POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Energy Engineering

Optimal distribution substations placement for an effective electrification strategy in developing countries

Relatore: Prof. Marco Merlo
Correlatrice: PhD Candidate Silvia Corigliano
Correlatore: Federico Rosato

Tesi di Laurea Magistrale di:
Carla ORTIZ DOMINGUEZ 916185

Anno Accademico 2019 - 2020

To the children of Kokuselei
Turkana, Kenya

Acknowledgments

First of all, I would like to thank my tutor, Prof. Merlo and my co-tutor, Silvia Corigliano, for putting their trust in me and let me participate in this amazing project of rural electrification. Their help and implication have been a referent for me.

I would like to thank Federico Rosato for his disinterested assistance and interest in the project.

To all the professors, administrative personnel and workers of the *Politecnico*, for their reception and attention during these two years.

To all the professors, administrative personnel and workers of the UPM, specially of the *ETSII,* my home university, where I have passed six of the best years of my life.

To my family. Specially my parents, Silvia and Alvaro. For their constant and unconditional support.

To my friends. Because they have always been there supporting my decisions and taking care of me.

I cannot but express a deep appreciation to Rocío Aguirre, Cecilia Puig and Eleni Tsegaw. They raised in me the passion for Turkana people, and the commitment to always look for real solutions to the problems of Africa.

Extended Abstract

I. INTRODUCTION

In 2019, slightly less than 1 billion people lacked access to electricity in the world. Approximately, 50% of them are found in Sub-Saharan Africa and 87% live in rural areas. In a global framework where the goal of the 7th SDG is to ensure access to affordable, reliable, sustainable and modern energy for all, rural electrification planning is needed [1].

This thesis work is centred on access to electricity and focuses on setting up of rural electricity infrastructure and providing connectivity to households. The objective is to design a procedure able to evaluate the optimal location of secondary substations from a topological perspective considering the population distribution given by georeferenced data. A two-step procedure has been developed.

The first step consists of clustering densely populated areas with the DBSCAN algorithm in order to find the more dense areas in terms of inhabitants, and the second step consists of applying another clustering algorithm to the groups obtained with DBSCAN in order to divide the population into areas supplied by a secondary substation.

To place secondary substations in the second step, two different algorithms have been studied in order to see the different advantages and disadvantages and choose the one which returns best results. These two algorithms are k-means, based on partition, and LUKES, based on graph theory.

II. DISTRIBUTION SYSTEMS

Electric power distribution is the final stage in the delivery of electric power; it carries electricity from the transmission system to individual consumers. It is composed by a primary substation, medium voltage lines, secondary substations and low voltage lines. In the primary substations, the energy is transformed from high voltage to medium voltage (HV/MV). The MV lines, directly supply the commercial and industrial consumers or the secondary substations which transform current from MV to LV.

The power distribution system is different from one country to another and it also differs from developed countries to developing countries.

In Tab.1, a comparison of five DSO indicators [2] between Europe and developing countries is presented:

ID	**DSOs indicators**	**Europe**	**Uganda**	**Mozambique**
1	LV circuit length per LV consumer	0.03 km	0.017 km	0.022 km
2	Number of LV consumer per MV/LV substation	86	100	110
3	MV/LV substation capacity per LV consumer	4.76 kVA	1.58 kVA	1.98 KVA
4	Number of MV supply points per HV/MV substation	126.75	-	156.8
5	Typical transformation capacity of MV/LV secondary substations in rural areas	100, 250, 400 kVA	160 kVA	220 kVA

Tab. 1 Comparison of DSO's indicators

The following conclusions are extracted:

- The LV circuit length per LV consumer is lower in developing countries.
- The number of LV consumers per MV/LV substations is higher in developing countries.

- The MV/LV capacity per LV consumer is lower in developing countries.

- The number of MV supply points per HV/MV substation is higher in developing countries.

- The typical transformation capacity of MV/LV secondary substations in rural areas is lower in developing countries.

These conclusions are backed by the lower population density, the lower power demand, lower energy efficiency, lower simultaneity factor and lower electricity density in developing countries.

The distribution system is an important part of the electric power system, accounting for almost 60% [2] of the voltage lines and hence most of the costs and losses. An introduction of it in the simulation tools is necessary to have an accurate estimation of the reality.

III. TOOLS FOR RURAL ELECTRIFICATION

Rural electrification planning is not an easy problem to overcome as connection to the grid requires very high investments, and a good planning and estimation of the future cost is necessary to be able to achieve the goal of bringing electricity to everybody.

Bearing in mind the characteristics and limits of distribution systems in rural areas of developing countries, it is important to have well defined strategies for planning new networks. In the literature, numerous tools for rural electrification planning developed from very different perspectives are found.

Lack of spatial information in rural and regional level is one of the main problems for development practitioners, government officials and local level planners. Geospatial planning facilitates the understanding of spatial aspects of social and economic development by relating socio-economic variables to natural resources and the physical world, providing a tool for targeting interventions and monitoring impacts on various scales over wide areas.

Some tools based on geospatial planning are:

- REM [3]
- RNM [4]
- Network Planner [5]
- GEOSIM [6]
- LAPER [7]

All of them have advantages but also disadvantages.

Currently, the *Politecnico di Milano* is developing a tool called GISEle. GISEle is rooted on the analysis of spatial data and runs through all the passages leading to the identification of the optimal techno-economic solution to bring the energy where it lacks.

GISEle is the result of an accurate analysis and wants to cover the important gaps missing in these tools portfolio. With GISEle a new trend is pursued, bringing from one side analytical innovation, and from the other promoting the development of an integration platform enhancing synergies among tools and identifying the direction needed to solve the three components of the energy conundrum: energy access, energy security, and climate change [8].

The procedure of this project is designed to be part of the clustering process of GISEle.

IV. SITING SECONDARY SUBSTATIONS

A power distribution network consists of a number of substations connected to each other via feeders. Distribution planners must ensure that there is adequate substation capacity (transformer capacity) and feeder capacity (distribution capacity) to meet the load forecasts within the planning horizon [9].

The planning of power distribution system includes [9]:

- Optimal location of substations
- Optimal location of feeders
- Optimal individual feeders design
- Optimal allocation of loads
- Optimal allocation of substation capacity
- Optimal mix of transformer by substation

Planning models to site secondary substations can be divided into two groups: planning under normal conditions or planning for emergency [9].

The solutions for planning under normal conditions are divided in three categories:

- Mathematical optimization: the models usually have strict optimality but are very difficult to solve when the size and complexity of the system grows substantially.
- Heuristic and algorithm optimization: the models simplify approaches to reduce the dynamic problem into a static one, thus allowing the problems to be solved more efficiently at the expense of getting an optimal solution.
- Intelligent optimization: these approaches can hardly handle large-scale problems with reasonable computation time due to parameter selection and the various annealing requirements [10].

Some examples of algorithms to site secondary substations are:

- GIS-based and Semi-Supervised Learning Algorithm [10]
- Genetic Algorithm [11]
- Imperialist Competitive Algorithm [12]
- PSO Algorithm with MST for feeder routine [13]
- K-means and Dijkstra's Algorithm [14]
- Weighted Voronoi Diagram and Transportation Model [15]

V. ALGORITHMS AND OPTIMIZATION METHODS

One of the final objectives of this thesis work is to help in the rural electrification planning. The project focus on the development of a procedure to site secondary substations.

The approach to site the substations is from a topological point of view, not an economic assessment. The idea is to place the substations where the people are. To do so, it is required to detect these populated areas by clustering the population. Hence, the election of the clustering algorithm has an important role, as it will determine which points are going to be considered in the electrification process, which ones are going to be connected to the grid and how.

The portfolio of algorithms is very extensive. However, in order to choose which clustering algorithm is the best one to cluster densely populated areas,

six different characteristics are considered:

- Time complexity;
- Shape of the cluster;
- Appropriate for large-scale data;
- Appropriate for high dimensional data;
- Sensitivity to input data;
- Sensitivity to noise/outliers.

In the problem addressed, the data are points (pixels) with geographic coordinates, with a population density associated.

After filtering the algorithms, giving priority to the ones with low or middle time complexity, the ones which deal with arbitrary shape of the cluster, for large scale data, with little or moderate sensitivity to input data and with little or moderate sensitivity to noise/outliers, the ones chosen as appropriate for the task of clustering populated areas are in Tab.2.

Algorithm	Time Complexity	Number of input parameters	Language of the code
K-MEANS	O (n)	1	Python
CURE	O (n^2log n)	1	Python, C++
DBCLASD	O($3n^2$)	0	Python
DBSCAN	O (n log n)// O(n^2)	2	Python
OPTICS	O (n log n)	2	C++, MATLAB
LUKES	O (W^2n)	1	Python
STING	O(n)	5	Python
DENCLUE	O (log |D|)	2	Java

Tab. 2 Comparison of the algorithms

For the procedure developed in this project, it is required low time complexity, the least number of inputs possible and Python as the coding language.

Taking these three features into account and the characteristics of the spatial data, k-means, DBSCAN and LUKES seems to be the more suitable option for clustering densely populated areas.

VI. PROCEDURE PROPOSED

The objective is to design a procedure able to evaluate the optimal location of secondary substations from a topological perspective.

On one hand, the mathematical methods and artificial intelligence methods have been discarded as the size of the managed data is big and this would lead to complex systems and high computational times.

On the other hand, the heuristic algorithms which look for minimizing a cost function, as the genetic algorithm or the ICA, have been discarded as this is not an economic study. However, as the location of consumers is known and the load can be estimated, the k-means and graph partitioning algorithms seem to fit with the data.

According to literature and the analysis of clustering densely populated areas, it has been decided to develop a two-step clustering procedure. The two steps consist of:

1. Clustering the population of the big area, detecting the densely populated areas and discarding outliers. For this step DBSCAN has been chosen.

2. Clustering the population of the clusters obtained through DBSCAN in order to divide the population obtaining sub-clusters inside each group that represents the area supplied by a secondary substation. For this step, k-means and LUKES algorithm have been chosen.

DBSCAN is useful at the beginning because it detects the areas with a specific population density. And, as the objective is to site substations where the people are, this algorithm finds these areas in an efficient way.

An analysis over the two algorithms chosen for the second step needs to be

done in order to choose the one with best results.

K-MEANS

K-means clustering is a method that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

The algorithm starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the cluster centres until a convergence criterion is met. The method is relatively scalable and efficient in processing large data sets and its time is linear O(n), so it is relatively small.

To evaluate the results and validity of the k-means algorithm, two different codes have been implemented and tested: a normal k-means and a weighted k-means, with three different versions of each one:

- With no loop: running the algorithm once and keep the first solution obtained.

- With a simple loop: imposing a distance constraint between substation and load and adding a cluster each time the algorithm finds a distance higher than the threshold.

- With a complex loop: imposing a distance constraint between substation and load and dividing in two only the clusters that do not meet the constraint.

Normal k-means

In the module *sklearn.cluster* of Python, the function *KMeans* is found. A code with this function has been written and implemented. The k-means problem is solved using either Lloyd's or Elkan's algorithm. The average complexity is given by O (knT), were n is the number of samples and T is the number of iterations.

For example, the structure of the code of normal k-means with complex loop is:

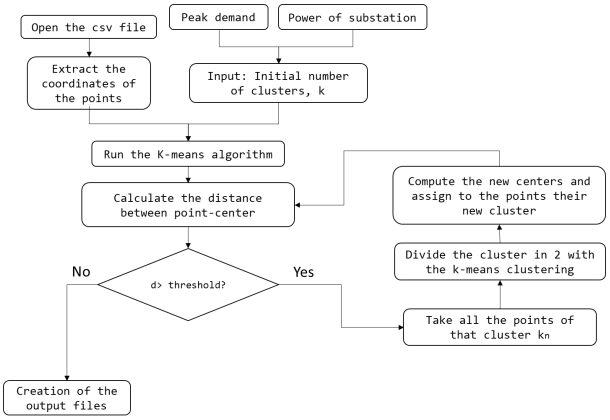

Fig. 1 Structure of normal k-means with complex loop

Weighted k-means

The "weighted" k-means problem is a natural extension of the k-means problem that allows to include some more information, namely, a set of weights associated with the data points. These might represent a measure of importance, a frequency count, or some other information. The intent is that a point with a weight of 5 is twice as "important" as a point with a weight of 2.5, for instance.

In the weighted k-means problem, a set of N points $X(I)$ in M-dimensions are given, and a corresponding set of nonnegative weights $W(I)$. The goal is to arrange the points into K clusters, with each cluster having a representative point $Z(J)$, usually chosen as the weighted centroid of the points in the cluster:

$$Z(J) = \frac{\sum_{all\ X(I) in\ cluster\ J} W(I) \cdot X(I)}{\sum_{all\ X(I) in\ cluster\ J} W(I)}$$

In this case, the weight W(I) is the power associated to each point. This will be obtained with the population of the point and the power per capita.

Weights assignment

In order to test the performance of the weighted k-means, different weights are assigned to the points. The criterion to assign different weight to some points is based on the hypothesis of higher power demand growth for those kinds of areas.

For instance, some features that could be considered are:

- Distance to the centre of the village: usually people tend to live closer to the centre of the villages, so more importance (more weight) is given to these areas.

- Size of the roof of the houses: the bigger the size of the roof, usually more people are living in that building and/or higher is the power demand, so more weight is given to that buildings.

- Density distance between buildings: areas with less distance between buildings are usually denser the areas in terms of population, so more weight is given to these areas.

- Distance to the road: usually in developing countries people tend to establish around the roads, and together with the fact that electric posts go along roads, these areas are given higher weights.

For example, the structure of the code of weighted k-means with complex loop is:

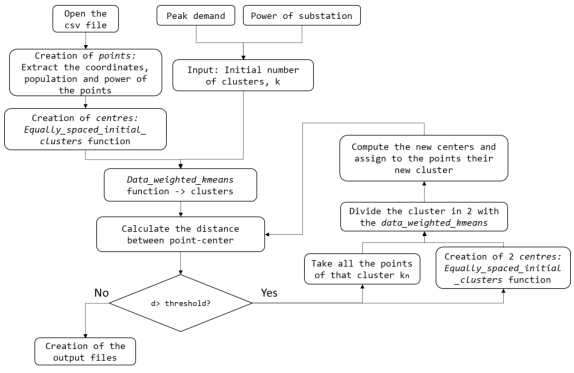

Fig. 2 Structure of weighted k-means with complex loop

LUKES

This graph partitioning algorithm has been chosen as its performance perfectly fits with the characteristics of the data available and objective pursued despite the high time complexity.

The steps associated with LUKES are [16]:

Step I. Label the tree T, and form the directed, order tree T'.

Step 2. For each leaf node u with weight w, form the partition $u_w = u_o =$ (u). The value of this partition is zero. For all nodes u of T' that are branch nodes (nodes having one or more sons), initialize (v) = v_j with value zero; here j is the weight of node u.

Step 3. Select some node x all of whose sons are leaf nodes, and form the optimal partitions for each weight equal to or less than the weight constraint of the subtree whose root is node x. To form these optimal partitions, follow these steps:

a. Let i = 1.

b. Form $x_j' = C [x_a, y_b]$ (for j = w, w + 1 , . . ., W), where the operator C $[x_a, y_b]$ forms partitions by either of the operations defined above; the particular partition C $[x_a, y_b]$ chosen is that of maximal value. Here y is the i^{th} son of node x and a + b = j, where $w \leq a \leq W$ and $0 \leq b \leq W$. The weight of node x is w, and the weight constraint is W.

c. Make all $x_j = x_j'$. If i = number of sons of node x, go to Step 4. Else, let i = i+1 and go to Step 3 (b).

Step 4. Denote by x_0 the partition of the subtree whose root is x that has maximal value from the set $\{x_w, x_{w+1}, ..., x_W\}$. Delete the sons of node x from the tree T'. If node x is the root of T', then x_0 represents the optimal partition of T' (hence of T). Otherwise, go to Step 3.

To understand how LUKES is implemented, the description of the properties of the graph are presented:

- Nodes (V): represent the populated points.

- Weight of the nodes (w): it is the power associated to the people of the node. The power is calculated by multiplying the number of people by the average power per capita.

- Weight constraint (W): it is the maximum power a cluster could have. This means the maximum power of the transformer, as a cluster represents an area supply by a secondary substation.

- Edges (E): represent the low voltage lines. They connect populated points.

- The weight of the edges (v): it is the inverse of the distance between points/nodes. As the objective is to minimize costs, and costs are directly proportional to the length of the lines, if the distance between nodes is taken as the weight of the edges, the shortest distances are going to be cut instead of the longest. It is desirable to keep the shortest distances as is where the low voltages lines are going to be potentially constructed.

The structure of the code of LUKES is:

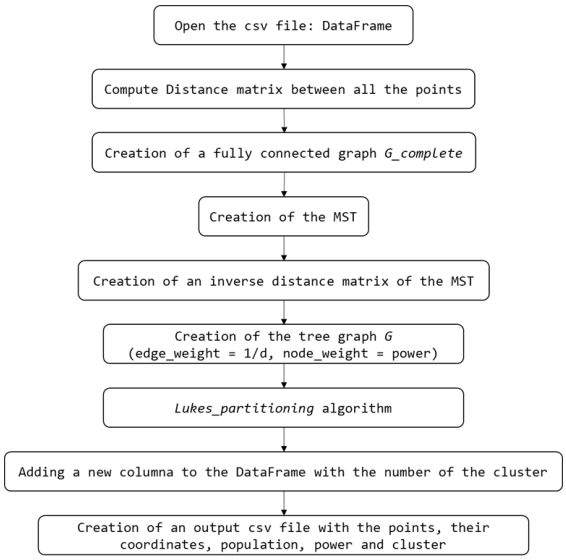

Fig. 3 Structure of LUKES

VII. CASE STUDIES

Two case studies have been conducted to test the siting substations procedure. The first one is conducted in Namanjavira, a rural administrative post in Zambezia, Mozambique. The second case study is conducted in Omereque, a municipality of Cochabamba, Bolivia.

NAMANJAVIRA

For the first step, the DBSCAN algorithm is implemented. The input data is the population data with its coordinates.

For the second step, the input data used to test the algorithms to site secondary substation are:

- The population data divided in the clusters obtained through the DBSCAN clustering algorithm: 4,729 points with 4 people per point (Fig.4).

- The power per capita consumed in that cluster: 0.025 kWh/capita [8].

- The distance constraint: 1000 m. [13]

- The power constraint (W): The maximum power per cluster is the power a transformer can supplied: W=50 kW is the chosen value.

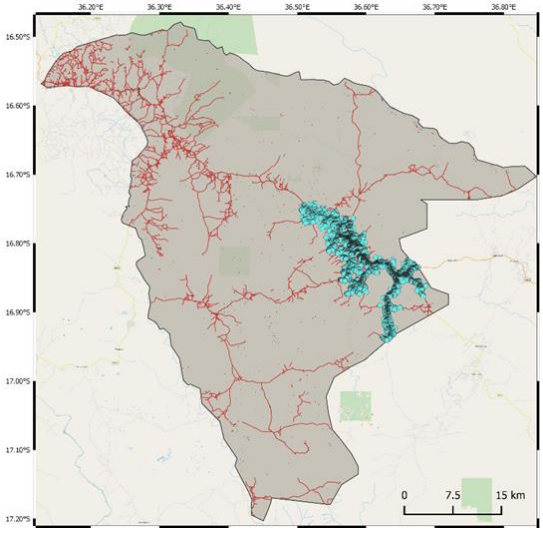

Fig. 4 Population data of Namanjavira

OMEREQUE

This case study skips the first step and analyses directly the site of secondary substations over an area in Omereque (Bolivia) with data provided by the Spanish NGO *Luces Nuevas*.

The input data used for the second step of the procedure to test the algorithms to site secondary substation are:

- The number and position of the households. This would be the output from the DBSCAN step. There are 139 points, with 5 people per point (Fig.5).

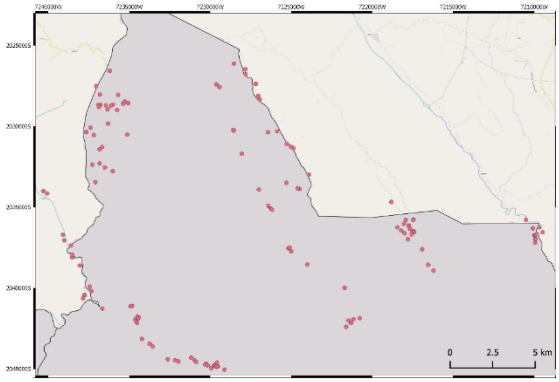

Fig. 5 Population data of Omereque

- The power per capita consumed in that cluster: 0.4 kWh/capita.

- The distance constraint: 600 m.

- The power constraint (W): Due to the rural area, the maximum power chosen as constraint are W=10, W=20 and W=25 kW.

VIII. RESULTS

The results are divided in the weighting process - for the weighted version of k-means - and the performance of the algorithms k-means and LUKES.

NAMANJAVIRA

Weighting Process

For the area of this case study, more weight has been given to (Fig.6):

i. Areas with higher population density, extracted from QGIS with the population layer. The population is represented with black dots.

ii. Areas where, according to data collected from OpenStreetMaps, there are recognized villages. The polygons are represented in green colour.

iii. Areas situated near the main road. The main road is represented with a yellow dashed line.

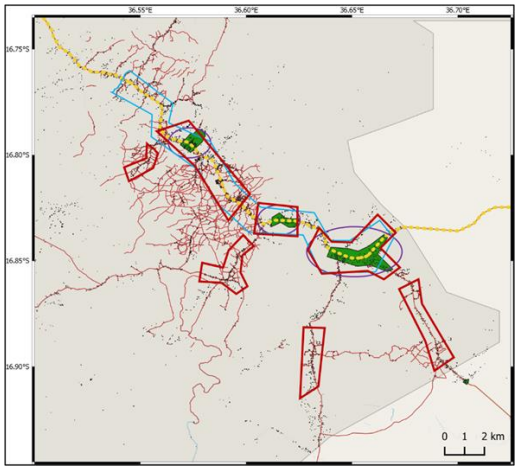

Fig. 6 Weighting process in Namanjavira

The weights (power per capita) given to the points are:

- 0.025: Power per capita according to [13]. Points outside the areas (i), (ii) and (iii) will remain with this weight.

- 0.05: weight given to points inside only one type of area (i), (ii) or (iii).

- 0.1: weight given to points inside two areas at the same time.

- 0.2: weight given to points inside the three areas (i), (ii) and (iii) at the same time.

As the population per point is 4, the power of the points is 0.1, 0.2, 0.4 and 0.8 kWh per point. Applying this criterion, the weights assigned to the points are presented in Fig.7.

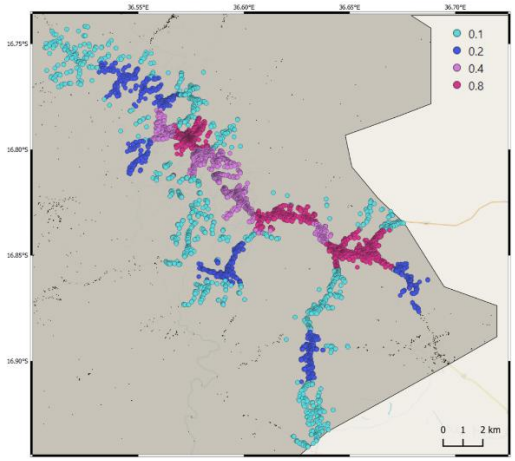

Fig. 7 Weights assigned in Namanjavira

Performance of the algorithms

A resume of the data obtained with the k-means with complex loop and LUKES algorithms in terms of performance are presented in Tab.3.

ID	Algorithm characteristics	Normal k-means	Weighted k-means	LUKES W=500
1	Number of clusters input	37	37	-
2	Final number of clusters	84	98	50
3	Running time [s]	6,865.9	1,426.71	87,060
4	Maximum distance between load and substation [m]	998.68	990.43	2,400
5	Average power [kW]	22.23	19.05	37.34

Tab. 3 Performance of the algorithms in Namanjavira

To compare the solution with reality, the 5 DSO indicators from section II have been selected. In the following table this information is shown and compare with the Mozambican values.

ID	Indicators	Normal k-means	Weighted k-means	LUKES W=500	Mozambique (Literature)
1	LV circuit length per LV consumer	1,000	1,000	2,400	22 m
2	Number of LV consumer per MV/LV substation	225	193	379	110
3	MV/LV substation capacity per LV consumer	0.22	0.26	0.13	1.98
4	Number of MV supply points per HV/MV substation	84	98	50	157
5	Typical transformation capacity of MV/LV secondary substations in rural areas	50 kW	50 kW	50 kW	176 kW*

*Value calculated with a power factor=0.8

Tab. 4 DSO's indicators from the algorithms vs. Literature

Considering the values of Mozambique both for rural and urban areas, and probably more representative for urban areas due to more access to information, the conclusions extracted are:

1. The LV circuit length is higher, which makes sense due to the lower population density in rural areas.

2. The number of LV consumers per substation is also higher for the same reason of population density along with the lower power demand in rural areas.

3. The capacity of the MV supply point is the typical transformation capacity of the substations in rural areas (ID 5). This value should be lower in rural areas, as usually smaller transformers are used, and the power demand is lower too.

4. The number of substations is usually lower in rural areas due to lower power demand and lower population density. Also, the value obtained in the case study is lower because in this study case only the MV/LV substations have been considered as MV supply points, neglecting the MV consumers.

