 we will propose an example of such tailoring of **HUGO** for the implementation of Chacas power system model.

As previously introduced, three different kind of input information are necessary to **HUGO**. The first data regards load demand, these data are expecting to be stored in Excel file. Load demand must be modelled as a P-Q couple, one for MV-LV transformer. Python code will use **openpyxl** and **pandas** library to read and write Excel file. The second input are GPKG file, so layers tables used by GIS to create a grid model. Thanks to **geopandas** library Python is able to read GPKG file and saving each layer's features as vectors that will be subsequently used.

The last and more complex source of data regards PowerFactory TYPEs, generators, external grid interconnection and a table which specifies codices used in GIS layers. All of these data will not be imported during the procedure but must be collected a priori and directly inserted in the code. TYPE data, as previously explained in chapter 2, are useful PowerFactory data set to collect many common data of electric utilities. Since the majority of utilities, such as lines and poles, have only one or few different typologies it is more convenient to save this common data in such object's TYPE. Any

Step	Description	Section
Importing Modules	This step is necessary in order to generate the appropriate Python environment of **Hugo** code.	3.1
Importing Dataset	Structure and Format of input data are explained, tools and procedures used in this process are described.	3.2
Preparing Excel File	DGS Excel file is generated, sheets are created and the default values compiled	3.3
Compiling Excel File	The main core of **Hugo** code, DGS file is compiled according to input file loaded.	3.4
Final Check and Output	A final check on results is performed and output DGS file saved.	3.5

Table 3.1: Steps of HUGO code

new object inserted in the grid will not need to store all these information but only object type. Generators and external grid information must be manually collected and inserted in the code.

To complete the list of information to be manually added we must take in consideration codices used in GIS tables. This codices are used to represent some information such as nominal voltage line or number of phases, they are chosen by the system operator or by the national authority. To be interpreted these codices must be explained during the code, a specific codes table must be compiled and integrated into **HUGO**.

HUGO code is subdivided into and initial module, the three main parts and a final module. This five steps are described in figure 3.2, each of them will be accurately described in the following sections but an overview of them is explained in table 3.1.

3.1 Importing Modules

Python projects always begin with importing modules, a Python **import** is a command to get access to code from another module by importing the file or function. Here we present a list of the main imported modules used by **HUGO**.

import numpy as np It is the fundamental package for scientific computing with Python. It contains among other things the N-dimensional array object, it frequently used by **HUGO**.

import openpyxl It is a Python library used to read from and write to Excel File, in this project it was especially used to create the DGS Excel file, to compile and save it.

import pandas as pd It is a library for data manipulation and analysis. In particular, it was also used to read data from an external Excel file.

import geopandas as gpd It is an extension of **pandas**, it is an open source project to make easier working with geospatial data in python. **HUGO** use GeoPanda for importing GPKG file and extracting from these layers many data arrays.

import math This module provides access to the mathematical functions defined by the C standard. It is used for the transposition of **UTM** coordinates in latitudes and longitudes, [14].

3.2 Importing Dataset

HUGO have as input an Excel file and a set of GPKG files. The Excel file contains P-Q couples of load demand. GIS layers store grid utilities information about electric properties and spatial coordinates and geometries. Import is made by **pandas** and **geopandas** packages.

3.2.1 Load Demand

As we previously introduced load demand must be stored in an Excel file. The file will have a row for each transformers divided in three columns. One column will contain transformers local names and the others P-Q couples, so active and reactive power to be assigned to the corresponding load. Excel file and GIS layer transformers must be the same, also local name of each element must be identical. During the following steps **HUGO** will connect transformers and loads, it is possible only if the names are the identical, otherwise the procedure will be interrupted.

The choice to not include load demand information in GIS layers was made mainly by two considerations. The first one assumes that normally this data are not available or difficult to be collected by the system operator. In addition when load information are present they are collected as active power only. The second consideration is that load demand change hour after hour. In our model we didn't evaluate dynamic or quasi-dynamic simulations

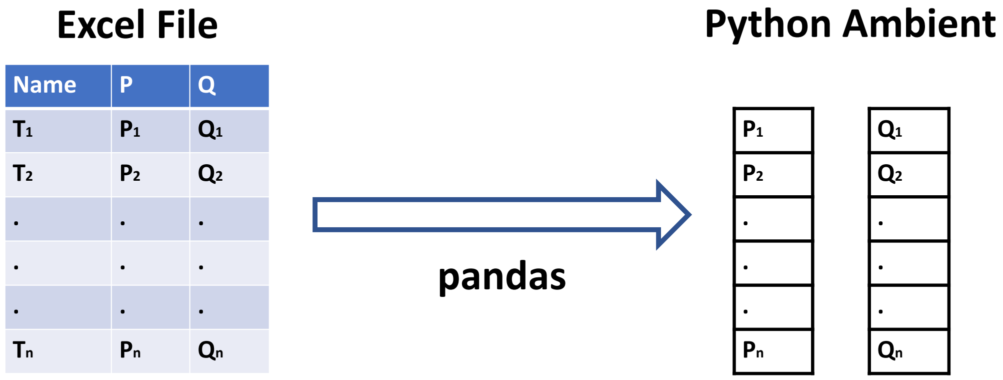


Figure 3.3: Load Import, P-Q values are imported by pandas package.

but only steady state power flow analysis. For this reason P-Q couples must be correctly chosen. Normally P-Q values considered are representative of the moment of the year with the maximum load demand, so when distribution lines are the most overloaded of the year.

Summarizing, this part of **HUGO** will generate two Nx1 vectors, where N is the number of loads, the simplified procedure is shown in figure 3.3.

3.2.2 GeoPackage import

Import of geospatial information is performed by **geopandas** package. As we previously said all information about the distribution power system are supposed to be stored in GPKG file. These data regards utilities geometry, geographic coordinates and electric properties. In the GIS software these data can be visualized in a specific **Attribute Table**. Normally a GIS grid model contains more information then the one we need to import so an accurate analysis of which attributes need to be imported have to be done. Names of attributes columns must be known a priori and inserted in the code.

Layers imported are related to the distribution grid, therefore medium voltage **LINES**, **PYLONS** and **TRANSFORMERS**. As we have written before generators and external grids will be manually added in another step. The following list describe all layers imported by **HUGO** and relatives attributes.

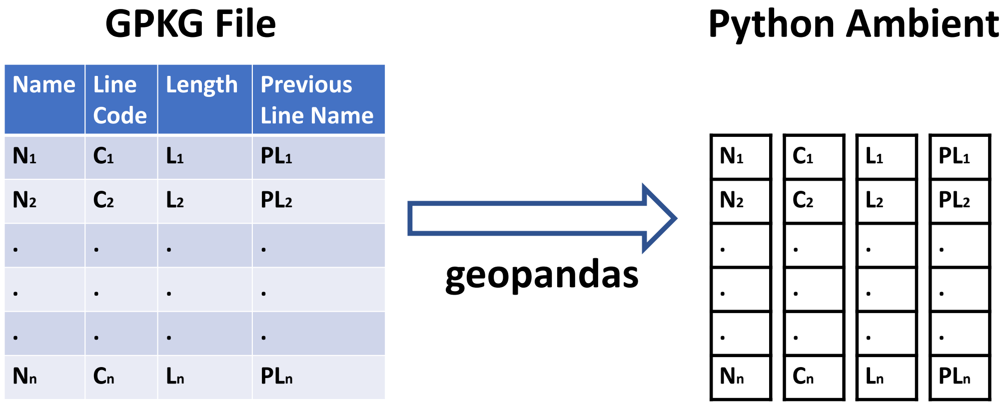


Figure 3.4: Example of a GPKG layer import, LINE.

LINES From this layer we receive information about MV lines. All of them are expected to have three phases and the neutral. It is not important to know a priori the number of lines expected to be imported, the code will calculate it. The list of fields we need is:

- **Name** Normally a line name is an alphanumeric code chosen by the system operator.
- **Line Code(s)** This attribute is a code, chose by the system operator or by a national supervisor organism, that will store information about line material, diameter, number of phases and the present of a neutral. These information can also be stored in different attributes columns so an accurate evaluation, and relative code alteration, must be performed.
- **Length** Line length. **HUGO** will expect to receive this data in kilometers, if not this data need to be changed.
- **Previous Line Name** It is the local name of the previous line. The grid is similar to a chain, each line have only the information about the previous one. This attribute will be fundamental during the next steps.

PYLONS This layer contain information about grid poles or pylon. Each pole represents a junction point of lines and physically support the three phases and the neutral. Only four attributes must be imported for pylons:

- **Name** Local name of a pole.
- **Code of Entering Line** Such as for lines if is fundamental to known which to which line the pylon is connected. It is not necessary that the logic direction of line-pole-line follows the power flow direction.
- **Longitude, Latitude** These fields represent geographic coordinates of poles, both are assigned to any system of coordinates. If they are expresses in another coordinate system they are imported as they are and later translated to Latitude and Longitude.

TRANSFORMERS This layer contains information about both MV-LV transformers properties and phases used by load. All transformers are represented as three-phases transformers. This is a strong assumption, since some of of them are mono-phase. These transformers are thus modelled as three-phases, while the load connected to them is modelled as absorbing only one phase. The need of such assumption is that it was considered the best, and easiest way, to model these load and at the same time correctly evaluate phase unbalances. Fields imported are:

- **Name** Local name of the transformer.
- **Nominal Power** Nominal power of the transformer. The power must be expressed in [MW].
- **Load Phases** A code storing information about which phases are used by costumers. All transformers are modelled to work with the three phases, but not all of them will be used. This code is a string composed by letters, normally a combination of **R**, **S** and **T**.
- **Previous Line or Pylon** The local name of the lines or pole over which (or close to) the transformer is installed. This data is fundamental to connect transformers and loads to the grid chain.
- **Status** This field give the information whether the load is active or not. It is a Boolean, where 1 means it is active and 0 that it is not yet connected or installed. This data will be imported and used in **PowerFactory** model.

- **Longitude, Latitude** Geographic coordinates of transformers, this field can not be necessary if the previous element indicated is a pylon, in that case its coordinates are used.

3.2.3 Coordinates variation

As we previously anticipated many elements have as a property a couple of geographic coordinates. These can be stored in many different coordinates system while **HUGO** is expected to work with Latitude and Longitude. An appropriate algorithm is thus created to modify these coordinates into Latitude and Longitude.

3.2.4 Checks

The last step, before starting to generate and compile the Excel file, is a series of useful check. In the first one we match the two lists transformers; The one imported from GIS layers and the other from Excel file. The aim of this match is to avoid problems during the creation of the grid model and loss of data. The two vectors must have the same length and exactly the same local name.

In the second group of checks local names of all objects are controlled. The aim of this operation is to avoid duplicates and missing data.

3.3 Preparing Excel file

This section is the third step described in table 3.1, so the central part of **Python Ambient** structure depicted in figure 3.2. In this part of **HUGO** code we will start to use **openpyxl** library. This library is necessary to create and modify Excel files. As we sad in chapter 2 PowerFactory DGS Excel file need to have different sheets with default names.

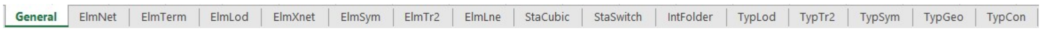


Figure 3.5: DGS Sheets to be created and compiled.

Each sheet, excluding the first one, is named according to a specific table of element, type or other instruction and contains just the data of that table. The first row of each sheet must contain the column definitions while the first column is always a **Foreign Identification** (FID) column, so the unique key name of any object. In all the sheets we will thus compile the first row. For some sheets, such as the **General** and the **Typ** sheets we will also insert the information we already have about these tables. Later we will proceed to compile also element sheets but up to now they will remain almost empty.

The first row is always the column definition. Each sheet need to store different type of information, so will have different column definition. Thus we will specify case by case how many columns are necessary to be created. Names of sheets and column definition must be written exactly as they are, otherwise during DGS import the information will not be recognised. The model developed use 19 sheets:

General This is the first sheet, it is mandatory, is composed by only three columns and contain the information about the DGS version we are willing to use. In addition other rows can be inserted containing instructions such as **PreCommand** or **PostCommand**.

- **FID** Foreign unique identifier
- **Descr** Setting name
- **Val** Setting value

ElmNet The sheet with information about the grid, the only information we want to specify are grid name and nominal frequency.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **frnom** Nominal frequency [Hz]

ElmTerm A sheet containing terminals of the grid, so all busbar, internal node and junction node. These element will not only coincide with the pylons imported by GIS layers but also some bus bar created in the next step. Eight columns will be generated:

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **iUsage** Node type
- **uknom** Nominal voltage [kV]
- **GPSlat** Latitude
- **GPSlon** Longitude
- **phtech** Phase Technology

ElmLod This sheet contains load objects, in our model load will coincide with the relative transformer.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **typ_id** Object Type FID
- **i_sym** Balancing (0=balanced, 1=umbalanced)
- **plinir** Phase R Active Power [MW]
- **qlinir** Phase R Reactive Power [Mvar]
- **plinis** Phase S Active Power [MW]
- **qlinis** Phase S Reactive Power [Mvar]
- **plinit** Phase T Active Power [MW]
- **qlinit** Phase T Reactive Power [Mvar]
- **outserv** Out of service
- **GPSlat** Latitude
- **GPSlon** Longitude

ElmXnet External network, if present, will be stored in this sheet.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **pgini** Active power operating point
- **GPSlat** Latitude
- **GPSlon** Longitude
- **iintgnd** Neutral conductor connection

ElmSym This sheet contains information about the synchronous machines installed in the power system.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **typ_id** Object Type FID

- **pgini** Active power operating point [MW]
- **av_mode** Local controller mode
- **cosgini** Power factor
- **GPSlat** Latitude
- **GPSlon** Longitude
- **mode_inp** Input mode
- **usetp** Operating voltage [p.u.]

ElmTr2 Transformers properties will be stored in this sheet. In this model of distribution grid the only transformers are those between MV and LV, external network and internal transformers of power plants.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **typ_id** Object Type FID
- **outserv** Out of service
- **GPSlat** Latitude
- **GPSlon** Longitude

ElmLne This sheet contains information about MV lines, this sheet will be explained later as an example, see figure 3.6.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **typ_id** Object Type FID
- **pCondCir** Line conductor FID
- **pCondGnd** Ground conductor FID
- **dline** Line length [km]
- **outserv** Out of service

StaCubic To connect objects such lines or load to terminals PowerFactory use a class of object called Cubicles. We will create these objects later on the code. In this sheet we will store all information about cubicles.

- **FID** Foreign unique identifier

- **loc_name** Object local name
- **fold_id** Folder name in which the object is stored
- **obj_bus** Input or output cubicle
- **obj_id** FID of the object connected with the cubicles

StaSwitch Inside a cubicle there is always a Switcher. Each switcher of the previous cubicles sheet are stored in these sheet.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder Name in which the object is stored
- **aUsage** Switch type
- **on_off** State, open (0) or closed (1)

IntFolder This sheet will generate a list of folders. It is not a necessary sheet but it will be useful to contain, inside DigSilent software, all object types that will be inserted in the next sheets.

- **FID** Foreign unique identifier
- **loc_name** Object local name

TypLod In the actual model created we will use only one type of Load, so a three phase with the neutral. Despite being connected the three phases not all load use all of them. When we will fill **ElmLod** sheet only the phases used will be assigned to the load, the others will remain empty.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder Name in which the object is stored
- **kpu** P voltage dependence exponent
- **kqu** Q voltage dependence exponent
- **systp** System type, AC (0) or DC (1)
- **phtech** Phase Technology

TypTr2 This sheet contains information about transformers types.

- **FID** Foreign unique identifier
- **loc_name** Object local name

- **fold_id** Folder Name in which the object is stored
- **strn** Rated power in [MW]
- **frnom** Nominal frequency [Hz]
- **utrn_h** Rated voltage, high voltage side [kV]
- **utrn_l** Rated voltage, low voltage side [kV]
- **uktr** Short circuit voltage uk
- **nt2ph** Technology

TypSym This sheet contains information about type of synchronous machines.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder Name in which the object is stored
- **sgn** Nominal apparent power [MVA]
- **ugn** Nominal voltage [kV]
- **cosn** Power factor
- **nslty** Connection technology

TypGeo Pylon geometry are saved in this sheet.

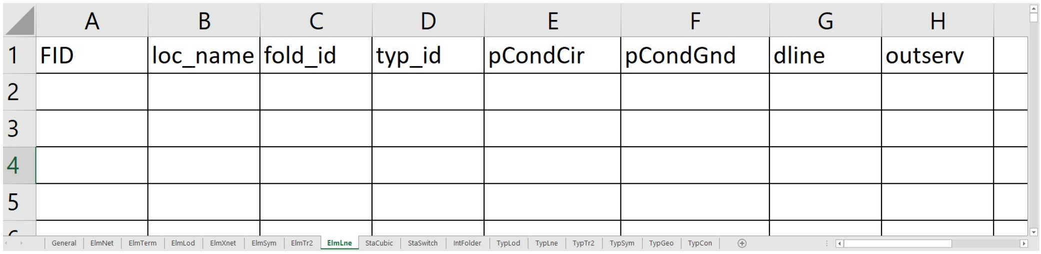
- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder Name in which the object is stored
- **nlear** Number of earth wires
- **nlcir** Number of line circuits
- **xy_c:0:0** Number of phases
- **xy_c:0:1** X coordinate of wire [m]
- **xy_c:0:2** X coordinate of wire [m]
- **xy_c:0:3** X coordinate of wire [m]
- **xy_c:0:4** Y coordinate of wire [m]
- **xy_c:0:5** Y coordinate of wire [m]
- **xy_c:0:6** Y coordinate of wire [m]
- **xy_e:0:1** X coordinate of ground [m]

- **xy_c:0:2** Y coordinate of ground [m]

TypCon Independently by geometry each pylon may support different cables. Information about conductor material, diameter nominal current etch are stored in this sheet.

- **FID** Foreign unique identifier
- **loc_name** Object local name
- **fold_id** Folder Name in which the object is stored
- **uline** Nominal voltage [kV]
- **sline** Nominal current [kA]
- **ncsub** Number of sub-conductor
- **iModel** Conductor model, solid (0) or tubolar (1)
- **rpha** DC resistance (20C) [Ohm/km]
- **rpha_tmax** DC resistance (80C) [Ohm/km]
- **erpha** GMR equivalent radius [mm]
- **diaco** Outher diameter [mm]
- **mlei** Conductor material

As an example, looking at figure 3.6, we can observe **ElmLne** sheet that will contain all MV lines. The first row contains the column definitions, which are FID and Loc_Name (for lines these properties coincides), pylon geometry (**typ_id**), phases and neutral conductors (**pCondCir** and **pCondGnd**), length in [km] (**dline**) and if it is out of service (**outserv**). The value inside each **Typ** definition (**typ_id**, **pCondCir** and **pCondGnd**) indicate the FID of the corresponding geometry and conductor type.



	A	B	C	D	E	F	G	H
1	FID	loc_name	fold_id	typ_id	pCondCir	pCondGnd	dline	outserv
2								
3								
4								
5								

Figure 3.6: Example of a DGS import sheet. ElmLne.

3.4 Compiling Excel file

This is the core part of **HUGO** code, it is the fourth code step of table 3.1, so the last section of **Python Ambient** before code conclusion, as depicted in figure 3.2. We will compile object tables starting from information we have extrapolated by GIS layers. More in detail we will compile sheets **ElmTerm**, **ElmLne**, **ElmLod**, **ElmTr2**, **ElmSym**, **ElmXnet**, **StaCubic** and **StaSwitch**.

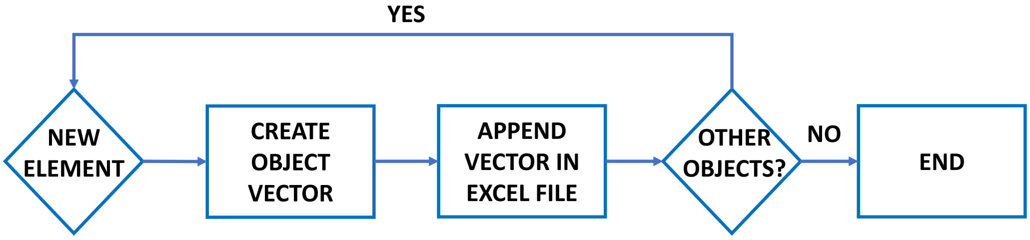


Figure 3.7: Procedure implemented for compiling the Excel file.

For each element we will add a new row, in the relative sheet, containing all information of that element we previously decided to import. The procedure will initially create a vector in Python ambient, using **openpyxl** package. This vector is then appended in the Excel file. Each information is taken from a different element list previously created during import phase.

3.4.1 ElmLne Example

As an example we will see in details only the procedure implemented for **ElmLne** sheet but the same logic can be adopted for each sheet.

FID, loc_name They are taken directly by the vectors created by GPKG lines layer. Both use line local name. As we can see in figure 3.8 it is not necessary to use any external codes table.

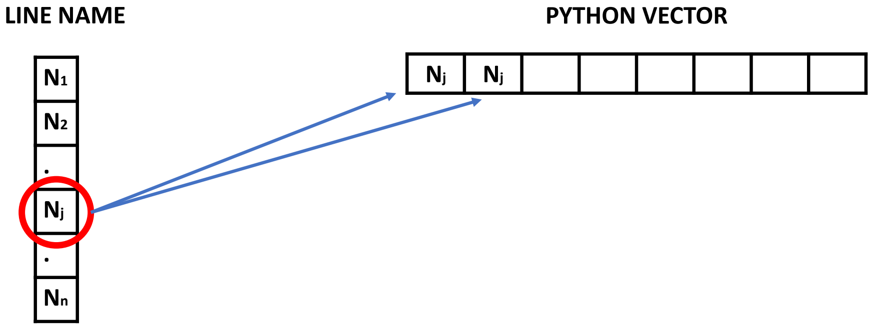


Figure 3.8: FID and loc$_{n}ameidentification$.

fold_id Only one grid is created in the model. It's **FID** name was saved in Python and copied at each loop so all elements of this column will have the same value. Figure 3.9.

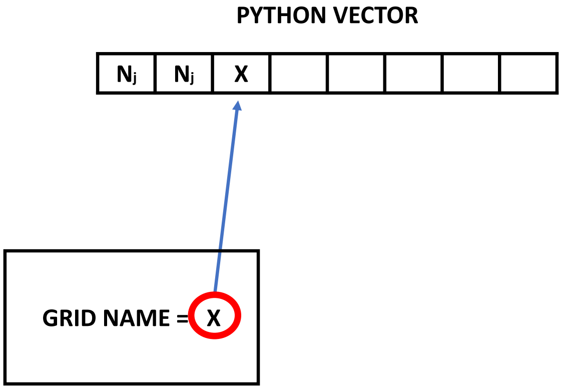


Figure 3.9: Grid name is equal to all objects.

typ_id, pCondCir, pCondGnd These three **FID** are necessary to identify geometry type and conductor type of phases and the neutral. All of these information can be obtained by a single code, such as in the example in figure 3.10, or by different ones. It is thus necessary to teach the code to recognize these codices and provide the correct type FID.