5. The value of the capacity of the transformers is the one used for the case study. The capacity of the transformers is lower for rural areas due to lower power demand, lower simultaneity factor and lower population density.

OMEREQUE

Weighting Process

For this area, more weight has been given to (Fig.8):

i. Areas where, according data collected from OpenStreetMaps, settlements already exist, so population density is higher. These areas are the big yellow polygons. Some of the small ones are water reserves, so they are not considered.

ii. Areas situated near the road. The main road is represented with a yellow line and secondary roads with a grey line.

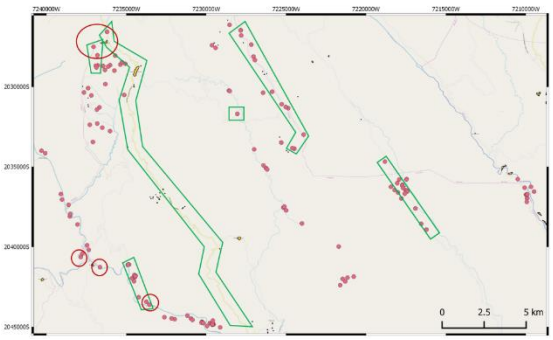

Fig. 8 Weighting process in Omereque

The weights (power per capita) given to the points are:

- 0.2: power per capita for the points outside the areas (i) and (ii).

- 0.4: weight given to points inside only one type of area (i) or (ii). It is the power per capita estimated by the Politecnico study.

- 0.8: weight given to points inside the two areas at the same time.

As the population per point is 5, the power of the points is 1, 2 and 4 kWh per point. Applying this criterion, the weights assigned to the points are presented in Fig.9:

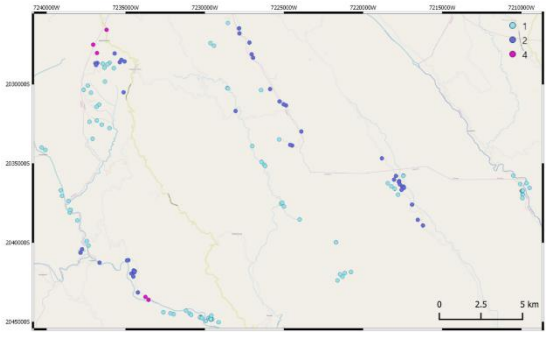

Fig. 9 Weights assigned in Omereque

Performance of the algorithms

A resume of the data obtained with the k-means and LUKES algorithms in terms of performance are presented in Tab.5.

ID	Algorithm characteristics	Normal k-means	Weighted k-means	LUKES W=10	LUKES W=20	LUKES W=50
1	Number of clusters input	10	10	-	-	-
2	Final number of clusters	44	49	28	14	5
3	Running time [s]	29.95	5.34	14.4	17.72	22.17
4	Maximum distance between load and substation [m]	598.8	599.92	2,700	4,800	8,400
5	Average power [kW]	4.61	4.17	7.25	14.5	40.6

Tab. 5 Performance of the algorithms in Omereque

To compare the solution with reality, the 5 DSO indicators from section II have been selected. In Tab.6 this information is shown and compare with the corresponding values of Bolivia.

ID	Indicators	Normal k-means	Weighted k-means	LUKES W=10	LUKES W=20	LUKES W=50	Bolivia (Literature)
1	LV circuit length per LV consumer	599 m	600 m	2700 m	4800 m	8400 m	500 -1000 m
2	Number of LV consumer per MV/LV substation	16	14	25	50	139	16-26
3	MV/LV substation capacity per LV consumer	1.27	1.41	0.40	0.40	0.36	0.958
4	Number of MV supply points per HV/MV substation	44	49	28	14	5	14 - 24
5	Typical transformation capacity of MV/LV secondary substations in rural areas	20 kW	20 kW	10 kW	20 kW	50 kW	12, 20 kW*

*Value calculated with a power factor=0.8

Tab. 6 DSO's indicators from the algorithms vs. Literature

Considering the values of Bolivia:

1. When the power constraint is met, the LV circuit length is higher, which makes sense due to the lower population density in rural areas.

2. The number of LV consumers per substation suits with the values from Literature when the power is similar (10-20 kW).

3. The capacity of the MV supply point is the typical transformation capacity of the substations in rural areas (ID 5). This value should be lower in rural areas, as usually smaller transformers are used, and the power demand is lower too. With k-means is higher due to the higher number of clusters. The systems would be highly oversized.

4. The number of substations is higher with k-means due to the distance constraint and the low population density. However, with the power constraint (LUKES) the values are aligned with literature.

5. The value of the capacity of the transformers is the one used for the case study. The capacity is usually lower in rural areas due to lower power demand, lower simultaneity factor and lower population density.

IX. CONLUSIONS

The results obtained are promising as they already fit with literature. And, as soon as more constraints are included in the procedure, better and more realistic results are going to be obtained.

Related to the performance of the algorithms, with k-means the distance is controlled, and it is faster than LUKES, while with LUKES the power is controlled despite the loss of control of the size of the clusters.

Regarding the comparison of the DSO's indicators, k-means returns best results in terms of length of the lines, while LUKES fits better with literature in terms of power. This is due to the different constraints imposed in both algorithms.

The future steps of this project are:

- Optimization of the k-means and LUKES algorithms in order to implement both distance and power constraints at the same time.
- Try different optimization options for the distance constraint loop.
- Look for the centre of the clusters with LUKES algorithm or a place to sit the secondary substation.
- Evaluate other algorithms considered in section V.
- Conduct an economic analysis.

X. BIBLIOGRAPY

[1] UNDG, "Background of the Sustainable Development Goals." [Online]. Available: https://www.undp.org/content/undp/en/home/sustainable-development-goals/background.html. [Accessed: 29-Mar-2020].

[2] G. Prettico, M. G. Flammini, N. Andreadou, S. Vitiello, G. Fulli, and M. Masera, *JRC Science for Policy Report; Distribution System Operators Observatory 2018*. 2019.

[3] R. Amatya *et al.*, "Computer-aided electrification planning in developing countries : The Reference Electrification Model (REM)," *Univers. Energy Access Lab*, pp. 1–111, 2018.

[4] C. Mateo Domingo, T. Gómez San Román, Á. Sánchez-Miralles, J. P. Peco González, and A. Candela Martínez, "A reference network model for large-scale distribution planning with automatic street map generation," IEEE Trans. Power Syst., vol. 26, no. 1, pp. 190–197, 2011.

[5] F. Kemausuor, E. Adkins, I. Adu-Poku, A. Brew-Hammond, and V. Modi, "Electrification planning using Network Planner tool: The case of Ghana," Energy Sustain. Dev., vol. 19, no. 1, pp. 92–101, 2014.

[6] S. Watchueng, R. Jacob, and A. Frandji, "Planning tools and methodologies for rural electrification," Rep. From Http//Www.Club-Er.Org/, no. December, p. 56, 2010.

[7] R. Fronius, "Rural electrication planning software (LAPER)," pp. v5-20-v5-20, 2005.

[8] T. Edeme, D. and Carnovali, "GISele: an innovative GIS-based

approach for electric networks routines.," Politecnico di Milano, 2019.

[9] S. K. Khator and L. C. Leung, "Power distribution planning: A review of models and issues," *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1151–1159, 1997.

[10] L. Yu *et al.*, "An efficient substation placement and sizing strategy based on GIS using semi-supervised learning," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 371–379, 2018.

[11] M. R. Haghifam, M. Esmaeeli, A. Kazemi, and H. Shayanfar, "Optimal placement of the distribution substations to improve reliability under load growth," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2015-Septe, pp. 1–5, 2015.

[12] S. Najafi and R. Gholizadeh, "On optimal sizing, siting and timing of distribution substations," *EPDC 2013 - 18th Electr. Power Distrib. Netw. Conf.*, pp. 1–6, 2013.

[13] I. J. Hasan, C. K. Gan, M. Shamshiri, M. Ruddin, A. Ghani, and R. Bin Omar, "Optimum Feeder Routing and Distribution Substation Placement and Sizing using PSO and MST," *Indian J. Sci. Technol.*, vol. 7, no. 0974–5645, pp. 1682–1689, 2014.

[14] G. C. Cabrera-celi and P. F. Vasquez-miranda, "using Clustering and Shortest Path Algorithms," 2017.

[15] S. Wang, Z. Lu, S. Ge, and C. Wang, "An improved substation locating and sizing method based on the weighted voronoi diagram and the transportation model," *J. Appl. Math.*, vol. 2014, 2014.

[16] J. A. Lukes, "Efficient Algorithm for the Partitioning of Trees.," *IBM J. Res. Dev.*, vol. 18, no. 3, pp. 217–224, 1974.

Abstract

In 2019, slightly less than 1 billion people lacked access to electricity in the world. Approximately, 50% of them are found in Sub-Saharan Africa and 87% live in rural areas. In a global framework where the goal of the 7th SDG is to ensure access to affordable, reliable, sustainable and modern energy for all, rural electrification planning is needed.

This thesis work is centred on access to electricity and focuses on setting up of rural electricity infrastructure and providing connectivity to households. The objective is to design a procedure able to evaluate the optimal location of secondary substations from a topological perspective considering the population distribution given by georeferenced data. A two-step procedure, which combines two different clustering algorithms, has been developed.

The starting point is a big populated area, with different population density. As it is not economically worthy to electrify areas where there is not a minimum number of people, the first step consists of clustering densely populated areas in order to find the denser areas in terms of inhabitants. The clustering algorithm chosen for this task is DBSCAN. The second step consists in applying another clustering algorithm to the groups obtained with DBSCAN in order to divide the populated areas in smaller areas supplied by a secondary substation.

To place secondary substations, two different algorithms have been studied in order to see different advantages and disadvantages and choose the one which returns best results. These two algorithms are k-means, based on partition, and LUKES, based on graph theory.

To evaluate the performance of the algorithms in actual case studies, two case studies have been considered: one in Namanjavira (Mozambique) and the other in Omereque (Bolivia).

Comparing the values of five DSO's indicators as LV circuit length per LV consumer, number of LV consumer per MV/LV substation, MV/LV substation capacity per LV consumer, number of MV supply points per HV/MV substation and the typical transformation capacity of MV/LV secondary substations in rural areas, the results obtained are promising as they already fit with literature, supporting the good performance of the algorithms.

Keywords: Rural Electrification Planning, Optimal Distribution Planning, Clustering Analysis, Substation Siting, GIS, k-means, LUKES Algorithm.

Sommario

Nel 2019, quasi un miliardo di persone non aveva accesso all'elettricità: circa il 50% si trova nell'Africa subsahariana e l'87% di esse vive nelle zone rurali. In un quadro globale in cui l'obiettivo del 7° SDG è quello di permettere a tutti di usufruire di un'energia accessibile, affidabile, sostenibile e moderna, è necessaria una pianificazione dell'elettrificazione rurale.

Questa tesi è incentrata sull'accesso all'elettricità e si concentra sulla creazione di infrastrutture elettriche rurali e sulla fornitura di connettività alle famiglie. L'obiettivo è quello di progettare una procedura in grado di valutare l'ubicazione ottimale delle sottostazioni secondarie dal punto di vista topologico. Lo scopo è nell'identificazione dei siti ottimi per le sottostazioni MT/BT, tenendo in conto la distribuzione spaziale della popolazione, ottenuta da dati georeferenziati. A tal fine è sviluppata una procedura in due fasi.

Il punto di partenza è una grande area popolata, con una diversa densità di popolazione. Poiché non è economicamente conveniente elettrificare le aree dove non c'è un numero minimo di persone, il primo passo consiste nel raggruppare le aree densamente popolate per trovare le aree più dense in termini di abitanti. L'algoritmo di clustering scelto per questo compito è il DBSCAN. Il secondo passo consiste nell'applicare un altro algoritmo di clustering ai gruppi ottenuti con DBSCAN per dividere le aree popolate in aree più piccole fornite da una sottostazione secondaria.

Per collocare le sottostazioni secondarie, sono stati studiati due diversi algoritmi per vedere i rispettivi vantaggi e svantaggi e scegliere quello che restituisce i migliori risultati. Questi due algoritmi sono k-means, basato sulla partizione, e LUKES, basato sulla teoria dei grafi.

Per valutare le prestazioni degli algoritmi, sono stati condotti due casi di studio: uno a Namanjavira (Mozambico) e l'altro a Omereque (Bolivia).

Confrontando i valori di cinque indicatori del DSO, ovvero la lunghezza del circuito BT per utente BT, il numero di utenti BT per sottostazione MT/BT, la capacità della sottostazione MT/BT per utente BT, il numero di punti di alimentazione MT per sottostazione AT/MT e la capacità di trasformazione tipica delle sottostazioni secondarie MT/BT nelle aree rurali, si ottengono risultati promettenti in quanto già in linea con la letteratura, a supporto delle buone prestazioni degli algoritmi.

Keywords: Pianificazione dell'elettrificazione rurale, pianificazione ottimale della distribuzione, analisi del clustering, localizzazione delle sottostazioni, GIS, k-means, algoritmo di LUKES.

Contents

List of Figures

List of Tables

List of Acronyms

AI = Artificial Intelligence

AICS = Italian Cooperation for Development Agency

BCSS = Between-Cluster Sum of Squares

CAPEX = Capital Expenditures

CRE = Cooperativa Regional de Electricidad

CRS = Coordinate Reference System

CURE = Clustering Using Representatives

DBCLASD = Distribution Based Clustering of Large Spatial Databases

DBSCAN = Density-Based Spatial Clustering of Applications with Noise

DENCLUE = Density Clustering

DSO = Distribution System Operator

EDM = Electricidade de Moçambique

ELFEC = Empresa de Luz y Fuerza Eléctrica Cochabamba

ENDE = Empresa Nacional de Electricidad

EPSG = European Petroleum Survey Group

GDP = Gross Domestic Product

GEOSIM = Geographic Simulation for Rural Electrification

GHI = Global Horizontal Irradiance

GIS = Geographic Information System

HCB = Hidroeléctrica de Cahora Bassa

HV = High Voltage

ICA = Imperialist Competitive Algorithm

JRC = Joint Research Centre

LAPER = Logiciel d'Aide à la Planification d'Électrification Rurale

LCOE = Levelized Cost of Energy

LREM = Local REM

LV = Low Voltage

MOTRACO = Mozambique Transmission Company

MS = Member States

MST = Minimum Spanning Tree

MV = Medium Voltage

ODSP = Optimal Distribution Substation Placement

OpenDSS = Open Distribution System Simulator

OPERX = Operating Expenses

OPTICS = Ordering Points To Identify the Clustering Structure

OSM = OpenStreetMaps

PLABER = Plan Bolivia de Electrificación Rural

PSO = Particle Swarm Optimization

QGIS = Quantum GIS

REM = Reference Electrification Model

RNM = Reference Network Model

SETAR = Servicios Eléctricos Trija, S.A.

SCCER-FURIES = Swiss Centre for Competence in Energy Research on the Future Swiss Electrical Infrastructure

SDG = Sustainable Development Goal

SIN = Sistema Interconectado Nacional

SSE = Sum of Square Errors

STING = Statistical Information Grid-based method

TDE = Transportadora de Electricidad

TSO = Transmission System Operator

WVD = Weighted Voronoi Diagram

Chapter 1

INTRODUCTION

A well-established energy system supports all sectors, from businesses, medicine, education to agriculture, infrastructure, communications and high technology. So, energy for all is essential. Access to electricity is a prerequisite for a society if it wants to move out of subsistence.

In 2000, the United Nations Millennium Declaration asserted that every individual has dignity; and hence, the right to freedom, equality, a basic standard of living that includes freedom from hunger and violence and encourages tolerance and solidarity. From this moment, the United Nations began to establish development goals. First, the Millennium Development Goals and then the 17 Sustainable Development Goals. The SDGs coincided with the historic agreement COP21 Paris Climate Conference. So, the SDGs not only reaffirm the commitment to end poverty but also to build a more sustainable, safer and more prosperous planet for all humanity.

The 7th SDG is "Affordable and Clean Energy", and its goal is to ensure access to affordable, reliable, sustainable and modern energy for all by 2030 [1].

In 2016, slightly less than 1 billion people did not have access to electricity in the world. Approximately, 50% of them are found just in Sub-Saharan Africa and 87% of these people live in rural areas. Although access to electricity in poorer countries has begun to accelerate, a lot more work still needs to be done, as for example 3 billion people lack access to clean cooking fuels, resulting in 4 million premature deaths each year. The need of clean, sustainable and affordable alternatives is real [1].

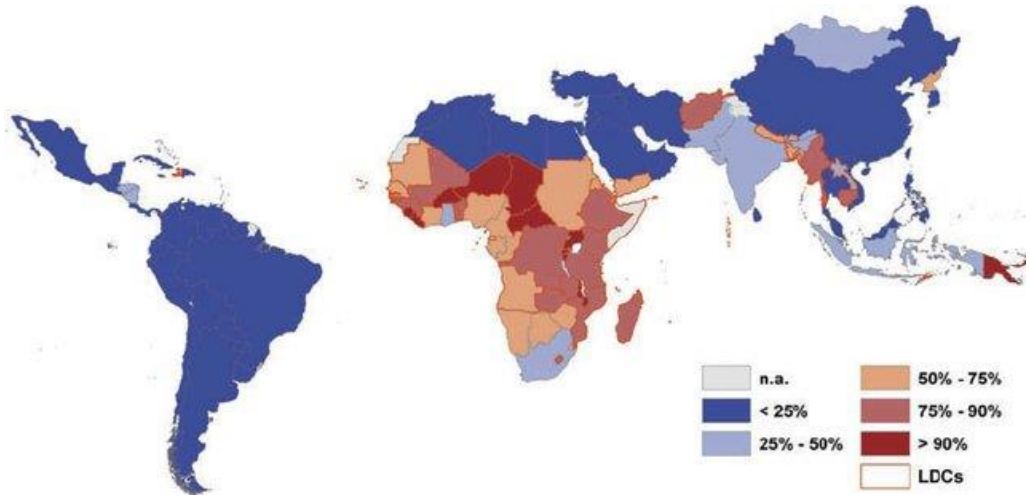

Figure 1. Share of people without electricity access for developing countries [2]

In order to bring electricity to rural areas, rural electrification planning is needed. Rural electrification plans are means of optimizing electricity services in a given area, within a given period of time in accordance with pre-defined strategic objectives. Rural electrification programs increase access to grid connections and the type of off-grid options available to rural households.

Electrification plans commonly depict intended developments in the field of electrification based on a spatial analysis of domestic demand, socio-economic activities and load forecasts, discerning between available electrification options like grid extension, mini-grids or off-grid systems.

The four foremost steps of rural electrification planning are [3]:

- *Planning approach*: can be technical-economic, based on the optimization of an economic criterion; or multi-sector approach, introducing also qualitative perspectives.
- *Primary Data Collection*: collection of information of existing and planned electricity grids structures, location and size of population centres, infrastructure in the fields of health, education and economic institutions; and the assessment of renewable energy potential.
- *Electricity Demand Modelling*: strongly related to the type and quality of available primary data. There are two basic approaches for demand modelling: the top-down approach, where demand is forecast through the application of econometric methods; and bottom-up models, which create demand profiles on the basis of socio-economic surveys, utilizing a higher degree of data input to achieve a higher level of precision in the modelling process.
- *Technical-economic Optimization*: assessment of electrification solutions (network expansion, mini-grid, stand-alone system) and the optimization of the corresponding installation set-ups in accordance with the criteria defined previously in the step of planning approach.

Rural electrification programs seem to be crucial to improve living conditions and promote development. However, to determine whether or not interventions are relevant and cost effective, an impact evaluation should be conducted over these programs [4].

According to [4], the expected outcomes and benefits from rural electrification are:

- Economic benefits: income benefits through new opportunities of work, increased productivity of home businesses and agricultural activities;
- Social benefits: time savings from household chores which can be used for leisure and productive activities, education benefits through higher earnings for children living in electrified households that have higher educational attainment, improved health outcomes and reduced mortality through improved indoor air quality from changes in lighting source and increased public security;
- Environmental benefits: lower environmental contamination.

Most of the econometric papers found in the literature are relative to specific case studies, so the impact is often evaluated in an area or region. This means that it is very difficult to capture all these benefits and identify causal relationships.

Although rural electrification planning seems to be the solution to reach the 7th SDG and promote development in rural areas, there are some problems:

- The limited and difficult access to data in developing countries. The data include geographic parameters such as the availability of energy resources, distance to grid and between localities, or the dispersal of housing in a given area, topologic expediency and the prevalence of natural constraints (forests, protected areas, etc.). Usually, in order to collect it, it is necessary to go directly to the places and do a manual collection by surveys and measurements, as there exist little literature and data available on the subject for developing countries.
- Selection of the area to electrify. The link of causality between a rural electrification program and the impacts is not identified by simple before-and after comparisons or connected and non-connected groups conditional on having access to the grid because households that connect to the grid are likely different in unobservable ways to the households that decided not to connect.
- Endogenous infrastructure placement. Program designers would place the electric grid in areas where they are likely to get higher paying customers, in denser population areas, etc., which would bias comparison between connected and non-connected areas.
- Objective function used by policymakers and programs designers when deciding what projects are cost effective. The evidence suggests that a planner using cost functions or profit functions as objective functions would make different decisions. This is due to the quasi-concavity of the production function and complete markets, situations that are not characteristic of the electricity sector.
- Difficulty in identifying properly measure indicators in the planning stage. Quantitative metrics are needed in order to evaluate the performance of the plans and identify the best solutions.

In essence, the high cost of providing electricity in low populated, remote places with difficult terrain and low consumption results in rural electricity schemes that are usually more costly to implement than urban schemes. In addition, low rural incomes can lead to problems of affordability, and the long distances mean greater electricity losses and more expensive customer support and equipment maintenance. Despite this, rural electrification has been claimed to have substantial benefits, promoting production and better health and education for households. Moreover, in the report of the Independent Evaluation Group of the World Bank, shows that consumer willingness to pay for electricity is almost always at or above supply cost [4]. So, there is a need to better analyse the situation.

In today's context, rural electrification has five major facets [5]:

- Setting up of rural electricity infrastructure;
- Providing connectivity to households;
- Adequate supply of desired quality of power;
- Supply of electricity at affordable rates;
- Providing clean, environmentally benign and sustainable power in efficient way.

As discussed, rural electrification planning is not an easy problem to overcome as connection to the grid requires very high investments, and a good planning and estimation of the future cost is necessary to be able to achieve the goal of bringing electricity to everybody.

Related to the setting up of rural electricity infrastructure, the power system planning is one of the important things for obtaining the economic operation of the power system. The power system is composed by the transmission and distribution system. In the power distribution system, the minimum cost of operation can be obtained by a proper planning and design of the distribution network in order to reduce the losses and give access to electricity to as many people as possible [6].

The distribution system is the closest network to the customer so a good design of it is basic for a good quality of service. The economic and reliable design of distribution networks is a main challenge of distribution network companies. A distribution system consists of medium voltage (MV) and low voltage (LV) networks. The objective of an MV network planning is to determine the location and rating of HV/MV substations and MV feeder routes and the objective of an LV network planning is to determine the locations and ratings of distribution transformers and LV feeder routes. However, most of these researches are focused on the MV level and less attention has been devoted to LV networks [7].

The main part of the losses associated with a distribution system belongs to its LV networks. Moreover, the cost of MV networks of a distribution system is comparable to that of its LV networks. Therefore, it is also necessary to pay more attention to the LV networks of the distribution system [7].

This project focuses on access to electricity and it is centred on providing connectivity to households by studying the siting of secondary substations, framed in the LV network planning. Being the LV network the closest network to the customer, it is needed to consider each single user/house, and this causes a huge amount of data to be processed. Consequently, KPI are necessary to describe the problem and the performances of the algorithms under test.

In particular, the objective of this work is to develop a procedure able to evaluate the optimal location of secondary substations from a topological perspective, not an economic assessment.

The main goal is to find the best approach to divide the area in small clusters that would be supplied by a secondary substation and how to site them. The aim of this clustering phase is to place the minimum number of substations that could meet the load of the cluster considering distance and power constrains.

Considering this information, the structure of this project is the following:

Chapter 2 presents how is the transmission and distribution systems in Europe versus developing countries to keep in mind how is the reality and how to implement the distribution system characteristics in the procedure.

Chapter 3 presents the state of the art of some of the tools for rural electrification planning found in the literature and presents an introduction to GISEle, a tool under development by the *Politecnico di Milano*.

Chapter 4 presents the state of the art relative to secondary substations siting, useful to understand the methods used in literature for the optimal placement of substations.

Chapter 5 continues with the state of the art of the different clustering algorithms and optimization methods. It exposes an analysis of the options for the clustering of densely populated areas to detect the areas where to place the substations.

In Chapter 6, the proposition of this work is presented with the motivation of this work. In order to site the secondary substations, two paths have been considered: one implementing the clustering algorithm k-means and the other with graph-partitioning theory, implementing LUKES algorithm.

To test the results of the procedure two case studies have been conducted: one in Namanjavira (Mozambique) and the other in Omereque (Bolivia). Chapter 7 presents the description of the case studies, with the context of the countries, the energy sectors and the input data.

Chapter 8 presents the results obtained in both case studies.

Finally, in Chapter 9 the conclusions obtained after the development of the project are discussed with the objective accomplished and the further direction of improvement for the next future.

Chapter 2

ELECTRIC POWER DISTRIBUTION SYSTEMS AROUND THE WORLD

Electric power systems are constituted by different components which allow the generation, transmission, distribution and consumption of the electrical energy.

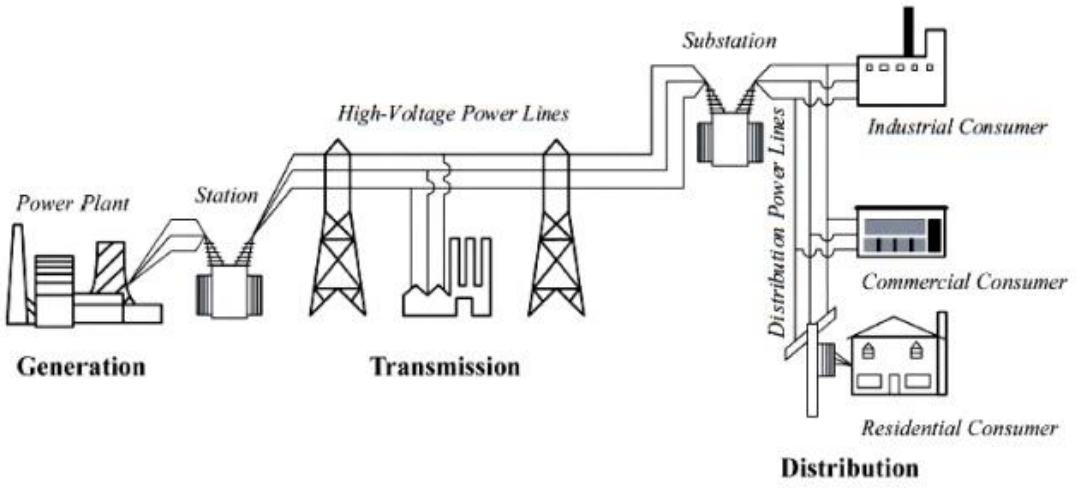

Figure 2. Electric Power System

Generation is the process of generating electric power from sources of primary energy. Electricity is obtained at 11 kV approximately and goes directly to a station. The station purpose is to bring the generation voltage (11 kV) up to the transport voltage (400 kV-250 kV).

The electric power transmission system is the bulk movement of electrical energy from the generation site to an electrical substation. The interconnected lines are known as the transmission network and they are part of the power grid. Electricity is transmitted at high voltages, usually 66 kV or more, to reduce energy losses due to the long distances of the lines.

Electric power distribution is the final stage in the delivery of electric power; it carries electricity from the transmission system to individual consumers. It is composed by a primary substation, medium voltage lines, secondary substations and low voltage lines. In the primary substations, the current is transformed from high voltage to medium voltage (HV/MV). The medium voltage level could be between 6 and 60 kV depending on the country. The MV lines, directly supply the commercial and industrial consumers or the secondary substations which transform current

from MV to LV. This low voltage is around 400-230 V and is the one used by residential consumers.

Historically, transmission and distribution were monopolies owned by the same company. But since the earlies 90s, many countries liberalized the regulation of the electricity market and these two systems separated from each other. The transmission system is operated by the TSO (transmission system operator) and the distribution system by the DSO (distribution system operator).

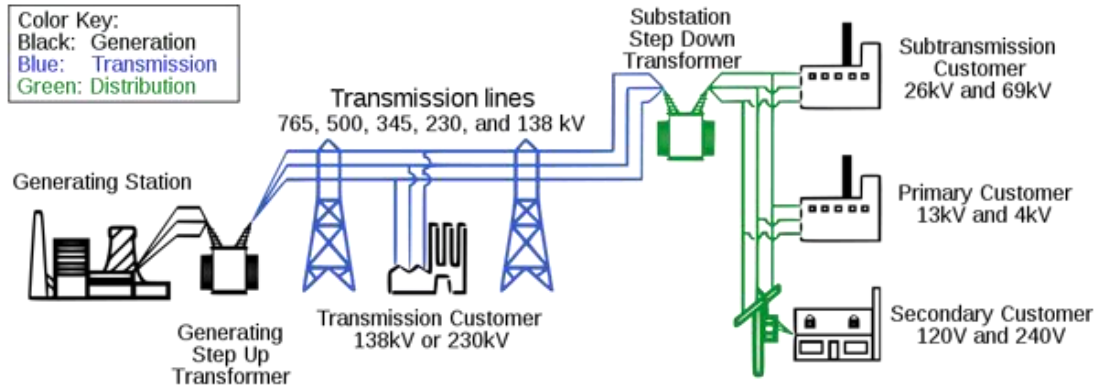

Figure 3. Electric Power System and typical voltages of each phase

The power distribution system is different from one country to another and it also differs from developed countries to developing countries. In this chapter, a general structure of the distribution system in Europe is presented and compared with the one of developing countries.

2.1 STANDARD DISTRIBUTION SYSTEMS IN EUROPE

The electric distribution system varies from one country to another. So, in Figure 4, a general description of the electric distribution system in Europe in presented.

Firstly, the primary substation, which converts HV to MV, is usually between 16 and 63 MVA. Secondly, the MV lines, at a voltage level between 15-20 kV, have a length of 20 km approximately. Then, the secondary substations, which converts MV to LV, are between 100 and 630 kVA, and finally the LV lines have a length of 1 or 2 kilometres.

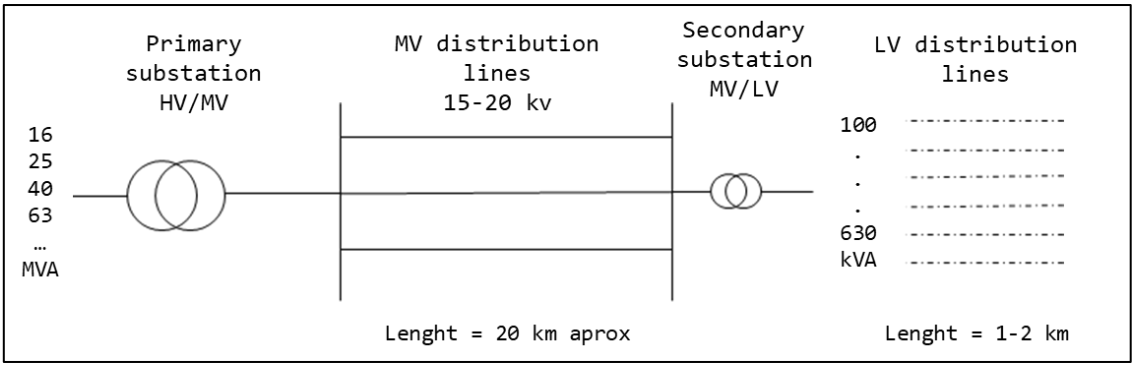

Figure 4. Scheme of the electric distribution system in Europe

According to recent studies [8], there are 2,400 electricity distribution companies in Europe which deliver a total of 2,700 TWh a year to 260 million connected customers, 99% of which are residential customers and small businesses. The total power lines length of all DSOs is around 10 million km, which are connected to the transmission system through 10,700 interconnection points. Furthermore, there are more than 4 million distribution transformers in the whole Europe which reduce the voltage levels from high to medium and low voltage [9].

There is huge variety regarding the number of DSOs in each member state. Whereas some countries have only one (e.g. Ireland, Lithuania) or a few (e.g. Slovakia, Bulgaria, Hungary), countries like France, Poland or Germany with more than 150, 180 and 800 distribution companies, respectively, have a sector structure being shaped by the presence of many small-scale DSOs supplying a relatively small area with a limited number of connected customers [10].

With the data given by 99 out of 191 European DSO [10], it has been possible to build 37 different indicators to compare the distribution systems among the different countries. These indicators are divided in three categories: network structure and reliability indicators, network design and distributed generation.

A subset of the total DSO indicators used to build the large-scale representative distribution networks are listed in Table 1. In the following figures the green and red lines will respectively show the average and median values of the parameters under analysis and each bar of the x axis represents a DSO [10].

Table 1. Subset of DSOs indicators [10]

ID	DSOs indicators
1	LV circuit length per LV consumer
2	Number of LV consumers per MV/LV substation
3	MV/LV substation capacity per LV consumer
4	Number of MV supply points per HV/MV substation
5	Typical transformation capacity of MV/LV secondary substations in rural areas

1. **LV circuit length per LV consumer**: this indicator is based on the location and distribution of LV consumers, as well as the distance among them. The figure shows a narrow gap among DSOs with a median (0.025 km/LV consumer) and an average (0.03 km/LV consumer) value almost identical.
   Higher values identify DSOs which are serving rural areas where indeed consumers (houses) are more spread (lower population density). Note that longer cables imply higher CAPEX and OPEX for the DSO which is obliged to distribute electricity even in areas where it is not cost-effective.

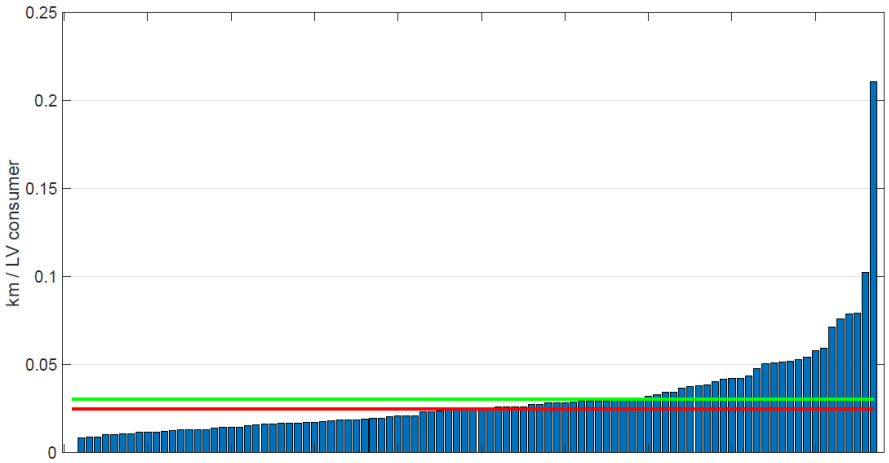

Figure 5. LV circuit length per LV consumer [10]

2. **Number of LV consumers per MV/LV substation**: The number of LV consumers per MV/LV substation is presented with a median value of 76 and an average value of 86. This ratio strongly depends on the spread of consumers in the supplied area. In urban areas, due to higher density, this ratio is higher compared to rural areas, where customers are less concentrated.

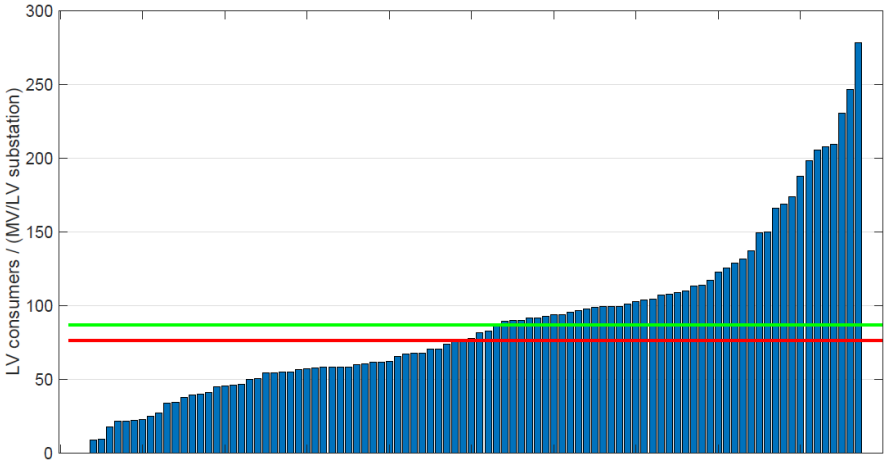

Figure 6. Number of LV consumers per MV/LV substation [10]

3. **MV/LV substation capacity per LV consumer**: it gives an indication on how much power (in kVA) is installed in a MV/LV substation for each LV consumer. This parameter depends on the typical peak average power of consumers, energy efficiency of the devices and the simultaneity factor. This latter depends on the size of the house and the number of people hosted per household. Furthermore, the higher the peak power of consumers the higher the capacity of the substation. The median capacity of MV/LV substation per LV consumer is 3.88 kVA with an average value of 4.76 kVA.
   The red dots represent the distribution transformer capacity utilization, which is defined as the distributed electricity in (MWh)*100 divided by total

distribution transfer capacity previously multiplied by 8,760 hours of one year. It indicates the effectiveness of distribution planning in matching transformer capacity with demand.

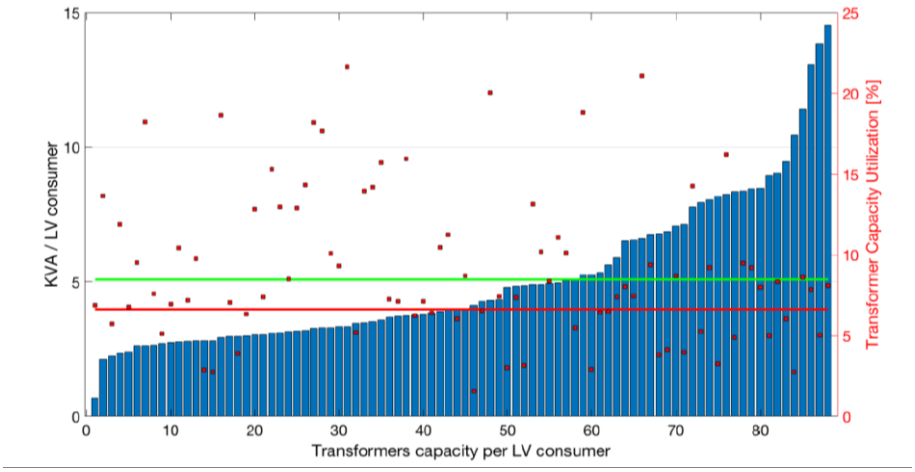

Figure 7. Transformers capacity per LV consumer [10]

4. **Number of MV supply points per HV/MV substation**: As known, the HV/MV substations supply electricity to MV supply points (as MV consumers and MV/LV substations) and are in charge of connecting MV distributed generation if present in the area. The MV consumers and MV/LV substations are distributed along feeders, and therefore the number of MV supply points per HV/MV substation is the product of the number of feeders of the substations and the average number of MV supply points per feeder. The median value is 126.75.

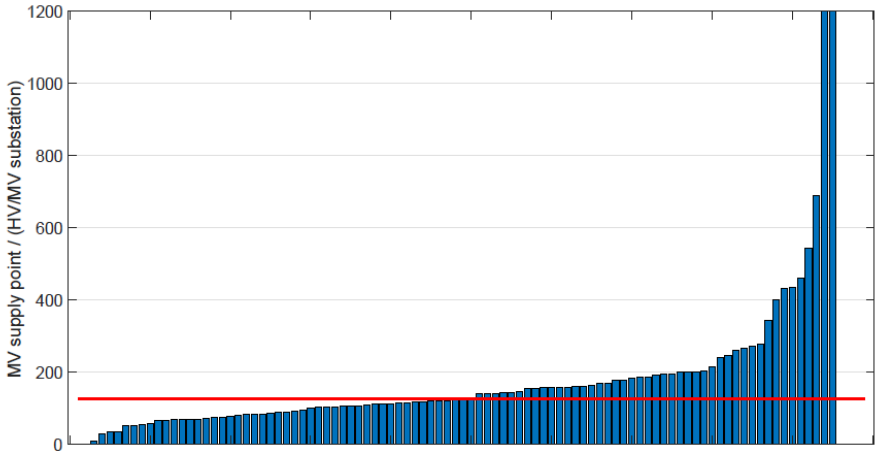

Figure 8. Number of MV supply points per HV/MV substation [10]

5. **Typical transformation capacity of MV/LV secondary substations in rural areas**: They are generally lower than in urban areas, due to the higher distances existing between consumers and the reduced electricity density.

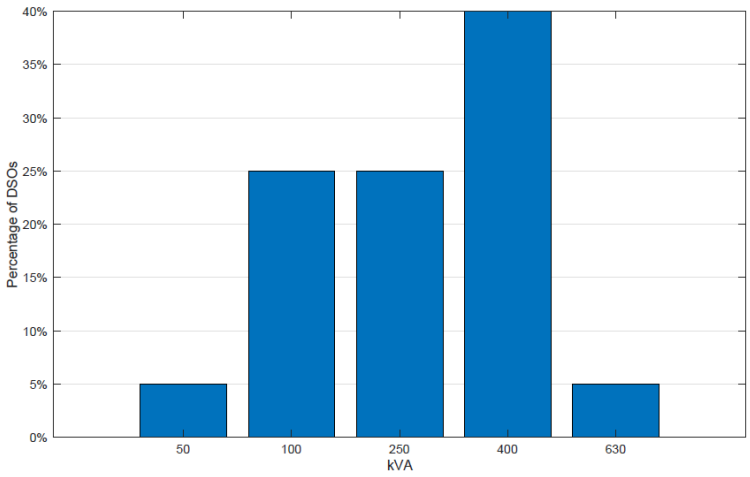

Figure 9. Typical transformation capacity of MV/LV secondary substations in rural areas (kVA)

Generally, the transformer's ratios for primary substations at transmission level and for secondary substations at distribution level are aligned among the member states. The most typical capacity for primary substations (HV/MV) is 25, 30, 40, 80 and 100 MVA. Based on the result of the DSO Observatory Survey the frequent values for transformation capacity of the MV/LV secondary substations are different for urban and rural areas. With respect to the urban case, the capacity is typically 400 kVA or 630 kVA. In rural areas due to a lower simultaneity factor considered in the planning of the grid but also due to the lower energy density the capacity values are 100 kVA, 250 kVA and in most of the cases up to 400 kVA [10].

A resume of the presented indicators can be found in the following table.

Table 2. Subset of the DSOs indicators and their values for Europe

ID	DSOs indicators	Average value	Median Value
1	LV circuit length per LV consumer	0.03 km	0.025 km
2	Number of LV consumers per MV/LV substation	86	76
3	MV/LV substation capacity per LV consumer	4.76 kVA	3.88 kVA
4	Number of MV supply points per HV/MV substation	-	126.75
5	Typical transformation capacity of MV/LV secondary substations in rural areas	50 (5%), 100 (25%), 250 (25%), 400 (40%), 630 (5%) kVA	

As it is shown in Figure 4, the typical values for the transformers of the MV/LV substations go from 50 kVA to 1000 kVA, varying from rural to urban areas, predominating the 400 and 630 kVA transformers. Approximately, between 76-86 people are connected to a MV/LV substation. However, this parameter has a wide range, with a maximum of more than 250 people connected to a minimum of less than 10 (see Figure 6). The length of LV lines can be calculated with the LV circuit

length per LV consumer and the number of LV consumers per MV/LV substation. This leads to a length of 1.9-2.5 km of LV line per secondary substation.

Figure 10 illustrates the shares of LV, MV and HV line lengths for each member state. In terms of percentage of the total length of the voltage lines, the LV lines represents 59.8% of the total lines, the MV 36.8% and the HV amounts for the 3.4%.

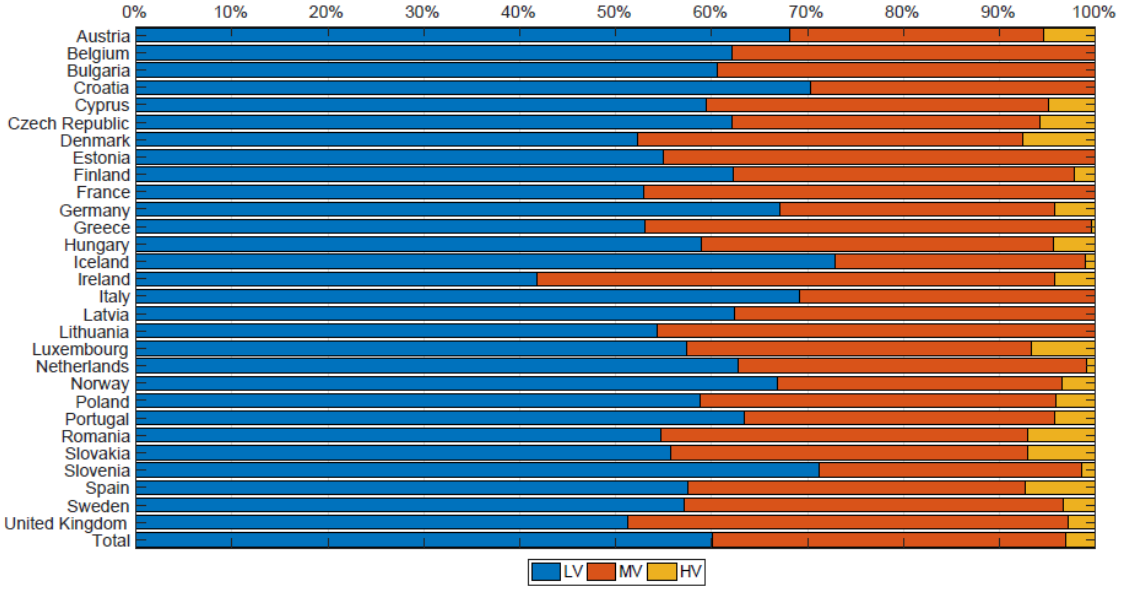

Figure 10. Share of distribution length lines per type (LV, MV, HV) [10]

2.2 DISTRIBUTION SYSTEMS IN DEVELOPING COUNTRIES

A typical scenario in developing countries is to have large power companies supported by the government, either through state ownership or by obtaining government-issued bids, and typically vertically integrated. The two main problems are power theft, that accounts for huge losses, and the market failure, as the electricity demand usually exceeds the production capacity. These two problems place a burden on the consumers, as they must cope with frequent outages and increased tariffs, and on the providers, who face difficult financial and infrastructural challenges [11].

The structure of the distribution system in developing countries is different from the European one and differs from one country to another. In 2015, 28% of the population in Europe lived in rural areas, whereas in developing countries was the 51% [12]. Due to the level of electrification and the predominance of rural areas in these countries, the length of the lines and the power of the transformers used are different. Rural electrification systems tend to use higher distribution voltages because of the longer distances covered by distribution lines.

A general description of the electric distribution system is presented in Figure 11. Firstly, the primary substation is usually between 6 and 10 MVA. Secondly, the MV

lines, which goes between 33-60 kV, have a length of 100 km approximately. Then, the secondary substations have a nominal power between 10 and 20 kVA, and finally the LV lines have a length of 100 meters approximately.

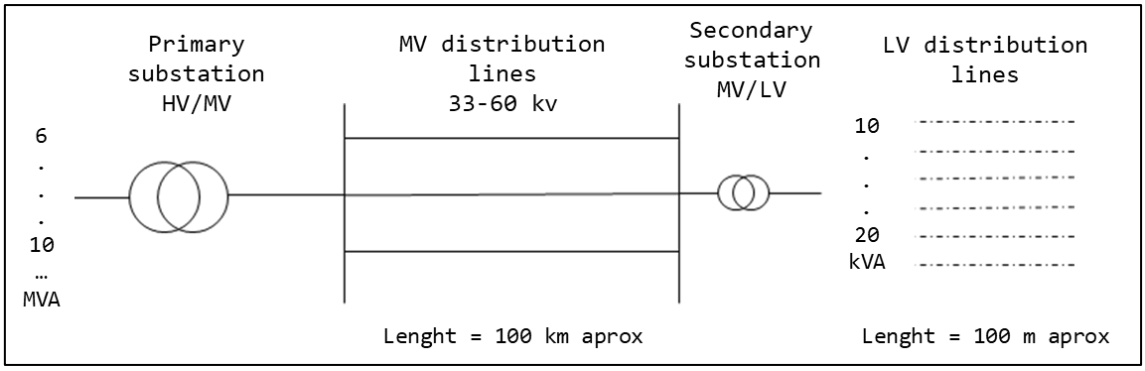

Figure 11. Scheme of the electric distribution system in developing countries

As it does not exist a report as the one of the JRC of the European commission, there is a need to better analyse the situation. To do so, the distribution system of two different sub-Saharan countries are presented: Uganda and Mozambique. Uganda has been selected due to the availability of detailed data related to its distribution system, not common in this context, while the analysis of Mozambique is helpful for one of the case studies of the thesis work, which focuses on a specific rural area of the country.

2.2.1 Uganda

Uganda is a developing country with 76% of rural population. The energy system is managed by both government and the private sector. Since 1954 to 2008, the number of consumers has increased from 11,432 to nearly 300,000 [11].

The world bank estimates the country has a generation capacity of 340 MW at its major generating facilities while the peak demand is nearly 380 MW. There is a market failure even without losses in the transmission and distribution system. Power is generated primarily by hydroelectric plants, which are subject to natural and man-made fluctuations. Because of this, the flow of electricity has become inconsistent and unpredictable [11].

The transmission system is handled by the Uganda Electricity Transmission Co. Ltd. (UETCL) at 132 kV, and the distribution system by the Uganda Electricity Distribution Co. Ltd. (UEDCL) at 33 kV. The average consumer interacts solely with the distributors, to whom they pay their electricity bills and issue any complaints [11].

Data has been extracted from the webpage of Energy Sector GIS Working Group Uganda [13] . Figure 12 shows transmission substations (blue dots), the distribution substations (pink dots) and the distribution lines (blue lines).