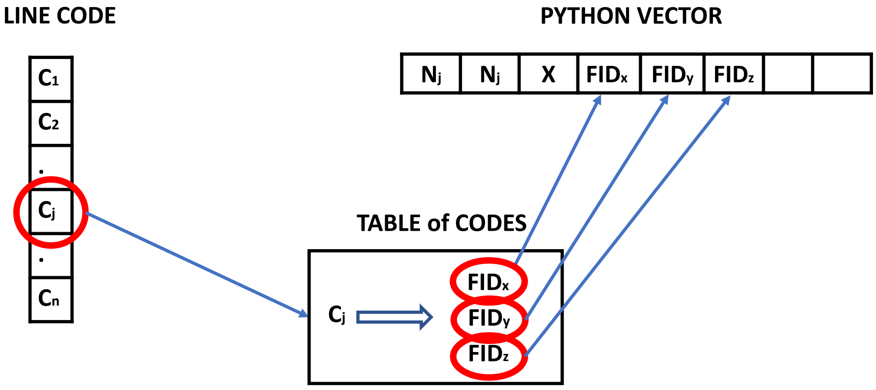


Figure 3.10: Identification of Line type through type tables.

length As for **FID** and **loc_name** lines length are directly taken by the vector created by GPKG layer. Figure 3.11.

outserv All lines are initially considered active, so this column will also have zeros. Figure 3.12.

Once the vector is created it is appended in the DGS Excel file. This operation is made by the command **.append** contained in **openpyxl** package. Figure 3.13.

3.4.2 Compiling Steps

These passages are included in a for loop so the procedure is repeated for each element. When a loop ends a new one starts to compile a new sheet since the procedure is not completed.

This section is divided in five parts. First of all starting from pylons vectors we appends terminals, than lines from lines vector, transformers and

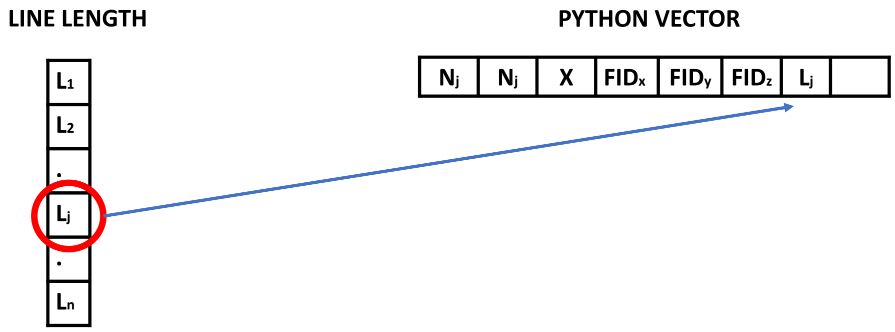


Figure 3.11: Line Length.

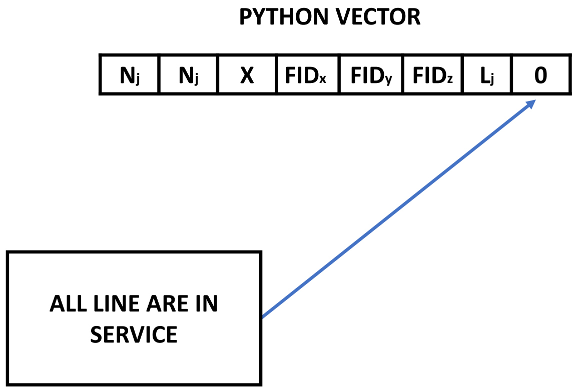
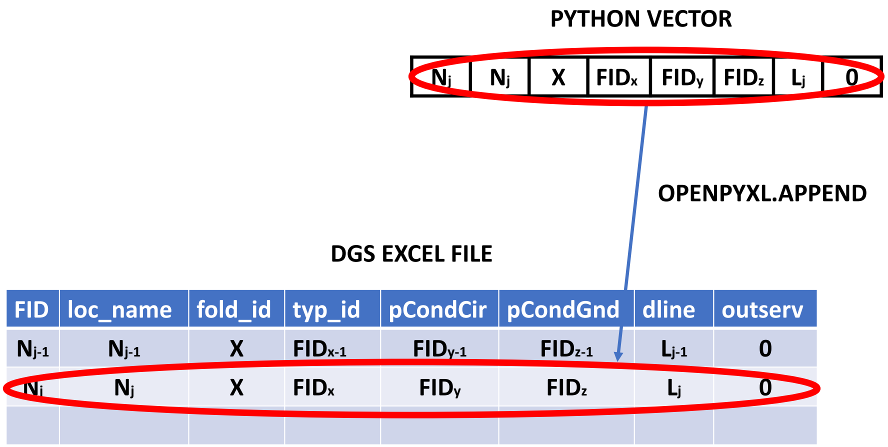


Figure 3.12: In service (0) or out of service (1).

load from transformers vector and load imported from Excel file. Generators and external networks are manually added. Every time an element is inserted on his specific sheet it is done with a new row, to do this we use the command **.append** contained in openpyxl library. This command add a new row with all properties we have identified.



N_j	N_j	X	FID_x	FID_y	FID_z	L_j	0

FID	loc_name	fold_id	typ_id	pCondCir	pCondGnd	dline	outserv
N_{j-1}	N_{j-1}	X	FID_{x-1}	FID_{y-1}	FID_{z-1}	L_{j-1}	0
N_j	N_j	X	FID_x	FID_y	FID_z	L_j	0

Figure 3.13: Appending the line vector just created in the apposite ElmLne sheet.

StaCubic and **StaSwitch** sheets is compiled during the whole section. Each element in **PowerFactory** is always connected between one or two terminals and it is always connected with these terminals by a **Cubicle**. In each **Cubicle** there is always a **Switcher**.

PYLONS Starting from the three imported vectors **Name**, **Previous Line Code**, **Longitude** and **Latitude** we can compile **ElmTerm** sheet. This is an easy step because terminals do not need a type, we only distinguish between terminals used as **Busbar**, **Internal Node** or **Junction Node**. Cubicles of lines entering or exiting from these terminals are created here. Not all of the terminals are created in this section, others will be added later with load, generators and the external grid.

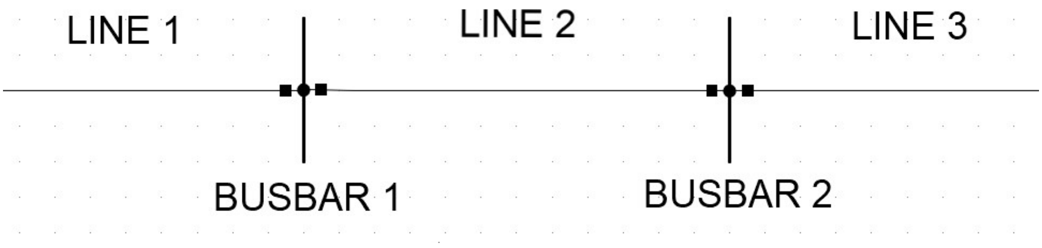


Figure 3.14: Example of grid branch once imported in PowerFactory.

MTTRAMO The proceeding used to compile this sheet are described in the previous section. Lines and poles is a grid are similar to a chain, they are logically connected thanks to the vector **Previous Line** by a **Cubicle**. The resulting grid is show in figure 3.14, both **BUSBAR1** and **LINE2** have as **Previous Line** property **LINE1**, while both **BUSBAR2** and **LINE3** have as **Previous Line** property **LINE2** etc. . . .

Transformers, Load In this part of the code we iterate a for loop for each of the MV-LV transformers. In each loop a new load and a new transformer is appended. Transformer type are identified through a control on its nominal power while this is not necessary for load since only one **TypLod** exist.

Load, vice versa, have to be processed in a different way. In the first part of **HUGO** we have assigned a couple of (P-Q) values to each transformer. This couple represent active and reactive power of the area dispatched by the transformer, independently by how many and which phases are consumed by the load. Now, from the transformer vector **FasiOut** the code evaluate which phases are consumed, so both active and reactive power will be split between the phases used. So, if an hypothetical 95 [kW] and 5[kVar] load absorb only phases **r** ans **s** it's active and reactive power, of both phases, are calculated as 47.5 [kW] and 0.25 [kVar]. No power will be assigned to phase **t**.

In each loop the code append in the relatives sheets a new transformer, a busbar, a load, three cubicles and three switchers. Once imported in PowerFactory the model of each Load should look like in figure 3.15.

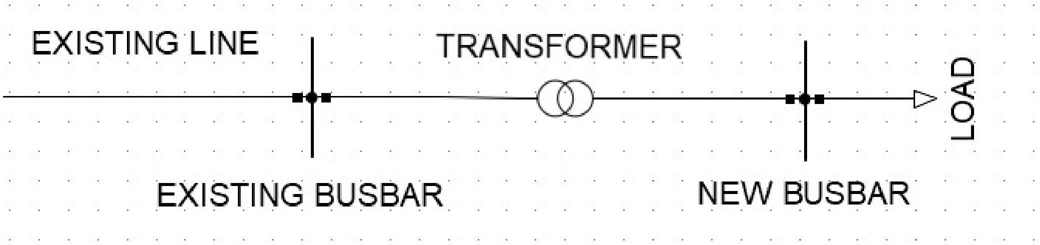


Figure 3.15: Example of Transformer and Load once imported in PowerFactory.

Generators Data regarding each generator present in the power system are not stored in QGIS layers. These information must be already known and manually inserted in the code. This part of the code is not an automatic procedure, if a new power plant is installed in the system the code must be modified. As it is possible to see in figure 3.16 the procedure is similar to the one used for **Load**, the code will add to the relative sheet a new transformer, a busbar, a generator, three cubicles and three switchers.

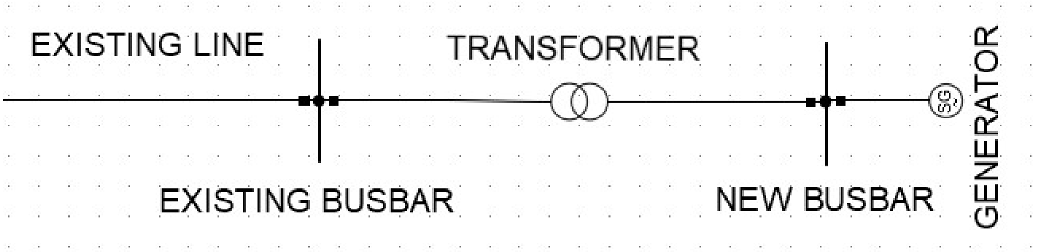


Figure 3.16: Example of Generator once imported in PowerFactory.

External Network Also information about the existing external network must be known a priori. From a physical point of view the interconnection with the national grid can have different configuration, depending whether the voltage level on the two side are equal or different. If voltage level is the same on both grids, such as in figure 3.17 a transformer is not necessary and only one object in ElmXnet sheet is appended. A Cubicle with the relative Switcher is also appended.

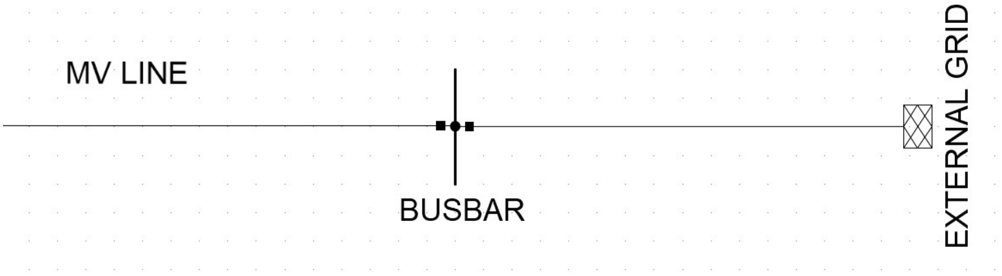


Figure 3.17: The external network once imported in PowerFactory, a transformer is not necessary since grid voltage is the same on both side.

3.5 Final Check and Output

During all the previous code lines **HUGO** have counted many different type of errors that may have incurred such as duplicates or missing data. In the last section of the code **HUGO** print and explain these errors to the console. Also a useful list of how many grid elements of each sheet is imported in **PowerFactory** is printed to user.

The code conclude saving the Excel file, renaming it, creating some copies and moving them to different folders.

Chapter 4

Introduction to The Case Study

In this chapter we are going to presents the Peruvian energy situation, poverty rate and electrification programs. Than it will be described Ancash region and the valley of Chacas, where the case study is carried out. After this Overview the power system **Sistema Eléctrico Chacas** will be described in details. We are going to analyse not only electric properties and costumers of the grid but also its history, stakeholders and the expected future developments. Grid is being expanding, load demand is growing and different interventions are going to be implemented or are in a planning stage. Since these future interventions will have a greater impact on the grid and will make more complex the grid operation, a power system analysis model of the grid is becoming increasingly necessary. In this context we will use **HUGO** procedure and a **PowerFactory** grid model will be created.

A special role will be given to the future development of both grid systems and to local load, once the grid model will be completed these considerations will be used for the simulation of future grid asset. These results will be explained in the following chapters.

Figure 4.1: A view of Belaùnde Lake, on the top of Chacas valley at 4500 meters.

4.1 Peru Overview

Peru is a developing country with a population of 32 million people [3]. Four people out of five lives in urban areas while 7.07 million people lives in rural areas [3] (see figure 4.2). During the last 60 years total population of Peru has increased more than three times wile rural population increased only by 30%. The Peruvian population has migrated, and is still migrating from rural areas to huge urban agglomerations. This migration is driven by a desire of better living condition but these expectations are in many cases disregarded. In 2014, the last data available about the rate of urban population living in slums indicated that more than ten millions people are living in these extremely poor areas [3].

Despite living in urban areas have some undeniable advantages, rural living standards are increasing year after years. As it is possible to see in figure 4.3 that rural access to electricity has almost reach 90% and rural poverty gap at national poverty line in 2014 has fallen below 14%. Rural poverty gap at national poverty lines is the rural population's mean shortfall from the poverty lines, so smaller the indicator closer is a rural population to be considered non-poor.

Gross National Income and Gross Domestic Product are increasing constantly during the last thirty years, both on a national and per capita level. The average of **Gini** coefficient during the last ten years (2007-2017) was around 45%. See figure 4.5, all of the three indicators are decreasing and represent poverty gap at 5.50$, 3.20$ and 1.90$ a day (US$ 2010 PPP).

4.1.1 Energy Supply

Total Primary Energy Supply of Peru in 2017 was 0.76 [toe/capita], half of world average value. National whole **TPES** in the same year was almost 24.6 [Mtoe]. Peruvian energy supply mainly come from Oil (44.2%) and Natural Gas (28.5%). An important contribution, far and away from the average of other countries is represented by Biofuel and waste (13.4%) and Hydro (10.2%). Coal (3.0%) is incredibly low for a developing country, no Nuclear power plants are present in the country (see figure 4.6).

4.2 Power Sector

Generation dispatch and network operations, of the Peruvian national **Power Sector**, are carried out by the independent system operator called *Comité de Operación Económica del Sistema Interconectado Nacional* (COES) and

Peru Overview (2018)	
Total Population	31.989.256
Urban Population	24.921.870
Urban Population Average Growth (1967-2017) %	2,58
Rural Population	7.067.386
Rural Population Average Growth (1967-2017) %	0,43
Total access to electricity %	96,36
Urban access to electricity %	100
Rural access to electricity %	83,68
Gini coefficient (2007-2017) %	44,65
Poverty gap at $5.50 a day (2011 PPP) (%) 2017	8,9
Poverty gap at $3.20 a day (2011 PPP) (%) 2017	3,2
Poverty gap at $1.90 a day (2011 PPP) (%) 2017	0,9

Table 4.1: Overview of Peruvian economic and social indicators

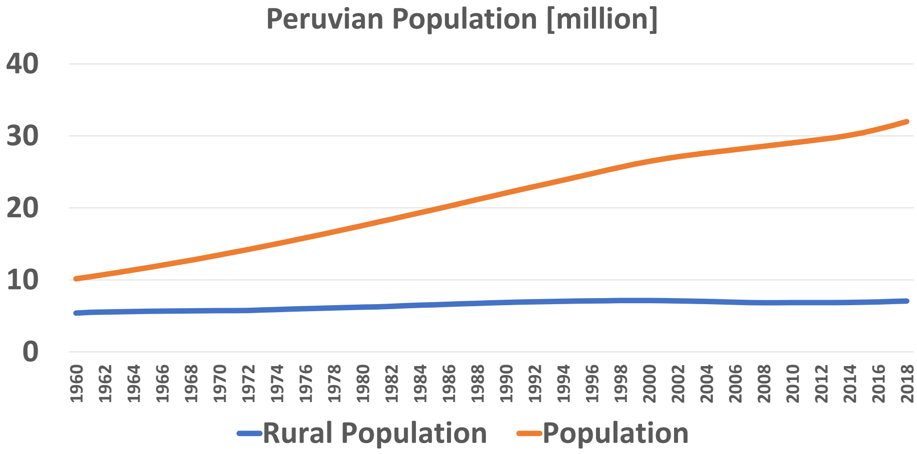


Figure 4.2: Peruvian population, total (red) and rural (blue), million [3].

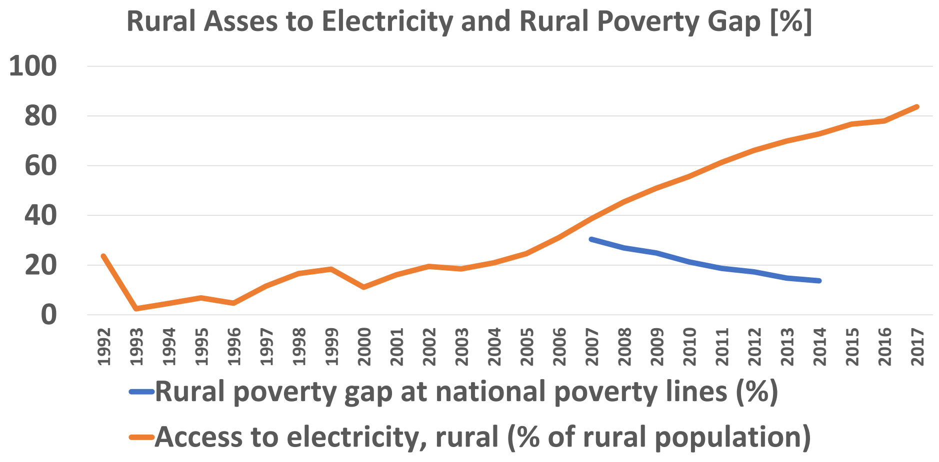


Figure 4.3: Access to electricity, rural (% of rural population) and Rural poverty gap at national poverty lines (%), Peru, World Bank [3].

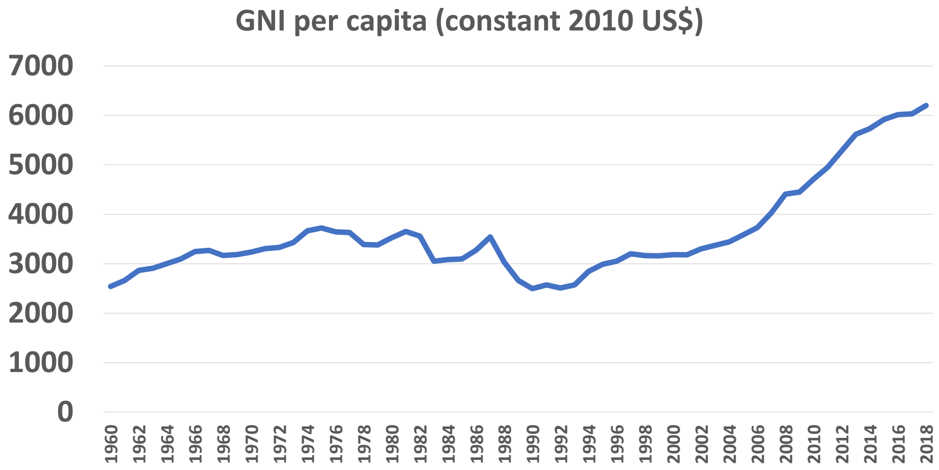


Figure 4.4: Peruvian Gross National Income from 1960 to 2018 [3].

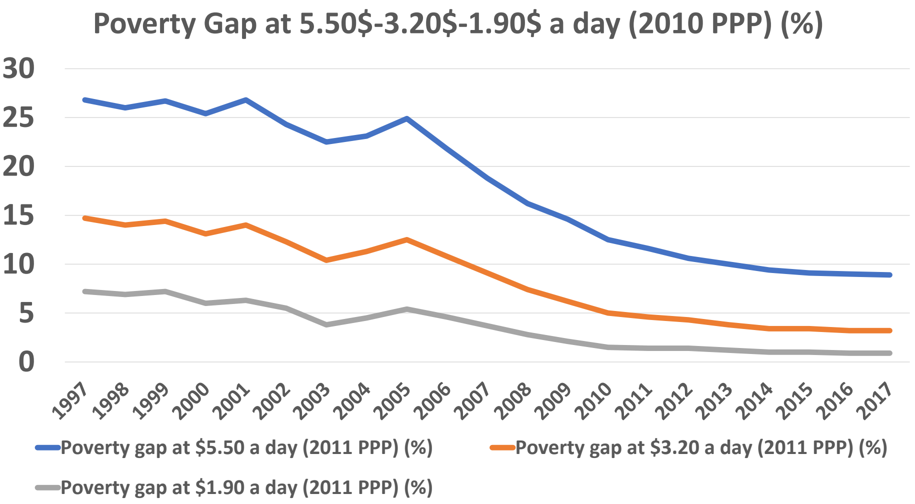


Figure 4.5: Poverty Gap at 5.50-3.20-1.90$ a day (2010 PPP) (%), Peru. World Bank [3].

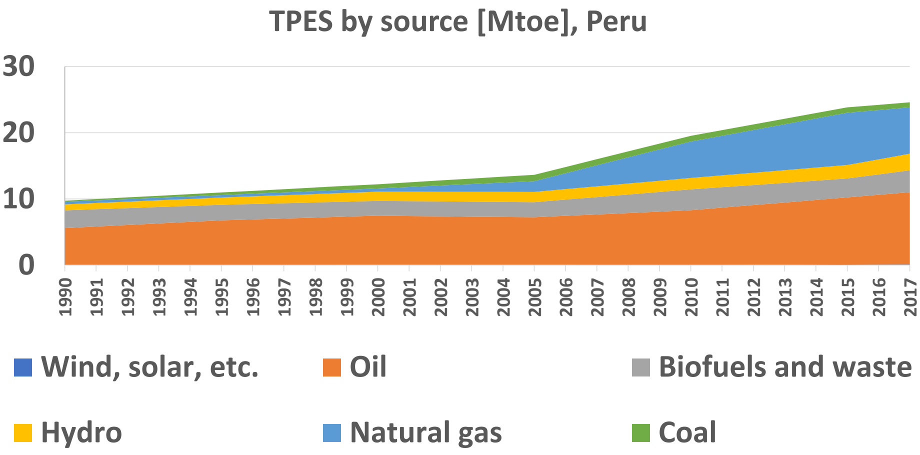


Figure 4.6: Total Primary Energy Supply (TPES) by source, Peru. International Energy Agency [8].

Energy Supply (2017)	
Total Primary Energy Supply [Mtoe]	24,6
TPES per capita [toe/capital]	0,76
Oil, % of TPES	44,24
Natural Gas, % of TPES	28,55
Biofuels and waste, % of TPES	13,40
Hydro, % of TPES	10,16
Coal, % of TPES	3,05
Wind,solar, ectc, % of TPES	0,61

Table 4.2: Energy Supply by Source, 2017

the national electric power system is called *Sistema Eléctrico Interconectado Nacional.*

Peruvian Power Sector is a relatively small market, with generation of about 53 [TWh] in 2017, nodal pricing and central dispatch are in a mandatory cost-based pool. **Power Purchase Agreements** are privately negotiated in the liberalized market and auctioned in the regulated market. Base load units in Peru are combined cycle units fueled by natural gas. Peaking units are combustion turbines fueled by diesel oil and hence significantly more expensive. Hydroelectric generation meets the remaining load, all hydro plants are dispatched if available [11].

The center and the southern parts of the country have traditionally weak transmission links, creating congestion conditions. The southern part of the country has insufficient local generation to cope with large mining demands. Although these congestion have been alleviated by transmission expansions in 220 kV and 500 kV, the center-southern transmission link presented congestion about 60% of the year during 2017.

4.2.1 Power Sources

Historically, Peru has relied mostly on hydropower (see table 4.3). In 1986 Shell gas company discovered a huge gas field in the **Camisea** province, 250 kilometers north than Cuzco, which entered into production in 2004. After the connection to this field thermoelectric generation started to be an increasing part of power supply. Between 2004 and 2008, new natural gas-fired electric power capacity, primarily less investment-intensive but also

Peruvian Energy Mix	GWh
Coal	650
Oil	1.117
Natural gas	19.699
Biofuels	1.032
Nuclear	0
Hydro	29.060
Solar PV	287
Wind	1.073
Total production	52.918

Table 4.3: Energy Mix for Electricity production, Peru 2017, IEA [8]

less efficient open-cycle technology, expanded rapidly in Peru. The abundant supply of cheap natural gas facilitated this expansion [11].

Over the past decade, installed generation capacity in Peru has grown from 4.790 [MW] in 2005 to 12.508 [MW] in 2017 (8.3% growing each year), while peak demand has grown from 3.305 [MW] in 2005 to 6.596 [MW] in 2017 (so 5.9% growing each year). The reserve margin has therefore grown from 35% in 2005 to 81% in 2017. The reserve margin has steadily increased since 2011 due to the sustained growth in gas fire plant installed capacity, which outpaced the growth in peak demand [11]. See figure 4.7.

4.2.2 Electricity Prices

This previously introduced electricity oversupply has positively affected electricity marginal prices. The marginal cost of energy has become the lowest in South American countries, especially among the countries that have competitive power markets. According to [11], the average liberalized electricity prices in 2017 was 46.93 [US$/kWh]. As it is possible to observe in figure 4.8 electricity marginal costs during the last ten years are constantly decreasing, while contract prices in the liberalized market is decreasing in a more soft pace.

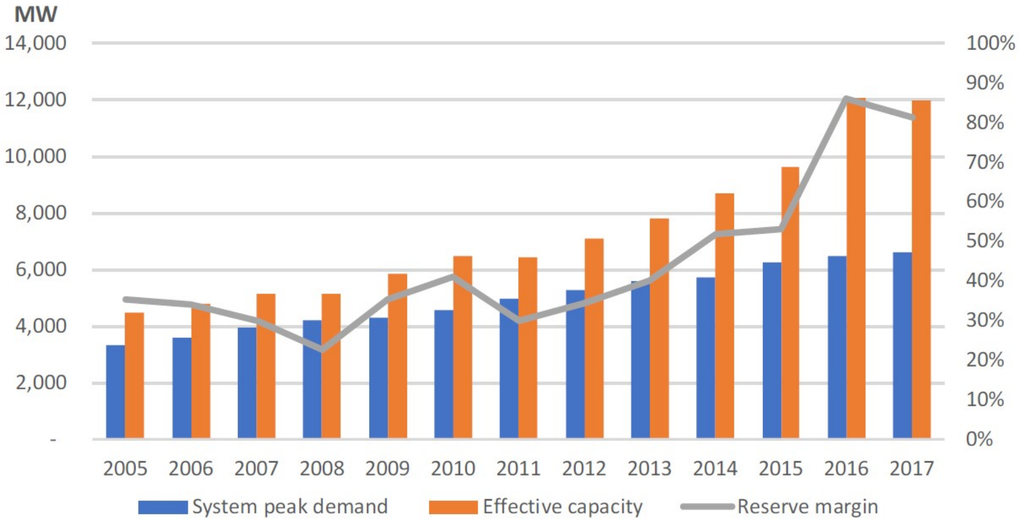


Figure 4.7: System adequacy in Peru: peak demand, installed capacity and reserve margin [11].

Power sector and Electricity prices, Peru (2017)	
Electricity generation [TWh]	53
Electricity final consumption [TWh]	46
Electricity losses [TWh]	5.5
Hydro, % of generation	54,9
Natural Gas, % of generation	37,2
Wind, Solar, biofuel, % of generation	4,5
Oil, % of generation	2,1
Spot electricity price (2005-2017) [USD/MWh]	39
Installed generation capacity [MW]	12.508
Peak demand [MW]	6.596
Reserve margin %	81

Table 4.4: Peruvian power market overview

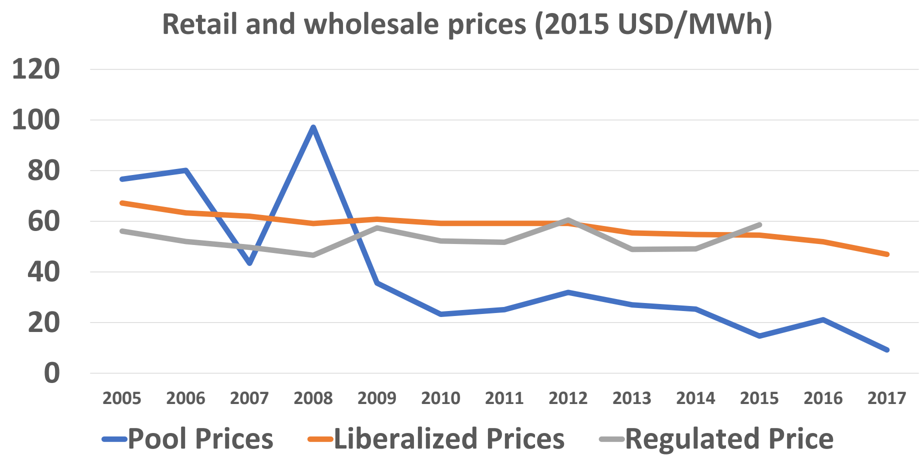


Figure 4.8: Electricity marginal costs and liberalized contracts prices. Peru [11]

4.3 Electrification Program

In 2013 the national rural electrification office (**DGER**) of energy ministry has implemented *Plan Nacional de Electrificacion Rural* (**PNER**). It is a national plan for rural electrification started in 2013 and expected to be concluded in 2022. The purpose of **PNER** by 2022 is to "reach a rural electrification coefficient of 95.8%, contributing to reduce poverty, as well as improving the level and quality of life of the inhabitants of rural, isolated households and border areas of the country in inclusion process" [15]. Main **PNER** targets are:

- The extension of the electrical border through the execution of rural electrical systems works.
- Propose the execution of such rural electrification with sustainable operations.
- Promote, through rural electrification, a sustainable socio-economic development of rural isolated areas, and border towns of the country, in order to improve the quality of life of the rural population.
- Promote the use of renewable energy sources in distributed generation systems and integrating it in the national electric distribution network.

The implementation of **PNER** will bring electricity to 6.2 million people and indirectly will promote rural development in the most remote areas of the country. The total investment is expected to cost 5.249 million Soles (almost 1.4 billion €). Direct expansion of national grid investments will amount to 3.205 million Soles (0.86 billion €).

4.3.1 Rural Incentives

As part of Peruvian rural electrification program in 2016 the Energy Ministry has created a social fund called *Fondo de Compensación Social Eléctrica* (**FOSE**). **FOSE**, increasing urban tariff, transfers funds to 26 system operators of rural areas, with the aim of decreasing electric bill of the most poor costumers [16]. In the only data available on 2016 **FOSE** has transferred to this 26 companies a monthly average of 15.4 million Soles (4.1 million €). Official statistics of MINEM say that three million people benefit of this found, with 68% reduction on bills.

Eilhicha, the official system operator in case study rural areas, have the possibility to use part of this fund. In 2016 **Eilhicha** received an average monthly amount of 149.000 Sol (40.000 €). **Eilhicha** calculate costumers bills by a monthly manual reading on all 5.804 household electric counter. In the next years the government is going to prepare a national modernization plan forcing electricity suppliers to install smart meters. This plan should be executed until 2025 but it has been criticized since estimations on cost of these meters is around 250 [€], so much more higher than the majority of annual rural tariffs.

2016	FOSE Transfer to Eilhicha	
Agust	S/ 171.887,00	46.409,49 €
September	S/ 144.958,00	39.138,66 €
October	S/ 144.958,00	39.138,66 €
November	S/ 144.958,00	39.138,66 €
December	S/ 138.505,00	37.396,35 €

Table 4.5: FOSE transfer to Eilhicha in 2016 [16].

4.4 Case Study Region

Ancash is a poor and mountain region, located 200 Km north of Lima. Geography is composed by two longitudinal valleys separated by a a dry mountain chain, Cordillera Negra, on East is present another mountain chain, Cordillera Blanca while on West the region is overlooking to the Pacific ocean. The economy is made up of mineral extractions, fishing near the coast and subsistence crops. Population of Ancash is 1.1 million people, the majority of people lives in poor urban agglomerates near the ocean and in small villages in the mountain regions. Inside Cordillera Blanca it was created a national park called Huascarán National Park, which has also been listed as a biosphere reservation and as a World Heritage Site by **UNESCO**.

The seasons are mainly divided between dry and raining one. The majority of raining is expected during this period, so between December and July. In mountain regions, especially close to glaciers, this fact is less acute. Glaciers and mountains reservoirs of waters will facilitate operation of off-river power plants.

4.4.1 Chacas Valley

Chacas valley is located on the westernmost area of Ancash region, the main river is Chacapata, although it is called with different names village by village. Chacapata river get into Rio Janamayo which get into Rio Maranon which is one of the two generating tributaries of Amazon river. To reach Chacas, from the other Peruvian region, only one paved road exist passing from a 4700 meter mountain pass. Due to the closeness to Cordillera Blanca the urban population live in an altitude around 3.000 meters, while the most remote rural villagers are located until an altitude of 5000 meters.

The valley is divided in two provinces, Asùncion and Carlos Fermìn Fitzcarrald. The total valley population is 17.717 [17], the majority of people lives in rural areas (83.12%) while urban population lives in two towns: Chacas and San Luis. Rural population of this area relies on agriculture and breeding. Main agricultural products are potatoes, maize and olloco (a kind of potato). Breeding is not intensive but made at household level, bread animals are bovine, pig, goat and sheep. Average income is 430 [Sol/month], so almost 112[€/month] [17].

Only two electricity distributors are present in the valley: **Eilhicha**, which we are going to better explain later, and Hidrandina. The second company supplies costumers only on the bottom part of the valley while **Eil-**

District	Total Population	Rural [%]
Chacas	4563	56,47
Acochaca	2815	100
San Luis	10493	78,57
San Nicolàs	3131	100
Yauya	4093	100
Whole Area	25095	83,12

Table 4.6: Population of case study area

hicha on the valley top, so whole valley grid can be split into two separated grids. National grid interconnection is therefore this conjunction between this two separate grids, physically it is a single line with a power meter, located near the village of Pommalucay. On the valley bottom, in Pomabamba the bottom grid part is connected with a 60 [kV] interconnection to high voltage national grid. Sistema Electrico Chacas, the power system studied in the present case study is therefore only the valley top section of the grid.

Agricultural Products	Rate
Potatos	33%
Mais	25%
Olloco	19%
Oca	11%
Broad Beams	2%
Barley	2%
Wheat	2%
Snatch	2%
Poroto	2%
Other Fruits	2%

Table 4.7: Agricultural products of of case study area

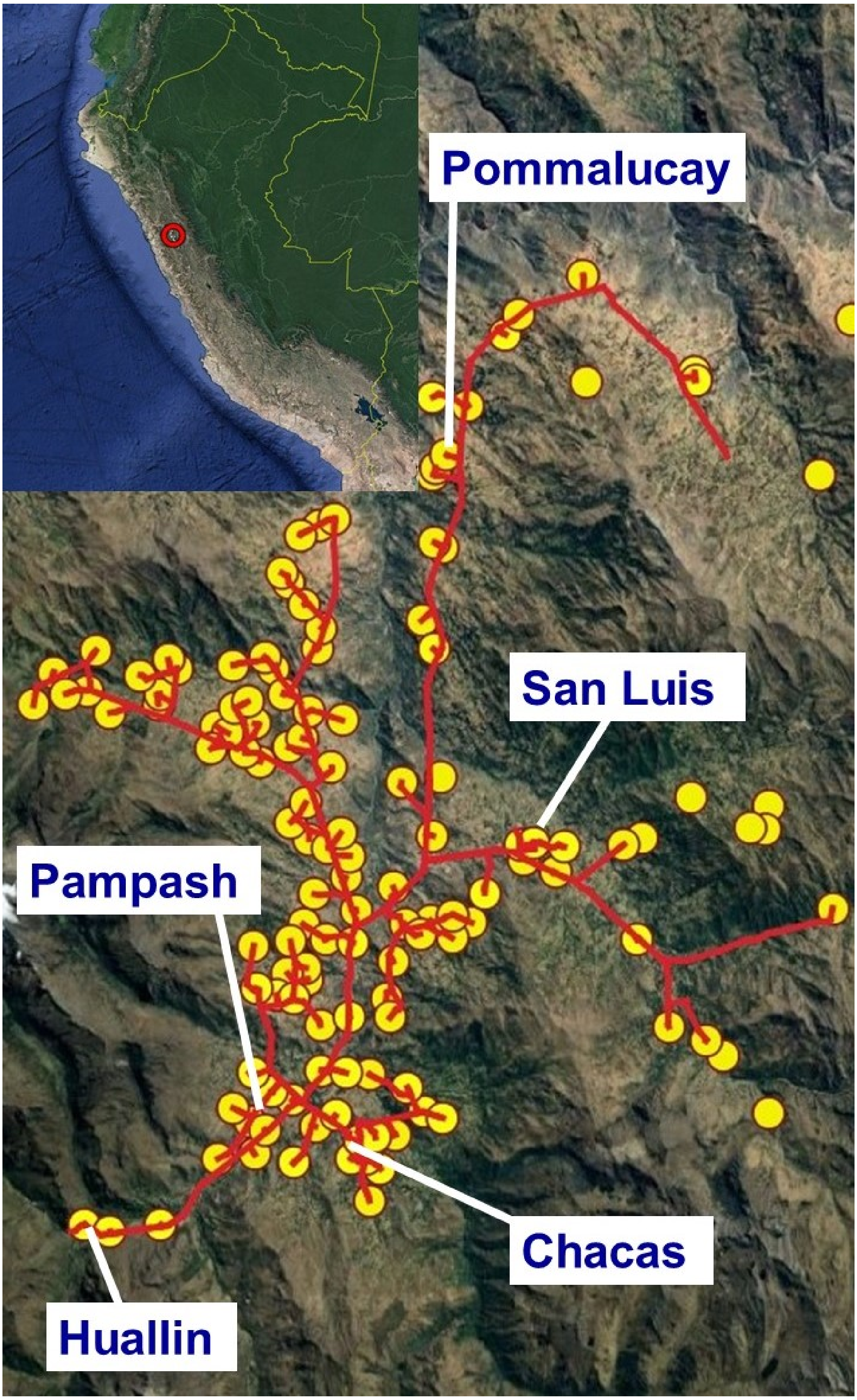


Figure 4.9: Chacas valley and GIS model of the grid.

4.5 Sistema Eléctrico Chacas

Case study grid is set in Chacas valley. The majority of grid utilities and costumers are located at an altitude higher than 3000 meters. A wide range of this population is connected to the grid (95%), in 2019 users was 5.804 so on the near future it is not expected an increase on the total number of users. The majority of users are poor households and, because of an improvement in their welfare is expected in the region, it is reasonable to think that each of them that will remain in the valley will increase their electricity demand. A great range of rural population, especially between the young ones, is migrating to the major cities near the coast or to closer urban areas. These two opposite trends make difficult to evaluate load estimation on the long period.

4.5.1 Grid Overview

Chacas grid is characterized by a radial topology, the four hydroelectric generators are located close to the mountains, costumers are spread through all the valley but the main consuming one are concentrated in the two main towns. Distribution grid nominal voltage is 22.9 [kV]. Total MV lines length is 135.2 [km], distance between valley top and bottom is more than 30 [km]. Installed capacity of the four generators is 6.370 [kW]. Distribution system is composed by 919 MV-branches, 792 MV-pylons and 138 MV-LV transformers.

Low voltage grid nominal voltage is 400 [V], it is composed of 4.539 LV-branches and 4.192 LV-pylons.

4.5.2 Stakeholders

Stakeholders involved in Chacas electric power system are the system operator **Eilhicha**, the competitor Hidrandina which supply valley bottom costumers, the **National Rural Electrification Office** (DGER) and the national transmission system operator (COES). In addition there are **Operazione Matogrosso** (OMG) and **Parroqua de Chacas**, both managed by Italian volunteers that 30 years ago created and expanded the grid.

Eilhicha S.A. The system operator of Sistema Electrico Chacas, it is also a local electricity producer and retailer. It was created in 1993 by Italian volunteers. **Eilhicha** has 26 employees, its main activities are producing electricity, operating and maintaining the grid, controlling energy flows, installing new equipments and collecting bills.

Population	25.095
Grid Users	5.804
Global MV length [km]	135,2
Number of MV lines	919
Number of MV pylons	792
Installed capacity [kW]	2.999
MV-LV transformers	138
MV Nominal Voltage [kV]	22,9
Frequency [Hz]	60

Table 4.8: Overview on Sistema Electrico Chacas

Hidrandina S.A. A public company which owns franchise contracts for distribution and retailing of electricity in Ancash, La Libertad and Cajamarca regions, serving more than 400.000 clients. It is controlled with a share of 95% by *Fondo Nacional de Financiamiento de la Actividad Empresarial del Estado*, a Peruvian group of public enterprises controlled by the Ministry of Economy and Finance. All electricity exchanged, by Sistema Electrico Chacas, through the so called "external grid" is sold or purchased by Hidrandina.

Santa Lucia de la Parroquia de Chacas Also this company was created by OMG, in 2015. It owns and control Huallin power plant, the biggest one connected to the grid of **Eilhicha**. All the electricity produced by Santa Lucia is sold to Hidrandina. Officially none is working in Santa Lucia since both the office and employees are the same of **Eilhicha**.

Operazione Matogrosso OMG is an Italian movement whose purpose is the education of young through free work given to poor, it is widespread very active in Latin America (Peru, Brazil, Ecuador and Bolivia). OMG is actively present in the valley since 1976 with the social mission to help the poor living in this rural area. In the valley OMG has built schools, hospitals, kindergartens, carpentry workshops and other activities. Thanks to charity donations OMG has installed the majority of MV and LV grid, the three power plants and other grid utilities.

Parroquia de Chacas The local parish, active in social and economic activities of Chacas. Its Italian voluntaries manage **Eilhicha**, Santa

Lucia, **Mama Ashu** hospital, some cooperatives, laboratories and schools. From the load point of view, these activities represent almost 30% of electricity demand.

Direction de Electrificacion Rural DGER is the office of the Ministry of Energy and Mines (MEM), responsible of expanding the electricity coverage, especially in the rural area. It has planned, designed and installed the most recent part of Chacas grid. After completed the installation all utilities have been given in concession to the system operator. In 2013 DGER has implemented Plan Nacional de Electrificacion Rural (PNER), the actual rural electrification program.

4.5.3 Grid History

Sistema Electrico Chacas was created in 1976 with a small hydropower plant in Collo. In the following years it was expanded, new generators and turbines were installed and load by load almost all people in the valley were connected to the grid. The main electrification steps of Chacas power systems were:

1976 Chacas power system was born with the installation of the first Collo hydropower plant and the connection of the first costumers. In the first period the grid was supplying electricity only to Chacas, San Luis and other big costumers such as **Mama Ashu** hospital and the parish. In this first years it was also installed a first 100 [kW] in Jambon.

1993 Chacas branch of distribution grid was replaced with a more robust one.

1996 The grid was strengthened also in the branch supplying San Luis.

2000 In Collo the existing generator was substituted by a more power one, it is still operative.

2005 In Jambon the existing generator was substituted by a 450 [kW] one.

2010 A second Jambon 1000 [kW] power plant was installed.

2012 In San Luis the diesel generator was installed.

2015 In Huallin it was installed a 4000 [kW] turbine, it is owned by Santa Lucia and all of electricity produced is sold to Hidrandina.

4.6 Costumers and Load Characteristics

The following data we are going to show are taken by **Eilhicha** database. Data period is between October 2017 and October 2018. Total number of costumers is 5.804, the majority of costumers is provided by a 3 [kW] electric connection, a value much more higher than the power they actually need.

4.6.1 Electricity Consumption

Total grid electricity consumption of the average on this period is 232.5 [MWh/month]. Total number of costumers is 5.804, almost ten percent (570) of these costumers have a negligible electricity consumption (less than 1 [kWh/month]) while a quarter of electricity is used by only 22 top-consumer costumers. These consumers are the hospital, municipality, local parish and some little local businesses. Ten percent of electricity is consumed by only **Mama Ashu** hospital.