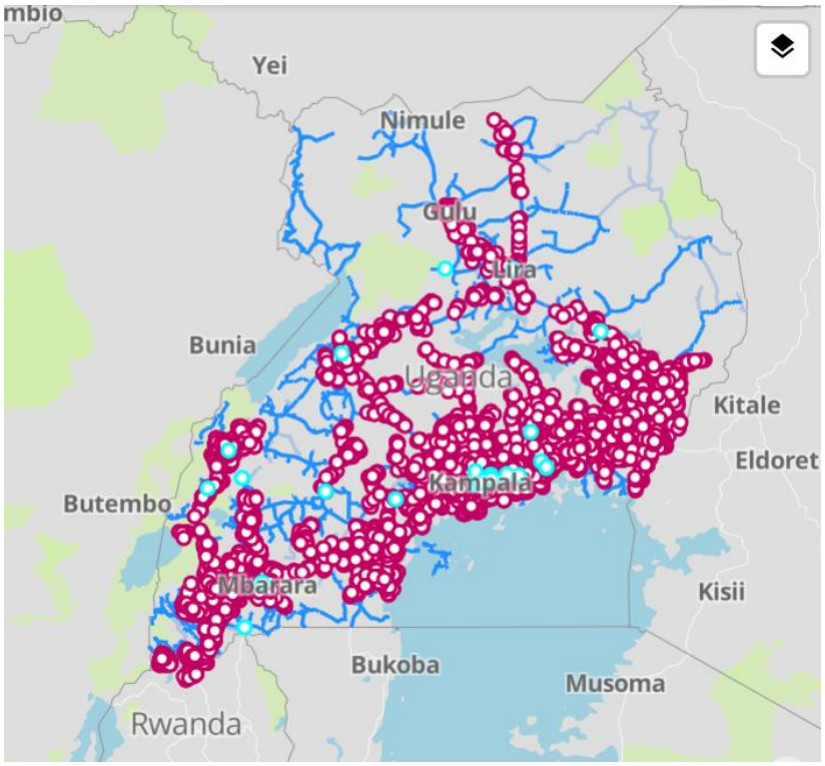

Figure 12. Transmission substations (blue dots), distribution lines (blue lines) and distribution substations (pink dots) of Uganda in 2018 [13]

Figure 13 presents a more precise image of the distribution system of Uganda.

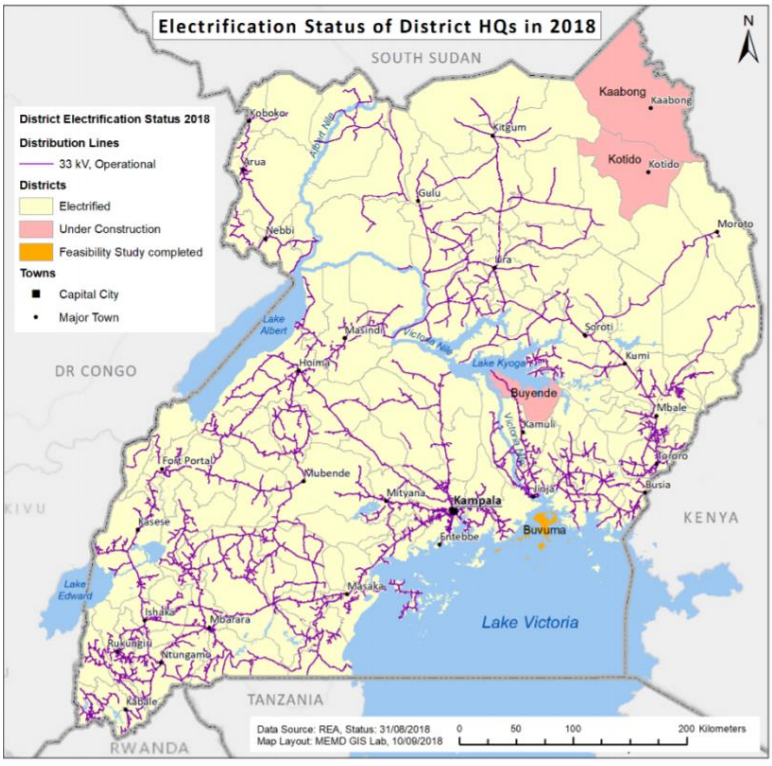

Figure 13. Electric distribution system of Uganda in 2018

According to [14], the characteristics of the distribution system in Uganda are the following:

Table 3. Distribution System characteristics of Uganda

Characteristics	Year 2018
Total Network line length (km)	34,000
MV Network line length (km)	14,105
LV Network line length (km)	20,200
Distribution transformers	12,000
Distribution transformer capacity (MVA)	1,900
Customers on the grid ('000)	1,200

From this data, the DSO indicators are calculated and shown in the following table:

Table 4. Subset of the DSOs indicators and their values for Uganda

ID	DSOs indicators	Value
1	LV circuit length per LV consumer	0.017 km
2	Number of LV consumers per MV/LV substation	100
3	MV/LV substation capacity per LV consumer	1.58 kVA
4	Number of MV supply points per HV/MV substation	-
5	Typical transformation capacity of MV/LV secondary substations in rural areas	158.3 kVA

2.2.2 Mozambique

Mozambique is a developing country with 64% of rural population [12]. Although it has one of the largest power generation potential in Southern Africa, only 29% of the population (15% in rural areas and 57% in urban areas) has access to electricity due to limited transmission and distribution networks and an unfavourable market [15].

The national grid is largely managed by state-owned utility Electricidade de Moçambique (EDM). A small proportion of the lines are owned by Hidroeléctrica de Cahora Bassa (HCB), the operator of the Cahora Bassa hydroelectric plant, and by Mozambique Transmission Company (MOTRACO), which supplies power to the Mozal aluminium smelter owned by BHP Billiton [16].

The transmission system is composed by three separate systems, northern, central and southern. The lines operate at 220 kV or 110 kV [16].

In 2017, the length of the MV lines was 17,580 km with a power of 2,378 MVA. However, EDM is carrying out numerous projects to develop the distribution system. These projects consist of 66 kV lines, 110/66 kV substations, distribution networks with 33 kV medium voltage lines, low voltage lines, 110/33 kV (10 MVA) and 66/33 kV (10MVA) substations, service connections, consultancy services for preparation of detailed design and supervision of works and project audit [17].

Figure 14 shows the transmission system of Mozambique.

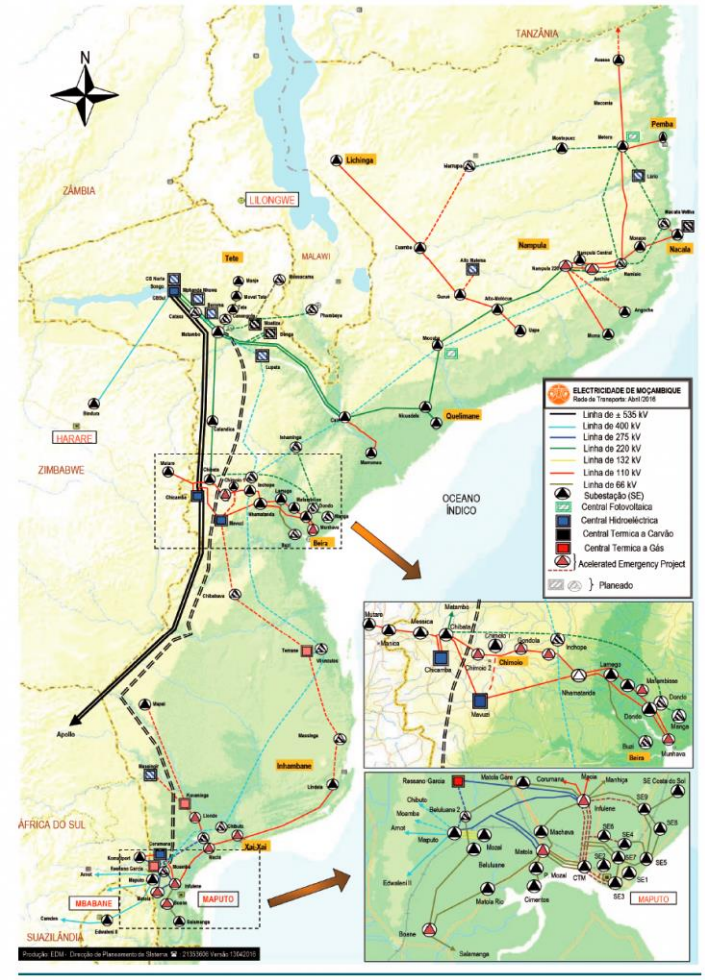

Figure 14. Transmission system of Mozambique

According to EDM, the characteristics of the transmission and distribution system in Mozambique are:

Table 5. Distribution System characteristics of Mozambique

Characteristics	Mozambique (2018)
LV Network line length (km)	26,109 *
MV Network line length (km)	17,580
HV Network line length (km)	5,420
Distribution transformers	10,822
Distribution transformer capacity (MVA)	2,378
Customers on the grid ('000)	1,200
Transmission transformers	69

* Estimated with the relation between the MV lines and LV lines in the province of Nampula (Data from [18])

From this data, the DSO indicators for Mozambique are calculated and shown in Table 6:

Table 6. Subset of the DSOs indicators and their values for Mozambique

ID	DSOs indicators	Value
1	LV circuit length per LV consumer	0.022 km
2	Number of LV consumers per MV/LV substation	110
3	MV/LV substation capacity per LV consumer	1.98 KVA
4	Number of MV supply points per HV/MV substation	156.8
5	Typical transformation capacity of MV/LV secondary substations in rural areas	220 kVA

2.3. DISTRIBUTION SYSTEMS IN EUROPE VS. DEVELOPING COUNTRIES

As can be seen in Figure 4 and Figure 11, and the data provided for different countries, the structure of the electric distribution system in Europe and developing countries are different.

The average voltage levels of transmission and distribution systems are reported in Table 7.

Table 7. Comparison of the power system of Europe and developing countries

Location	HV lines	MV lines	LV lines
Europe	220 – 400 kV	11–12 kV	230/400 V
Developing countries	110 – 500 kV	20 – 66 kV	220/230/400 V
Uganda	132 kV	33 kV	230/400 V
Mozambique	220 –110 kV	66 – 33 kV	230/400 V
Indonesia [6]	500 kV, 275 kV	20 kV	220 V

The comparison of the DSO indicators between Europe and developing countries is presented in Table 8:

Table 8. Comparison of the DSOs indicators in Europe vs. developing countries

ID	DSOs indicators	Europe	Uganda	Mozambique
1	LV circuit length per LV consumer	0.03 km	0.017 km	0.022 km
2	Number of LV consumers per MV/LV substation	86	100	110
3	MV/LV substation capacity per LV consumer	4.76 kVA	1.58 kVA	1.98 KVA
4	Number of MV supply points per HV/MV substation	126.75	-	156.8
5	Typical transformation capacity of MV/LV secondary substations in rural areas	100, 250, 400 kVA	160 kVA	220 kVA

The following conclusions are extracted:

- The LV circuit length per LV consumer is lower in developing countries.
- The number of LV consumers per MV/LV substations is higher in developing countries.
- The MV/LV capacity per LV consumer is lower in developing countries.
- The number of MV supply points per HV/MV substation is higher in developing countries.
- The typical transformation capacity of MV/LV secondary substations in rural areas is lower in developing countries.

These conclusions are backed by the lower population density, the lower power demand, lower energy efficiency, lower simultaneity factor and lower electricity density in developing countries.

The distribution system is an important part of the electric power system, accounting for almost 60% [10] of the voltage lines and hence most of the costs and losses. An introduction of it in the simulation tools is necessary to have an accurate estimation of the reality.

Chapter 3

STATE OF THE ART: TOOLS FOR RURAL ELECTRIFICATION

Stated the characteristics and limits of distribution systems in rural areas of developing countries, it is important to have well defined strategies for planning new networks. In the literature, numerous tools for rural electrification planning developed from very different perspectives are found.

Some examples of tools are: REM, RNM, Network Planner, GEOSIM, LAPER, SOLARGIS, ENERGIS, LEAP, GISELEC, NORIA, NEPLAM, GIPSY, POWERWORLD, Electrification Planning Decision Tool RAPS, LAP, VIPOR, Calliope, ECOWREX, OnSSet, Homer Energy Pro, RE2Naf, RETSCREEN.

Lack of spatial information in rural and regional level is one of the main problems for development practitioners, government officials and local level planners. Geospatial planning facilitates the understanding of spatial aspects of social and economic development by relating socio-economic variables to natural resources and the physical world, providing a tool for targeting interventions and monitoring impacts on various scales over wide areas.

Due to the importance of geospatial planning, REM, RNM, Network Planner, GEOSIM and LAPER have been chosen among the literature as example of rural electrification planning tools, and they are analysed and compared.

3.1 REM

The Reference Electrification Model (REM) [19], is a computer model developed by the MIT and Comillas Universal Energy Access Laboratory. Its purpose is to support large-scale electrification planning and local electrification projects (LREM).

The inputs for the model are information about building locations, solar irradiance, topography, grid extent and reliability, expected consumer demand, fuel costs and infrastructure costs. After running a series of clustering and optimization algorithms specifically designed for electrification planning, REM produces lowest-cost system designs.

Individual consumers are grouped into electrification clusters so that total system costs (actual and social costs) are minimized. These clusters may denote groups of customers to be connected to separate mini-grid systems, groups to be connected to

the existing grid, or clusters of single customers to be supplied with stand-alone systems.

The customers are grouped into a hierarchical structure of off-grid and grid extension clusters. The clustering process consists of a bottom-up greedy algorithm that join customers into groups if the expected cost of being connected is lower than the expected cost of being electrified separately. The only goal of the clustering process is to deliver a well-defined, compact, and meaningful structure of clusters to be thoroughly explored in the final design phase.

When evaluating the cost of the internal network of a mini-grid or of a grid extension, REM has to design the minimum cost network that meets all prescribed technical requirements. This network-design process has to be done numerous times (many alternative subsystems or clusters), even for problems of moderate size. For this task, REM employs the greenfield network-design software called the Reference Network Model (RNM) [19].

However, a different strategy, currently under development [19], is to start with a large grid-extension cluster that contains all the consumers, and then proceed with a bottom-up evaluation of disconnections.

The first step of this top-down strategy is to calculate a detailed network design connecting everyone to the network. Then, REM would systematically evaluate the removal of lines and transformers, considering the least cost scenario. Cost reductions in the grid would be compared with the cost of electrifying the corresponding downstream consumers with off-grid systems.

This clustering algorithm would determine which consumers are better electrified with grid extension designs and which are left in off-grid systems. However, it would still be necessary to find the optimal off-grid solution.

The following outputs are typically obtained when REM is applied to some specific territory [19]:

- The optimal groupings of individual consumers into electrification clusters so that total system costs are minimized. These clusters may denote groups of customers to be connected to separate mini-grid systems, groups to be connected to the existing grid, or clusters of single customers to be supplied with stand-alone systems.
- The optimal generation mix and network layout for each of the off-grid mini-grids.
- The optimal network layout for each cluster that will be connected to the grid.
- A detailed description of the optimal plan with information pertinent to decision-making including total cost breakdowns, expected reliability data, GIS files specifying network layouts, generation and storage specifications, bills of materials, summary charts, geo-referenced maps of system designs, and reports in text and spreadsheet formats.

3.2 RNM

The Reference Network Model (RNM) is a large-scale distribution planning tool that can help regulators to estimate efficient costs in the context of incentive regulation applied to distribution companies developed by the *Instituto de Investigación Tecnológica* of Comillas University [20].

The model designs the minimum cost network that meets quality-of-service specifications, using a user-provided catalogue of equipment to specify distribution infrastructure down to the individual consumer-level, and also considering geographical constraints.

Because the RNM uses the location coordinates of final customers as input data, it is possible to design an algorithm to automatically generate the corresponding street map from this information. The street map is represented by a graph G (V, E). The nodes are the electrical points to be supplied, the customer locations, whereas the branches E are links between nodes that correspond to feasible electrical links between customers. The algorithm developed to obtain the street map follows the steps represented in the following figure [20]:

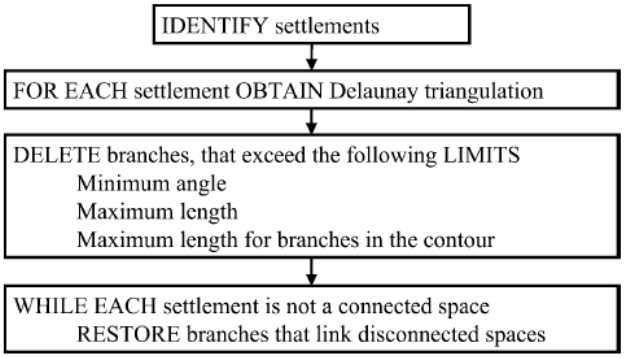

Figure 15. RNM algorithm to obtain the street map [20]

To begin with, nearby customers are grouped, and the urban areas where customers are located, are identified as settlements. A settlement is defined as a set of customers in which the maximum distance to the nearest customer is not greater than a specified threshold, 200 m for example. Then, for each settlement, a Delaunay triangulation is obtained, using the location of the electrical points to be supplied to generate all the initial branches of the graph in that particular settlement. The Delaunay triangulation is used because of the following properties [20]:

1) Every node is connected to the nearest neighbours.
2) The branches never intersect.
3) The minimum spanning tree is contained in the graph comprised of the Delaunay branches.

The third step is devoted to removing some of the initially proposed branches. Maximum length and minimum angle criteria are applied. Finally, if there are disconnected zones remaining inside a settlement, some of the branches that were previously removed, have to be restored in order to remedy this situation.

Although RNM resemble a distribution planning tool, it has significant differences with them. Its principle objective is to optimize the already existing networks, not planning from scratch.

3.3 NETWORK PLANNER

Network Planner is an online tool for planning grid, mini-grid, and off-grid electricity from the community scale to national scale. Network planner takes a host of inputs, including geo-spatial population distribution, costs of energy technologies, electricity demand and population grow-fluidity, and existing grid network, and output the least-cost solution [21].

The Network Planner is a decision support tool for exploring costs of different electrification technology options in un-electrified communities. The web-accessible model is written in Python, and developed by a team from Modi Research Group, at the Earth Institute in Columbia University, based in New York, USA. The model combines data on electricity demands and costs with population and other socio-economic data to compute detailed demand estimates for all communities in a dataset. Then, the model computes cost projections of three electrification options and proposes the most cost-effective option for electrifying communities within a specified time horizon. This helps planners both to understand costs and time frames for electrification overall, as well as to prioritise areas where grid expansion is a cost-optimized option and where other stand-alone options are preferred.

One important aspect of this modelling approach is that it predicts costs for different electricity generation technologies for each of the communities involved and thus gives the planner the freedom to explore the most cost-effective technology based on existing conditions in the community and price trend of electrification inputs during the planning period [21].

The budget available for the external components of the grid connection for the communities, namely the MV-line to connect to the nearest grid location, is divided by the cost of MV-line per metre, obtaining a key decision metric, 'MV_{max}' for each community. The MV_{max}, expressed in metres, represents the maximum length of MV-line which can be installed for each community before the cost of grid extension exceeds the cost of the least-cost stand-alone option [21].

The model applies a geospatial algorithm to compare the MV_{max} values with the actual distances between the location of unconnected communities (identified by latitude and longitude coordinates) and identifies those sites with MV_{max} values that justify grid connection. Those communities where grid extension is the most cost-effective technology for electrification are recommended by the model; in other words, they are 'grid-compatible'. The communities beyond the MV_{max} values are instead recommended for electrification using the least-cost stand-alone option [21].

3.4 GEOSIM

GEOSIM® [22] is a decision-making tool for rural electrification planning. The software is based on Geographic Information System (GIS) technology and operates with a manifold environment.

Its main innovation consists in the optimisation of energy services covering a given territory, within a given time horizon, with a view of improving the economic and social impact of rural electrification. Consequently, as it is based on the logic of land-use management, GEOSIM is initially used to select and hierarchically arrange the localities according to their own dynamism and impact on neighbouring localities [22].

GEOSIM consists of 4 interdependent modules:

- *GEOSIM Spatial Analyst*: it is a module dedicated to the analysis of local dynamics within the studied area, prior to the socioeconomic optimisation phase of electrification solutions. Through the concepts of development poles and hinterlands (or attraction areas), GEOSIM Spatial Analyst identifies and analyses settlements with potential for social and economic development and that should be electrified first, to improve the impact of rural electrification. Additionally, GEOSIM Spatial Analyst enables to identify localities with low access to socioeconomic services and requiring specific attention.

- *GEOSIM Demand Analyst*: it aims at modelling and forecasting the demand for electricity of the studied area at the planning horizon. Input data can be collected through field visits or issued from extrapolations or hypothesis done by the user. By its "bottom-up" approach the model allows to predict reliable demand in the medium and long term in an accurate and realistic way. As outputs, GEOSIM Demand Analyst provides typical load duration curves and key figures related to the number of clients (MV/LV), the yearly consumption and the peak demand for each locality of the studied area.

- *GEOSIM Network Options*: the module finds the best decentralised options to supply electricity to previously identified development poles and their surrounding settlements, using one of the following methods:
    - MV grid extension
    - Diesel hybrid solutions (with PV or wind generation)
    - Solar powerplant (with or without energy storage)
    - Biomass powered mini-grids (gasification or cogeneration)
    - Mini-hydro powered mini-grids

- *GEOSIM Distributed Energy*: it is a module for pre-electrification programs. Access to basic energy services is a necessity for settlements located far from development poles and not electrified at the planning horizon. Based on GEOSIM Spatial Analyst and GEOSIM Network Options results, GEOSIM Distributed Energy identifies localities that should benefit from projects oriented towards access to basic energy services from alternatives sources

(photovoltaic systems, multifunctional platforms, etc...). These modules provide systems sizing and related required investments for targeted localities.

3.5 LAPER

The LAPER software is a tool for sustainable rural electrification of vast regions. It calculates the most economic masterplan for electrification of an area not yet supplied with electricity. It uses all available geographical data, creates villages-types in order to be able to use standardised electrical equipment for electrification, and compares the costs of all possible solutions of electrification (i.e. mini-grids, diesel gensets, solar panels, small hydro- or wind generators) [23].

LAPER, determines the villages which would economically benefit from being connected to the power grid and those for which a decentralised method of electrification is preferable. After this computation of the "target" solution, LAPER determines the master plan based on given annual budgets and various non-technical criteria (political, environmental...) which influence the order in which villages will be electrified. The software is based on GIS (ArcView from ESRI) [22].

The GIS provides the user with a direct access to the location of the villages and their respective geographical context (levelling of the ground, rivers, roads, etc…). The planner may then draw what he considers as the most suitable power grid to which the maximum number of villages can be connected. She may choose between several kinds of lines, since the main line is automatically set as a three phased one, and the secondary lines as single phased ones. The software then carries out an electric check of the designed grid.

The optimisation algorithm is rather simple. It consists in minimising the global cost of the whole area's electrification. This cost consists of the sum of investments and running expenses.

During the first step, it was possible for the planner to gather villages in separate groups (according to their characteristics). LAPER then splits these groups according to generation mode, to gather all villages that are supplied by the same one.

As far as the MV network is concerned, each village bears the entire expenses incurred for lines to which it is the only one connected. Expenses for shared portions of lines are broken down according to the consumption of each village. The first step consists of selecting the most economical dispersed mode of electrification for each village. Then, for every village which was initially connected to the network, the network is replaced by the selected dispersed mode (this is to be done whenever the selected dispersed mode is more economical than the network for a given village). The global cost born by every disconnected village will then dwindle, whereas the MV network expenses born by all villages that will remain connected will increase, since they may bear additional shared expenses formerly shared with the newly disconnected village. The global cost for all villages that are already supplied through dispersed generation remains, of course, unchanged. Hence, it should

compare, at each step (whenever a given village *i* is disconnected), two possible costs for the whole community [23].

3.6 COMPARISON OF THE TOOLS

In Table 9, the strengths and weaknesses of the different tools are presented:

Table 9. Strengths and weaknesses of the existing tools

TOOL	STRENGHTS	WEAKENESSES
REM	- GIS based - Complete analysis - Optimal network layout	- Limited variety of energy resources considered (PV and genset) - Under development - Needs RNM to run
RNM	- GIS based - Electrical balance	- Just designs the network of the grid - Based on a catalogue - Focus on urban areas
Network Planner	- GIS based - From community to national scale - Tested in Kenya, Senegal and Ghana - Written in Python	- Only top-down approach - Not basic input data (previous deep electrification analysis) - Seems abandoned - No population density accounting - Only grid extension
GEOSIM	- For decision-makers and planners - Time horizon considered - Connection to the grid or decentralised solutions - Social impact considered - Own load estimation module	- Commercial software
LAPER	- GIS based - Non-technical criteria considered - Valid for areas without electricity	- The planner draws the grid and the software only checks it

GISEle is a new tool underdevelopment by a research team in the *Politecnico di Milano* and it is the result of an accurate analysis and wants to cover the important gaps missing in these tools portfolio. With GISEle a new trend is pursued, bringing from one side analytical innovation, and from the other promoting the development of an integration platform enhancing synergies among tools and identifying the direction needed to solve the three components of the energy conundrum: energy access, energy security, and climate change.

3.7 INTRODUCTION TO GISELE

A research team in the *Politecnico di Milano* is currently working on the development of GISEle, a tool written in Python, that is a new procedure for the optimal design of the distribution grid topology in rural areas of developing countries.

Starting from GIS data analysis, the goal is to define the optimal techno-economic solution to bring the energy where it lacks. In addition to the technical configuration, exploiting the potentialities of GIS environment, GISEle is able to consider in its solution spatial characteristics as:

- the spatial distribution of consumers and generation plants.
- the detailed topology of the electric grid which would connect them together and, optionally, to the existing national grid.