Many other costumers (28.2%) have a very low electricity consumption (between 1 and 10 [kWh/month]). The majority of costumers (57.0%) has an electricity consumption range between 10 and 100 [kWh/month]. This range of users represent almost half (44.4%) of total consumption. The three lowest consuming categories (from zero to 10 [kWh/month]) represent 37.9% of consumers but only 3.6% of electricity consumption. Year after years electricity consumption is slowing increasing but up to now peak load is lower than installed capacity. For the majority of the year there is always one or more turbines turned off, since **Eilhicha** cannot sell electricity to the national grid. To to make up for load demand growth, it is planned the installation of a new power plant in Collo, at the present timings are not yet known.

It can be interesting to highlight this fact, the majority of such low-consuming costumers lives in sparse, low populated and difficult access areas, while the majority of electricity consumption is concentrated close the main towns and to generators. A great amount of grid kilometers was thus installed only for reaching the less-consuming (and poor) users. See figure 4.10 and 4.11.

4.6.2 Bills

The average of **Eilhicha** electric bills between October2017 and October2018 is 20.2 [sol/month] (5.4 [€/month]). The majority of costumers (65.5%) pay less than 10 [Sol/month] (2.7 [€/month]), one third pay between 10 and 100 [Sol/month], while 2% of them pay more than 100 [Sol/month]. Only 9

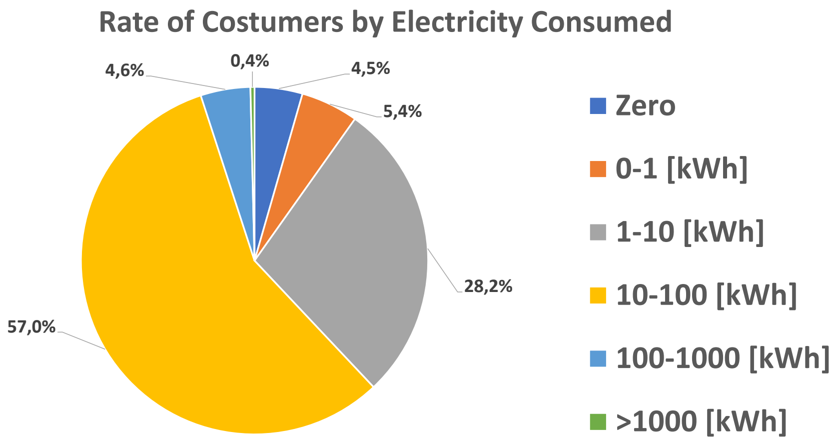


Figure 4.10: Eilhicha costumers by Electricity Consumed, monthly average between Oct. 2017 and Oct. 2018.

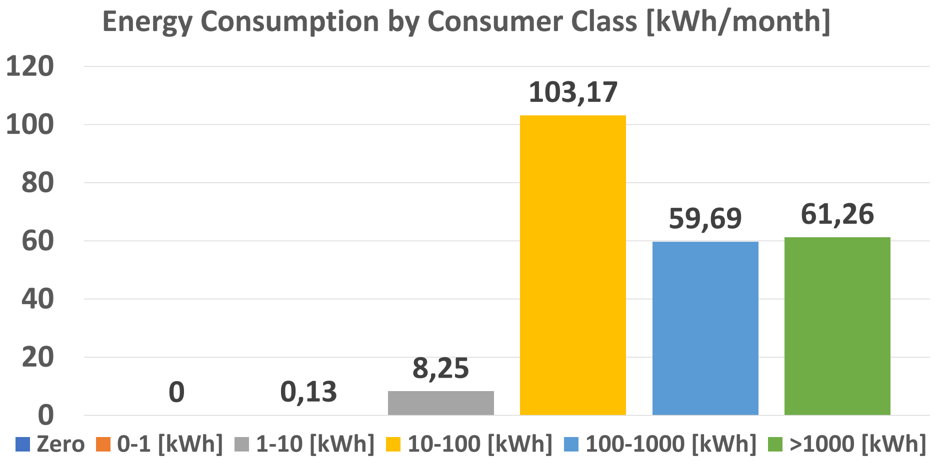


Figure 4.11: Electricity consumption of Eilhicha costumers classes, monthly average between Oct. 2017 and Oct. 2018.

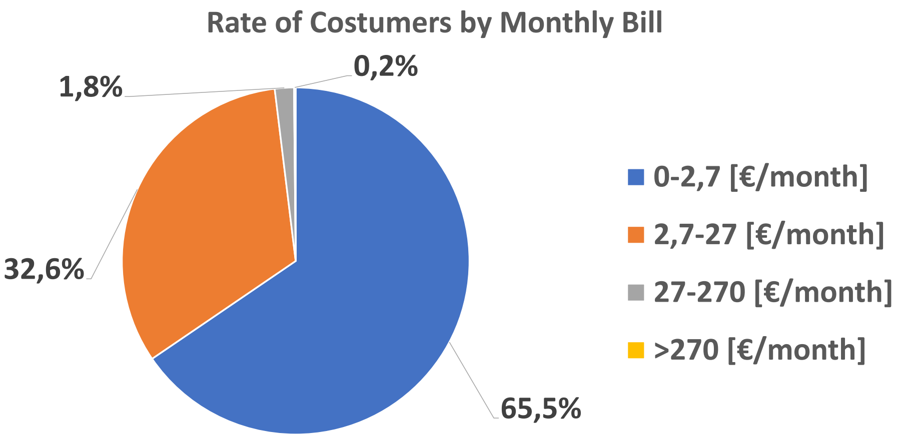


Figure 4.12: Electric bills distribution of Eilhicha costumers, monthly average between Oct. 2017 and Oct. 2018.

costumers have an electric bill higher than 1000 [Sol/month] (270 [€/month]). It is important to highlight that the top-consuming users are not the top-paying, since many top-consumers are activities related to **OMG** and local parish. Chacas grid was initially created to give electricity to these activities, so the bill they pay is very reduced.

4.6.3 New Costumers: V Etapa

As we previously introduced DGER is installing 16 new transformers and all LV utilities in new villages of Chacas valley. All of these transformers have 5 [kW] nominal power, once this connection will be realized the ministry expected a 43.3 [kW] increase of load demand in Chacas distribution grid. Figure 4.9

Village	Power Expected [kW]	Phase
CANINACO ALTO	1,06	R
CARASH ALTO	1,26	R
ESPADIN BAJO	1,93	R
HUACUY	1,93	S
VILLA SUR	3,65	S
LLAMELLIN OCO	4,41	S
UCHUPUQUIO	2,2	T
TASHTA ALTO	3,69	R
OCSHAPAMPA ALTO	1,73	R
ALTO ANDINO	2,4	R
BUENOS AIRES ALTO	3,29	R
PULIAG DE ILLAURO	3,69	T
SIRHUARURI	3,07	R
VILLA SUR	3,65	S
TAMBILLOS	2,83	S
CANINACO CHICO	2,5	R

Table 4.9: New load installed in V Etapa. Transformers nominal power is 5 [kW], power are calculated by [15].

Chapter 5

HUGO Application to The Case Study

In order to create the electric grid model **HUGO** need to receive as input set of data regarding the structure of real grid. These information are about grid structure, load and generators characteristics, external grid and about the future grid expansion. In the following sections we are going to see in detail which information we need, how they are stored in the existing **GIS** model and how **HUGO** imports them. In the last section we explain how was adapted **HUGO** code to the case study, in order to include these information in the model.

5.1 Grid structure

Chacas power system model of distribution grid is divided in 919 lines supported by 792 poles. Since distribution network was installed in different time periods many differences in material, diameters and phases used are present between grid branches. The majority of lines are made of aluminum, the older branches also of copper. Cable areas range from 16 to 70 square millimeter. Most branches bring all of three phases, while some of them only one or two. The neutral is present in all branches bringing three phases. Pylons are made by local Eucalyptus wood and by concrete. They are high between 6 and 10 meters. The eight different cables encountered are explained in table 5.1. Chacas power system is composed of 138 MV-LV transformers 16 of which will be connected in the next few months. Range of nominal power is between 5 to 400 [kW], for the most part they are mono-phase 5 [kW].

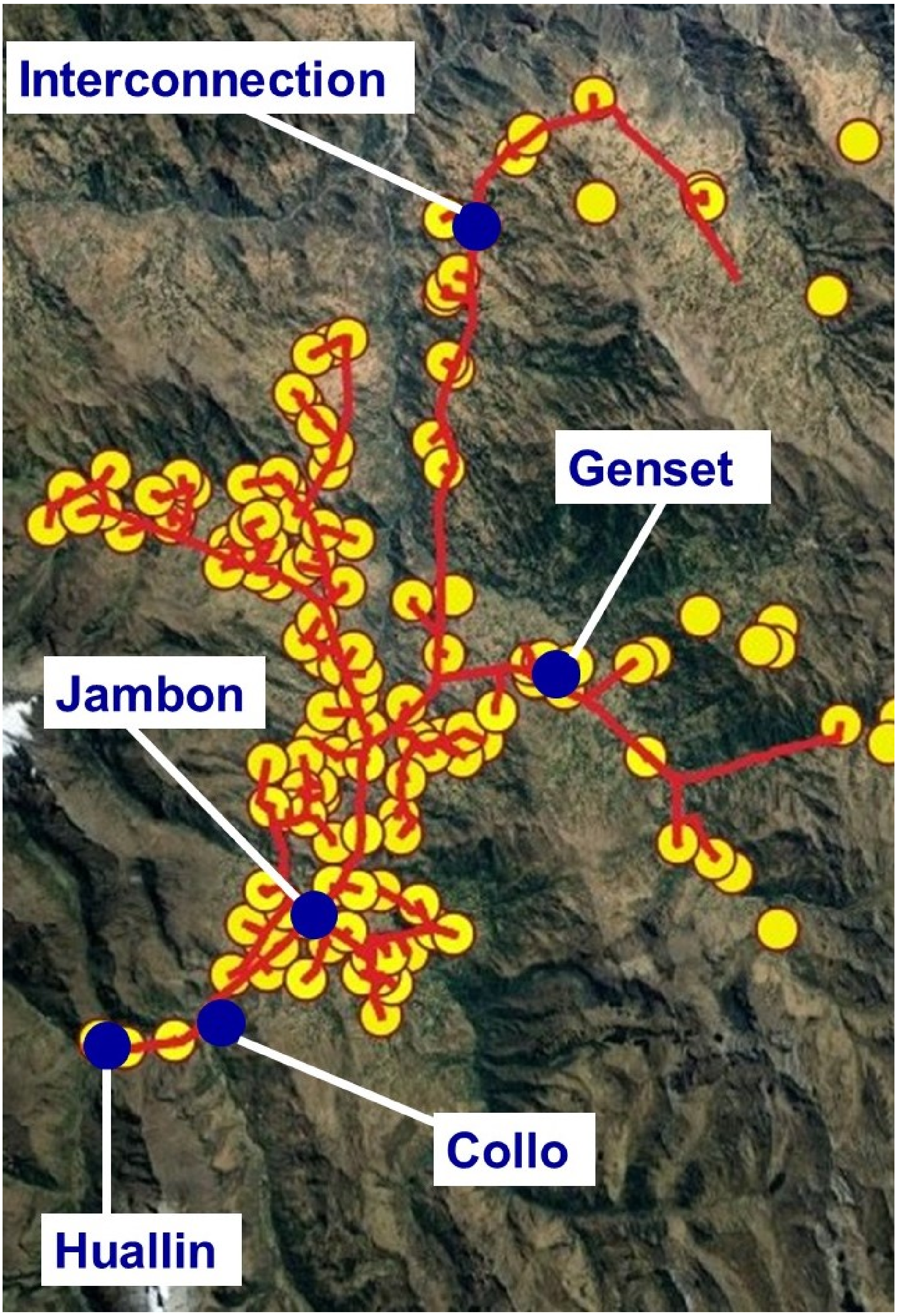


Figure 5.1: Chacas valley, GIS model of the grid and the main sites of interest in the distribution system.

Local Name	Material	Cross Area [mm2]	Nominal Current [kA]
Al_016_mm2	Aluminium	16	0,1
Al_025_mm2	Aluminium	25	0,125
Al_035_mm2	Aluminium	35	0,16
Al_050_mm2	Aluminium	50	0,195
Al_070_mm2	Aluminium	70	0,235
Cu_016_mm2	Copper	16	0,1
Cu_025_mm2	Copper	25	0,188

Table 5.1: Cables typology present in lines of Sistema Electrico Chacas.

5.1.1 Generators

Sistema Electrico Chacas have four generators located in Huallin, Collo and Jambòn. All of them are run-off river hydro power plants, they are located on valley top, close to mountains and theirs glaciers and rivers. The main power plant is Huallin, it was installed recently in 2015, nominal power is 4 [MW]. Actually this generator is producing only 1.1 [MW], since a higher charge cannot be sustained by the grid. Collo power plant was connected in 2000, the generator have 0.92 [MW] of nominal power. Jambon power plant is composed of two turbines: Jambon_A with 1 [MW] and Jambon_B with 0.45 [MW]. Jambon_A was installed in 2010 wile Jambon_B in 2005. In table 5.2 are described grid generators and their properties while in figure 5.1 there they are located.

As we previously introduced, clime in the region has more or less two seasons, dry season during summer and rainy during winter. In Huallin and Collo plants the amount of water necessary for turbines is much less than rivers availability, so water supply is not an important issue, also during dry season. In Jambon power plant normally only one of the two turbine is working, water is collected for both in a unique catchment basin and than sent with different pipelines into turbines. In the present year a new channel was built near Jambon to supply an additional water supply, from a different river, to this basin.

5.1.2 San Luis Genset

In addition to hydro power plants in 2012 a genset was donated to the parish and installed in San Luis. **Eilhicha** have the possibility to use it whenever

Generator	Nominal Power [MW]	Machine	Installation year
Jambòn A	1	Francis	2005
Jambòn B	0,45	Francis	2010
Collo	0,92	Francis	2002
Huallin	4	Pelton	2015
San Luis	0,8	Genset	2012

Table 5.2: Chacas power system's Generators

necessary. Nominal power of this diesel generator is 0.8 [MW]. The reason of this machine installation is to serve as a backup generators. When an unexpected outage incur in power plants the generator can start to supply electricity to the grid. It must be said that, according to the experience of local system operator, time necessary to reach and start this machine, frequencies and time of outages make unjustified the purchase and installation of this machine. A complete and affordable set of outages data is not present, so it is difficult to actually estimate numbers and length of service interruption. Relying on **Eilhicha's** operators experience, grid's outages occurs almost once a week and outages time are normally 2 or 3 hours. Time necessary to reach genset, from Chacas offices to San Luis is almost one hour.

5.1.3 Interconnection with the National Grid

National grid interconnection is located in Pommalucay, on valley bottom. From a physical point of view the interconnection with the national grid is simply a busbar with a power meter. Since voltage level on grid both side is the same, a transformer is not necessary. This interconnection is mainly used to sell to Hidrandina all the electricity generated by Huallin power plant. Only in rare circumstances this interconnection is used to purchase electricity from the national grid. The following data 5.3 were directly read by Pommalucay power meter with one hour time step.

In the next few years it is planned to be generated a new interconnection with the national grid. This 60 [kV] connection will have nominal power of 6 [MW] and will be realized in Huallin, directly connected with the power plant. National high voltage trellis actually are 20 [km] south than Chacas valley, so also the national grid network will be extended. Civil works will start in 2020 and will be completed in two years.

Power [kW]	Sell	Purchased
July	1049,5	0
September	803,9	1,64
October	1044,7	0
November	1028,9	0
December	1033,5	0

Table 5.3: Official exchanges in Pommalucay interconnection between Hidrandina and Eilhicha.

5.1.4 Planned Expansions

In the future it is expected the installation of a new power plant near the one existing in **Collo**. A group of retirees Italian volunteers, with years of experiences on hydro plants, are preparing the project and restoring an old turbine to use in the plant. This new plant is expected to generate around 1 [MW] and will be owned by **Eilhicha**, so generated electricity will only be used to make up local increase in load demand.

5.2 GPKG Input file

All of information described in the previous section are stored in specifics GPKG files. **Eilhicha** employees use the software QGIS to visualize and editing these data. In figure 5.1 we can see the QGIS model, distribution branches are the red lines while the yellow dots are MV-LV transformers. Not all layers are visible since many utilities are not of interest.

In the following subsections we are going to describe all code alterations necessary to make **HUGO** works with the actual case study. To be precise, all changes in the code that specifies GPKG layers the code must deal with. As described in chapter 3 to import GPKG layers **HUGO** needs to know a priori the mane of import file, layers and attributes of each layer, so these names must be manually insert in the code.

5.2.1 MTTRAMO

This is the layer that stores information about MV lines. The grid is made of 919 medium voltage lines with a total length of 135.2 [km]. All MV line have three phases and the neutral. It's attribute table contains the following fields list.

- **CodTramo** Local Code of every single line. It start with the incipit TMT followed by six numbers. An example is TMT000772. In python this layer will be saved as both local name (**Tramo_LocName**) and foreign identifier (**FID**).
- **Localidad1** Locality, where a line begin. It is an incomplete field so it is not imported.
- **CodNorma** VNR code for cables. It contain information about material, diameter and number of phases.
- **CodNormaNe** VNR code for the neutral, if present. It contains information about material and diameter. If a neutral is not present the field is empty.
- **Length** Line length in meters, it is calculated as 2D difference between geographic coordinates. In a mountain landscape such Chacas a simplification such that can generate non negligible errors.
- **CodTramo_1** It is the code of the previous line. The grid is similar to a chain, each line have only the information about the previous one. This attribute will be fundamental during the next steps.

5.2.2 MTPOSTE

This layer contain information about grid poles. Each pole represent a junction point of lines and physically support the three phases and the neutral. There are 792 poles. Generators and transformers bus bar are not stored in this layer. The main fields are:

- **CodEstruct** Local Code of every single pole. It start with the incipit STR followed by six numbers. An example is STR000123. This layer will be saved as **Poste_LocName**.
- **Localidad** Locality of a pole, since it is an incomplete field it is not imported.
- **CodTramoMT** The code of the entering line in the pole. The logic direction is not necessary following power flow direction.

- **Position X, Position Y** These fields represent geographic coordinates of poles, both are assigned according to Universal Transverse Mercator (UTM) system. They are imported as they are and later translated from UTM to Latitude and Longitude coordinates.

- **Position Z** Pole altitude, the grid is mainly located between 2.500 and 4.000 meters. This data will not be considered and not imported due to incompleteness of data.

5.2.3 MTNODO

Transformers busbar are not modelled as a MTPOSTE but as different objects called MTNODO. There are 138 objects, each for each SED. Fields considered are similar to MTPOSTE:

- **CodNodoMT** Local code. It start with the incipit NOD followed by six numbers. An example is NOD000102. This layer will be saved as **Nodo_LocName**.

- **CodTramoMT** The code of the entering line.

- **Position X, Position Y** Geographic coordinates, such as MTPOSTE.

- **Position Z** Altitude, such ad MTPOSTE it is ignored.

5.2.4 SED

This is the transformers layer. It contains information about both transformers and phases used by load. All transformers have three phases but in the majority of villages only one phase is connected to loads. There are 138 SED, 16 of which are not yet installed or connected, anyhow they are imported and will be considered during the modelling of the future grid operation. In the first part of the import procedure we already imported a list of SED but the previous one was only useful for load assignment. Fields imported and necessary for the model are:

- **CodSed** Local code of the transformer. It start with the incipit SED followed by one or two numbers for the zone identification and other four numbers. An example is SED030007, where 03 is the identifier for Chacas zone. It is important to highlight that these zones are not the ones considered in the previous load estimation. This layer will be saved as **Tr2_LocName**.

- **Localidad** Locality of the transformer. In this layer the locality information will be saved, since it will be useful in PowerFactory model.
- **PotInstal** Nominal power of the transformer. The range varies from 5 to 400 [kW] but the majority of transformers are equal to or less than 25 [kW].
- **FasiOut** A code storing information about which phases will be used by costumers. All SED transform the three phases, but not all of them will be used. This code is a string composed by letters, as an example RN means that this specific load is connected to phase R and the neutral (N) while RSTN means that all phases are used.
- **CodNodo** The code of the pole over which (or close to) the transformer is installed. This data is fundamental to connect transformers and loads to the chain of lines.
- **Active** This field give the information whether the SED is active or not. It is a Boolean, where 1 means it is active and 0 that it is not yet connected or installed. This data will be imported and used in **PowerFactory** model.
- **Position X, Position Y** Geographic coordinates of transformers.

5.2.5 Coordinate Variation

Coordinates of many grid objects imported up to now are stored as UTM coordinates systems. Since **HUGO** need to work with Longitude and Latitude we need to transform these coordinates into the correct reference system. An appropriate function was found and inserted in Python code [14]. Each object couple of coordinates is thus converted into Latitude and Longitude.

5.2.6 Codigos VNR

As we previously explained many codices are used to identify different object properties. These codes (Còdigos VNR) were created by the *Organismo Supervisor de la Inversión en Energía y Minería* (OSINERGIM) a supervisory body of Peruvian state [18]. From VNR codices **HUGO** will identify object type, so their meaning must be manually inserted in the code. As an example, if a line have as a property **AA02503** in **CodNorma** field it means that this specific line is made of Aluminum, have a section of 25 square millimeters and all of three phases are present. The line in this example will thus be recognized having **41** as **TypCon** FID (see figure 5.2).

```
for e in Tramo_LocName:
    if Tramo_CodNorma[j] == 'AA01602':
        tipcond = 40
    elif Tramo_CodNorma[j] == 'AA02501':
        tipcond = 41
    elif Tramo_CodNorma[j] == 'AA02502':
        tipcond = 41
    elif Tramo_CodNorma[j] == 'AA02503':
        tipcond = 41
    elif Tramo_CodNorma[j] == 'AA03503':
        tipcond = 42
```

Figure 5.2: Identification TypCon of each line.

5.3 Measured Power Flow

Figure 5.3: San Luis panel

One of daily tasks of **Eilhicha** employees is to collect a set of data necessary to evaluate performance of the power system. In the panel of figure 5.3 it is possible to see voltage level, frequency, phase displacement and power flow between the plant and San Luis zone of the grid. Data acquired are power Generation of each power plant, fluxes to or by the interconnection with Hidrandina grid and lectures of Jambon panels. An example of these panels can be observed in figure 5.3, the data collected by such panels are power flows entering and exiting from Jambon power plant. Jambon plant is located in the middle of distribution grid, from this plant come out the four main branches of the grid. Data collection is made by hand by **Eilhicha** employees, written in a data book and then reported in a specific Excel file called **PowerFlows**, in this non-automatic process some human mistakes commonly happened, so check was made during Python data elaboration. Time step is one hour and data acquired represent only a fraction of the years.

To be precise only the months of July, September, October, November and December of 2018 had enough reliable data to be taken in consideration.