Those aspects allow GISEle to supply accurate evaluations about the final costs of different possible solutions. The overall GISEle's structure is presented in the following figure.

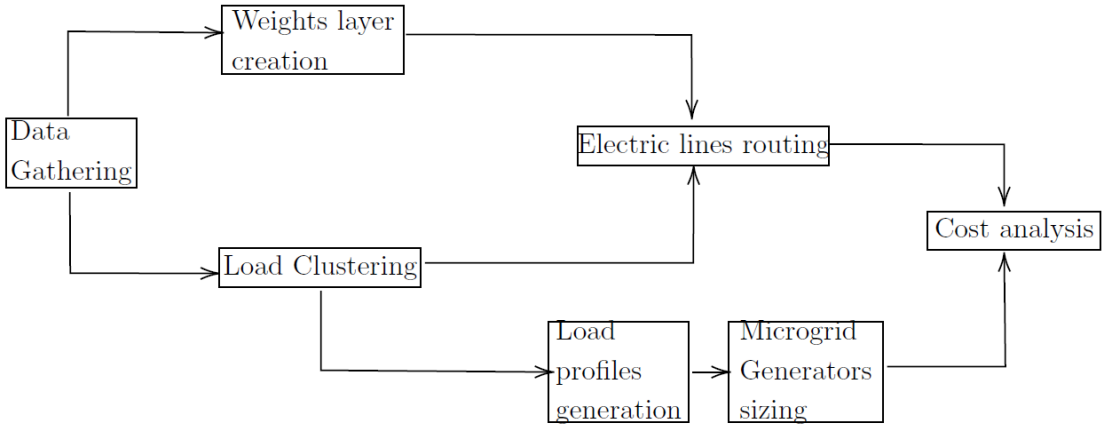

Figure 16. Description of the structure of GISEle [24]

Data gathering

The procedure starts gathering the data needed for the analysis: population density, terrain physical characteristics, natural resources availability, the network of roads, rivers' and lakes' position, are some of the geographical aspects that will be considered for the definition of the optimal line routing.

It is not always easy to find all the data needed, and for developing countries it is even more difficult, as sometimes digital datasets do not exist at all. All the datasets exploited by GISEle are presented in the following table:

Table 10. Geo-datasets exploited in GISEle [24]

DATASET	TYPE
Population distribution	Raster
Administrative boundaries	Vector Polygon
Existing grid network	Vector line
Planned grid network	Vector line
Electric substations	Vector Point
Roads network	Vector line
Water bodies (lakes)	Vector Polygon
Protected areas	Vector Polygon
GHI	Raster
Wind speed	Raster
Elevation map	Raster
Land cover	Raster
Hydrologic soil type	Raster
Monthly rainfall precipitation	Raster
Monthly mean Temperature	Raster
Monthly mean minimum Temperature	Raster
Monthly mean maximum Temperature	Raster

Some information, as the river network is not there. This is because the hydro-power potential is the most difficult to estimate. However, to assess the hydropower potential and create the layer, three steps are accomplished: determination of the water runoff, determination of how the surface water moves across the territory and estimation of the hydraulic head available in a reasonable neighbourhood of each river's points.

Data management

Once all the datasets are prepared, they are combined in QGIS providing the desired resolution and the EPSG code of the Projected Coordinate System to which project the datasets.

First, all the datasets are clipped on the region's administrative borders, then comes the nodes creation and the allocation of information as attributes to each cell, as it is the slope, elevation, land-cover, river-flow rate, road distance, water bodies and protected areas and substations.

The information gathered and embedded in each node constitutes a set of the spatial aspects and ground characteristics which impact on the costs of an electric line realization. This information is managed in order to assign a "penalty factor", an

index of the difficulty of building an electric line through that specific point. In GISEle, the contribution of the following attributes has been considered:

- Distance from road
- Land cover
- Slope
- River flow rate
- Water bodies
- Protected areas

In each node, these values are analysed and combined to return a penalty coefficient affecting the electric line which will run through it. The penalty coefficient is initially set to 1 and is augmented by the contributions of each single voices. The final value responds to the question of how many times more than the basic the electric line would cost.

$$Penalty\ Factor = 1 + \sum_{i}^{penalty\ aspects} penalty_i$$

Clustering

After collecting and processing the necessary data, the first step consists in clustering the population and hence the electric loads of the entire area. This process is done considering the population attribute of the previously created regular points grid.

The population clustering is a fundamental step of GISEle procedure to reduce the computational burden of the model. The routing procedure applied to the whole region of interest, indeed, would require a big amount of time because of the high number of points to be considered; furthermore it would create an electric line connecting each populated point, independently from its location, included the most isolated ones. Clustering solves these issues, by classifying those single isolated points as outliers, and grouping all the others in different clusters which will be considered as unique, separated energy communities.

As mentioned in Chapter 4, GISEle's clustering algorithm is a slightly modified version of DBSCAN, with a particular adaptation based on an important input data property: points do not represent a singular element like in general clustering problems but have a fundamental property which is the population. Therefore, the *minPts* input parameter changes its meaning from minimum number of points to be found in a neighbourhood to define it as core neighbourhood, o minimum number of people. Firstly, this allows the algorithm to create clusters having the exact communities' extension shape. Additionally, the ability of this algorithm of identifying scarcely populated areas allows to neglect vast zones with few people, prioritizing highly populated ones.

DBSCAN's input parameters, *eps* and *minPts*, need to be defined by the user. To facilitate this task, the model asks the user to define ranges for both, and displays

three tables showing the results of each couple of parameters' combinations in terms of:

- number of resulting clusters.
- % of clustered people over the target area total population.
- % of clustered area over the total target area.

In this way, the user can choose the preferred parameters' combination to be entered in GISEle, clusters are created, and finally the model asks the user to combine (if necessary) the output clusters, based on a graphic interface showing their distribution in the total space.

Electric grid routine

In order to design the electric grid topology which will connect all the consumers of a cluster, the allowed interconnections between the nodes, together with their intrinsic cost, have to be defined.

There are two methods to create the grid, which after are compared in order to choose the one of least cost. The first one, to reach the solution, adopts the Steiner-tree function, the MST approach, and the theory behind it. The other one combines the potentialities of MST and Dijkstra algorithms.

So, for the first method, a regular graph is built allowing each node to be connected only with its eight neighbouring nodes, considering the penalty factor as the mean of the penalty factors of the two terminals and the cost of a single graph's edge connecting two nodes i and j defined as:

$$C_{ij} = L_{ij} \cdot \frac{UC}{1000} \cdot \frac{p_i + p_j}{2}$$

Where:

- C_{ij} is the cost of the connection ij in US dollars.
- L_{ij} is the distance between two terminals i and j.
- UC is the unitary base cost of an electric MV line expressed in [$/m].
- p_i and p_j the penalty factors associated to the two terminals.

With the equation of C_{ij}, an edge cost matrix is created with the cost of the selected edges and the cost of the remaining connections are set to almost an infinite value. Now the graph is ready to be employed in the Minimum Path Analysis.

Considering a single cluster, every populated point within it (with a value of population greater than 0) becomes part of the set of terminal nodes of a Steiner Tree problem. Its solution will lead to the definition of the electric line topology with minimum cost, connecting all the populated point of the cluster in a single energy system. The code in Python will return the graph composed by only the edges which constitute the final electric grid able to connect all the cluster's populated point in a unique energy community.

The second method, developed by two students of the *Politecnico di Milano* [24], adopts a new approach for the electric routine combining the potentialities of MST and Dijkstra algorithms to produce an approximate solution to the Steiner Problem. After creating the MST and classifying the lines as short and long, it reduces the size of the problem by limiting the area under analysis. Once defined the borders of the graph, edges connecting neighbouring nodes are defined and weighted, to in the end launch Dijkstra path and obtain the shortest path connecting source and target.

In some individual cases, the algorithm could lead to the formation of internal cycle in the final grid. To face this problem, a Python routine able to detect cycle structure has been implemented. It identifies the imperfection and deletes the costliest connection which is part of the cycle.

Whether the final grid is calculated with the Steiner function or through the new, modified approach, the resulting graphs are analysed in the same way and the total length and cost of the grid are so calculated.

To finish this step, the model also evaluates if it is possible and worthy to connect the cluster to the closer HV/MV substation of the existing grid. The Dijkstra algorithm, adopted for the routing of long lines, is also useful to establish the path of the electric line which could connect the cluster's internal grid to the nearest HV/MV transmission substation. In other words, with a single analysis it is possible to identify both:

- The HV transmission line substation which is closest to the cluster in object.
- The cluster's point, connected to the internal grid, which is as close as possible to the defined substation.

Load curves estimation in GISEle

Being GISEle designed to be as much as possible autonomous in its exploratory purpose, it is evident how the definition of loads cannot be done before running the model: considering that it is impossible to know clusters' number, geographic positions and extents, in advance. Accordingly, loop inferences must be performed inside the model itself, in order to determine each cluster's load profile based on valuable proxies.

The energy needs assessment of the clusters is based on the definition of reference profiles: in the initialization phase, the user will be required to insert previously computed reference load profiles. This approach assumes that geographically close communities share socio-economic analogies in terms of basic energy needs, so in first approximation, their load curves will approximately have similar shapes.

GISEle takes the reference load profile as input, so the user is free to use the preferred approach in computing it. In addition to the profile, the user will have to specify a proxy, the most suitable being the number of households in the community to be used as reference. The reason behind this choice relies on assumption that, in

developing countries' rural areas, communities are basically household-centric, meaning that their core is made of rural households.

The load curves are going to be computed with the number of people per household and the number of appliances per category and functioning times.

Generation sizing

After having defined the cluster energy needs, the successive step is sizing an optimal power system capable of supplying the energy when needed. Being the final comparison of this proposed methodology between off and on-grid configuration, it is paramount to size an optimal system based on available resources. Solar, wind and hydro have been identified as the most cost-effective renewable sources for off-grid rural electrification.

As long as sizing is not the main goal of this tool and being a problem that has already been widely addressed in the literature, this part has been "outsourced". Between Homer Energy Pro and Calliope, two tools for generation sizing, Calliope has been chosen. The reason behind is Homer has a proprietary license, so the code would have to pause at every iteration to insert the information in the tool. However, Calliope has its code accessible and written in Python language, so it is possible to customize it in order to make it complementary to GISEle's purposes. The main purpose of Calliope is not sizing autonomous systems but optimizing medium and large-scale interconnected systems.

The result of the analysis will be a set of different possible energy solutions able to fulfil the energy demand of the cluster grid for the future 25 years. Each setup is presented with embedded information about:

- Number and characteristics of each item which will be part of the energy system.
- LCOE.
- Initial capital expenditure.
- Overview of operating costs over the entire operating horizon.
- Fraction of renewable energy penetration.
- Amount of pollutant emissions.
- Amount of fuel required.

Among other minor information detailing both technical and economical characteristics of the solution proposed.

The proposed configurations are usually sorted in an increasing LCOE order. The primary objective of this specific step in electrification strategy planning is stated by the available technologies, the energy resources and the amount of energy required along the year, to find the optimal combination of energy technologies able to fulfil the demand pursuing the minimization of the energy cost. In each cluster, GISEle selects three different setups which will be subjected to the final evaluation:

- Solution 1: the cheapest one.
- Solution 2: the one still exploiting fossil fuels, but with the highest fraction of renewable penetration.
- Solution 3: the cheapest solution providing 100% renewable energy.

Costs evaluation and decision making

The final output delivered by GISEle is a cluster specific comparison between the two main possible electrification strategies:

- Isolated Micro-Grid.
- Grid connected energy system.

Isolated micro-grids are not connected to the HV national grid, therefore they do not have to sustain the costs related to the siting of the power line linking the cluster's internal grid to the nearest HV substation but require a dedicated power supply system to feed the demand. The new final expression of the energy cost becomes:

$$LCOE_{micro-grid} = \frac{C_{grid}}{Total\ Energy} + LCOE_{gen}$$

- C_{grid}: Capital cost for cluster internal electric network.
- Total Energy: Forecast of the total amount of energy produced and sold by the energy system in 25 years of operation.
- $LCOE_{gen}$: Levelized Cost of Electricity returned by sizing process.

The alternative to the micro-grid structure consists in connecting the cluster network directly to the national HV transmission line. This solution gives access to electric energy at lower cost (since production by big power plants can benefit from economies of scale) but requires realizing the further electric line linking the cluster to the closest substation. The energy cost related to this option is therefore:

$$LCOE_{connected-grid} = \frac{(C_{grid} + C_{con})}{Total\ Energy} + COE_{NG}$$

- C_{con}: Capital cost of electric connection between cluster and HV line
- COE_{NG}: Cost of Energy provided by TSO

Finally, the two alternatives are compared, and the optimal strategy is defined.

Chapter 4

STATE OF THE ART: SITING SECONDARY SUBSTATIONS

As mentioned before, the objective of this work is to develop a procedure able to evaluate the optimal location of secondary substations. To do so, a brief research over the state of the art of siting secondary substations has been done to understand how the substations are placed in the literature.

A power distribution network consists of a number of substations connected to each other via feeders. Distribution planners must ensure that there is adequate substation capacity (transformer capacity) and feeder capacity (distribution capacity) to meet the load forecasts within the planning horizon [25].

The planning of power distribution system includes [25]:

- Optimal location of substations
- Optimal location of feeders
- Optimal individual feeders design
- Optimal allocation of load
- Optimal allocation of substation capacity
- Optimal mix of transformer by substation

The siting of secondary substation is not a simple process. According to [26], two of the most critical factors in the design of a substation are its location and siting. Failure to carefully consider these factors can result in excessive investment in the number of substations and associated transmission and distribution facilities.

The selection of the location of a substation must consider many factors. The substation site must be appropriate to contain the high-side equipment, the transformers and their support equipment, the low-side equipment and anything else required, in whatever configurations are decided upon. Substations are normally built above ground with the equipment exposed to elements.

When selecting a substation site, environmental considerations as weather, altitude, earthquakes, wildlife and livestock; and interfacing considerations should be taken into account. Also it should consider coordinating the location, design, and construction with other utilities operating in the area, as telecommunications, cable television, water and sewer, gas and radio and television stations [26]. Purchase and preparation of the actual site is a significant portion of substation cost. It also varies

greatly from site to site, making site cost a factor in siting and planning a substation [10]. Elements of site cost are:

- Land
- Civil/electrical/mechanical preparation
- Feeder getaway
- Public safety and esthetical site preparation
- Taxes and permits

Planning models to site secondary substations can be divided into two groups: planning under normal conditions or planning for emergency [25]. This section will explain in more detail the first group, presenting a short introduction of the second one.

4.1 PLANNING UNDER NORMAL CONDITIONS

Extensive research has been conducted in the field of optimal substation siting and sizing. In general, existing solutions can be divided into three major categories [27]:

i. Mathematical optimization.
ii. Heuristic and algorithm optimization.
iii. Intelligent optimization (AI/Expert-System approaches [25]).

Mathematical optimization

Mathematical optimization models usually have strict optimality but are very difficult to solve when the size and complexity of the system grows substantially.

The mathematical optimization problems are divided in single-period models and multi-period models. Single-period models are static models which assume that the load demand would not change during the horizon, divided in four subgroups: individual feeder models, system-feeder models, two-phase substation-then-feeder model, and substation-feeder models. This kind of problems deals with the design deciding on the length, conductor size, and gradation, and by addressing the economic trade-off between capital and operating costs. They use the transportation model framework, linear programming, 0-1 linear programming, and non-linear programming. Multi-period models formulate correlated time-dynamic decisions during the modelling [25].

Heuristic and algorithm optimization

Heuristic and algorithm optimization models simplify approaches to reduce the dynamic problem into a static one, thus allowing the problems to be solved more efficiently at the expense of getting an optimal solution [25]. Most of the heuristic search algorithms encounter with some convergence related problems especially in parameter sensitivity to a specific problem [28].

Comprehensive reviews of the trend in the field of distribution system planning are presented in the literature. Several methods have been used to solve the distribution planning problem [7]:

- Genetic algorithm [7]
- Imperialist Competitive Algorithm (ICA) [28]
- Supervised Learning Algorithms [25] [28]
- Particle Swarm Optimization [30]
- Weighted Voronoi Diagram [31]
- Graph Theory [30]
- Others: Ant Colony Optimization [6], Dynamic Programming [7], Evolution Strategies [7], Spatial Information-based Self-Adaptive Differential Evolution (ISADE) [6], Flower Pollination Algorithm [6], Multi-objective fuzzy models.

Intelligent optimization

In recent years, a growing number of intelligent optimization techniques have been applied to solve this problem. Simulated annealing algorithms have been applied to determine substations' location, capacity, and regional division with relatively good solution quality. However, these approaches can hardly handle large-scale problems with reasonable computation time due to parameter selection and the various annealing requirements [27].

AI/Expert Systems Approaches have been reported based on PROLOG, an artificial -intelligent programming language. These approaches look for load allocation in distribution substations, minimizing power loss and investment costs [25].

Some examples of the algorithms to site secondary substations are detailed in the following paragraphs.

Genetic Algorithm

In [7], the author developed a genetic algorithm considering a factor for annual growth rate of the demand over the analysis interval. The size, number, and placement of distribution transformers are optimally determined in order to improve system reliability and to minimize the losses under load growth.

The objective function of the problem includes the costs of the investment, maintenance, and the losses. The investment cost includes the capital cost of transformers installation and the capital cost of low voltage feeder construction. The maintenance cost includes the operation and maintenance cost associated to the distribution transformer and maintenance cost of low voltage feeders. The loss cost has two parts. One part is the energy loss cost which is proportional to the cost per kWh and the other part is the peak power cost which is proportional to the cost saving per kW reduction in the peak power. The interruptions cost is not usually considered in the optimization of the distribution substation placement. In this paper, the important of this reliability worth is investigated. The reliability cost is

determined using the outage cost of the distribution transformers. The outage cost is calculated based on customer damage function. The time horizon of this optimization problem is *y* years. This objective function should be minimized.

$$\min OF = \sum_{yr=1}^{N_y} \left(\frac{1+Infr}{1+Intr}\right)^{yr} \cdot \left(C_{Inv}(yr) + C_{OM}(yr) + C_L(yr) + \xi \cdot C_{Int}(yr)\right)$$

Where:

- C_{Inv}: The investment cost for year yr (\$)
- C_{OM}: The operational and maintenance cost for year yr (\$)
- C_L: The losses cost for year yr (\$)
- C_{Int}: The interuptions cost for year yr (\$)
- N_y: Number of years in the study timeframe (year)
- $Infr$: Inflation rate
- $Intr$: Interest rate
- ξ: The decision variable for considering interruptions cost.

The input data of the model are the low voltage of the lines, the position of load points and candidates' positions for the transformers, the fixed cost of the LV line construction, the loss factor, energy loss cost and peak power cost, ΔV_{max}, the overload time of the transformer, the study period, inflation and interest rate, the load growth rate, failure rate of pole mounted and pad mount transformers and the restoration time.

The constraints of the optimization problem are radial structure of the distribution network, transformer loading, and voltage drop limits. The load concentration points are dummy points; therefore, their path cannot be found through the streets. The distance is calculated using the summation of the vertical and horizontal distances. The load curve associated with the worst day of the year is selected to represent yearly load. It should be noted that the proposed model is in accordance with the transformer loading to indicate peak and off-peak loads. However, the losses are calculated based on the maximum active power losses multiply a suitable factor. Two approaches to load a transformer are considered: normal continuous loading and normal cyclic loading.

The results show that the considering reliability in the planning approach reduces the total cost of the distribution transformer placement. However, the associated investment cost increases while the reliability cost decreases.

Imperialist Competitive Algorithm

In [28], the author applies an Imperialist Competitive Algorithm (ICA) for the optimal expansion planning of large distribution network, solving the optimal sizing, siting and timing of medium voltage substation, using related fixed and variable costs subject to any operating and optimization constraints.

The algorithm aims to minimize capital investment and operating costs of expanded and new developed installation, considering electrical, geographical and other constraints in ODSP (Optimal Distribution Substation Placement). The presented model modifies the existing installations and finds the necessary new substations' type, size and location regarding the required future load growth.

The paper uses a new optimization algorithm and its main contribution is the implementation of multistage planning of distribution network under load growth.

Like other evolutionary algorithms, ICA starts with an initial population, which is called country and is divided into two types; colonies and imperialists, which together form empires. Every country could be defined as a vector with socio-political characteristics such as culture, language, and religion. In this work, ICA is applied to search for the optimal distribution substation placement problem.

The major and important information for the distribution substation placement is the value and geographical distribution of load density in the study region for horizon year.

From the multi-objective optimization formulation view of the problem, the optimal MV substation problem is solved such that the following constraints are satisfied:

- Maximum loading capacity of all substations.
- All loads or of the network should be supplied.
- The voltage drop should have acceptable limits.
- Total costs should be minimum.
- Geographical and asset constraints should be checked.

Optimal placement of substation is done first for mid-term forecasted load (5 year). After mid-term substation placement, the long-term case is run with previously optimal located substations which are found at mid-term study.

For each substation, an allowable and acceptable radius is defined. If the distance between load and transformer is greater than the radius defined, the load is not connected to the transformer.

The ICA is applied to minimizing the objective function:

$$\min CF = \sum_{n=1}^{N} (CCNS_n + CALS_n) + \lambda \sum_{n=1}^{N} \left[\sum_{m=1}^{M} IL_{nm} \right]$$

Where:

- $CCNS_n$: Cost of construction of new substation n in $ (set to zero for existing substations).
- $CALS_n$: Cost of resistive and core loss of substation n.
- λ: energy loss cost factor calculated according to electrical energy cost in planning year.
- IL_{nm}: loss index of low voltage feeder connecting substation n to load m.

The goal of the optimization in this stage is to find the best location, size and installation of candidate MV substations, subject to predefined criterion. In this stage of the study, each country is defined as a binary vector. The length of the vectors is equal to the number of candidates MV substations, which is determined by the system engineers in the feasible topological locations. The feasibility of the geographical constraints is checked by GIS system.

GIS-based and Semi-Supervised Learning Algorithm

A study conducted in China [27] presents a computationally efficient methodology to handle land cost in the substation siting problem. Assisted with GIS, the proposed approach is capable of modelling feasibilities of regions in the substation siting and sizing process and generates better solution as compared to existing approaches.

When geographic information is not considered, a substation can be placed/built anywhere within the planned area as long as the spot gives the lowest total cost. When geographic information is considered, as certain regions within the planned area are not available, the siting algorithm has to avoid these areas, as well as lakes, rivers, canyons, etc. When setting up the mathematical models, the geographic constraints can be modelled as regions with different land prices. To stay away from one particular region, a very high land price can be assigned to it. In addition, ordinary areas can also be distinguished by land prices.

The load clustering problem is essential to group the loads according to their characteristics (size, geographical location, etc.). The most popular clustering algorithm is the k-means algorithm, which belongs to unsupervised learning. As it is unsupervised, k-means is not reliable, and convergence starts to become an issue as the number of sample increases. In addition, k-means is very sensitive to initial seeding. To solve the convergence problem, the k-means++ and the bisecting k-means methods are tested in the paper. The k-means ++ is the one with best performance.

The k-means++ algorithm is an extension of the k-means algorithm. The basic idea of this algorithm is that by carefully selecting the initial seeds (centroids), convergence can be sped up and better results can be obtained as measured by the Sum of Square Errors (SSE), which is the sum of the squared difference between each sample and the corresponding cluster's centroid.

The bisecting k-means algorithm is a straightforward extension of the k-means approach. The basic idea of the bisecting k-means is to first split the entire system into two clusters using the k-means algorithm and then choose the worse cluster (in terms of the SSE) and split it into two new clusters.

In addition, to supervise the centroid selection process, the clustering algorithm needs to ensure that the total capacity of a single load group cannot exceed capacity of the corresponding substation. Therefore, the load clustering approach needs to consider the total capacity of each substation. As the k-means++ clustering algorithm itself does not consider the capacity at each substation, the k-means++

algorithm is improved to incorporate this factor, c_{max}. The algorithm calculates the total load and if it is lower than c_{max} ends, and if it exceeds, it remove the loads from that group one by one starting from the farthest load, until the sum of the remaining load falls below the capacity of the substation.

The proposed clustering algorithm does not require uniform load distribution. When the load distribution is extremely uneven, the proposed load clustering algorithm can determine the substation capacity required for each load group according to its density without the need to specify the capacity of each substation in advance. This is also an advantage of the proposed algorithm as compared to existing approaches.

With GIS and land price information included, the proposed approach significantly reduces the investment cost.

PSO algorithm with MST for feeder routine

In [30], an algorithm to find the optimum distribution substation placement and sizing by utilizing the PSO (Particle Swarm Optimization) algorithm and optimum feeder routing using modified MST is proposed.

In the swarm intelligent algorithm, the movement towards the optimal position is obtained from the best information of each particle which is included in the initial population (Best Personal Position) and the optimal position that is found by the neighbour's positions (Best Global Position).

Based on graph theory, MST is the graph with the minimum weight on branches. This paper uses the Prim's algorithm to solve the optimum feeder routing on LV and MV networks.

The power flow calculation is required in planning of distribution network to evaluate the network. The Open Distribution System Simulator (OpenDSS) is an open source comprehensive electrical system simulation tool for electric utility distribution systems, developed by the Electric Power Research Institute.

In this paper, OpenDSS engine is utilized as a power flow solution in order to find the power system parameters such voltage profile, power factor, real and reactive power flowing in each line, power losses and etc. which can be used in optimum substation placement and sizing problem in distribution networks.