5.4 Hypothesis for Grid Model and HUGO Code

Some hypothesis and assumption have to be made to crate the electrical model of the grid. A great amount of time and efforts was used also to adapt the ideal **HUGO** code to the case study. We will start this section with a description of hypothesis regarding grid utilities and load estimation, than we will analyze code peculiarity.

Since the existing GIS grid model was created, and daily used, mainly by Spanish-speaking users it will be used up to now a specific Peruvian technical language. Many layer's properties in GIS model and Excel File adopt this vocabulary, in table 5.4 are explained the main words and codes also used by us in the next section.

TRAMO	line
MTTRAMO	MV line
POSTE-NODO	pole
MTPOSTE	MV pole
SED	MV-LV transformer

Table 5.4: Peruvian technical vocabulary adopted in the thesis

5.4.1 Type Hypotheses

The following list introduce the main hypotheses used to model the distribution grid of the case study.

Lines In the real grid different type of line geometry exists. Some network branches bring to load only one or two phases and the neutral is not always present. In our grid model we will assume that all lines have three phases and the neutral. Also pylon geometry is constant, the pole is ten meters height and branches are three meters wide.
In **PowerFactory** model this assumption will generate only one geometry type, so only one raw will be present in sheet **TypGeo**.

Load Type To model loads we considered only one **TypLod**, this load type have three phases, the neutral and Y connection. In the real grid almost half transformers are mono-phase and others have two phases, as will be better explained later, this problem is solved assigning the load to only the correct phases used by each transformers.

5.4.2 Load Hypoteses

Load import is performed assigning to each transformers a list of (P-Q) couples. Since **Eilhicha** did not collect electricity consumption of each transformer, load estimation is performed trough the following procedure, based on grid zones subdivision observed in figure 5.4.

PowerFlows Excel file are reported lectures regarding power flows entering into grid (from power plants dispatching), exchanged with Hidrandina (so sold to or purchased by the interconnection to the national grid) and sent to grid branches. From these lectures it was also possible to split load flows into four zones. For each of such zone we calculated load demand with a simple power flow balance between the entering power, into the zone, and the exiting one. In this simple model we will not consider grid losses and at the moment they will be ignored. The four zones considered begin or conclude in Jambon power plant and are shown in figure 5.4. This zones are:

- **Huallin** It is the smaller branch with the lowest load demand. This zone include two power plants (Huallin and Collo) and the load is calculated as a difference between the injections by the power plants and the reading of the panel in Jambon (see equation (5.1)).

$$P_{Huallin}^{load} = P_{Huallin}^{gen} + P_{Collo}^{load} - P_{H-J}^{flow} \tag{5.1}$$

- **Chacas** It is the closer zone to Jambon, the flow is directly read in the panel (5.2). Thanks to these considerations it can be considered a reliable data.

$$P_{Chacas}^{load} = P_{J-C}^{flow} \tag{5.2}$$

- **Pampash** As the flow to Chacas it is directly red in the panel (5.3) and can be considered reliable.

$$P_{Pampash}^{load} = P_{J-P}^{flow} \tag{5.3}$$

- **San Luis** It is the less reliable data. It is calculated as the difference between the power flowing from Jambon power plant and the exchange with the national grid (5.4). In this zone no power plant are present and load represent the majority of costumers and power consumption. All planned SED and costumers to be connected in V ETAPA are in San Luis zone.

$$P_{SanLuis}^{load} = P_{J-SL}^{flow} - P_{Hidrandina}^{flow} \tag{5.4}$$

Day and Hour to be Modeled Once imported in Python ambient each zone demand we sorted power demand, of the whole grid, in ascending order. Than we removed some values obviously wrong and selected, as load to be imported in the model, the higher one. Load Flow we will later perform in DigSilent will thus represent Chacas power system in this specific moment, which is July 11 2018 at 20:00.

Load Assignment Once we have obtained zones load we need to split in between the active transformers. As we said in chapter 3 **HUGO** needs as a load import a list of P-Q couples for each transformer.

From **PowerFlows** Excel file besides power absorbed by each zone we also imported SED properties about belonging zone, nominal power and if the transformer is active or not. Inside python ambient each SED load is thus calculated as a weighted average, on nominal power, of zone load.

$$P^{load}_{Huallin,i} = P^{load}_{Huallin} \frac{P^{nominal}_{i}}{\sum_{j=1}^{Huallin} P^{nominal}_{j}} \tag{5.5}$$

$$P^{load}_{Chacas,i} = P^{load}_{Chacas} \frac{P^{nominal}_{i}}{\sum_{j=1}^{Chacas} P^{nominal}_{j}} \tag{5.6}$$

$$P^{load}_{Pampash,i} = P^{load}_{Pampash} \frac{P^{nominal}_{i}}{\sum_{j=1}^{Pampash} P^{nominal}_{j}} \tag{5.7}$$

$$P^{load}_{SanLuis,i} = P^{load}_{SanLuis} \frac{P^{nominal}_{i}}{\sum_{j=1}^{SanLuis} P^{nominal}_{j}} \tag{5.8}$$

Some SED are not yet installed or connected to the main grid. They are part of the next electrification step (V Etapa) and a load estimation was directly calculated by the ministry. We will use this value as it is. In figure 5.5 it is possible to see actives SED in yellow and the planned ones in red.

Active and Reactive Power Since the **PowerFlows** Excel file only take into account of Active power we need to split each SED value, calculated on the previous passages into active and reactive power. We will assume that load will be represented 95% by active power and the remaining by reactive. So inside Python ambient a generic 10 [kVA] load will be spited into 9.5 [kW] and 0.5 [kVar].

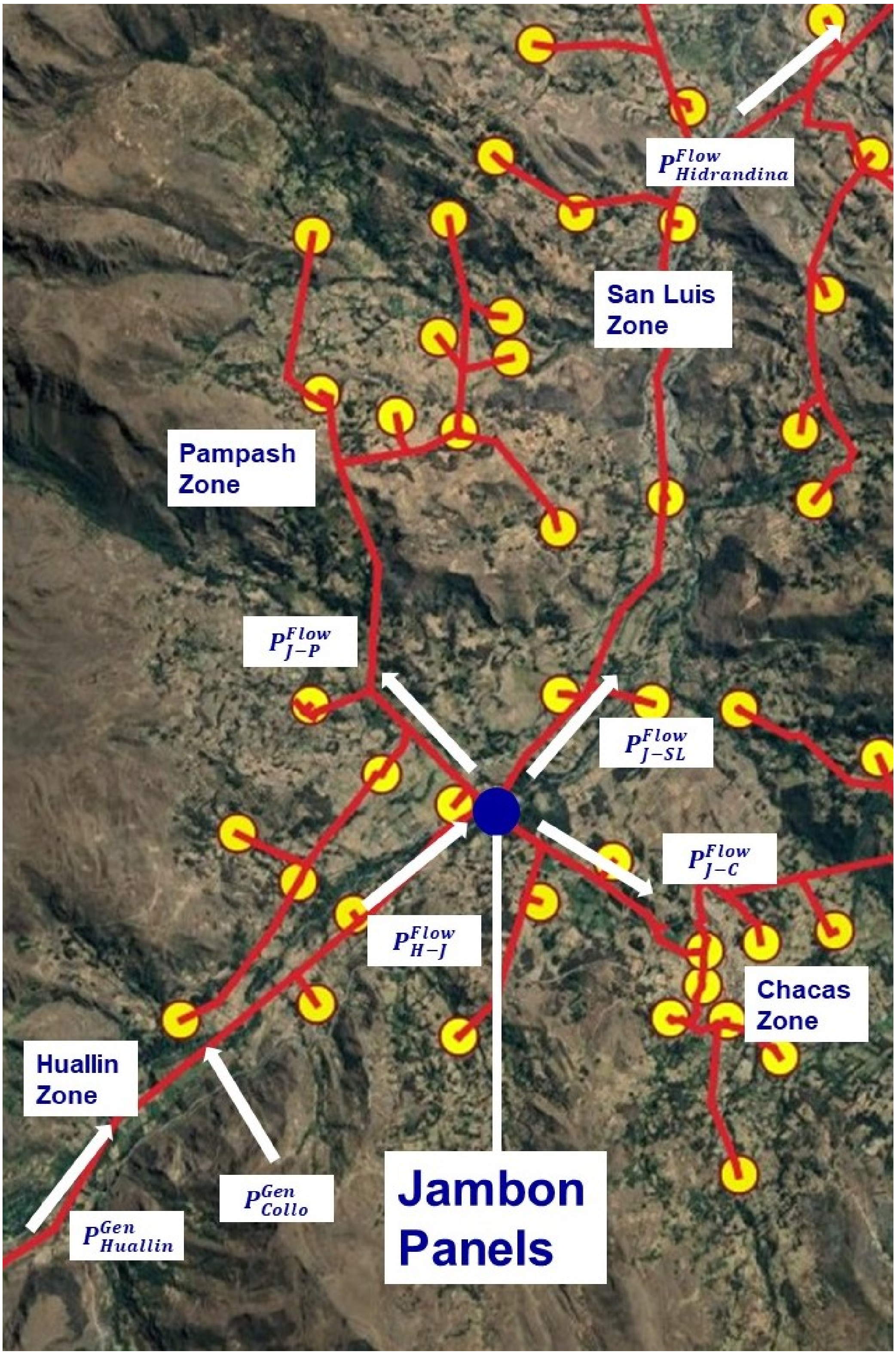


Figure 5.4: Jambon power plant panels and grid partition into zones

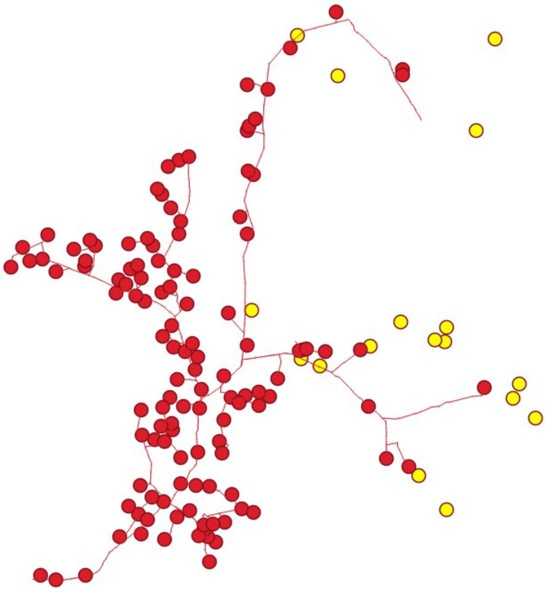


Figure 5.5: MV-LV Transformers Installed (red) and Planned (yellow)

Chapter 6

Simulations and Results

In order to validate the procedure developed in the thesis, we performed a set of simulations with Chacas grid model created by **HUGO**. Once established, with the first simulation, that the electric grid model crated by the procedure is actually an affordable model, we perform three future grid alteration to demonstrate its potentiality. Scenarios created by these simulations will give to the system operator technical information for easier daily operation and a more conscious future planning.

Thus we are going to perform simulations using the grid model created in Chapter 5. The model was prepared in a specific DGS Excel file, once imported in **PowerFactory** project library it can be saved and we didn't need anymore to import DGS file each time we want to use it. On the contrary, when one or more objects in GIS model are modified, eliminated or added **HUGO** must be run, a new DGS Excel file will be generated and a new import must be executed.

In the following simulations we are not going to modify GPKG file, so **HUGO** will not be ran anymore. All grid alteration will be implemented by hand. New utilities, lines, pylons will be manually added. V ETAPA information about transformers and new load are already present in GIS layers, so also on **PowerFactory** model of the grid, as NOT-ACTIVE objects. Therefore we don't need to add these information but only to manually activating them.

In each of this cases, the main results to be analyzed are bus voltage profile, line losses and loading on each phase and others variables that will be explained case by case. Buses considered are all internal nodes, junction nodes and bus bars in generators, external grid and LV transformers. Unless otherwise specified, the values of the loads will be the ones introduces in the previous chapter, so the most loaded hour of available historic data: July 11 2018 at 20:00

6.0.1 Peak Condition

In this specific hour of the year, the total load of the distribution grid was 950 [kW]. Load requests of each zone were: 50 [kW] in Huallin, 57 [kW] in Pampash, 324 [kW] in Chacas and 519 [kW] in San Luis. To supply this load (and, according to contracts, to sell to Hidrandina) three of the four generators were in operation, while Jambon B was closed. Huallin power plant was producing 1.100 [kW], Collo 920 [kW] and Jambon A 340 [kW].

More in detail we can see generators dispatching in the next two tables 6.1 and 6.2.

	11/7/2018 20:00 [kW]	Nominal Power [kW]
Jambòn A	340	1000
Jambòn B	0	450
Collo	540	920
Huallin	1100	1100
Global	1980	3470

Table 6.1: Case study, Generators dispatching on July 11 2018 at 20:00.

July 11 2018 at 20:00	Load Demand [kW]
Huallin	50
Pampash	57
Chacas	324
San Luis	519
Global	950

Table 6.2: Zone and global load demand on July 11 2018 at 20:00.

6.0.2 Load Flow Parameters

For all of the following simulations we decided to perform a Load Flow analysis, since we need to evaluate voltages and losses for each phase it is necessary to perform an **unbalanced AC Load Flow**. To set up **PowerFactory** to solve unbalanced power plow we will use options described in table 6.3:

Load Flow Parameters	
Calculation method	AC Load Flow, unbalanced, 3 phase (RST)
Load flow mothod	Newton-Raphson (Power Equations, classical)
Max. number of iterations	Newton-Raphson iteration: 30
Iteration step size	Fixed relaxation, Relaxation Factor: 0,5
Others	For all other options defaults values are used

Table 6.3: Load Flow parameters used to Power Flow Equations convergence

With these options simulations converge, with an average number of iterations between 10 and 15.

6.0.3 Cases Simulated

Cases presented in the following sections are described in table 6.4, the first simulation is made to the model of the actual grid, while the others are performed according to future expansion expected to be implemented in the following years. In GIS grid map depicted in figure 6.1 we can observe where the following interventions are located.

Case Name	Description	Section
Base Case	Grid model is be used as it is, a load flow is performed. This is the most important simulation since it represents the actual grid status at his maximum load registered: July 11 2018 at 20:00. As results will be shown grid voltage profile, line losses and line loading on each phase.	6.1
60 [kV] Interconnection	In the next two years it is expected to be realized a 60 [kV] interconnection with the national grid. The external grid, and a relative transformer, will be manually added near Huallin power plant. Since Huallin electric charge will be removed by grid backbone along the valley, the resulting loading and losses reduction will be evaluated.	6.2
V Etapa	New villages on the valley bottom are going to be connected to the grid, load expected has already been evaluated by the government. These new loads are connected to distribution grid and a load flow is performed.	6.3
Load Increasing	After Huallin Interconnection and V Etapa loads will be connected we will simulate Chacas grid load flow in the long period. Load is expected to increase with 5\% rate each year. Time step evaluated are 5, 10, 15 and 20 years from now (2019).	6.4

Table 6.4: Cases simulated with the electric grid model developed.

6.1 Base Case

In this first simulation we perform a Load Flow analysis of the actual state of the grid. V Etapa loads, 60 [kV] interconnection and Collo_2 plant are not yet connected. In the next figures we can observe the resulting Heatmaps of Voltage distribution across the grid (left) and line loading (right).

As we expected, voltage is higher on the valley top and constantly decrease up to Hidrandina interconnection. The fact that in some nodes, voltage is higher than 1 [p.u.] is congruent to the presence of the three operating generators, all grouped in this grid zone. The more we move away from generators, the more the voltage decreases, since to Pommalucay interconnection where a value close to 1 [p.u.] is computed.

6.1.1 Voltage Magnitude

Maximum bus voltage value calculated is 1.046 (phase T). As we expected maximum voltage occurs on the first line exiting from Huallin power plant. Minimum value is 0.9997 (phase T), on valley bottom, so close to the last load and far away from generators. In table 6.5 we can better observe minimum and maximum voltages, printed also in figure (a). The majority of line is therefore over nominal voltage, not more than 60 lines are under 1 [p.u.].

	Phase R	Phase S	Phase T
Max Magnitude	1,044863	1,042061	1,045807
Min Magnitude	0,99971	0,999702	0,999701

Table 6.5: Max and Min bus voltage magnitude [p.u.], Base Case

6.1.2 Line Loading

As we can see in figure (b) the maximum values of line loading are along the grid backbone. The figure was modified to make results only values higher than 10%. An important consideration can be considered on this figure; the highest line loading values are not close to power plant, where line cables has high cross area (35 and 50 square millimeter), but after the town of San Luis. In this valley bottom part of the grid, cables are made of aluminium, with 25 square millimeter of cross area.

The maximum loading is 38.22%, but the majority of lines are underexploited. As we can see on figure 6.2, 75% of lines, during the hour of

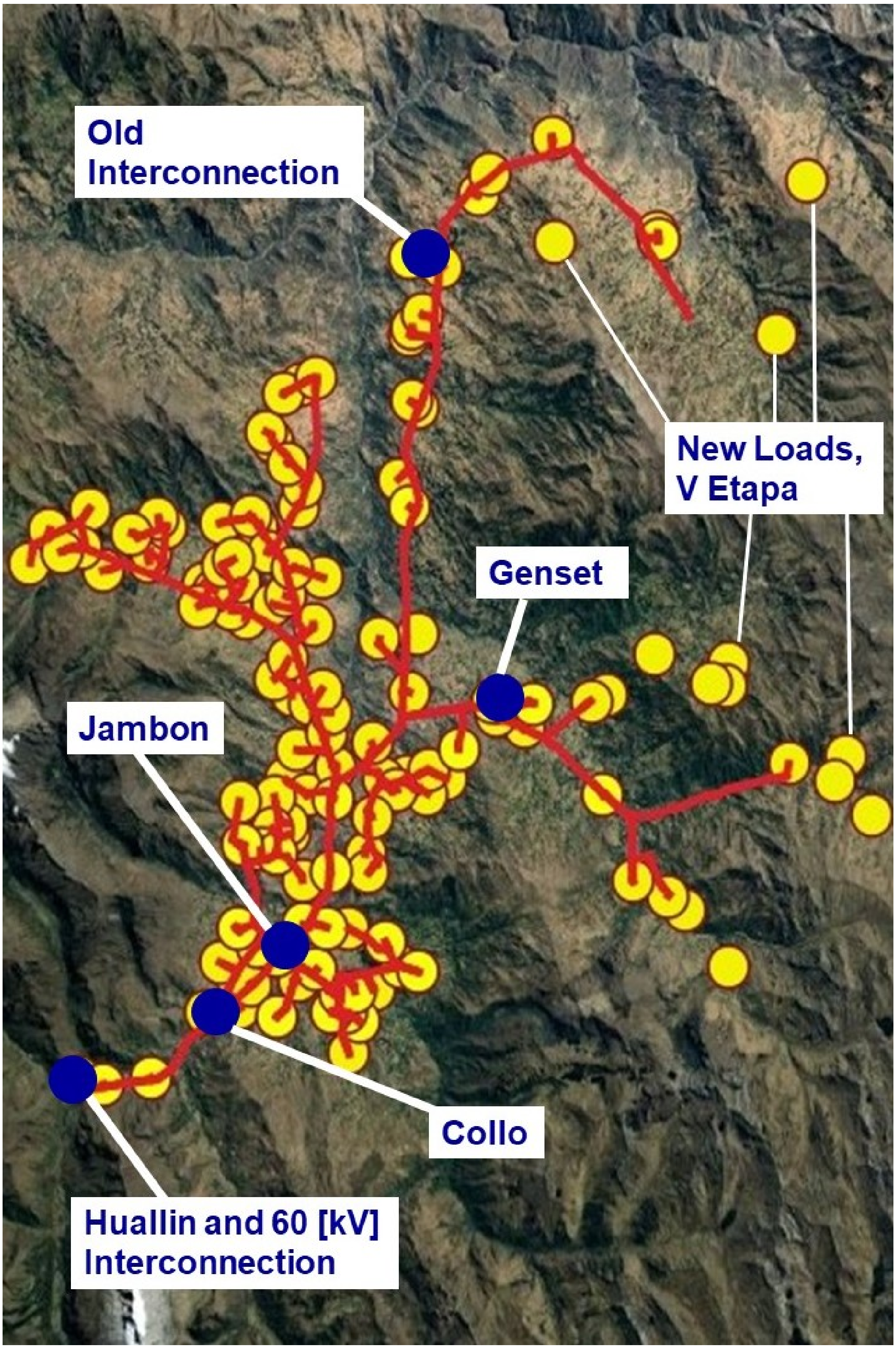
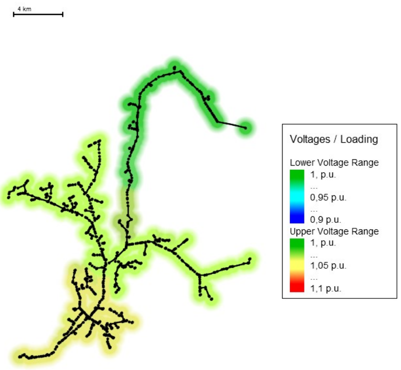
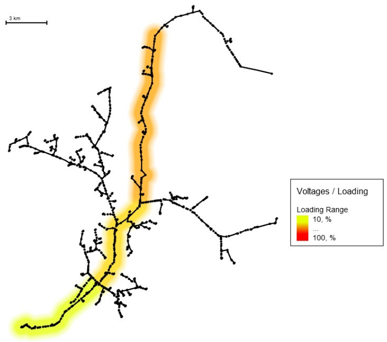


Figure 6.1: Grid structure, power plants and future interventions.



(a) Base Case Voltage level



(b) Base Case Line loading

maximum load, are loaded less than 1% of theirs line nominal current. Only 6% of lines have a loading factor higher than 30%.

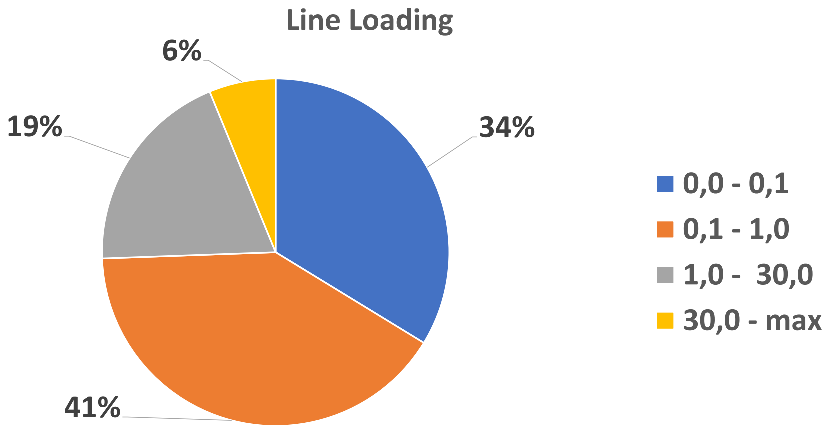


Figure 6.2: Line Loading Rate [%], Base Case

6.1.3 Losses

Overall line losses is 162.2 [kW]. Also losses are more concentrated in the grid backbone, in this case it is more evident observing figure 6.3. There are 152 lines that loose more than 1 [kW/km], but they represent 99.88% of overall losses. They are located between Huallin power plant and Pommalucay interconnection. The majority of lines (77%), due to such a low current passing through them, have a losses less than 0.01 [kW/km]. It is impressive to see that any line have a loss concentration value between 0.01 and 0.1 [kW/km].

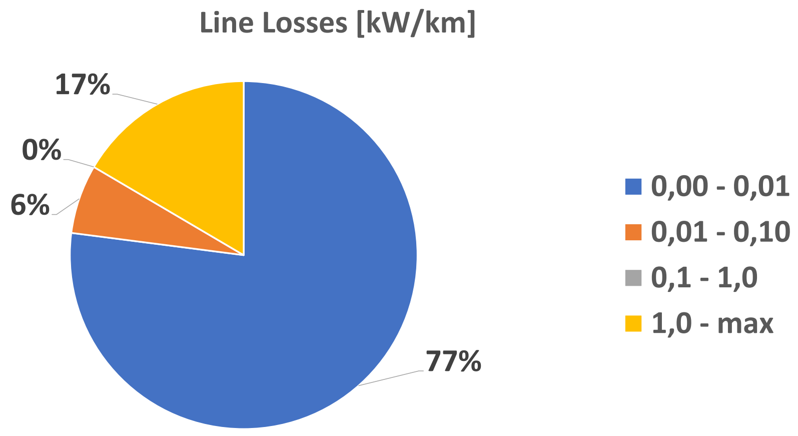


Figure 6.3: Line Losses, Base Case [%]

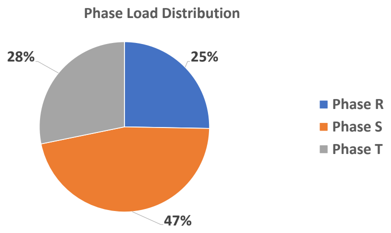
Therefore also this data confirm that the power flowing from Huallin to Pommalucay generates the main grid problems, in this stretch are dissipated 99.9% of losses.

6.1.4 Phases Balance

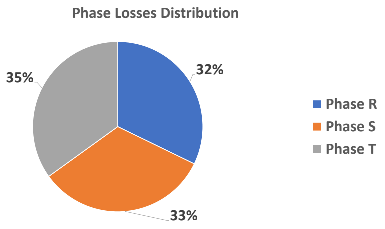
We already spoken about differences on Voltage Magnitude. In addition, taking a look to the following figure (a) we can see that a great unbalance is present in load. Almost half load demand is consumed by **phase S**. Losses instead are well balanced, figure (b).

6.2 60 [kV] Interconnection

In this case study we are going to simulate the grid after that 60 [kV] interconnection between Huallin power plant and the national grid is realized. The results obtained by this simulation are then compared with **Base Case** ones. As we previously said in chapter 4 Huallin power plant has 4 [MW] of nominal power but, due to distribution grid technical limits it is not possible to generate the maximum power and and only 1.1 [MW] are used.



(a) Load demand distribution between phases



(b) Losses distribution between phases

Connecting Huallin power plant with the national grid will have two main type of advantages:

Economical Santa Lucia will generate and sell more electricity. An increase in power production will increase revenues of almost 232.000 € per year. This data was calculated considering 8.000 operation equivalent hours and a marginal pool price of 10 [€/MWh] [19]

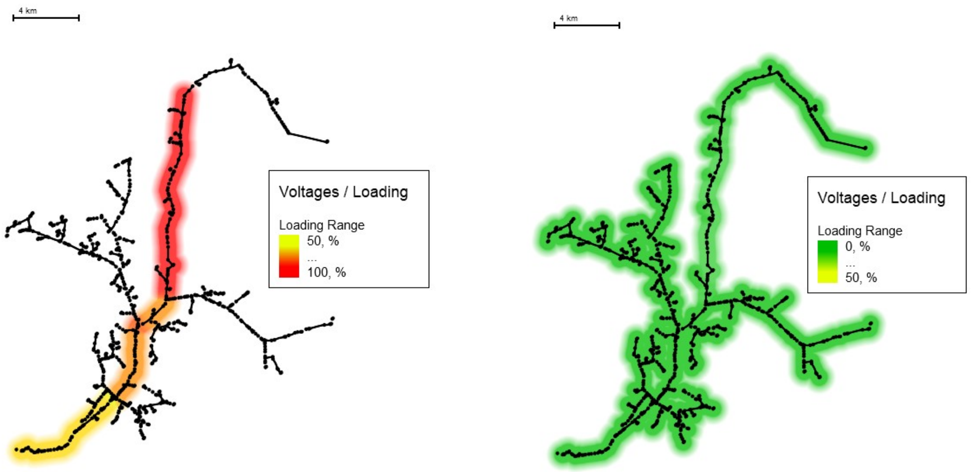
Operational As we explained in the previous case study 6.1 a huge amount of losses and issues are due to the necessity of make Huallin power passing through the whole grid before arriving in Pommalucay. Eliminating this excessive flux we will lighten grid line loading.

As we can see in figure (c) if the whole power of Huallin was used, line loading between Huallin and Pommalucay would be excessive. In a contest like this, the maximum line loading would be 109.9% and even 54 lines would exceed 100%. Line losses would be 1.39 [MW], so almost 30% of electricity production. In addition voltage magnitudes, especially in the areas near power plants, would exceed 1.1 [p.u.] inducing many complex technical problems.

Instead, as we can see in figure (d), deviating Huallin power flows from the grid will result in a grid lightening.

6.2.1 Voltage Magnitude

As we can see from the comparison between figure (e) and (f) bus grid bus voltage magnitude will slightly benefit from Huallin Interconnection. The maximum line to line voltage is 1.036 [p.u.] while the minimum is 0.999 [p.u.]. In table 6.6 we can better read maximum and minimum bus voltage magnitude [p.u.]. Also in this case study only a small fraction of buses (56)



(c) Loading at maximum operation limit (d) Loading with HV interconnection

are under the unit value, in particular those on valley bottom. The maximum value, with respect to Base Case is decreased from 1.046 to 1.036.

	Phase R	Phase S	Phase T
Max Magnitude	1,031458	1,03294	1,035711
Min Magnitude	0,999707	0,999698	0,999708

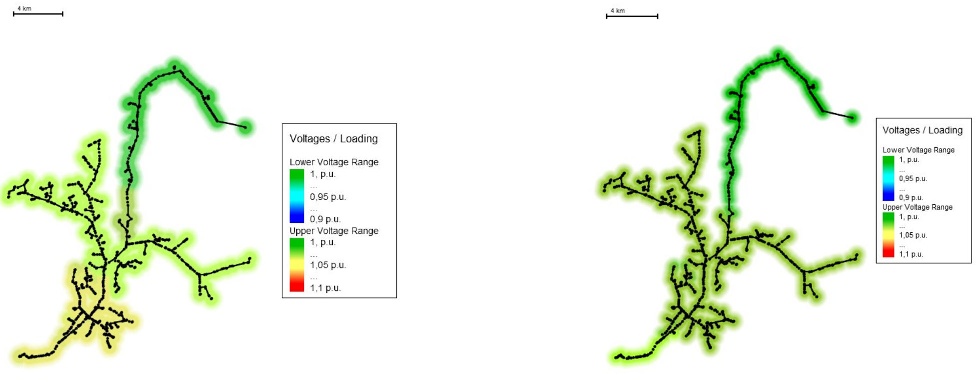
Table 6.6: Max and Min bus voltage magnitude [p.u.], Huallin Case

6.2.2 Line Loading

We previously discussed about hypothetical decreasing in line loading, in case of full Huallin power loading. A comparison between Base Case and the present one is made between figures (a) and (b) of the previous page. Also in this case the maximum loading (16.3%) occurs near San Luis interconnection, when the line cable decrease its cross area. Another important data to be show is that only 202 lines have an loading higher than 1 [%].

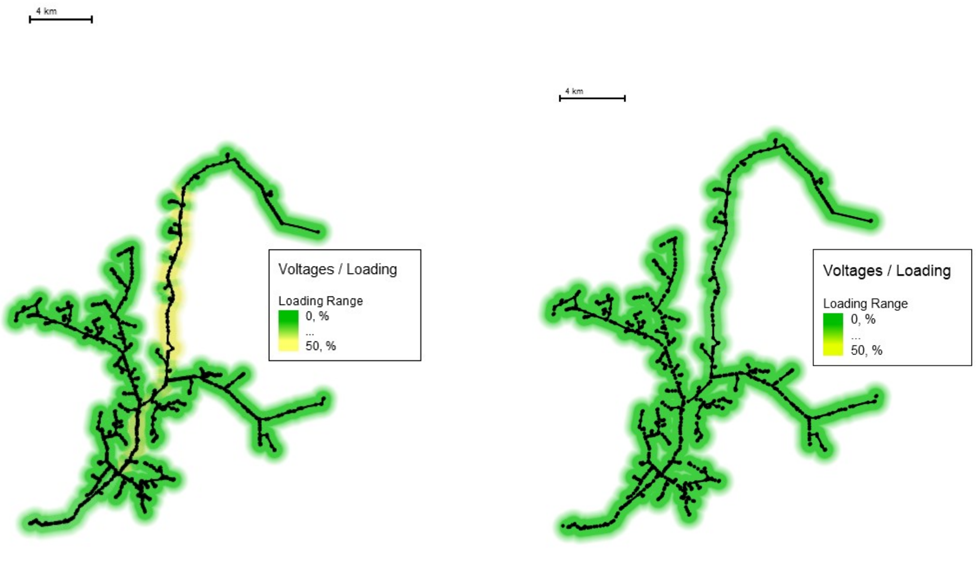
6.2.3 Losses

Since losses are directly correlated to line loading also these values decrease. Total grid losses decreases from 162.2 to 25.5 [kW]. As we can see from the comparison between figures (c) and (d) also the distribution of losses



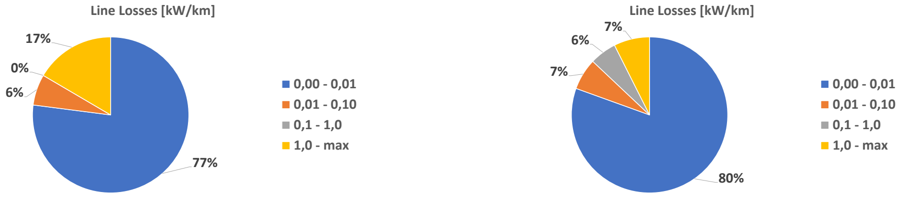
(e) Voltage magnitude before 60 kV

(f) Voltage magnitude after 60 kV



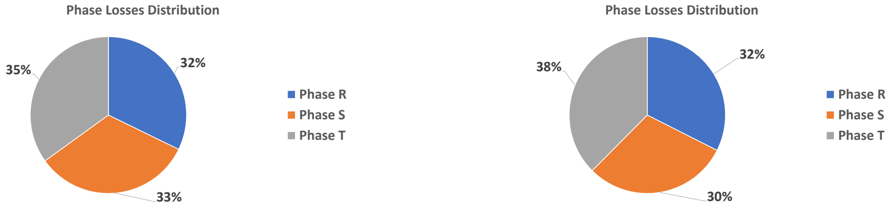
(g) Loading in Base Case

(h) Loading with HV interconnection



(i) Losses distribution in Base Case

(j) Losses distribution with HV interconnection



(k) Phase Losses Distribution in Base Case

(l) Losses distribution with HV interconnection

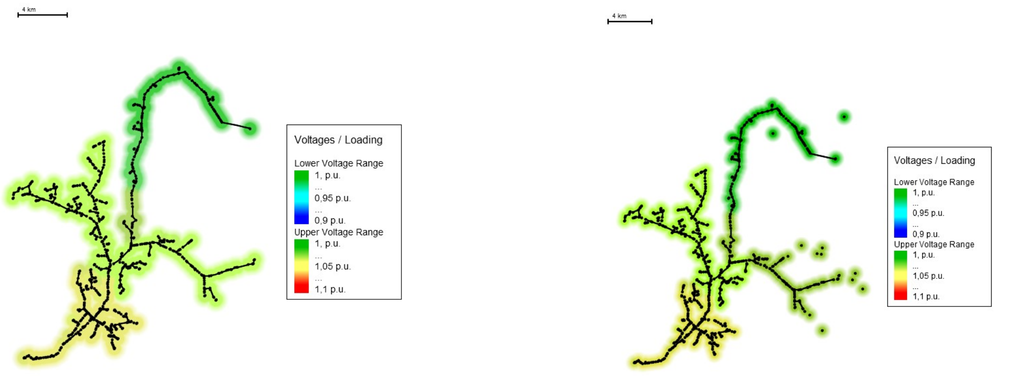
concentration changes. An higher rate of lines have average loss below 0.01 [kW/km] while less lines have it above 1.0 [kW/km].

6.2.4 Phase Balance

We already said that from a comparison between **Base Case** and **60 [kV] Interconnection** the distribution in voltage magnitude is more constant. Losses distribution instead increase disparities between phases, in this simulation **Phase T** have an higher rate of losses (38%), while **Phase S** the lowest (30%).

6.3 V ETAPA

In this case study we are going to simulate Chacas distribution grid once the new villages planned to be connected by V Etapa will be installed. This simulation is thus compared with **Base Case**, but since we don't know when 60 [kV] interconnection will be realized, the present case is performed without such high voltage interconnection. It is an hypothetical simulation since load will not be connected all in a once but step by step. Right now some of these



(m) Voltage magnitude before V Etapa load connection.

(n) Voltage magnitude after V Etapa load connection.

connections have already been installed while others not.

July 11 2018 at 20:00	Load Demand [kW]
Huallin	50
Pampash	57
Chacas	324
San Luis	562
Global	993

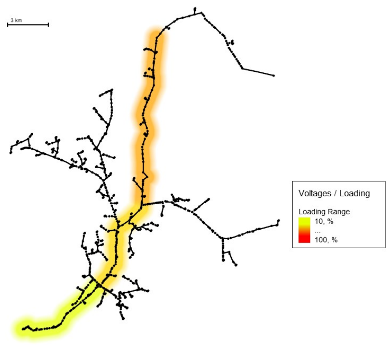
Table 6.7: Zone and global load demand on July 11 2018 at 20:00.

As we already explained in table 4.9 with the implementation of V Etapa 16 new transformers will be connected to the grid. All of these transformers, and the relative loads, are located in San Luis area. Therefore we expect to observe an increase in voltage magnitude and losses in this grid branch. The expected maximum load demand is depicted in table 6.7.

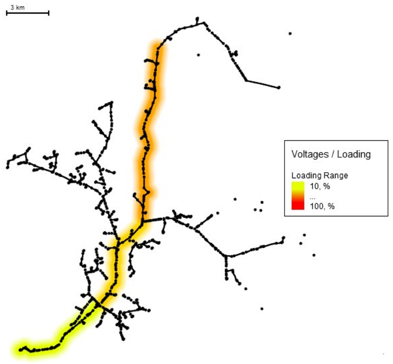
As we can see in table 4.9 new loads are mainly connected to **phase R** (21.8 [kW]), only five are connected to **phase S** (15.6 [kW]) and two to **phase T** (5.9 [kW]). This choice is coherent considering that phase R load is the most under exploited.

6.3.1 Voltage Magnitude

As we expected, load growth due to V Etapa (4.5%) will not deeply modify bus voltage magnitude. The maximum value of line to line voltage magnitude



(o) Line loading before V Etapa load connection.



(p) Line loading after V Etapa load connection.

is 1.0459, so almost equal to 1.0458, the one of Base Case. The minimum is still 0.999. In table 6.8 are printed maximum and minimum voltage magnitude values.

	Phase R	Phase S	Phase T
Max Magnitude	1,04312	1,04318	1,045907
Min Magnitude	0,999706	0,999699	0,999699

Table 6.8: Max and Min bus voltage magnitude [p.u.], V Etapa Case

As we can see from a comparison between figures (m) and (n) voltage magnitude along the grid is almost unchanged. New load can be observed on the upper and right grid areas.

6.3.2 Line Loading

Also line loading results are very similar to Base Case ones. Maximum value is 38.29 [%], a small increase with respect to 38.22 [%]. Average value increase from 4.936 [%] to 4.944 [%]. The minimum loading is not changed, since this specific line is not interested by the new loads.

6.3.3 Losses

Also line losses will slightly increase. From 162.2 [kW] to 163.9 [kW]. As we saw before in table 4.9, new load are mainly connected to **phase R** and **phase S**. This increase in phase load is observed also for each single phases in

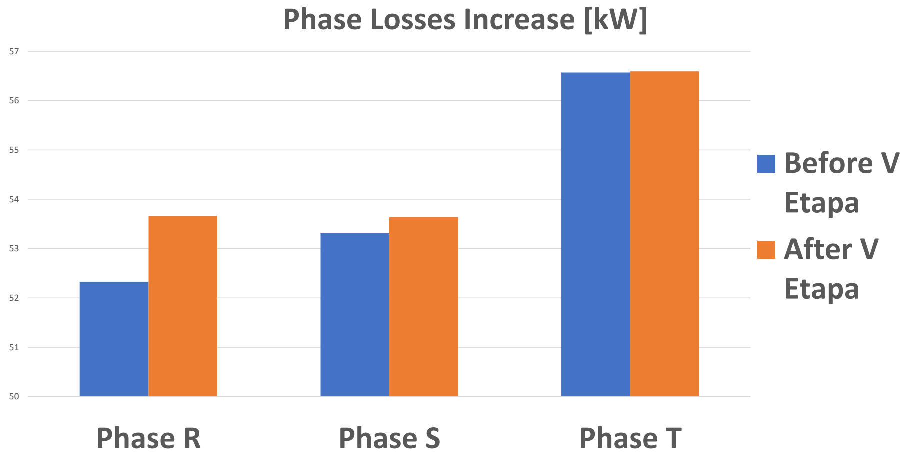


Figure 6.4: Increase in phase losses after V Etapa.

figure 6.4. We can see that **phase R** increases his losses from 52.33 to 53.66 [kW], **phase S** from 53.31 to 53.63 [kW] while losses in **phase T** increases only from 56.57 to 56.59 [kW].

6.3.4 Phase Balance

We already spoken about phase balance in term of voltage magnitude and losses increase. No further considerations are needed.

6.4 Load Increasing

In this Case Study we are going to simulate how Chacas distribution grid behaves with load increasing in the next years. Each year we expected a 5% increase in Maximum Load Demand, we calculated load of each transformer directly in **PowerFactory**. The periods we choose are 5, 10, 15 and 20 years up to now. After each simulations we reported values regarding Load Flows, Losses and Line Loading.

Before each simulation we need to calculate which combination of power plant can be used as generators. It is important to highlight that Huallin plant, once connected to the national grid can be considered separated from Chacas grid and that **Eilhicha** cannot sell electricity to Hidrandina, so Pommalucay interconnection will be used only to balancing the grid. Therefore we need to choose how many plants, and combination of them must be used to met load demand.

6.4.1 Combinations of Power Plants

From table 6.9 we can evaluate nominal power of generators and their possible combinations. The maximum power that can be generated by **Eilhicha** is 2.37 [MW], in the following list we will not take into account losses, since they are a small fraction of power generation.

Plant	Power
Collo	0,92
Jambon A	1
Jambon B	0,45
C-JA	1,92
C-JB	1,37
JA-JB	1,45
C-JA-JB	2,37

Table 6.9: Combination of plant and the relative active power supplied to the grid

This simulation will also be useful to have an estimation by when to install ColloB, the new Hydropower plant planned to be connected in the long term. As we said before in section 5.1.4, at the present time it is not possible to evaluate time and technical information about this plant.

Now As we can see in table 6.10 at the present state of the grid all plant combinations can be used to met Load demand. In normal operation **Eilhicha** use only two machines at the same time, both in a reduced capacity. Theoretically for the majority of the year only on power plant should be necessary but for security reasons two plants are turned on.

5 Years Load demand will increase only until 1.37 [MW]. Only Collo-JambonA combination is not anymore sufficient to met load demand. This fact means that in peak load JambonB must be operative, without this plant **Eilhicha** would be forces to use some power from Huallin power plant or from Hidrandina grid.

10 Years After 10 years peak load is expected to increase since 1.75 [MW], therefore the only allowed combinations are Collo-JambonA and the one using all plant.

15 Years The only possibility to met load peak demand after 15 years will be to switch on all plant. Also in this case, installed capacity should be theoretically enough but, considering losses and a range of unpredictability, we can say that the actual installed capacity will not be sufficient. ColloB power plant should be installed before this year.

20 Years After 20 years peak load is expected to be much beyond installed capacity. A new power plant, or more, must be installed in the grid.

	Peak Load	Power Plant Combination			
		C-JA	C-JB	JA-JB	C-JA-JB
Now	1,07	O	O	O	O
5 Years	1,37	O	X	O	O
10 Years	1,75	O	X	X	O
15 Years	2,22	X	X	X	O
20 Years	2,84	X	X	X	X

Table 6.10: Power plant combinations available (0) or not (X), to met Peak Load Demand in the next years.

Regardless which plant combination is used to met load demand, following results will be obtained with and adequate generation, enough to meet load demand without exporting to Pommalucay. For the last simulation (20 Years) all plants are switched on and a flowing import is forced.

6.4.2 Results

As previously said, the main results we are going to see are Line Loading and Losses increase in the next years. As we can see in table 6.11 and in figures 6.5 and 6.6, the average of line loading is expected to increase from 2.00 to 4.84%, the maximum value from 16.35% to 39.07%. Therefore we can conclude that also on the long period line loading will not be the main priority of the grid. Also during the maximum peak load line loading is expected not to reach 40%.

Losses will increase from 25.5 to 159.6 [kW]. Considering also the increase in load demand we observe an increase in rate between losses and peak loading. Actually losses (25.5 [kW]) represent 2.4% of peak load (1070 [kW]), while in 20 years from now this rate will increase till 5.6%. Nevertheless such increase we can conclude that losses also in the future will represent a small fraction of loading and will not be an important issue for grid operation.

	Peak Load [MW]	Average Loading [\%]	Max Loading [\%]	Losses [kW]	Losses over Peak Load
Now	1,07	2,00	16,35	25,5	2,4%
5 Years	1,37	2,37	18,90	34,6	2,5%
10 Years	1,75	3,08	25,67	62,5	3,6%
15 Years	2,22	4,14	35,44	120,6	5,4%
20 Years	2,84	4,84	39,07	159,6	5,6%

Table 6.11: future projections of Peak Load, Line Loading and Losses.