This paper tried to look for the best location of the substations with the objective of reducing the distribution network losses. Therefore, the minimum number of distribution substations are estimated based on the maximum branching rate and number of consumer (load centres) for each branch.

The Particle Swarm Optimization (PSO) method is performed to optimize the best possible placement of distribution substations. Then, the Prim's algorithm is implemented to find the optimum feeder routing of LV and MV networks.

The constraints that must be satisfied to find the optimal substation allocation and feeder routing, are:

- Supplying all the consumers of the network.
- Acceptable voltage drops at the receiving bus.
- Maximum load capacity of all substation.
- Cost minimization of new substation construction.

The main objective function that needs to be minimized is:

$$\min Z = CL + CC + PF$$

Where Z is the total cost function, CL the total losses cost, CC the annualized capital cost and PF the penalty factor which is calculated by the optimization constraints.

The first constraint of distribution network planning is acceptable voltage drop at receiving bus (V_i), and the second constraint of distribution network planning is the longest distance of each consumer from the distribution substation which is introduced by the substation radius based on a standard.

K-means and Dijkstra's algorithm

In [29], the secondary circuit of a rural distribution network is planned in two stages. Firstly, the optimal allocation and sizing of the step-down transformer is performed using the k-means algorithm and validated using Davies-Bouldin index. Secondly, once the transformer is placed, the secondary circuit path is found based on the Dijkstra's algorithm to find the shortest path.

The k-means algorithm can be easily implemented to find the optimal location of a transformer if the locations of the consumers is known.

Clustering is performed for different number of centroids and DBI evaluates the distances in order to analyse the best option, minimizing the distances into the same cluster, but larger distances between clusters. When the index is smaller the cluster is better.

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max\left\{\frac{\sigma_i + \sigma_j}{D_{i,j}}\right\} K = 2, \dots N-1$$

Where $D_{i,j}$ is the distance between centroids c_i and c_j, K is the number of clusters, and σ_k is the average distance of all the elements of the clusters x to the centroid c_x.

Once the number of costumers is allocated to each of the transformers, the number and the path of secondary circuits is found using Dijkstra's algorithm. This algorithm is based on the shortest path problem minimizing distances. The graph obtained starts with all the path from one vertex to all other points, in this case finds the shortest path between the transformers and their respective consumers, representing the electrical network.

Weighted Voronoi Diagram and Transportation Model

In [31], an improved method based on the weighted Voronoi diagram (WVD) and transportation model for substation planning is proposed, which can optimize the location, capacity, and power supply range for each substation with the minimum investment which contains the cost of the lines, substations, and annual operation expense.

The initial weight of the WVD is the relation between the power supply radius of each substation – according to the load of the substation and the average load density of the planning area - and the average power supply radius of each substation - according to its rated capacity and average load density of the planning area.

The weighted Voronoi diagram, whose weights can be adaptively adjusted with two experimental parameters, can calculate the location and the capacity for each substation with good performance of global convergence and better convergence speed. Transportation model can simulate the best correspondence relationship between the loads and substations.

When the load distribution of the target year is known, the number, capacity, location, and power supply range of the construction substations are determined to minimize the cost of substation, network investment, and annual operating costs under the constraint of meeting the load carried by the substations.

The main problems and solutions are as follows:

- The convergence, rationality, and running time of the algorithm were challenged by the constant expansion of the grid. The solution is self-adjustment of the weights and capacity optimization.
- Geographical factors, like rivers, lakes, and mountains, had a great impact on substation locating and sizing. The areas that could not build substation were marked in advance, and the location of substations avoided these areas automatically by the program.
- The power supply range of each substation is usually conformed after substation locating by the WVD, but facts have proved that it can hardly meet the load rate demand in a finite number of iterations; that is, the load rates of some substations are always bigger than their maximum. Transportation model has been used to calculate the power supply range of each substation after substation siting by the WVD.

The weighted Voronoi diagram reflects the effect on power supply range of an uneven load and the different nominal capacity and load rate of each substation.

4.2 PLANNING FOR EMERGENCY

Although not a common phenomenon, transformers and feeders do fail, and the cost of power outage can be very significant. The emergency models can be divided in two groups: contingency planning where the environment is generally at the substation

level, and problems in load restoration and load balance both of which are at the feeder level [25].

The contingency models are divided in five subgroups:

- Single contingency capacity: this policy stipulates that at any given time the substation capacity of a service area should be able to handle the load even when one transformer in the area fails.
- Load reallocation: re-allocation of unsatisfied load under the single-contingency environment.
- Multi-period feeder expansion: construction of feeders to facilitate further load allocation.
- Multi-period transformer allocation: transformer procurement when the addition of feeders does not resolve the demand shortfall.
- Systematic planning scheme: an integrated planning with the four previous models.

4.3 COMPARISON OF THE METHODS

In this study, the objective is to design a procedure able to evaluate the optimal location of secondary substations from a topological perspective considering the population distribution given by georeferenced data.

On one hand, the mathematical methods and artificial intelligence methods have been discarded as the size of the managed data is big and this would lead to complex systems and high computational times.

On the other hand, to evaluate the heuristic and algorithm methods, a comparison among the options presented in the literature has been done.

Table 11. Comparison of methods to site secondary substations

Algorithm	Cost Function	Geographic Information	Input data
Genetic Algorithm	✓	×	LV lines, candidate positions and fixed costs
Imperialist Competitive Algorithm	✓	✓	Number of substations, costs and losses
K-means + GIS	×	✓	Number of substations, location of consumers, power constraint
PSO + MST	✓	✓	Maximum branching rate and number of consumers for each branch
K-means + Dijkstra's algorithm	×	✓	Number of substations, location of consumers
WVD + Transportation Model	✓	✓	Capacity of the substations, location of the loads

The heuristic algorithms that aim at minimizing a cost function have been discarded as this is not an economic study. However, as the location of consumers is known and the load can be estimated in the data managed in this problem, the k-means and graph partitioning algorithms, as MST and Dijkstra's algorithm, seem to fit with the data.

Chapter 5

ALGORITHMS AND OPTIMIZATION METHODS

One of the final objectives of this thesis work is to help in the rural electrification planning in order to bring electricity to the highest number of people with the least cost possible bearing the energy mix in mind. The project focuses on the development of a procedure to site secondary substations.

The approach to site the substations is from a topological point of view, not an economic assessment. The idea is to place the substations where the people are. To do so, it is required to detect these populated areas by clustering the population. Hence, the election of the clustering algorithm has an important role, as it will determine which points are going to be considered in the electrification process, which ones are going to be connected to the grid and how.

In the literature, many tools for cluster analysis can be found, although each clustering algorithm has its own strengths and weaknesses due to the complexity of information. But what is cluster analysis? Cluster analysis is the task of grouping a set of objects in such a way that objects in the same group are more similar, in some sense, to each other than to those in other groups.

In this chapter, some clustering algorithms are presented, and the ones chosen to cluster densely populated areas.

5.1 CLASSIFICATION OF CLUSTERING ALGORITHMS

According to [32], the clustering algorithms can be studied from two perspectives: traditional and modern ones. The traditional algorithms are the first developed more established; while the modern algorithms are the ones still developing due the progress in science and technology and are more suitable to address specific problems.

On one hand, the traditional clustering algorithms can be divided into 9 categories [32]. These categories are clustering algorithms based on:

- Partition
- Hierarchy
- Fuzzy theory
- Distribution
- Density

- Graph theory
- Grid
- Fractal theory
- Model

Clustering Algorithms based on Partition

Given a database of n objects, clustering algorithms based on partition construct k partitions of the data. Each object must belong to only one group, and each group will have at least one object [33]. The basic idea of this kind of clustering algorithms is to regard the centre of data points as the centre of the corresponding cluster. For example, in k-means a cluster is represented by its centroid, which is the mean of the points of the cluster [33].

The search of the centre is an iterative process that attempts to improve the partitioning by moving objects from one group to another. The strategy is to optimize an objective function, for example, minimize the total square-error [34].

The data size of this type of algorithm is small to medium, with spherical data shape and the clustering based on distance [34].

The typical clustering algorithms based on partition include k-means, k-medoids, PAM, CLARA and CLARANS.

The advantages of partition algorithms are the relatively low time complexity and the high computing efficiency in general [32].

The disadvantages are that they are not suitable for non-convex data, they are relatively sensitive to the outliers, easily drawn into local optimal, the number of clusters need to be pre-set, and the clustering result is sensitive to the number of clusters [32].

Clustering Algorithms based on Hierarchy

Hierarchical clustering method creates a hierarchical decomposition of the database D and yields a dendrogram which represent the hierarchical relationship among the data to be clustered. These algorithms are divided into two categories: agglomerative (bottom-up approach), which start considering each point of data as an individual cluster and then merge the points by proximity until there is only one cluster left, called 'nested set of partitions'; and divisive (top-down approach), the reverse process, which starts with only one cluster and divide it until each point is a cluster itself [33].

Typical algorithms based on hierarchy are BIRCH, CURE, ROCK or Chameleon.

The advantages of these algorithms are their suitability for data sets with arbitrary shape and attribute of arbitrary type, the easy detection of hierarchical relationships among clusters, and the relatively high scalability in general [32].

The disadvantages comprehend the relatively high time complexity in general, which leads to a not proper scale, and the need for pre-setting of a termination condition [32].

Clustering Algorithms based on Distribution

The basic idea is to cluster objects according to the similarity between the probability distribution of the data [35]. Objects belong to different clusters if there exist several distributions in the original data. The distribution difference cannot be captured by geometric distances, so the difference can be captured by Kullback-Leibler divergence [35].

For example, a cluster based on the distribution of the nearest neighbour distance is set as follow [36].

Def. Let *DB* be a set of points. A *cluster C* is a non-empty subset of *DB* with the following properties:

i. *NNDistSet(C)*, the nearest neighbour distance set of a set of points *C*, has the expected distribution with a required confidence-level.
ii. *C* is *maximal*
iii. *C* is *connected*

The typical algorithms are DBCLASD and GMM.

The advantages are the relatively high scalability by changing the distribution, the support by the well-developed statistical science and the absence of input parameters [32].

The disadvantages are that the probability distribution does not represent with the same accuracy all the data and the relatively high time complexity [32].

Clustering Algorithms based on Density

The basic idea of density-based clustering methods is that the data which is in the region with high density of the data space is considered to belong to the same cluster [32].

The typical ones include DBSCAN, DENCLUE, OPTICS, HDBSCAN and Mean-shift.

The advantages are the high efficiency in clustering, its suitability for data with arbitrary shape and the ability to handle noise [34].

The disadvantages are the low-quality resulting clustering when the density of data space is not even, the need of a memory with big size when the data volume is big, and the need of input parameters.

Clustering Algorithms based on Graph Theory

The graph-based clustering is a cluster analysis based on graph theory. The data is used to form a similarity graph in which the elements correspond to the vertices or nodes of the graph, and the edges of the graph connect the elements with similar values above some threshold [37].

This type of methods have two steps: firstly, the construction of a neighbourhood graph using the similarities between all the points, and secondly, a graph clustering algorithm is applied to this graph, where the partitioning occurs and some edges are cut in order to obtain the final clusters [38].

The final clusters are not always the same. The results of graph-based clustering algorithms are affected by the type of graph and the connectivity parameters that are chosen for the construction of it [38].

Typical algorithms of this kind of clustering are CLICK, LUKES and MST-based clustering.

The advantages are the high efficiency in clustering, the high accuracy in the clustering result, and good results even in the presence of high noise levels [32].

The main disadvantage is that the time complexity increases dramatically with the increasing of graph complexity [32].

Clustering Algorithms based on Grid

The basic idea of this kind of clustering algorithms is to divide the data space into a finite number of cells, constructing a grid structure. Then, the density of each cell is calculated, and cells are classified according to their densities to find the centre of the clusters [33].

The typical algorithms of this kind of clustering are STING and CLIQUE.

The advantages are the low time complexity due to the fast processing time, the high scalability and suitability for parallel processing and increment updating [32].

The disadvantages are the sensitivity of the result clustering to the granularity (the mesh size), the high calculation efficiency at the cost of reducing the quality of clusters and reducing the clustering accuracy [32].

Clustering Algorithms based on Fractal Theory

Fractal stands for the geometry that can be divided into several parts which share some common characters with the whole [32].

A fractal has no characteristic scale and cannot be described with traditional measures such as length, area, volume, and density. The basic parameter used for fractal description is fractal dimension. Fractal dimension can be defined on the base of entropy and correlation function [39].

Scale-invariant systems are usually characterized by non-integer dimensions. The dimension tells how some property of an object or space changes as it is viewed in increased detail [40].

The typical algorithm of this kind of clustering is FC of which the core idea is that the change of any inner data of a cluster does not have any influence on the intrinsic quality of the fractal dimension [32].

The advantages are the high efficiency in clustering, high scalability, deals effectively with outliers and suitability for data with arbitrary shape and high dimension [32].

The disadvantages are the hypothesis on the data are not completely correct and the clustering result is sensitive to the parameters [32].

Clustering Algorithms based on Model

The model-based clustering algorithm is based on hypothesizing a model for every cluster to find best fit of the data according to the mathematical model [33].

There are mainly two kinds of model-based clustering algorithms, one based on statistical learning method and the other based on neural network learning method [32].

The typical algorithms, based on statistical learning method, are COBWEB and GMM. The typical algorithms, based on neural network learning method, are SOM and ART.

One of the main advantages is that the models are diverse and well-developed providing means to describe data adequately and each model has its own special characters that may bring about some significant advantages in some specific areas [32].

The disadvantages are the relatively high time complexity in general, the hypothesis of the models over the data are not completely correct, and the clustering result is sensitive to the parameters of selected models [32].

On the other hand, the modern clustering algorithms can be divided into 10 categories [32].These categories are clustering algorithms:

- Based on Kernel
- Based on Ensemble
- Based on Swarm intelligence
- Based on Quantum theory
- Based on Spectral Graph Theory
- Based on Affinity Propagation
- Based on Density and Distance
- For Spatial Data

- For Data Stream
- For Large-Scale Data

As they are very specific, not all of them are interesting to cluster populated areas. Only the ones based on ensemble, swarm intelligence, density and distance, spatial data and large-scale data are detailed in the following paragraphs.

Clustering Algorithms based on Ensemble

A clustering ensemble aims to combine multiple clustering models to produce a better result than that of the individual clustering algorithms in terms of consistency and quality [41].

The main idea is to cluster the data with different clustering algorithms, generate a set of initial results and in the end integrate all the results to obtain the final clustering result. The initial clustering results are integrated by means of the consensus function. These consensus functions can be divided in nine categories. Some, for example, are based on co-association matrix, graph partition, hybrid model or fuzzy theory.

The typical algorithms of this type of clustering are CSPA, HGPA, MCLA, VM, HCE, LAC, WPCK, sCSPA, sMCLA and sHBGPA.

The advantages of this method are the robustness, scalability, and its ability to be parallel and take advantage of the strengths of the involved algorithms [32].

The disadvantages are the inadequate understanding about the difference among the initial clustering results and the existing deficiencies of the design of the consensus function [32].

Clustering Algorithms based on Swarm Intelligence

Swarm Intelligence is defined as any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of the social insect colonies and other animal societies [42].

The basic idea of this type of clustering method is to simulate the changing process of the biological population [32].

The typical algorithms of this type are ACO_based(LF), PSO_based, SFLA_based and ABC_based.

The advantages of the algorithm are the ease of overcoming a local optimal and get the global optimal and the ease to understand the algorithm [32].

The disadvantages are the low scalability, low operating efficiency and the not suitability for high dimensional or large-scale data [32].

Clustering Algorithms based on Density and Distance

In [43], the authors propose an approach based on the idea that cluster centres are characterized by a higher density than their neighbours and by a relatively large distance from points with higher densities. This idea forms the basis of a clustering procedure in which the number of clusters arises intuitively, outliers are automatically spotted and excluded from the analysis, and clusters are recognized regardless of their shape and of the dimensionality of the space in which they are embedded [43].

The idea is to figure out, based on distance function, the local density of each data point and the shortest distance among each data point and other data points with higher local density in order to construct the decision graph first, select the cluster centres based on the decision graph then, and put the remaining data points into the nearest cluster with higher local density at last [32].

The advantages are the simplicity and clearness of the idea, the suitability for data sets with arbitrary shape and the insensitivity to the outliers [32].

The disadvantages are the relatively high time complexity, the relatively strong subjectivity for the selection of the cluster centre based on the decision graph and the sensitivity of the clustering result to the parameters involved in DD algorithm [32].

Clustering Algorithms for Spatial Data

Spatial data, also known as geospatial data, is a term to describe any data related to or containing information about a physical object that can be represented by numerical values in a geographic coordinate system [44].

In this case, spatial data refers to the data with the two dimensions, time and space, at the same time, sharing the characteristics of large in scale, high in speed and complex in information [32].

Typical algorithms are DBSCAN, STING, Wavecluster, CLARANS.

Clustering Algorithms for Large-Scale Data

Large scale data analysis is the process of applying data analysis techniques to a large amount of data, typically in big data repositories. It uses specialized algorithms, systems and processes to review, analyse and present information in a form that is more meaningful for organizations or end users [45].

Big data shares the characteristics of large in volume, rich in variety, high in velocity and doubt in veracity, while large scale data, although, it is not a standard term but can be associated with data that grows to a huge size over time and is held by conventional data warehousing solutions. The main basic ideas of clustering for big data can be summarized in the following 4 categories: sample clustering, data merged clustering, dimension-reducing clustering and parallel clustering.

Typical algorithms are K-means, BIRCH, CLARA, CURE, DBSCAN, DENCLUE, Wavecluster, FC.

Table 12 presents a resume of the principal characteristics of the clustering algorithms presented in [32]:

Table 12. Characteristics of the clustering algorithms [32]

Category based on	Clustering algorithm	Time complexity	Shape of cluster	For Large Scale Data	For high dimensional data	Sensitivity to input data	Sensitivity to noise/ outliers
Partition	K-means	Low	Convex	Yes	No	High	High
	K-medoids	High	Convex	No	No	Moderate	Little
	PAM	High	Convex	No	No	Moderate	Little
	CLARA	Middle	Convex	Yes	No	Moderate	Little
	CLARANS	High	Convex	Yes	No	High	Little
	AP	*	*	*	*	*	*
Hierarchy	BIRCH	Low	Convex	Yes	No	Moderate	Little
Hierarchy	CURE	Low	Arbitrary	Yes	Yes	Moderate	Little
	ROCK	High	Arbitrary	No	Yes	Moderate	Little
	Chamaleon	High	Arbitrary	No	No	Moderate	Little
Fuzzy Theory	FCM	Low	Convex	No	No	Moderate	High
	FCS	High	Arbitrary	No	No	Moderate	High
	MM	Middle	Arbitrary	No	No	Moderate	Little
Distribution	DBCLASD	Middle	Arbitrary	Yes	Yes	Little	Little
	GMM	High	Arbitrary	No	No	High	Little
Density	DBSCAN	Middle	Arbitrary	Yes	No	Moderate	Little
	OPTICS	Middle	Arbitrary	Yes	No	Little	Little
	Mean-Shift	High	Arbitrary	No	No	Little	Little
	DENCLUE	*	*	*	*	*	*
Graph Theory	CLICK	Low	Arbitrary	Yes	No	High	High
	MST	Middle	Arbitrary	Yes	No	High	High
	SM	*	*	*	*	*	*
	NJW	*	*	*	*	*	*
Grid	STING	Low	Arbitrary	Yes	Yes	Little	Little
	CLIQUE	Low	Convex	No	Yes	Little	Moderate
	Wavecluster	*	*	*	*	*	*
Fractal Theory	FC	Low	Arbitrary	Yes	Yes	High	Little
Model	COBWEB	Low	Arbitrary	Yes	No	Little	Moderate
	GMM	*	*	*	*	*	*
	SOM	High	Arbitrary	No	Yes	Little	Little
	ART	Middle	Arbitrary	Yes	No	High	High
Kernel	Kernel K-means	High	Arbitrary	No	No	Moderate	Little
	Kernel SOM	High	Arbitrary	No	No	Little	Little
	Kernel FCM	High	Arbitrary	No	No	Moderate	Little
	SVC	High	Arbitrary	No	No	Little	Little
	MMC	High	Arbitrary	No	No	Little	Little
	MKC	High	Arbitrary	No	No	Little	Little
Ensemble	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Swarm intelligence	ACO_based	High	Arbitrary	No	No	Little	High
	PSO_based	High	Arbitrary	No	No	Moderate	Moderate
	SFLA_based	High	Arbitrary	No	No	Moderate	Moderate
	ABC_based	High	Arbitrary	No	No	Moderate	Moderate
Quantum Theory	QC	High	Convex	No	No	Little	Little
	DQC	Middle	Convex	No	No	Little	Little
Spectral clustering	SM	High	Arbitrary	No	Yes	Little	Little
	NJW	High	Arbitrary	No	Yes	Little	Little
Affinity Propagation	AP	High	Convex	No	No	Moderate	Little
Density and Distance	DD	High	Arbitrary	No	No	Little	Little
Spatial Data	DBSCAN	*	*	*	*	*	*
	STING	*	*	*	*	*	*
	Wavecluster	Low	Arbitrary	Yes	No	Little	
	CLARANS	*	*	*	*	*	*
Data Stream	STREAM	Low	Convex	Yes	No	High	High
	CluStream	Low	Convex	Yes	No	High	High
	HPStream		Convex	Yes	Yes	High	High
Data Stream	DenStream		Arbitrary	Yes	No	High	Little
	K-means	*	*	*	*	*	*
Large-scale data	BIRCH	*	*	*	*	*	*
	CLARA	*	*	*	*	*	*
	CURE	*	*	*	*	*	*
	DBSCAN	*	*	*	*	*	*
Large-scale data	DENCLUE	Middle	Arbitrary	Yes	Yes	Moderate	Little
	Wavecluster	*	*	*	*	*	*
	FC	*	*	*	*	*	*

5.2 CLUSTERING OF DENSELY POPULATED AREAS

The area under study in rural electrification planning is a large spatial database. So, the clustering method chosen should meet three requirements [46]:

i. Minimal number of input parameters: it is difficult to identify initial parameters as number of clusters, shape and density in advance.
ii. Discovery of cluster with arbitrary shape.
iii. Good efficiency for large databases (more than just a few thousand objects).

The basis for constructing clustering algorithms are the concept of distance and similarity [32]. For quantitative data features, distance is preferred and for qualitative data similarity is preferred. In this case the data is population, so clustering algorithms based on distance are preferred.

The portfolio of algorithms is very extensive. However, in order to choose which clustering algorithm is the best one to cluster densely populated areas, six different characteristics are considered:

- **Time complexity:** If the runtime of the code is high or low.
- **Shape of the cluster:** If the shape needs to be convex or could be arbitrary. A convex polygon is a simple polygon (not self-intersecting) in which no line segment between two points on the boundary ever goes outside the polygon.
- **Appropriate for large scale data:** If the algorithm could deal with a big amount of data or not.
- **Appropriate for high dimensional data:** In physics and mathematics, the dimension of a mathematical space is informally defined as the minimum number of coordinates needed to specify any point within it.
  Clustering high-dimensional data is the cluster analysis of data with anywhere from a few dozen to many thousands of dimensions. Such high-dimensional spaces of data are often encountered in areas such as medicine, where DNA microarray technology can produce many measurements at once, and the clustering of text documents, where, if a word-frequency vector is used, the number of dimensions equals the size of the vocabulary.
- **Sensitivity to input data:** If there is need of choosing input data before the algorithm runs and if these inputs influence or not in the solution. Usually, the value of the input data is not very precise, but approximations or hypothesis have to be made, so low sensitivity to the input data is preferable.
- **Sensitivity to noise/outliers:** The data is not always very aggregate. Sometimes outliers or points far away of the concentration of data exist. Algorithms that could recognise these points and not consider them in the clustering process are necessary, as outliers could change the value and shapes of clusters and in the end have a not very representative division of the data.

In the problem addressed, the data are points (pixels) with geographic coordinates, with a population density associated.

According to the definition of high-dimensional data given above, it can be concluded that the data is not a high-dimensional data. So, this feature will not be necessary when choosing the algorithm. The population can be considered as one dimension of the data and this dimension can be chosen to cluster the points.

It can be said that the population data is spatial data because the data of the density of population in each point has geographic coordinates. Also, the data can be considered as large-scale data, as the area under study is not small and the number of points is considerable.

As mentioned before, only areas with a minimum population density are considered as they must be worthy to be electrified, economically speaking. As the area must have a minimum density of the population there are some points that are not going to be consider. Because of this, the algorithm needs to deal with noise or outliers and not include all the point in the clusters.

Moreover, these areas do not have a convex shape necessarily, so an algorithm which finds arbitrary shapes should be chosen.

Considering these six characteristics, the filters to consider the algorithms are:

- Low or middle time complexity.
- Only the ones with arbitrary shape of the cluster.
- Large scale data.
- Not filtering the high-dimensional data.
- Little or moderate sensitivity to input data.
- With little or moderate sensitivity to noise/outliers.

Applying these filters, the considered algorithms to cluster populated areas are presented in Table 13 and explained in detail later.

In addition to these ones, k-means and LUKES are included in the study. Although k-means is sensitive to input data and returns convex shapes, it is widely used and easy to implement with interesting results. For its part, LUKES is based on graph theory and its structure and implementation matches with the population data, although it has high time complexity.

Table 13. Filtered algorithms for clustering densely populated area

Category based on	Clustering algorithm	Time complexity	Shape of cluster	For Large Scale Data	For high dimensional data	Sensitivity to input data	Sensitivity to noise/ outliers
Partition	K-means	Low	Convex	Yes	No	High	High
Hierarchy	CURE	Low	Arbitrary	Yes	Yes	Moderate	Little
Distribution	DBCLASD	Middle	Arbitrary	Yes	Yes	Little	Little
Density	DBSCAN	Middle	Arbitrary	Yes	No	Moderate	Little
	OPTICS	Middle	Arbitrary	Yes	No	Little	Little
Graph Theory	LUKES	High	Arbitrary	Yes	No	High	High
Grid	STING	Low	Arbitrary	Yes	Yes	Little	Little
Model	COBWEB	Low	Arbitrary	Yes	No	Little	Moderate
Large-scale data	DENCLUE	Middle	Arbitrary	Yes	Yes	Moderate	Little

K-MEANS

K-means clustering is a method that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

Given a set of observations (x_1, x_2, ..., x_n), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($\leq n$) sets $S = \{S_1, S_2, ..., S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg min_S \sum_{i=1}^{k} \sum_{x \epsilon S_i} \|x - \mu_i\|^2 = \arg min_S \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Where μ_i is the mean of points in S_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\arg min_S \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{x,y \, \epsilon S_i} \|x - y\|^2$$

The equivalence can be deduced from identity

$$\sum_{x \epsilon S_i} \|x - \mu_i\|^2 = \sum_{x \neq y \, \epsilon S_i} (x - \mu_i)(\mu_i - y)$$

Because the total variance is constant, this is equivalent to maximizing the sum of squared deviations between points in different clusters (between-cluster sum of squares, BCSS), which follows from the law of total variance.

There are three steps in the algorithm. It starts with the initialisation and then it proceeds by alternating between the assignment and update steps:

- *Initialization*: usually generate at random, giving an initial set of *k* means $m_1^{(1)}$, …, $m_k^{(1)}$. However, initialization methods, as the Forgy or Random Partition methods exist.
- *Assignment step*: Assign each observation to the cluster whose mean has the least squared Euclidean distance; this is intuitively the "nearest" mean. Mathematically, this means partitioning the observations according to the Voronoi diagram generated by the means:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\}$$

  where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

- *Update step*: Calculate the new means (centroids) of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

In essence, the algorithm starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the cluster centres until a convergence criterion is met. The method is relatively scalable and efficient in processing large data sets and its time is linear O(n), so it is relatively small.

Nevertheless, k-means has a couple of disadvantages:

- The number of clusters have to be selected a priori. This is not always trivial and ideally for a clustering algorithm that wants to figure that out for you.
- K-means also starts with a random choice of cluster centres and therefore it may yield different clustering results on different runs of the algorithm. Thus, the results may not be repeatable and lack consistency.
- The method often terminates at a local optimum, not a global one.
- This algorithm is not suitable for discovering clusters with non-convex shapes or clusters of very different size.

Pseudocode

```
Input: Number of clusters
>> Initialization
>>Assignment step
>>Update step
```

CURE

CURE (Clustering Using Representatives) is a bottom-up hierarchical clustering algorithm, but instead of using a centroid-based approach or an all-points approach

it employs a method that is based on choosing a well-formed group of points to identify the distance between clusters.

In fact, CURE begins by choosing a constant number, *c*, of well scattered points from a cluster. These points are used to identify the shape and size of the cluster. The next step of the algorithm shrinks the selected points toward the centroid of the cluster using some predetermined fraction α. Varying the fraction α between 0 and 1 can help CURE to identify different types of clusters. Using the shrunken position of these points to identify the cluster, the algorithm then finds the clusters with the closest pairs of identifying points. These are the clusters that are chosen to be merged as part of the hierarchical algorithm. Merging continues until the desired number of clusters, k, an input parameter, remain [47].

CURE is less sensitive to outliers since shrinking the scattered points toward the mean reduces the adverse effects due to outliers, since outliers are typically further away from the mean and are thus shifted a larger distance due to the shrinking [48].

The runtime is $O(n^2 \log n)$, making it rather expensive, and space complexity is $O(n)$.

Because of the high runtime complexity, the algorithm cannot be directly applied to large databases. So, there are three steps to deal with this problem: doing a random sampling, a partitioning and labelling data on disk.

Pseudocode

Input: A set of points S
Output: k clusters

>> For every cluster *u* (each input point), in *u.mean* and *u.rep* store the mean of the points in the cluster and a set of *c* representative points of the cluster (initially *c* = 1 since each cluster has one data point). Also *u.closest* stores the cluster closest to *u*.

>> All the input points are inserted into a k-d tree *T*

>> Treat each input point as separate cluster, compute *u.closest* for each *u* and then insert each cluster into the heap *Q* (clusters are arranged in increasing order of distances between *u* and *u.closest*).

>> While size(Q) > k

>> Remove the top element of *Q* (say *u*) and merge it with its closest cluster *u.closest* (say *v*) and compute the new representative points for the merged cluster *w*.

>> Remove *u* and *v* from *T* and *Q*.

>> For all the clusters *x* in *Q*, update *x.closest* and relocate *x*

>> Insert *w* into *Q*

>> Repeat

The code of CURE is available in Python and C++ and can be found in GitHub.

DBCLASD

Distribution Based Clustering of Large Spatial Databases (DBCLASD) is a locality-based clustering algorithm. The authors observed that the distance from a point to its nearest neighbours is smaller inside a cluster than outside that cluster. Each cluster has a probability distribution of points to their nearest neighbours, and this probability set is used to define the cluster [47].

Three parameters are defined in the algorithm: $NN_S(q)$, $NN_{Dist}(q)$, and $NN_{DistSet}(S)$. Let q be a query point and S be a set of points. Then the nearest neighbour of q in S, denoted by $NN_S(q)$, is a point p in $S-\{q\}$ which has the minimum distance to q. The distance from q to its nearest neighbour in S is called the nearest neighbour distance of q, NN_{Dist} (q) for short. Let S be a set of points and e_i be the elements of S. The nearest neighbour distance set of S, denoted by $NN_{DistSet}(S)$, or distance set for short, is the multi-set of all values. The probability distribution of the nearest neighbour distances of a cluster is analysed based on the assumption that the points inside of a cluster are uniformly distributed. A grid-based representation is used to approximate the clusters as part of the probability calculation [48]. DBCLASD is an incremental algorithm. Points are processed based on the points previously seen, without regard for the points yet to come which makes the clusters produced by DBCLASD dependent on input order. In order to ameliorate this dependency, the algorithm uses two techniques. First, unsuccessful candidates for a cluster are not discarded, but tried again later, and second, points already assigned to a cluster may switch to another cluster later [47].

The major advantage of DBCLASD is that it requires no outside input which makes it attractive for larger data sets and sets with larger numbers of attributes, although, unlike DBSCAN, the algorithm assumes that the points inside each cluster are uniformly distributed [48].

The runtime is $O(3n^2)$ [49]. The internal loops of DBCLASD are computationally expensive and the runtime is between 1.5 and 3 times the runtime of DBSCAN [47].

Pseudocode [36]

```
procedure DBCLASD (database db)
  initialize the points of db as being assigned to no cluster;
  initialize an empty list of candidates;
  initialize an empty list of unsuccessful candidates;
  initialize an empty set of processed points;
  for each point p of the database db do
   if p has not yet been assigned to some cluster then
     create a new cluster C and insert p into C;
     reinitialize all data structures for cluster C;
     expand cluster C by 29 neighboring points[1];
     for each point p1 of the cluster C do
        answers := retrieve_neighborhood(C,p1)[2];
        update_candidates(C,answers)[3];
     end for each point p1 of the cluster C;
     expand_cluster I;
```

```
   end if p has not yet been assigned to some cluster;
  end for each point of the database;
```

[1]The χ^2-test can only be applied to clusters with a minimum size of 30. Therefore, the current cluster is expanded to the size of 30 using k nearest neighbour queries without applying the χ^2-test.
[2]The list of answers is sorted in ascending order of the distances to point *p1*.
[3]Each element of the list of answers not yet processed
is inserted into the list of candidates.

```
Procedure expand_cluster (cluster C)
  change := TRUE;
  while change do
    change := FALSE;
    while the candidate list is not empty do
       remove the first point p from the candidate list
          and assign it to cluster C;
       if distance set of C still has the expected distribution
       then
          answers:= retrieve_neighborhood(C,p);
          update_candidates(C,answers);
          change := TRUE;
       else
          remove p from the cluster C;
          insert p into the list of unsuccessful candidates;
      end if distance set still has the expected distribution;
    end while the candidate list is not empty;
    list of candidates := list of unsuccessful candidates;
  end while change;

listOfPoints procedure retrieve_neighborhood(cluster C;point p);
  calculate the radius m;
  return result of circle query with center p and radius m;

procedure update_candidates(cluster C; listOfPoints points);
   for each point in points do
      if point is not in the set of processed points then
         insert point at the tail of the list of candidates;
         insert point into the set of processed points;
      end if point is not in the set of processed points;
   end for each point in point
```

The code of DBCLASD is written in Python and can be found in GitHub.

DBSCAN

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a locality-based clustering which its density-based notion of clustering states that within each cluster, the density of the points is significantly higher than the density of points outside the cluster [47].

The algorithm uses two input parameters, *Eps* and *MinPts* to control the density of the cluster. *MinPts* is the minimum number of points in any cluster and the *Eps*-neighbourhood of a point is defined by N_{Eps} (p)={$q \in D \mid dist(p,q) \leq Eps$ }. The distance function dist(*p,q*) determines the shape of the neighbourhood [47].

To find the clusters in the database using these two values, DBSCAN starts with an arbitrary point *p* and retrieves all points density-reachable from the point *p* using *Eps* and *MinPts* as controlling parameters. If the point *p* is a core point, then the procedure yields a cluster. If the point *p* is on the border, no points are density-reachable from *p* and then DBSCAN goes on to the next point in the database. The algorithm may need to be called recursively with a higher value for *MinPts* if "close" clusters need to be merged because they are within the same *Eps* threshold [47].

The major drawbacks of the algorithm are the need of two input parameters, and the time it takes to calculate them, as it is not factored in the runtime.

The runtime is $O(n \cdot log\ n)$ according to [32] and $O(n^2)$ according to [49].

Pseudocode

```
DBSCAN(DB, distFunc, eps, minPts) {
   C = 0                                                  /* Cluster counter */
   for each point P in database DB {
      if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
      Neighbors N = RangeQuery(DB, distFunc, P, eps)      /* Find neighbors */
      if |N| < minPts then {                              /* Density check */
         label(P) = Noise                                 /* Label as Noise */
         continue
      }
      C = C + 1                                           /* next cluster label */
      label(P) = C                                        /* Label initial point */
      Seed set S = N \ {P}                                /* Neighbors to expand */
      for each point Q in S {                             /* Process every seed point */
         if label(Q) = Noise then label(Q) = C            /* Change Noise to border point */
         if label(Q) ≠ undefined then continue            /* Previously processed */
         label(Q) = C                                     /* Label neighbor */
         Neighbors N = RangeQuery(DB, distFunc, Q, eps)   /* Find neighbors */
         if |N| ≥ minPts then {                           /* Density check */
            S = S ∪ N                                     /* Add new neighbors to seed set */
         }
      }
   }
}
```

where RangeQuery can be implemented using a database index for better performance, or using a slow linear scan:

```
RangeQuery(DB, distFunc, Q, eps) {
   Neighbors = empty list
   for each point P in database DB {                      /* Scan all points in the database */
      if distFunc(Q, P) ≤ eps then {                      /* Compute distance and check epsilon */
         Neighbors = Neighbors ∪ {P}                      /* Add to result */
      }
   }
   return Neighbors
}
```

The code of DBSCAN is written in Python and has already being tested in GISEle.

OPTICS

OPTICS (Ordering Points To Identify the Clustering Structure) creates an augmented ordering of the database representing its density-based clustering structure. Its basic idea is similar to DBSCAN, but it addresses one of DBSCAN's major weaknesses: the problem of detecting meaningful clusters in data of varying density. To do so, the points of the database are (linearly) ordered such that spatially closest points become neighbours in the ordering.

Let *DB* be a database containing *n* points. The OPTICS algorithm generates an ordering of the points $o:\{1..n\} \rightarrow$DATABASE and corresponding reachability-values $r:\{1..n\} \rightarrow R \geq 0$. OPTICS does not assign cluster memberships. Instead, the algorithm stores the order in which the objects are processed and the information which would be used by an extended DBSCAN algorithm to assign cluster memberships. This information consists of only two values for each object: the core-distance and a reachability distance. The core-distance of an object *p* is simply the smallest distance ε' between *p* and an object in its ε -neighbourhood such that *p* would be a core object with respect to ε' if this neighbour is contained in $N_\varepsilon(p)$. Otherwise, the core-distance is undefined. The reachability-distance of an object *p* with respect to another object *o* is the smallest distance such that *p* is directly density-reachable from *o* if *o* is a core object. Depending on the size of the database, the cluster-ordering can be represented graphically for small data sets or can be represented using appropriate visualization technique for large data sets [48].

The parameter ε (maximum distance (radius) to consider) is, strictly speaking, not necessary. It can simply be set to the maximum possible value. OPTICS abstracts from DBSCAN by removing this parameter, at least to the extent of only having to give the maximum value.

The key parameter to DBSCAN and OPTICS is the "minPts" parameter. It roughly controls the minimum size of a cluster. If set it too low, everything will become clusters. If set it too high, at some point there will not be any clusters anymore, only noise. However, the parameter usually is not hard to choose [50].

The more difficult parameter for DBSCAN is the radius. In some cases, it will be very obvious. In other cases, the parameter will not be obvious, or it might need multiple values. That is when OPTICS comes into play [50].

OPTICS is based on a very clever idea: instead of fixing MinPts and the Radius, only the MinPts is fixed, and the radius at which an object would be considered dense is plotted by DBSCAN. In order to sort the objects on this plot, they are processed in a priority heap, so that nearby objects are nearby in the plot [50].

Using a reachability-plot (a special kind of dendrogram), the hierarchical structure of the clusters can be obtained easily. It is a 2D plot, with the ordering of the points as processed by OPTICS on the x-axis and the reachability distance on the y-axis. Since points belonging to a cluster have a low reachability distance to their nearest neighbour, the clusters show up as valleys in the reachability plot. The deeper the valley, the denser the cluster.

In Figure 17 this concept is illustrated. In its upper left area, a synthetic example data set is shown. The upper right part visualizes the spanning tree produced by OPTICS, and the lower part shows the reachability plot as computed by OPTICS. Colours in this plot are labels, and not computed by the algorithm; but it is well visible how the valleys in the plot correspond to the clusters in above data set. The yellow points in this image are considered noise, and no valley is found in their reachability plot. They are usually not assigned to clusters, except the omnipresent "all data" cluster in a hierarchical result.

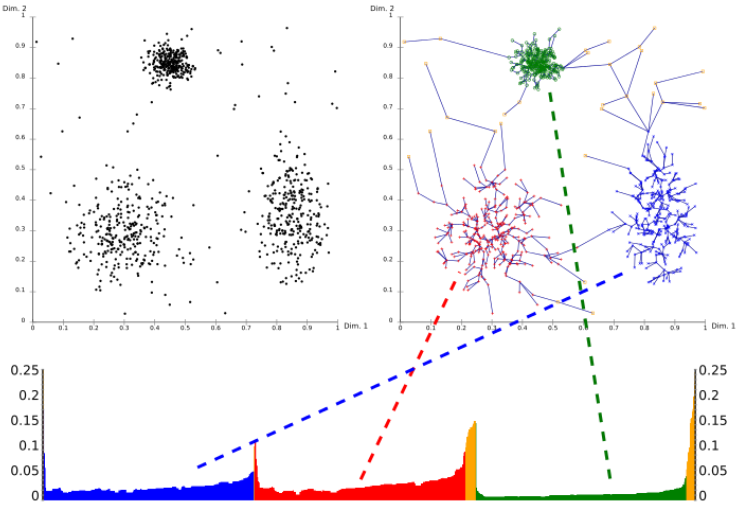

Figure 17. Reachability-plot processed by OPTICS algorithm

This could be a very interesting characteristic as not all the populated areas have the same population density.

The runtime is $O(n \cdot \log n)$. Although, the authors of the original OPTICS paper report an actual constant slowdown factor of 1.6 compared to DBSCAN.

Pseudocode

```
OPTICS(DB, eps, MinPts)
   for each point p of DB
      p.reachability-distance = UNDEFINED
   for each unprocessed point p of DB
      N = getNeighbors(p, eps)
      mark p as processed
      output p to the ordered list
      if (core-distance(p, eps, Minpts) !=
UNDEFINED)
         Seeds = empty priority queue
         update(N, p, Seeds, eps, Minpts)
         for each next q in Seeds
            N' = getNeighbors(q, eps)
            mark q as processed
            output q to the ordered list
            if (core-distance(q, eps, Minpts) !=
UNDEFINED)
               update(N', q, Seeds, eps, Minpts)
```

In update(), the priority queue Seeds is updated with the ε -neighbourhood of p and q, respectively:

```
 update(N, p, Seeds, eps, MinPts)
    coredist = core-distance(p, eps, MinPts)
    for each o in N
       if (o is not processed)
          new-reach-dist = max(coredist, dist(p,o))
          if (o.reachability-distance == UNDEFINED)
// o is not in Seeds
             o.reachability-distance = new-reach-
dist
             Seeds.insert(o, new-reach-dist)
          else               // o in Seeds, check
for improvement
             if (new-reach-dist < o.reachability-
distance)
                o.reachability-distance = new-
reach-dist
                Seeds.move-up(o, new-reach-dist)
```

The code is written in C++ and MATLAB; and can be found in GitHub.

GRAPH THEORY

A "graph" in mathematics and computer science consists of "nodes", also known as "vertices". Nodes may or may not be connected with one another [51].

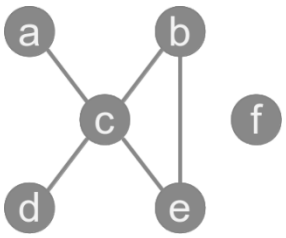

Figure 18. Example of a graph [51]

Figure 18 is a pictorial representation of a graph, where the node "a" is connected with the node "c", but "a" is not connected with "b". The connecting line between two nodes is called an edge. If the edges between the nodes are undirected, the graph is called an undirected graph. If an edge is directed from one vertex (node) to another, a graph is called a directed graph. A directed edge is called an arc.

Though graphs may look very theoretical, many practical problems can be represented by graphs. They are often used to model problems or situations in physics, biology, psychology and above all in computer science. In computer science, graphs are used to represent networks of communication, data organization, computational devices, the flow of computation…

Paths in Graphs

In graph theory, the shortest path from one node to another wants to be found. Some definitions to understand this concept are the following:

- Adjacent vertices: Two vertices are adjacent when they are both incident to a common edge.

- Path in an undirected Graph: A path in an undirected graph is a sequence of vertices
$$P = (v_1, v_2, \ldots v_n) \in V \times V \times \ldots \times V$$
such that v_i is adjacent to $v_{\{i+1\}}$ for $1 \leq i < n$. Such a path P is called a path of length n from v_1 to v_n.
- Simple Path: A path with no repeated vertices is called a simple path.

In the example of Figure 18, (a, c, e) is a simple path in the graph, as well as (a,c,e,b). (a,c,e,b,c,d) is a path but not a simple path, because the node c appears twice.

Degree

The degree of a vertex v in a graph is the number of edges connecting it, with loops counted twice. The degree of a vertex v is denoted *deg(v)*. The maximum degree of a graph G, denoted by $\Delta(G)$, and the minimum degree of a graph, denoted by $\delta(G)$, are the maximum and minimum degree of its vertices.

In the graph of Figure 18, the maximum degree is 4 at vertex c and the minimum degree is 0 at the isolated vertex f.

If all the degrees in a graph are the same, the graph is a regular graph. In a regular graph, all degrees are the same, and it can be called the degree of the graph.

The degree sum formula (Handshaking lemma):

$$\sum_{v \in V} \deg(v) = 2|E|$$

This means that the sum of degrees of all the vertices is equal to the number of edges multiplied by 2. It can be concluded that the number of vertices with odd degree has to be even (handshaking lemma).

Graph density

The graph density is defined as the ratio of the number of edges of a given graph, and the total number of edges the graph could have. In other words: It measures how close a given graph is to a complete graph.

The maximal density is 1, if a graph is complete. This is clear, because the maximum number of edges in a graph depends on the vertices and can be calculated as:

$$\max number\ of\ edges\ = \frac{1}{2} \cdot |V| \cdot (|V| - 1)$$

On the other hand, the minimal density is 0, if the graph has no edges: an isolated graph.

For undirected simple graphs, the graph density is defined as:

$$D = \frac{2|E|}{|V| \cdot (|V| - 1)}$$

A dense graph is a graph in which the number of edges is close to the maximal number of edges. A graph with only a few edges, is called a sparse graph.

Connected graphs

A graph is said to be connected if every pair of vertices in the graph is connected. Figure 19 is an example of a connected graph.

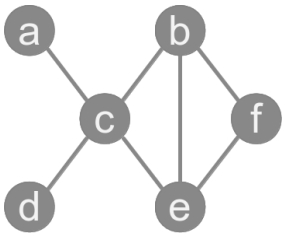

Figure 19. Example of connected graph [51]

It possible to determine with a simple algorithm whether a graph is connected:

- Choose an arbitrary node x of the graph G as the starting point.
- Determine the set A of all the nodes which can be reached from x.
- If A is equal to the set of nodes of G, the graph is connected; otherwise it is disconnected.

Distance and Diameter of a graph

The distance *dist* between two vertices in a graph is the length of the shortest path between these vertices. No backtracks, detours, or loops are allowed for the calculation of a distance.

In Figure 20, the distance between the vertex a and the vertex f is 3: dist(a,f) = 3, because the shortest way is via the vertices c and e (or c and b alternatively).

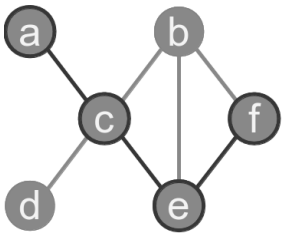

Figure 20. Example of distance between a and f in a given graph G [51]

The eccentricity e of a vertex s of a graph G is the maximal distance to every other vertex of the graph, where V is the set of all vertices of G:

$$e(s) = max(\{dist(s,v)|v \in V\})$$

The diameter d of a graph is defined as the maximum eccentricity of any vertex in the graph. This means that the diameter is the length of the shortest path between the most distanced nodes. To determine the diameter of a graph, first find the shortest path between each pair of vertices. The greatest length of any of these paths is the diameter of the graph.

In the example of Figure 20, the diameter is 3, because the minimal length between a and f is 3 and there is no other pair of vertices with a longer path.

Tree and Forest

A tree is an undirected graph which contains no cycles. This means that any two vertices of the graph are connected by exactly one simple path.

A forest is a disjoint union of trees. Contrary to forests in nature, a forest in graph theory can consist of a single tree. A graph with one vertex and no edge is a tree (and a forest).

An example of a tree is Figure 21:

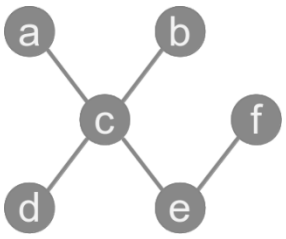

Figure 21. Example of a tree [51]

While the previous example depicts a graph, which is a tree and forest, the following picture shows a graph which consists of two trees. The graph is a forest but not a tree:

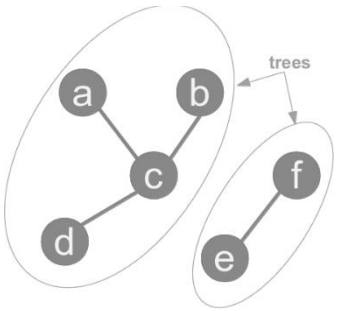

Figure 22. Forest Graph composed by two trees [51]

Spanning tree

A spanning tree T of a connected, undirected graph G is a subgraph G' of G, which is a tree, and G' contains all the vertices and a subset of the edges of G. G' contains all the edges of G, if G is a tree graph. Informally, a spanning tree of G is a selection of edges of G that form a tree spanning every vertex. That is, every vertex lies in the tree, but no cycles (or loops) are contained.

For example, in Figure 23 a fully connected graph is presented:

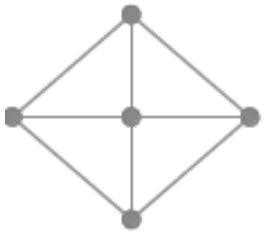

Figure 23. Fully connected graph [51]

And two spanning trees from the previous fully connected graph could be the ones presented in Figure 24:

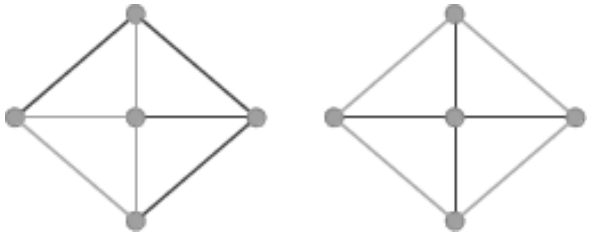

Figure 24. Spanning trees of the graph of Figure 23 [51]

A minimum spanning tree (MST) is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

LUKES

This algorithm, exposed by [52], is for partitioning a graph that is in the form of a tree. The algorithm has a growth in computation time and storage requirements that is directly proportional to the number of nodes in the tree.