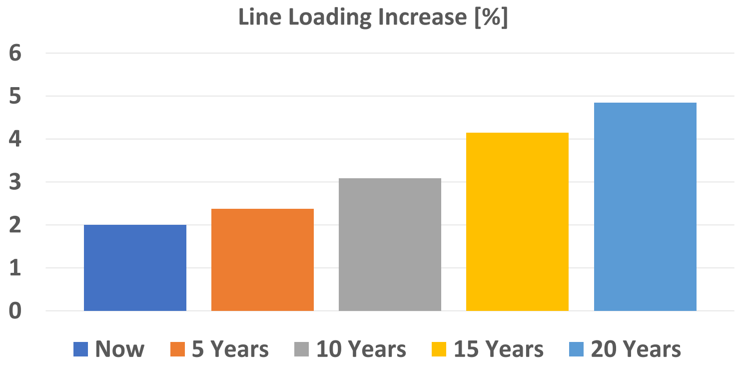


Figure 6.5: Increase in Line Loading [%] expected in the next years.

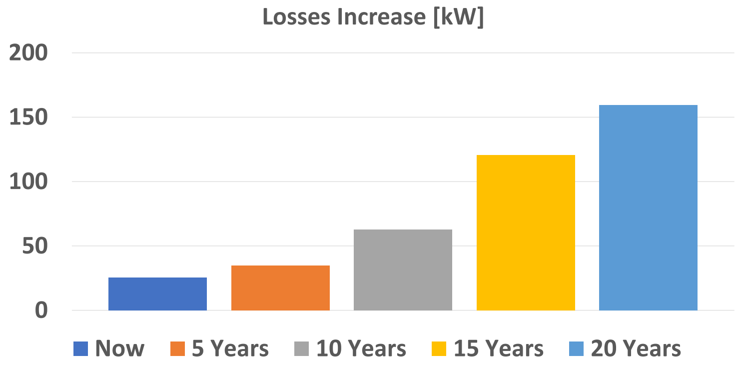


Figure 6.6: Increase in Losses [kW] expected in the next years

Conclusions

Around the world 840 million people are living without access to electricity while 1.0 billion people are connected to unreliable electric grid. To fill this gap mini grid have become one of the best option, due to recent improvement in technologies, falling of costs and exploitation of renewable energy sources. One of the main challenge developing countries mini grid system operators are facing, is the growing in operation complexity and need of planning with the increasing of grid size and number of costumers. To accomplish this purpose, modelling the electric grid through a power system analysis software is considered the best solution. With such a model it is possible to evaluate grid weakness, electric losses, excessive line loading, faults simulations and above all it is a useful tool that give to the system operator tangible information for long period planning.

Since an electric grid model is not normally available to developing country's rural system operators, the aim of this thesis was to create a semi-automatic procedure that, starting from an existing GIS grid model generates an electric one. The procedure developed, HUGO, take as input data the grid structure, in GPKG files, and a set of (P,Q) couples for each MV-LV transformers, so information about load demand. These data are imported in a Python ambient that generates an output file, tailored to a specific system analysis software, that once imported inside this software creates the electric grid model expected. The main difficulty of this procedure is writing the Python code; since each grid have its own peculiarities and since grid information are stored in GIS tables with names chosen case by case, the code must be adjusted in each specific case. Nevertheless, once the grid is created, every time an alteration is implemented in the real grid (so on the GIS model) it is easily possible to execute the code and update the electric grid model.

In order to validate HUGO procedure it was tested on a real life case study: Chacas electric grid. It is a mini grid located in a Peruvian mountain rural area, the system operator is Eilhicha. The procedure was applied and an electric grid model is generated, the software used to import and analyze this model is PowerFactory (a well known commercial tool).

Once PowerFactory grid model was generated some simulations were performed: An initial Base case load flow was used to evaluate if the model actually correspond to the real model, in addition this simulation was used to suggest to the system operator the electric state of the grid. Since in the next years a high voltage interconnection is expected to be performed to the main power plant, this grid change is added to the model and the impact on the grid is observed. The main result of this simulation is a lightening on line loading. A third simulation evaluates the impact of the connection of new villages, expected to be implemented in the next few months. The resulting grid condition is a little more loaded but basically it is close to the base case, impact on the grid is minimal. The last simulation, incorporating also the previous two grid improvements, is a load prevision during the next 20 years, with 5 years time steps and a 5% rate of in peak load increasing. The result of this last simulation suggests the system operators that the main issue, it need to face in the long term period, are not line losses but rather lack of power capacity to cover peak load demand.

List of Figures

List of Tables

Acronyms

AC	Alternate Current
CAD	Computer Aided Design
COES	Comité de Operación Económica del Sistema Int. Nac.
DC	Direct Current
DG	Distributed Generators
DGER	Direction de Electrificacion Rural
DGS	DigSilent Data Exchange Interface
EDI	Energy Development Index
FID	Foreign Identification
FOSE	Fondo de Compensación Social Eléctrica
GIS	Geographic Information Systems
GNI	Gross National Income
GPKG	Geopackaging Format
HDI	Human Development Index
HV	High Voltage
IEA	International Energy Agency
LCOE	Levelized Cost of Electricity
LV	Low Voltage
MINEM	Ministerio de Energìa y Minas
MV	Medium Voltage
NGO	Non-Governative Organizations
OECD	The Org.n for Economic Co-operation and Development
OMG	Operazione Matogrosso
PNER	Plan Nacional de Electrification Rural
PPP	Purchased Power Parity
PSAS	Power System Analysis Software
PV	Photovoltaic
RES	Erenewable Energy Sources
RISE	The World Bank's Regulatory Indicators for SDG's
SDG7	Sustainable Development Goal n 7
TPES	Total Primary Energy Supply
UTM	Universal Transverse Mercator
VNR	Codigos Valor Nuevo de Reemplazo

Bibliography

[1] ESMAP. *Tracking SDG7 Report*, 2019.

[2] International Energy Agency. *World Energy Outlook*. IEA, 2012.

[3] World bank open data, 2017. https://data.worldbank.org/indicator/EG.ELC.ACCS.ZS.

[4] World Bank and International Energy Agency. *Global Tracking Framework Report*, 2015.

[5] ESMAP. *Mini Grids for Half a Billion People, Market Outlook and Handbook for Decision Makers*, 2019.

[6] World Bank. *World Bank's Regulatory Indicators for Sustainable Energy (RISE)*, 2018.

[7] Josè Luis Rueda J.M. Roldan-Fernandez, Francisco M.Gonzalez-Longatt and H.Verdejo. *Modelling of Transmission Systems Under Unsymmetrical Conditions and Contingency Analysis Using DIgSILENT PowerFactory*. Springer, 2014.

[8] Statistics data, 2017. https://www.iea.org/statistics/?country=PERisISO=true.

[9] Country profile: Peru, 2017. https://www.hydropower.org/country-profiles/peru.

[10] Mimem, peru, 2017.

[11] *Learning From Developing Country Power Market Experience: The Case of Peru*, 2019. Policy Research Working Paper.

[12] William D. Stevenson John J. Grainger and Gary W. Chang. *Power System Analysis*. McGraw-Hill, 2016.

[13] Wikipedia, 2019. https://en.wikipedia.org/wiki/MainPage.

[14] melppmene. Python code for utm-latlong transposition, 2018.

[15] *Plan Nacional de Electrificacion Rural (PNER)*, 2012.

[16] Fondo de inclusion social energetico, 2016. http://www.fise.gob.pe/que-es-fise.html.

[17] *Censos Nacionales 2017*, 2017.

[18] OSINERGIM. *Guia de Elaboracion del Valor Nuevo de Remplazo (VNR) de las Intalaciones de Distribucion Electrica*, 2004.

[19] Comité de operación económica del sistema interconectado nacional. http://www.coes.org.pe/Portal/.

Appendices

```
import numpy as np
import openpyxl
import xlrd
import pandas as pd
from pandas import DataFrame
import geopandas as gpd
import time
from openpyxl import load_workbook
import shutil
import os
import math

start_time = time.time()
print('Inizio')
file_import = 'C:/Users/Toto-/Dropbox/Poli/Tesi/Passerella/
    Letture_Trasformatori.xlsx'
LDSE = pd.read_excel(file_import, sheet_name='SED')
Year_Dati = pd.read_excel(file_import, sheet_name='Year')
LDSE_SED = LDSE['SED']._values
LDSE_power = LDSE['Nominal Power [kW]']._values
LDSE_zona = LDSE['Zona']._values
LDSE_active = LDSE['Active']._values
LDSE_3ETAPA = LDSE['III Etapa prevision [kW]']._values
LDSE_power_active_Huallin = []
j=0
for e in LDSE_power:
    if LDSE_active[j]=='Y' and LDSE_zona[j]=='H':
        LDSE_power_active_Huallin.append(e)
    else:
        LDSE_power_active_Huallin.append(0)
    j = j + 1
LDSE_power_active_Pampash = []
j=0
for e in LDSE_power:
    if LDSE_active[j] == 'Y' and LDSE_zona[j]=='P':
        LDSE_power_active_Pampash.append(e)
    else:
        LDSE_power_active_Pampash.append(0)
    j = j + 1
LDSE_power_active_Chacas = []
j=0
for e in LDSE_power:
    if LDSE_active[j] == 'Y' and LDSE_zona[j]=='C':
        LDSE_power_active_Chacas.append(e)
    else:
        LDSE_power_active_Chacas.append(0)
    j = j + 1
LDSE_power_active_SanLuis = []
j=0
```

```
for e in LDSE_power:
    if LDSE_active[j] == ’Y’ and LDSE_zona[j]==’S’:
        LDSE_power_active_SanLuis.append(e)
    else:
        LDSE_power_active_SanLuis.append(0)
    j = j + 1
LDSE_power_unit_Huallin=LDSE_power_active_Huallin/sum(
   LDSE_power_active_Huallin)
LDSE_power_unit_Pampash=LDSE_power_active_Pampash/sum(
   LDSE_power_active_Pampash)
LDSE_power_unit_Chacas=LDSE_power_active_Chacas/sum(
   LDSE_power_active_Chacas)
LDSE_power_unit_SanLuis=LDSE_power_active_SanLuis/sum(
   LDSE_power_active_SanLuis)
#---------------Potenze Zone Excel-----------
Year_Load_Huallin = Year_Dati[’Huallin’]._values
Year_Load_Pampash = Year_Dati[’Pampash’]._values
Year_Load_Chacas = Year_Dati[’Chacas’]._values
Year_Load_SanLuis = Year_Dati[’San Luis’]._values
Year_Load_Huallin_low = Year_Load_Huallin[np.int(np.round(len
   (Year_Load_Huallin)/50))]
Year_Load_Pampash_low = Year_Load_Pampash[np.int(np.round(len
   (Year_Load_Pampash)/50))]
Year_Load_Chacas_low = Year_Load_Chacas[np.int(np.round(len(
   Year_Load_Chacas)/50))]
Year_Load_SanLuis_low = Year_Load_SanLuis[np.int(np.round(len
   (Year_Load_SanLuis)/50))]
Year_Load_Huallin_mid = Year_Load_Huallin[np.int(np.round(len
   (Year_Load_Huallin)/2))]
Year_Load_Pampash_mid = Year_Load_Pampash[np.int(np.round(len
   (Year_Load_Pampash)/2))]
Year_Load_Chacas_mid = Year_Load_Chacas[np.int(np.round(len(
   Year_Load_Chacas)/2))]
Year_Load_SanLuis_mid = Year_Load_SanLuis[np.int(np.round(len
   (Year_Load_SanLuis)/2))]
Year_Load_Huallin_top = Year_Load_Huallin[len(
   Year_Load_Huallin)-1]
Year_Load_Pampash_top = Year_Load_Pampash[len(
   Year_Load_Pampash)-1]
Year_Load_Chacas_top = Year_Load_Chacas[len(Year_Load_Chacas)
   -1]
Year_Load_SanLuis_top = Year_Load_SanLuis[len(
   Year_Load_SanLuis)-1]
#------------------Elaboro Dati-----
Load_Huallin_low=Year_Load_Huallin_low*
   LDSE_power_unit_Huallin
Load_Pampash_low=Year_Load_Pampash_low*
   LDSE_power_unit_Pampash
Load_Chacas_low=Year_Load_Chacas_low*LDSE_power_unit_Chacas
```

```
Load_SanLuis_low=Year_Load_SanLuis_low*LDSE_power_unit_SanLuis
Load_Huallin_mid=Year_Load_Huallin_mid*LDSE_power_unit_Huallin
Load_Pampash_mid=Year_Load_Pampash_mid*LDSE_power_unit_Pampash
Load_Chacas_mid=Year_Load_Chacas_mid*LDSE_power_unit_Chacas
Load_SanLuis_mid=Year_Load_SanLuis_mid*LDSE_power_unit_SanLuis
Load_Huallin_top=Year_Load_Huallin_top*LDSE_power_unit_Huallin
Load_Pampash_top=Year_Load_Pampash_top*LDSE_power_unit_Pampash
Load_Chacas_top=Year_Load_Chacas_top*LDSE_power_unit_Chacas
Load_SanLuis_top=Year_Load_SanLuis_top*LDSE_power_unit_SanLuis
Load_low=Load_Huallin_low+Load_Pampash_low+Load_Chacas_low+Load_SanLuis_low+LDSE_3ETAPA
Load_mid=Load_Huallin_mid+Load_Pampash_mid+Load_Chacas_mid+Load_SanLuis_mid+LDSE_3ETAPA
Load_top=Load_Huallin_top+Load_Pampash_top+Load_Chacas_top+Load_SanLuis_top+LDSE_3ETAPA
# -----------IMPORT of vectors from QGIS----------
SED = gpd.read_file(r"C:\Users\Toto-\Documents\Pyton\Progetti\GIS\GIS Layers\SED.gpkg")
MTTRAMO = gpd.read_file(r"C:\Users\Toto-\Documents\Pyton\Progetti\GIS\GIS Layers\MTTRAMO.gpkg")
MTSALIDA = gpd.read_file(r"C:\Users\Toto-\Documents\Pyton\Progetti\GIS\GIS Layers\MTSALIDA.gpkg")
MTNODO = gpd.read_file(r"C:\Users\Toto-\Documents\Pyton\Progetti\GIS\GIS Layers\MTNODO.gpkg")
MTPOSTE = gpd.read_file(r"C:\Users\Toto-\Documents\Pyton\Progetti\GIS\GIS Layers\MTPOSTE.gpkg")
# ------------------CREATING useful VECTORS------------
# POSTE
Poste_LocName = MTPOSTE['CodEstruct']._values
Poste_CodTramoMT = MTPOSTE['CodTramoMT']._values
Poste_north_utm = MTPOSTE['geometry'].y._values
Poste_east_utm = MTPOSTE['geometry'].x._values
Poste_alti = MTPOSTE['Position Z']._values
Poste_num = len(Poste_LocName)
# LOAD
Load_LocName = SED['CodSed']._values
Load_Locality = SED['Localidad']._values
Load_north_utm = SED['geometry'].y._values
Load_east_utm = SED['geometry'].x._values
Load_CodTenNom1 = SED['CodTenNom1']._values
Load_CodTenNom2 = SED['CodTenNom2']._values
Load_PotInstal = SED['PotInstal'].values
```

```python
Load_FasiOut = SED['FasiOut'].values
Tr2_LocName = Load_LocName
Tr2_PotInstal = SED['PotInstal']._values
Tr2_Active = SED['Active']._values
Tr2_CodTenNom1 = Load_CodTenNom1
Tr2_CodTenNom2 = Load_CodTenNom2
Tr2_CodNodo = SED['CodNodo']._values
ElmLoad_FID = Load_LocName
Load_num = len(Load_LocName)
# TRAMO
Tramo_LocName = MTTRAMO['CodTramoMT']._values
Tramo_CodNorma = MTTRAMO['CodNorma']._values
Tramo_CodTenNomi = MTTRAMO['CodTenNomi']._values
Tramo_CodNormaNe = MTTRAMO['CodNormaNe']._values
Tramo_LocName_Pater = MTTRAMO['CodTramo_1']._values
Tramo_Length = MTTRAMO['Length']._values
Tramo_num = len(Tramo_LocName)
# NODO
Nodo_LocName=MTNODO['CodNodoMT']._values
Nodo_CodTramoMT=MTNODO['CodTramoMT']._values
Nodo_north_utm = MTNODO['Position Y']._values
Nodo_east_utm = MTNODO['Position X']._values
Nodo_alti = MTNODO['Position Z']._values
Nodo_num=len(Nodo_LocName)
# -------------from UTM to Lat_Long----------
def utmToLatLng(zone, easting, northing, northernHemisphere=True):
    if not northernHemisphere:
        northing = 10000000 - northing
    a = 6378137
    e = 0.081819191
    e1sq = 0.006739497
    k0 = 0.9996
    arc = northing / k0
    mu = arc / (a * (1 - math.pow(e, 2) / 4.0 - 3 * math.pow(e, 4) / 64.0 - 5 * math.pow(e, 6) / 256.0))
    ei = (1 - math.pow((1 - e * e), (1 / 2.0))) / (1 + math.pow((1 - e * e), (1 / 2.0)))
    ca = 3 * ei / 2 - 27 * math.pow(ei, 3) / 32.0
    cb = 21 * math.pow(ei, 2) / 16 - 55 * math.pow(ei, 4) / 32
    cc = 151 * math.pow(ei, 3) / 96
    cd = 1097 * math.pow(ei, 4) / 512
    phi1 = mu + ca * math.sin(2 * mu) + cb * math.sin(4 * mu) + cc * math.sin(6 * mu) + cd * math.sin(8 * mu)
    n0 = a / math.pow((1 - math.pow((e * math.sin(phi1)), 2)), (1 / 2.0))
    r0 = a * (1 - e * e) / math.pow((1 - math.pow((e * math.sin(phi1)), 2)), (3 / 2.0))
```

```
    fact1 = n0 * math.tan(phi1) / r0
    _a1 = 500000 - easting
    dd0 = _a1 / (n0 * k0)
    fact2 = dd0 * dd0 / 2
    t0 = math.pow(math.tan(phi1), 2)
    Q0 = e1sq * math.pow(math.cos(phi1), 2)
    fact3 = (5 + 3 * t0 + 10 * Q0 - 4 * Q0 * Q0 - 9 * e1sq) * math.pow(dd0, 4) / 24
    fact4 = (61 + 90 * t0 + 298 * Q0 + 45 * t0 * t0 - 252 * e1sq - 3 * Q0 * Q0) * math.pow(dd0, 6) / 720
    lof1 = _a1 / (n0 * k0)
    lof2 = (1 + 2 * t0 + Q0) * math.pow(dd0, 3) / 6.0
    lof3 = (5 - 2 * Q0 + 28 * t0 - 3 * math.pow(Q0, 2) + 8 * e1sq + 24 * math.pow(t0, 2)) * math.pow(dd0, 5) / 120
    _a2 = (lof1 - lof2 + lof3) / math.cos(phi1)
    _a3 = _a2 * 180 / math.pi
    latitude = 180 * (phi1 - fact1 * (fact2 + fact3 + fact4)) / math.pi
    if not northernHemisphere:
        latitude = -latitude
    longitude = ((zone > 0) and (6 * zone - 183.0) or 3.0) - _a3
    return (latitude, longitude)
j = 0
Poste_lat = Poste_north_utm
Poste_long = Poste_east_utm
for e in Poste_LocName:
    (Poste_lat[j], Poste_long[j]) = utmToLatLng(18, Poste_east_utm[j], Poste_north_utm[j], northernHemisphere=False)
    j = j + 1
j = 0
Load_lat = Load_north_utm
Load_long = Load_east_utm
for e in Load_LocName:
    (Load_lat[j], Load_long[j]) = utmToLatLng(18, Load_east_utm[j], Load_north_utm[j], northernHemisphere=False)
    j = j + 1
Tr2_lat = Load_lat
Tr2_long = Load_long
j = 0
Nodo_lat = Nodo_north_utm
Nodo_long = Nodo_east_utm
for e in Nodo_LocName:
    (Nodo_lat[j], Nodo_long[j]) = utmToLatLng(18, Nodo_east_utm[j], Nodo_north_utm[j], northernHemisphere=False)
    j = j + 1
```

```
#----------FID check-----------------
#POSTE
fid_error = 0
b = 0
c = 0
for e in Poste_LocName:
    for f in Poste_LocName:
        if b == c:
            pass
        else:
            if e == f:
                print('Two POSTE have the same Loc_Name: ', e
   )
                fid_error = fid_error + 1
        c = c + 1
    b = b + 1
    c = 0
#NODE
b = 0
c = 0
for e in Nodo_LocName:
    for f in Nodo_LocName:
        if b == c:
            pass
        else:
            if e == f:
                print('Two NODO have the same Loc_Name: ', e)
                fid_error = fid_error + 1
        c = c + 1
    b = b + 1
    c = 0
#LOAD
b = 0
c = 0
for e in Load_LocName:
    for f in Load_LocName:
        if b == c:
            pass
        else:
            if e == f:
                print('Two LOAD have the same Loc_Name: ', e)
                fid_error = fid_error + 1
        c = c + 1
    b = b + 1
    c = 0
n = 0
b = 0
c = 0
Cime_LocName=[]
```

```
#TRAMO
for e in Tramo_LocName:
    for f in Tramo_LocName:
        if b == c:
            pass
        else:
            if e == f:
                print(’Two TRAMO have the same Loc_Name: ’, e)
                fid_error = fid_error + 1
            if e == Tramo_LocName_Pater[c]:
                n = 1
        c = c + 1
    if n == 0:
        Cime_LocName.append(e)
    b = b + 1
    c = 0
    n = 0

numcim = len(Cime_LocName)
b = 0
c = 0
xlserror=0
LDSE_power_order = []
if len(LDSE_SED) != len(Tr2_LocName):
    print(’QGIS_SED list and Excel_SED list have different lenght’)
    xlserror = xlserror + 1
else:
    for e in Tr2_LocName:
        for f in LDSE_SED:
            if e==f:
                b = b + 1
                LDSE_power_order.append(LDSE_power[c])
            c = c + 1
        if b == 0:
            print(e, ’ is not present in the Excel FILE’)
            xlserror = xlserror + 1
        elif b == 1:
            pass
        else:
            print(’There are ’, b, ’ SED in the Excel FILE with the same name: ’, e)
            xlserror = xlserror + 1
        b = 0
        c = 0
b = 0
c = 0
#-------GENERATING EXCELL FILE-------
```

```python
wb = openpyxl.Workbook()
del wb['Sheet']
General = wb.create_sheet('General')
ElmNet = wb.create_sheet('ElmNet')
ElmTerm = wb.create_sheet('ElmTerm')
ElmLod = wb.create_sheet('ElmLod')
ElmXnet = wb.create_sheet('ElmXnet')
ElmSym = wb.create_sheet('ElmSym')
ElmTr2 = wb.create_sheet('ElmTr2')
ElmLne = wb.create_sheet('ElmLne')
StaCubic = wb.create_sheet('StaCubic')
StaSwitch = wb.create_sheet('StaSwitch')
IntFolder = wb.create_sheet('IntFolder')
TypLod = wb.create_sheet('TypLod')
TypLne = wb.create_sheet('TypLne')
TypTr2 = wb.create_sheet('TypTr2')
TypSym = wb.create_sheet('TypSym')
TypGeo = wb.create_sheet('TypGeo')
TypCon = wb.create_sheet('TypCon')
#-----COMPILE PRE-IMPOSTED SHEETS--------
Mat_General = (
    ('FID', 'Descr', 'Val'),
    (1, 'Version', '6.0')
)
for row in Mat_General:
    General.append(row)
Mat_ElmNet = (
    ('FID', 'loc_name', 'frnom'),
    (9, 'Rete', 60)
)
for row in Mat_ElmNet:
    ElmNet.append(row)
Mat_ElmTerm = (
    ('FID', 'loc_name', 'fold_id', 'iUsage', 'uknom', 'GPSlat', 'GPSlon', 'phtech'),
)
for row in Mat_ElmTerm:
    ElmTerm.append(row)
Mat_ElmLod = (
    ('FID', 'loc_name', 'fold_id', 'typ_id', 'i_sym', 'plinir', 'qlinir', 'plinis', 'qlinis', 'plinit', 'qlinit', 'outserv', 'GPSlat', 'GPSlon'),
)
for row in Mat_ElmLod:
    ElmLod.append(row)
Mat_ElmXnet = (
    ('FID', 'loc_name', 'fold_id', 'pgini', 'GPSlat', 'GPSlon', 'iintgnd'),
)
```

```python
for row in Mat_ElmXnet:
    ElmXnet.append(row)
Mat_ElmSym = (
    ('FID', 'loc_name', 'fold_id', 'typ_id', 'pgini', 'av_mode', 'cosgini', 'GPSlat', 'GPSlon', 'mode_inp', 'usetp'),
)
for row in Mat_ElmSym:
    ElmSym.append(row)
Mat_ElmTr2 = (
    ('FID', 'loc_name', 'fold_id', 'typ_id', 'outserv', 'GPSlat', 'GPSlon'),
)
for row in Mat_ElmTr2:
    ElmTr2.append(row)
Mat_ElmLne = (
    ('FID', 'loc_name', 'fold_id', 'typ_id', 'pCondCir', 'pCondGnd', 'dline', 'outserv'),
)
for row in Mat_ElmLne:
    ElmLne.append(row)
Mat_StaCubic = (
    ('FID', 'loc_name', 'fold_id', 'obj_bus', 'obj_id'),
)
for row in Mat_StaCubic:
    StaCubic.append(row)
Mat_StaSwitch = (
    ('FID', 'loc_name', 'fold_id', 'aUsage', 'on_off'),
)
for row in Mat_StaSwitch:
    StaSwitch.append(row)
Mat_IntFolder = (
    ('FID', 'loc_name'),
    (10, 'Load Type'),
    (11, 'Transformer Type'),
    (12, 'Line Type'),
    (13, 'Syncronous Machine Type'),
    (14, 'Tower Type'),
    (15, 'Conductor Type'),
    (16, 'External Network')
)
for row in Mat_IntFolder:
    IntFolder.append(row)
Mat_TypLod = (
    ('FID', 'loc_name', 'fold_id', 'kpu', 'kqu', 'systp', 'phtech'),
    (20, '3PH-Y', 10, 1.6, 1.8, 0, 3),
)
for row in Mat_TypLod:
```

```
    TypLod.append(row)
Mat_TypLne = (
    ('FID', 'loc_name', 'fold_id', 'uline', 'sline', 'cohl_',
    'frnom', 'rline', 'xline', 'nlnph', 'nneutral'),
    (100, 'noname1', 12, 60, 1, 1, 60, 0.3059, 0.434, 3, 0),
    (101, 'noname2', 12, 22.9, 0.235, 1, 60, 0.484033,
   0.44499, 3, 1)
)
for row in Mat_TypLne:
    TypLne.append(row)
Mat_TypTr2 = (
    ('FID', 'loc_name', 'fold_id', 'strn', 'frnom', 'utrn_h',
    'utrn_l', 'uktr', 'nt2ph'),
    (400,'Huallin', 11, 4, 60, 22.9, 6, 7, 3),
    (401,'Collo', 11, 0.8, 60, 22.9, 0.44, 4.07, 3),
    (402,'Jambon500', 11, 0.5, 60, 22.9, 0.44, 5, 3),
    (403,'Jambon630', 11, 0.63, 60, 22.9, 0.44, 4.2, 3),
    (404,'Interconnection', 11, 6, 60, 60, 22.9, 6.61, 3),
    (405, 'LowVoltage_5', 11, 0.005, 60, 22.9, 0.44, 5, 3),
    (406, 'LowVoltage_10', 11, 0.01, 60, 22.9, 0.44, 5, 3),
    (407, 'LowVoltage_15', 11, 0.015, 60, 22.9, 0.44, 5, 3),
    (408, 'LowVoltage_16', 11, 0.016, 60, 22.9, 0.44, 5, 3),
    (409, 'LowVoltage_25', 11, 0.025, 60, 22.9, 0.44, 5, 3),
    (410, 'LowVoltage_37.5', 11, 0.0375, 60, 22.9, 0.44, 5,
   3),
    (411, 'LowVoltage_40', 11, 0.04, 60, 22.9, 0.44, 5, 3),
    (412, 'LowVoltage_50', 11, 0.05, 60, 22.9, 0.44, 5, 3),
    (413, 'LowVoltage_60', 11, 0.06, 60, 22.9, 0.44, 5, 3),
    (414, 'LowVoltage_75', 11, 0.075, 60, 22.9, 0.44, 5, 3),
    (415, 'LowVoltage_100', 11, 0.1, 60, 22.9, 0.44, 5, 3),
    (416, 'LowVoltage_200', 11, 0.2, 60, 22.9, 0.44, 5, 3),
    (417, 'LowVoltage_400', 11, 0.4, 60, 22.9, 0.44, 5, 3)
)
for row in Mat_TypTr2:
    TypTr2.append(row)
Mat_TypSym = (
    ('FID', 'loc_name', 'fold_id', 'sgn', 'ugn', 'cosn', '
   nslty'),
    (300, 'Huallin', 13, 4, 6, 0.8, 2),
    (301, 'Collo', 13, 0.92, 0.44, 0.8, 2),
    (302, 'Jambon380', 13, 0.45, 0.44, 0.85, 2),
    (303, 'Jambon785', 13, 1, 0.44, 0.8, 2)
)
for row in Mat_TypSym:
    TypSym.append(row)
Mat_TypGeo = (
    ('FID', 'loc_name', 'fold_id', 'nlear', 'nlcir', 'xy_c
   :0:0', 'xy_c:0:1', 'xy_c:0:2', 'xy_c:0:3', 'xy_c:0:4', '
   xy_c:0:5', 'xy_c:0:6', 'xy_e:0:1', 'xy_e:0:2'),
```

```
    (32, 'Three_phase', 14, 0, 1, 3, -1.2, -0.8, 1.2, 10, 11,
    10, 0.8, 11)
)
for row in Mat_TypGeo:
    TypGeo.append(row)
Mat_TypCon = (
    ('FID', 'loc_name', 'fold_id', 'uline', 'sline', 'ncsub',
    'iModel', 'rpha', 'rpha_tmax', 'erpha', 'diaco', 'mlei'),
    (40, 'Al_016_mm2', 15, 33, 0.1, 1, 0, 2.09, 2.09,
   1.8513, 5.1, 'Al'),
    (41, 'Al_025_mm2', 15, 33, 0.125, 1, 0, 1.31, 1.31,
   2.3595, 6.5, 'Al'),
    (42, 'Al_035_mm2', 15, 33, 0.16, 1, 0, 0.952, 0.952,
   2.7588, 7.6, 'Al'),
    (43, 'Al_050_mm2', 15, 33, 0.195, 1, 0, 0.663, 0.663,
   3.3033, 9.1, 'Al'),
    (44, 'Al_070_mm2', 15, 33, 0.235, 1, 0, 0.484, 0.484,
   4.09, 10.8, 'Al'),
    (45, 'Al_120_mm2', 15, 33, 0.34, 1, 0, 0.275, 0.275, 5.4,
    14.3, 'Al'),
    (46, 'Cu_016_mm2', 15, 27, 0.1, 1, 0, 1.5, 1.5, 2, 5.1, '
   Cu'),
    (47, 'Cu_025_mm2', 15, 27, 0.188, 1, 0, 0.741, 0.741,
   2.3232, 6.4, 'Cu'),
)
for row in Mat_TypCon:
    TypCon.append(row)
elmnet = 9
#-------COMPILING EXCELL file--------
k= 30001
#------------POSTE-----------------
b = 0
j = 0
n = 0
inout = 0
iusage = 0
phtech = 1
for e in Poste_LocName:
    for f in Tramo_LocName:
        if Tramo_LocName_Pater[b] == Poste_CodTramoMT[j]:
            inout = 0
            StaCubic.append([k, k, e, inout, f])
            StaSwitch.append([k + 10000, k + 10000, k, 'cbk',
    1])
            k = k + 1
            n = n + 1
        b = b + 1
    phtech = 1
    if n == 0:
```

```
        iusage = 0
    elif n == 1:
        iusage = 2
    else:
        iusage = 1
    ElmTerm.append([e, e, elmnet, iusage, 22.9, Poste_lat[j], Poste_long[j], phtech])
    inout = 1
    StaCubic.append([k, k, e, inout, Poste_CodTramoMT[j]])
    StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
    k = k + 1
    b = 0
    n = 0
    j = j + 1
#---------NODO---------------
Nodo_LocName_useful = list(dict.fromkeys(Tr2_CodNodo))
b = 0
j = 0
tram = 0
lat=0
long=0
iusage = 0
Nodo_Num_Phases = []
Nodo_Neutral = []
Nodo_Phases = []
    phtech = 1
    ElmTerm.append([e, e, elmnet, iusage, 22.9, lat, long, phtech])
    inout = 1
    StaCubic.append([k, k, e, inout, tram])
    StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
    k = k + 1
    for f in Tramo_LocName:
        if Tramo_LocName_Pater[b] == tram:
            inout = 0
            StaCubic.append([k, k, e, inout, f])
            StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
            k = k + 1
        b = b + 1
    j = j + 1
    b = 0
#------------MTTRAMO-----------
j=0
nfas = 0
codgeo = 32
tipcond = 40
tipnet = 40
cnm = 0
```

```
lenm = 0
for e in Tramo_LocName:
    if Tramo_CodNorma[j] == 'AA01602':
        tipcond = 40
    elif Tramo_CodNorma[j] == 'AA02501':
        tipcond = 41
    elif Tramo_CodNorma[j] == 'AA02502':
        tipcond = 41
    elif Tramo_CodNorma[j] == 'AA02503':
        tipcond = 41
    elif Tramo_CodNorma[j] == 'AA03503':
        tipcond = 42
    elif Tramo_CodNorma[j] == 'AA05003':
        tipcond = 43
    elif Tramo_CodNorma[j] == 'AA07003':
        tipcond = 44
    elif Tramo_CodNorma[j] == 'CU01603':
        tipcond = 46
    elif Tramo_CodNorma[j] == 'CU02503':
        tipcond = 47
    else:
        tipcond = 41
        cnm = cnm + 1
    if Tramo_CodNormaNe[j] == 'AA01604':
        tipnet = 40
    elif Tramo_CodNormaNe[j] == 'AA02504':
        tipnet = 41
    elif Tramo_CodNormaNe[j] == 'AA03504':
        tipnet = 42
    else:
        tipnet = 40
    ElmLne.append([e, e, elmnet, codgeo, tipcond, tipnet,
   Tramo_Length[j]/1000, 0])
    j = j + 1
#------TR2 and LOAD---------
j=0
outserv = 0
type = 400
i = 20001
pim = 0
aggi = 0.0002
plini = 0
plinir = 0
plinis = 0
plinit = 0
qlini = 0
qlinir = 0
qlinis = 0
qlinit = 0
```

```
for e in Tr2_LocName:
    if Tr2_Active[j] == 0:
        outserv = 1
    else:
        outserv = 0
    if Tr2_PotInstal[j] == 5:
        type = 405
    elif Tr2_PotInstal[j] == 10:
        type = 406
    elif Tr2_PotInstal[j] == 15:
        type = 407
    elif Tr2_PotInstal[j] == 16:
        type = 408
    elif Tr2_PotInstal[j] == 25:
        type = 409
    elif Tr2_PotInstal[j] == 37.5:
        type = 410
    elif Tr2_PotInstal[j] == 40:
        type = 411
    elif Tr2_PotInstal[j] == 50:
        type = 412
    elif Tr2_PotInstal[j] == 60:
        type = 413
    elif Tr2_PotInstal[j] == 75:
        type = 414
    elif Tr2_PotInstal[j] == 100:
        type = 415
    elif Tr2_PotInstal[j] == 200:
        type = 416
    elif Tr2_PotInstal[j] == 400:
        type = 417
    else:
        pim = pim + 1
        print(’PotInstal missing for: ’, e)
    ElmTr2.append([Load_LocName[j] + ’T’, Load_Locality[j] + Load_LocName[j], elmnet, type , outserv, Load_lat[j] + aggi, Load_long[j] + aggi])
    inout = 0
    StaCubic.append([k, k, Tr2_CodNodo[j], inout, Load_LocName[j] + ’T’])
    StaSwitch.append([k + 10000, k + 10000, k, ’cbk’, 1])
    k = k + 1
    iusage = 0
    phtech = 1
    voltage = 0.44
    ElmTerm.append([i, Load_Locality[j] + Load_LocName[j] + ’phantom’, elmnet, iusage, voltage, Load_lat[j] + 2*aggi, Load_long[j] + 2*aggi, phtech])
    inout = 1
```

```
    StaCubic.append([k, k, i, inout, Load_LocName[j] + 'T'])
    StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
    k = k + 1
    b = 0
    for f in LDSE_SED:
        if f==e:
            plini=0.95*Load_top[b]/1000
            b=b+1
    b=0
    qlini = 0
    if Load_FasiOut[j] == 'RN':
        plinir = plini
        plinis = 0
        plinit = 0
    elif Load_FasiOut[j] == 'SN':
        plinir = 0
        plinis = plini
        plinit = 0
    elif Load_FasiOut[j] == 'TN':
        plinir = 0
        plinis = 0
        plinit = plini
    elif Load_FasiOut[j] == 'RSN':
        plinir = plini/2
        plinis = plini/2
        plinit = 0
    elif Load_FasiOut[j] == 'RTN':
        plinir = plini/2
        plinis = 0
        plinit = plini/2
    elif Load_FasiOut[j] == 'STN':
        plinir = 0
        plinis = plini/2
        plinit = plini/2
    else:
        plinir = plini/3
        plinis = plini/3
        plinit = plini/3
    qlinir=0.05*plinir/0.95
    qlinis = 0.05 * plinis / 0.95
    qlinit = 0.05 * plinit / 0.95
    i_sym = 1
    typ_id = 20
    ElmLod.append([Load_LocName[j], Load_LocName[j], elmnet, typ_id, i_sym, plinir, qlinir, plinis, qlinis, plinit, qlinit, outserv, Load_lat[j] + 3*aggi, Load_long[j] + 3*aggi])
    inout = 0
    StaCubic.append([k, k, i, inout, Load_LocName[j]])
```

```
    StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
    i = i + 1
    k = k + 1
    j = j + 1
#------GENERATORS-------
#----HUALLIN------
Huallin_lat = -9.18101
Huallin_long = -77.4322
iusage = 0
voltage = 22.9
phtech = 1
ElmTerm.append([i, 'HuallinMT_Bus', elmnet, iusage, voltage, Huallin_lat, Huallin_long, phtech])
inout = 1
StaCubic.append([k, k, i, inout, 'TMT001070'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
outserv = 0
type = 400
ElmTr2.append(['Huallin_Tr2', 'Huallin_Tr2', elmnet, type , outserv, Huallin_lat + aggi, Huallin_long + aggi])
inout = 0
StaCubic.append([k, k, i, inout, 'Huallin_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
i = i + 1
iusage = 0
voltage = 6
phtech = 1
ElmTerm.append([i, 'HuallinGEN_Bus', elmnet, iusage, voltage, Huallin_lat + 2*aggi, Huallin_long + 2*aggi, phtech])
inout = 1
StaCubic.append([k, k, i, inout, 'Huallin_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
fid = 600
type = 300
pgini = 1.1
cosgini = 0.98
usetp = 1.03
ElmSym.append([fid, 'Huallin_SYM', elmnet, type, pgini, 'constv', cosgini, Huallin_lat + 3*aggi, Huallin_long + 3*aggi, 'PC', usetp])
inout = 0
StaCubic.append([k, k, i, inout, fid])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
i = i + 1
k = k + 1
#----COLLO------
```

```
Collo_lat = -9.1707
Collo_long = -77.4028
iusage = 0
voltage = 22.9
phtech = 1
ElmTerm.append([i, 'ColloMT_Bus', elmnet, iusage, voltage, Collo_lat, Collo_long, phtech])
inout = 0
StaCubic.append([k, k, i, inout, 'TMT000326'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
inout = 0
StaCubic.append([k, k, i, inout, 'TMT000304'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
outserv = 0
type = 401
ElmTr2.append(['Collo_Tr2', 'Collo_Tr2', elmnet, type, outserv, Collo_lat + aggi, Collo_long + aggi])
inout = 0
StaCubic.append([k, k, i, inout, 'Collo_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
i = i + 1
k = k + 1
iusage = 0
voltage = 0.44
phtech = 1
ElmTerm.append([i, 'ColloGEN_Bus', elmnet, iusage, voltage, Collo_lat + 2*aggi, Collo_long + 2*aggi, phtech])
inout = 1
StaCubic.append([k, k, i, inout, 'Collo_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
fid = 601
type = 301
pgini = 0.734
cosgini = 0.98
usetp = 1.03
ElmSym.append([fid, 'Collo_SYM', elmnet, type, pgini, 'constv', cosgini, Collo_lat + 3*aggi, Collo_long + 3*aggi, 'PC', usetp])
inout = 0
StaCubic.append([k, k, i, inout, fid])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
i = i + 1
k = k + 1
#----JAMBON------
Jambon_lat = -9.1523
Jambon_long = -77.3800
```

```
outserv = 0
type = 402
ElmTr2.append(['Jambon500_Tr2', 'Jambon500_Tr2', elmnet, type
    , outserv, Jambon_lat - aggi, Jambon_long - aggi])
inout = 0
StaCubic.append([k, k, 'STR000550', inout, 'Jambon500_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
iusage = 0
voltage = 0.44
phtech = 1
ElmTerm.append([i, 'Jambon500GEN_Bus', elmnet, iusage,
   voltage, Jambon_lat - 2* aggi, Jambon_long - aggi, phtech
   ])
inout = 1
StaCubic.append([k, k, i, inout, 'Jambon500_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
fid = 602
type = 302
pgini = 0.38
cosgini = 0.98
usetp = 1.03
ElmSym.append([fid, 'Jambon500_SYM', elmnet, type, pgini, '
   constv', cosgini, Jambon_lat - 3*aggi, Collo_long - aggi,
   'PC', usetp])
inout = 0
StaCubic.append([k, k, i, inout, fid])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
i = i + 1
k = k + 1
outserv = 0
type = 403
ElmTr2.append(['Jambon630_Tr2', 'Jambon630_Tr2', elmnet, type
    , outserv, Jambon_lat - aggi, Jambon_long + aggi])
inout = 0
StaCubic.append([k, k, 'STR000550', inout, 'Jambon630_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
iusage = 0
voltage = 0.44
phtech = 1
ElmTerm.append([i, 'Jambon630GEN_Bus', elmnet, iusage,
   voltage, Jambon_lat - 2 * aggi, Jambon_long + aggi, phtech
   ])
inout = 1
StaCubic.append([k, k, i, inout, 'Jambon630_Tr2'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
```

```
fid = 603
type = 303
pgini = 0.785
cosgini = 0.98
usetp = 1.03
ElmSym.append([fid, 'Jambon630GEN_SYM', elmnet, type, pgini,
   'constv', cosgini, Jambon_lat - 3*aggi, Collo_long + aggi,
    'PC', usetp])
inout = 0
StaCubic.append([k, k, i, inout, fid])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
i = i + 1
k = k + 1
#-------EXTERNAL NETWORK-----
pgini = 10
xnet_lat = -8.9875
xnet_long = -77.3422
iintgnd = 1
ElmXnet.append(['Xnet', 'Xnet', elmnet, pgini, xnet_lat,
   xnet_long, iintgnd])
inout = 0
StaCubic.append([k, k, 'STR000730', inout, 'Xnet'])
StaSwitch.append([k + 10000, k + 10000, k, 'cbk', 1])
k = k + 1
#-------PRINTING RESULTS AND ERRORS-------
print('number of POSTE', Poste_num)
print('number of LOD:', Load_num)
print('number of TRAMO:', Tramo_num)
print('number of NODO', Nodo_num)
if fid_error != 0:
   print('number of reference errors: ', fid_error)
print('number of cime: ', numcim)
print('number of cubicles', k-30001)
if pim != 0:
    print('number of PotInstalada missing in SED is: ', pim)
if cnm != 0:
    print('number of CodNorma missing in MTTRAMO is: ', cnm)
if lenm != 0:
    print('number of Length missing in MTTRAMO is: ', lenm)
if xlserror != 0:
    print('number of Excel_SED error is: ', xlserror)
totm =fid_error + pim + cnm + lenm + xlserror
if totm != 0:
    print('total number of missing data is: ', totm)
#---------END--------
#----saving-----
wb.save(filename = 'HUGO.xlsx')
#---delete the previous file---
delpc = 'C:/Users/Toto-/Dropbox/Poli/Tesi/Passerella/HUGO.
```

```
    xlsx'
os.remove(delpc)
#--mooving to new folder---
nompc = 'C:/Users/Toto-/Documents/Pyton/Progetti/HUGO.xlsx'
destpc = 'C:/Users/Toto-/Dropbox/Poli/Tesi/Passerella'
shutil.move(nompc,destpc)
print("Fine, tempo: {:.2f}s".format(time.time() - start_time)
    )
```