Consider a graph G whose nodes have nonnegative integer weights and whose edges have positive values. A familiar combinatorial problem is the partitioning of G into subgraphs such that the sum of the node weights in any subgraph does not exceed a given maximum and the sum of the values of the edges joining the different subgraphs is minimal.

Assume a tree T = (V, E) with node set V and edge set E. A partition of T is defined as a collection of k clusters of nodes $\{c_i\}$, i= 1,2, ..., k, such that:

$$\bigcup_{i=1}^{k} c_i = V$$

$$c_i \cap c_j = \emptyset \; for \; all \; i \neq j$$

A nonnegative integer weight w_i is associated with each node i of T. A weight constraint W is imposed on each cluster of T, such that the sum of the weights of the nodes of any cluster does not exceed W. An edge *(i, j)* of T is said to be cut by a partition of T if nodes i and j are in different clusters. A positive value v_{ij} is associated with each edge *(i, j)* of T. The value of a partition of T is equal to the sum of the values of the edges of T that are within its clusters (intracluster edges); the cost of a partition of T is equal to the sum of the values of the edges of T that are cut by the partition of T (intercluster edges). Thus, the value plus the cost of a partition of T is equal to the sum of the values of the edges of T.

An optimal partition of T, p_T (opt) = $\{c_1, c_2, \ldots, c_k\}$ is one in which each cluster c_i satisfies the weight constraint:

$$\sum_{j \in c_j} w_j \leq W$$

And in which cost $\sum v_{ij}$ is minimal, where $i \in c_f, j \in c_g$, f, g=1, 2, …, k and f ≠g.

To identify the different nodes of a tree, the tree is labelled by the assignment of a unique integer to each node. To represent a partition, it is used a collection of lists in which each list represents a cluster and the contents of the list are the nodes in that cluster.

Each cluster formed by a partition of tree T represents a subtree of T. However, attention is restricted to partitions of T each of whose clusters forms a connected subtree of T, where connected means that every pair of nodes in the subtree is joined by a path.

For notational convenience, the given tree T is changed into a directed and ordered tree T'. A directed, ordered tree is defined to be a finite set T' of one or more nodes such that:

i. There is a distinct node called the root of T'.
ii. The remaining nodes (excluding the root) are, separated into $m \geq 0$ disjoint sets $T'_1, T'_2, ..., T'_m$, and each of these sets is in turn a directed, ordered tree.

The trees $T'_1, T'_2, ..., T'_m$, are called the subtrees of the root, where T'_1, is the first subtree, T'_2, the second, etc.

The particular directed, ordered tree used does not affect the growth in computational complexity of the algorithm. By convention T' is formed by selecting node 1 as the root of T' and the sons of node 1 become those nodes adjacent to (sharing an edge with) node 1. The sons of node 1 are ordered by increasing label value. If node *i* is a son of node 1, those nodes adjacent to node *i* (excluding node 1) are again ordered as the sons of node *i* by increasing label values. This process is repeated for each node in T resulting in the ordered, directed tree T', as presented in Figure 25.

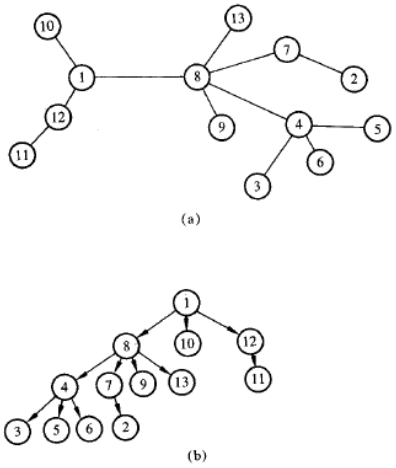

Figure 25. Transformation of a tree T (a) into a directed, ordered tree T' (b) [52]

The basis of the tree partitioning algorithm is a dynamic programming technique that takes advantage of a basic property of a tree, its acyclic nature, to find a globally optimal partition based on local information. The algorithm generates the optimal partition of the tree T' by finding the partitions of increasingly larger subtrees of T' until the partitioned subtree is T' itself.

The first step in the partitioning algorithm is to generate the trivial partitions of the leaf nodes of T'. Then, it is determined the set of nodes in T' all of whose subtrees have been partitioned. Assume that there is a node in this set with the label *x*. To generate the partitions of *x* the following sequence of steps are followed:

1. Find the partitions of node *x* and its first subtree.
2. Combine these partitions and the partitions of the second subtree of *x* to generate the partitions of, the subtree composed of node *x* and its first two subtrees.
3. Combine the partitions created in step 2 with the partitions of the third subtree of *x*. The result is the set of partitions of the subtree composed of node *x* and its first three subtrees.
4. Continue this procedure such that on the (i + 1)st step the partitions of the subtree composed of node *x* and its first i + 1 subtrees are generated by combining the partitions generated on step i with the partitions of the (i + 1)st subtree of x.
5. Finally, it is reached a point in the algorithm when the tree with root *x* is partitioned.

As a result of the partitioning of the tree with root *x*, it may happen that the node that is the father of *x* becomes qualified as a node all of whose subtrees are partitioned. If such is the case, the father of node *x* is added to this set.

Upon finishing the partitioning of the tree with root node *x*, node *x* is removed from the set of nodes sharing the property that all of their subtrees are partitioned. Another node from this set is selected, and the tree for which this node is the root is partitioned in the manner described above. At some point in the algorithm, the set of nodes each of whose subtrees is partitioned is exhausted, whereupon the optimal partition of T' has been generated.

A method of generating the partitions of a subtree U', given the partitions of subtrees U and V, where the nodes of U and V comprise the nodes of U' is considered.

In Figure 26, two subtrees, U and V, whose partitions have been generated previously, are presented. To generate the partitions of U', the subtree created by combining U and V, exist two operations:

1. Concatenate the clusters of a partition of U, u_i, with those of a partition of V, v_j, merging the nodes in the clusters containing nodes *u* and *v* into a single cluster.
2. Concatenate the clusters of u_i and v_j.

Any other combination of the partitions u_i and v_j violates the connectivity constraint.

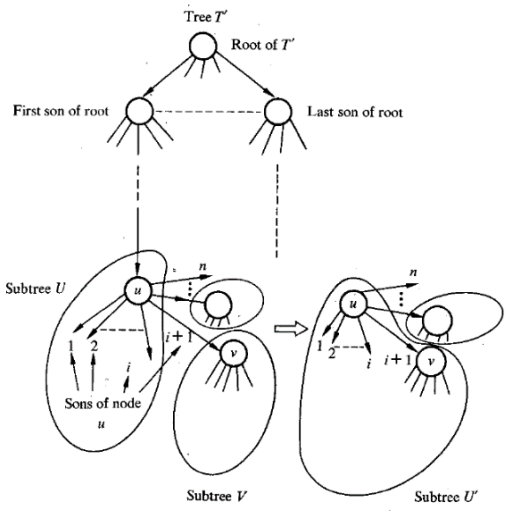

Figure 26. Generation of partitions by combining the partitions of two subtrees [52]

The notation C [u_i, v_j] is used to denote the partition created by the first operation above. The weight of the cluster of C [u_i, v_j] containing node u is $i + j$, and its partition value is the sum of the value of u_i, v_j, and edge (i, j) . The partition created by the second operation can also be denoted by the above notation if v_0 is defined to be the partition of V that has the maximal value of all partitions of V. Then C [u_i, v_0] represents a partition of U' whose cluster containing node u is of weight i and whose value is equal to the sum of the values of u_i and v_0.

Summarizing the steps associated with the partitioning algorithm:

Step 1. Label the tree T, and form the directed, order tree T'.

Step 2. For each leaf node u with weight w, form the partition $u_w = u_o = (u)$. The value of this partition is zero. For all nodes u of T' that are branch nodes (nodes having one or more sons), initialize (v) = v_j with value zero; here j is the weight of node u.

Step 3. Select some node x all of whose sons are leaf nodes, and form the optimal partitions for each weight equal to or less than the weight constraint of the subtree whose root is node x. To form these optimal partitions, follow these steps:

a. Let i = 1.

b. Form x_j' = C [x_a, y_b] (for j = w, w + 1 , . . ., W), where the operator C [x_a, y_b] forms partitions by either of the operations defined above; the particular partition C [x_a, y_b] chosen is that of maximal value. Here y is the ith son of node x and a + b = j, where $w \leq a \leq W$ and $0 \leq b \leq W$. The weight of node x is w, and the weight constraint is W.

c. Make all $x_j = x_j'$. If i = number of sons of node x, go to Step 4. Else, let i = i+1 and go to Step 3 (b).

Step 4. Denote by x_0 the partition of the subtree whose root is x that has maximal value from the set {x_w, x_{w+1}, ..., x_W}. Delete the sons of node x from the tree T'. If node

x is the root of T', then x_0 represents the optimal partition of T' (hence of T). Otherwise, go to Step 3.

In forming the optimal partition of T', it is required to combine the partitions of some U with the partitions of some V once for every edge in the tree. Since an n-node tree has n-1 edges, the computational complexity grows as W^2n and is independent of the particular structure of the tree under consideration.

STING

Statistical Information Grid-based method (STING) exploits the clustering properties of index structures. It divides the spatial area into rectangular grid cells which forms a hierarchical structure. This makes STING order independent. Each cell at level *I* is partitioned into a fixed number *k* of cells at the next level [47].

The algorithm stores parameters in each cell which are designed to aid in answering certain types of statistically based spatial queries. In addition to storing the number of objects or points in the cell, STING also stores some attribute dependent values. STING assumes that the attributes have numerical values, and stores the following data [47]:

- *m*: the mean of all the values in the cell.
- *s*: the standard deviation of all values of the attribute in the cell.
- *min*: the minimum value of the attribute in the cell.
- *max*: the maximum value of the attribute in the cell.
- *dist*: the type of distribution followed by the attribute value in the cell.

Clustering operations are performed using a top-down method, starting with the root. The relevant cells are determined using the statistical information, and only the paths from those cells down the tree, are followed. Once the leaf cells are reached, the clusters are formed using a breadth-first search, by merging cells based on their proximity and whether the average density of the area is greater than some specified threshold. This is similar to DBSCAN but using cells instead of points. Thus, STING finds an approximation of the clusters found in DBSCAN [47].

In addition to the granularity of the grid which reduces the quality of the clusters, STING also does not consider the spatial relationship between a cell and its siblings when constructing the parent cell. This also may cause a degradation in the quality of the clusters.

The runtime complexity for STING is O(n) where n is the number of grid cells at the lowest layer, which is usually much lower than the number of objects. The smaller the n, the more approximate are the clusters. The lower the granularity, the higher the n, the slower the algorithm will run [47].

Pseudocode [53]

1. Determine a layer to begin with.
2. For each cell of this layer, calculate the confidence interval (or estimated range) of probability that this cell is relevant to the query.
3. From the interval calculated above, label the cell as relevant or not relevant.
4. If this layer is the bottom layer, go to Step 6; otherwise, go to Step 5.
5. Go down the hierarchy structure by one level. Go to Step 2 for those cells that form the relevant cells of the higher-level layer.
6. If the specification of the query is met, go to Step 8; otherwise, go to Step 7.
7. Retrieve those data fall into the relevant cells and do further processing. Return the result that meet the requirement of the query. Go to Step 9.
8. Find the regions of relevant cells. Return those regions that meet the requirement of the query. Go to Step 9.
9. Stop.

The code of STING could be found written in Python in GitHub.

COBWEB

COBWEB is an incremental system for hierarchical conceptual clustering. The core idea is to build a classification tree, based on some heuristic criteria, in order to realize hierarchical clustering on the assumption that the probability distribution of each attribute is independent.

The four basic operations COBWEB employs are merging two nodes, splitting a node, inserting a node and passing an object down the hierarchy.

This algorithm does not seem to be the best for the problem of clustering population as the objective is not creating a tree.

The runtime depends on the distribution [32].

DENCLUE

DENCLUE (Density Clustering) is a generalization of partitioning, locality-based and hierarchical or grid-based clustering approaches. The algorithm models the overall point density analytically using the sum of the influence functions of the points. Determining the density-attractors causes the clusters to be identified. DENCLUE can handle clusters of arbitrary shape using an equation based on the overall density function. The authors claim three major advantages for this method of higher-dimensional clustering [47].

- Firm mathematical base for finding arbitrary shaped clusters in high-dimensional data sets
- Good clustering properties in data sets with large amounts of noise
- Significantly faster than existing algorithms

The approach of DENCLUE is based on the concept that the influence of each data point on its neighbourhood can be modelled mathematically. The mathematical

function used, is called an impact function. This impact function is applied to each data point and the density of the data space is the sum of the influence function for all the data points. In DENCLUE, since many data points do not contribute to the impact function, local density functions are used. Local density functions are defined by a distance function, in this case, Euclidean distance. The local density functions consider only data points which actually contribute to the impact function. Local maxima, or density-attractors identify clusters. These can be either centre-defined clusters, similar to k-means clusters, or multi-centre-defined clusters, linked by a particular path which identify clusters of arbitrary shape. Clusters of arbitrary shape can also be defined mathematically. The mathematical model requires two input parameters, α and ξ. α is a parameter which describes a threshold for the influence of a data point in the data space and ξ is a parameter which sets a threshold for determining whether a density-attractor is significant [47]. The density parameter and the noise threshold need to be selected carefully as it significantly affects the quality of results [49].

The algorithm DENCLUE first performs a pre-clustering step, which creates a map of the active portion of the data set, used to speed up the density function calculations. The second step is the clustering, including the identification of density-attractors and their corresponding points. Using a cell-based representation of the data allows the algorithm to work very efficiently [47].

The runtime is $O(log|D|)$ [49].

Pseudocode [49]

1. Take Data set in Grid which each side is of 2σ
2. Find highly dense cells
3. Find out the mean of highly populated cells.
4. If d(mean(cl), mean(c2)) < 4a then two cubes are said to be connected.
5. Now highly populated or cubes that are connected to largely populated cells will be taken into consideration in determining clusters.
6. Find Density Attractors using a Hill Climbing procedure.
7. Randomly pick point r.
8. Compute Local 4 σ density.
9. Pick another point (r+1) close to previous computed density.
10. If denI < den(r+1) climb.
11. Put points within (σ /2) of path into cluster.
12. Connect the density attractor-based cluster.

The code is written in Java and can be found here in GitHub.

ENSEMBLE BASED CLUSTERING TECHNIQUES

In the first filtering step, this type of algorithm has not been considered as in the table there is not information provided about its features. So, in this section, a deeper overview is done to see if this algorithm could be also a good option for the problem addressed.

The goal of any machine learning problem is to find a single model that will best predict the wanted outcome. Rather than making one model and hoping this model to be the best/most accurate predictor, ensemble methods take a myriad of models into account, and average those models to produce one final model.

Ensemble techniques are used mostly for supervised learning, however, they have been used also in unsupervised learning scenarios, for example in consensus clustering or in anomaly detection.

Consensus clustering, also called cluster ensembles or aggregation of clustering (or partitions), refers to the situation in which a number of different (input) types of clustering have been obtained for a particular dataset and it is desired to find a single (consensus) clustering type which is a better fit in some sense than the existing clustering algorithms.

An important issue of the clustering analysis is the validation of the clustering results. Lacking an external objective criterion, consensus clustering provides a method that represents the consensus across multiple runs of a clustering algorithm, to determine the number of clusters in the data, and to assess the stability of the discovered clusters.

The **pseudocode** of an example of an ensembled clustering based on mixture model consensus algorithm proposed by [54] is:

```
begin
  for i=1 to H // H – number of clusterings
    cluster a dataset X: π←k-means(X)
    add partition π to the ensemble Π = {Π, π}
  end
  initialize model parameters Θ = {α1,...αM,...θ1 ... θM}
  do until convergence criterion is satisfied
    compute expected values E[zim], i=1..N, m=1..M
    compute E[zim yimis] for missing data (if any)
    re-estimate parameters (k)jm ϑ, j=1..H, m=1..M, ∀k
  end
  πC (xi) = index of component of zi with the largest expected value, i=1..N
  return πC // consensus partition
end
```

The fact that the number of clusters is needed *a priori* is not the best for the problem studied here, as it is not known which the best number of clusters is to divide the area.

5.3 COMPARISON OF THE ALGORITHMS

To select the best algorithm among the ones described in detail, they will be compared (COBWEB is discarded) in terms of time complexity, the input parameters and the program language in which the code is already written, giving priority to those written in Python.

Table 14. Comparison of the clustering algorithms for densely populated areas

Algorithm	Time Complexity	Number of input parameters	Language of the code
K-MEANS	O (n)	1	Python
CURE	O ($n^2 \log n$)	1	Python, C++
DBCLASD	O($3n^2$)	0	Python
DBSCAN	O ($n \cdot \log n$)// O(n^2)	2	Python
OPTICS	O ($n \cdot \log n$)	2	C++, MATLAB
LUKES	O ($W^2 n$)	1	Python
STING	O(n)	5	Python
DENCLUE	O (log |D|)	2	Java

According to the programming language, k-means, CURE, DBCLASD, DBSCAN, LUKES and STING are good options for the clustering densely populated areas as they are written in Python.

This procedure aims to be introduced in GISEle, in order to improve the tool and improve the rural electrification process. As GISEle is written in Python, the procedure to sit secondary substations is going to be written in Python too.

In terms of time complexity, the runtime of DBCLASD doubles the one of DBSCAN, and CURE is more complex than the others. Although the time complexity of OPTICS seems acceptable, the code is not available in Python and still need DBSCAN algorithm. The time complexity of LUKES depends on the input data and the constraint W.

Finally, although STING has a low runtime complexity and it is available in Python, it is discarded due to the high number of input parameters that it needs.

So, taking these three features into account, k-means, DBSCAN and LUKES seems to be the more suitable option for clustering densely populated areas.

In Table 15, pros and cons of each algorithm are presented to deep in this analysis.

Currently, GISEle is running a slightly modified version of DBSCAN to cluster the populated areas.

Table 15. Pros and cons of the clustering algorithms

ALGORITHM	PROS	CONS
K-MEANS	· Low runtime · Only one input · Easy to implement and introduce constraints	· Only convex shapes · High sensitivity to input data · Falls in local optima
CURE	· Only one input · Low sensitivity to outliers	· High runtime complexity · Intermediate steps needed for large databases · Number of clusters as input needed · Merges points only by distance
DBCLASD	· No outside input · Solves the dependency of the input order	· Assumes points inside each cluster uniformly distributed · High runtime complexity · Computationally expensive · Need to set a distribution to follow the clustering process
DBSCAN	· Acceptable runtime complexity · Good deal with outliers	· Two input parameters
OPTICS	· Detects clusters with varying density · One of the parameters could be a maximum possible value instead of a fixed value · Acceptable runtime complexity	· Needs DBSCAN as intermediate step · Two input parameters · Not written in Python
LUKES	· The input is a constraint · Detects clusters with varying density	· Slow running times, increasing with the number of nodes
STING	· Low runtime complexity · Order independent · Capacity to store attributes in addition to the number of objects	· Loss of quality as taking cells instead of points · High number of inputs · Does not consider spatial relationship between cells and siblings
DENCLUE	· Firm mathematical base · Good deal with noise · Faster than other algorithms	· Not written in Python · Two input parameters

Chapter 6

PROPOSED METHODOLOGY

6.1 MOTIVATION OF THIS WORK

As presented in Chapter 2, the power system is divided into the transmission and distribution system, at different voltages levels. In real projects, the consumers, here the cells, are connected through low voltage lines to the secondary substations (MV/LV), these secondary substations are connected through medium voltage lines to the primary substations (HV/MV) and these substation are connected to the generation sites through high voltage lines.

Moreover, the LV lines account for 60% of the length of the voltage lines and the losses in LV network are approximately 50% of the total losses of the power system [30].

As explained in the previous section, for the moment, GISEle clusters the population and then evaluates if it is worthy to connect each cluster to the national grid or to create an isolated micro-grid. However, GISEle is connecting the clusters directly to the primary substation, skipping all the distribution system. Furthermore, to connect the cluster to the HV/MV substation, GISEle takes the closest cell of the cluster to the existing substation and does the grid routine from that cell to the substation.

As can be noted, this is not a very realistic interpretation. So, as this procedure aims to be introduced in GISEle, in order to improve the tool and improve the rural electrification process, a study to site MV/LV substations has been done. This is achieved by an improvement of the load clustering step:

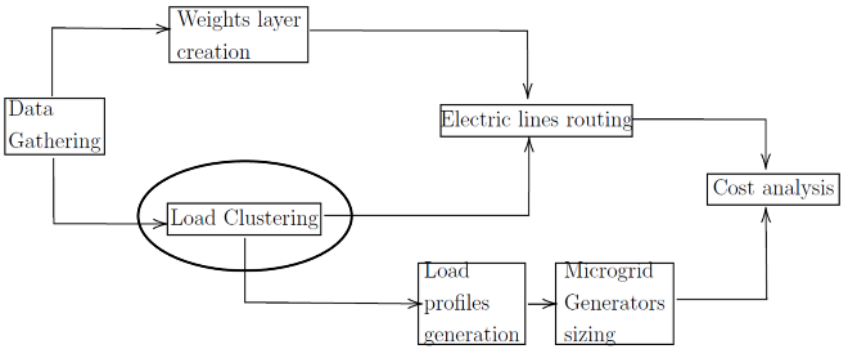

Figure 27. Step to improve in GISEle

6.2 PROPOSITION TO SITE SECONDARY SUBSTATIONS

In this study, the objective is to design a procedure able to evaluate the optimal location of secondary substations from a topological perspective considering the population distribution given by georeferenced data. The approaches presented in Chapter 4 and algorithms of Chapter 5 have been considered.

As mentioned before, the mathematical methods and artificial intelligence methods have been discarded as the size of the managed data is big and this would lead to complex systems and high computational times.

On the other hand, the heuristic algorithms that aim at minimizing a cost function, as the genetic algorithm or the ICA, have been discarded as this is not an economic study.

Considering the input data as population distribution given by georeferenced data, the location of consumers and the load can be estimated in the problem. The k-means and graph partitioning algorithms, as MST and Dijkstra's algorithm, seem to fit with the data.

According to literature and the analysis of clustering densely populated areas of sections 4.3, 5.2 and 5.3, it has been decided to develop a two-step clustering procedure. The two steps consist of:

1. Clustering the population of the big area, detecting the densely populated areas and discarding outliers. For this step DBSCAN has been chosen.
2. Clustering the population of the clusters obtained through DBSCAN in order to divide the population obtaining sub-clusters inside each group that represents the area supplied by a secondary substation. For this step, k-means and LUKES algorithm have been chosen.

DBSCAN is useful at the beginning because it detects the areas with a specific population density. And, as the objective is to site substations where the people are, this algorithm finds these areas in an efficient way.

An analysis over the two algorithms chosen for the second step needs to be done in order to choose the one with best results.

In this section, the codes implemented to conduct the analysis are presented. Both steps and algorithms have been written in Python.

In the LV network, the length of the lines is important and also the power of the substations should meet the power demand of the households. Because of this, a distance and a power constraint have been implemented in the codes.

The distance constraint has been introduced in the k-means algorithm. This means, the distance between the points and the centres of their clusters is calculated and evaluated imposing a maximum value that should not be overcome.

Conversely, in the LUKES algorithm a power constraint is inherent to the code. This means, the creation of the clusters is based on adding points to a group until a maximum power is reached.

6.3 K-MEANS ALGORITHM

The solution of k-means is the points gathered in groups named clusters and represented by the mean value of all the points, named the centre. This centre represents the location of the substation.

To evaluate the results and validity of the k-means algorithm two different codes have been implemented and tested: a normal k-means and a weighted k-means, with three different versions of each one:

- *With no loop:* running the algorithm once and keep the first solution obtained.
- *With a simple loop*: imposing a distance constraint between substation and load and adding a cluster each time the algorithm finds a distance higher than the threshold.
- *With a complex loop*: imposing a distance constraint between substation and load and dividing in two only the clusters that do not meet the constraint.

The objective of introducing a loop is to automatize the optimal number of clusters and solve the problem of selecting the number of clusters a priori. The main difference between the version of the simple loop and the complex loop is that with the complex loop only the sub-clusters that do not meet the distance constraint are divided. By contrast, the simple loop takes all the points and divides them from the beginning, instead of focusing only in the sub-clusters with problems.

The procedure is explained in detail in the following paragraphs.

NORMAL K-MEANS

In the module *sklearn.cluster* of Python, the function *KMeans* is found. A code with this function has been written and implemented. The k-means problem is solved using either Lloyd's or Elkan's algorithm. The average complexity is given by O (knT), were n is the number of samples and T is the number of iterations. In practice, the k-means algorithm is very fast (one of the fastest clustering algorithms available), but it falls in local minima. That is why it can be useful to run it several times.

The method called in the code for this function is: *fit_predict (self, X [, y, sample_weight])*, which computes the cluster centres and predict cluster index for each sample.

With the normal k-means only the distance constraint is considered.

No loop version

This first version of the code takes the input file with the points of a certain cluster and their coordinates, asks the number of sub-clusters *k* in which the big cluster wants to be divided as an input. The number of initial clusters is set dividing the peak demand of all the area by the power of the substation. Then, it applies directly the k-means algorithm. When it finishes, it creates two output files:

- One for the points with the coordinates and the number of their cluster.
- Other for the centres and their coordinates.

The flow chart of this code is presented in Figure 28:

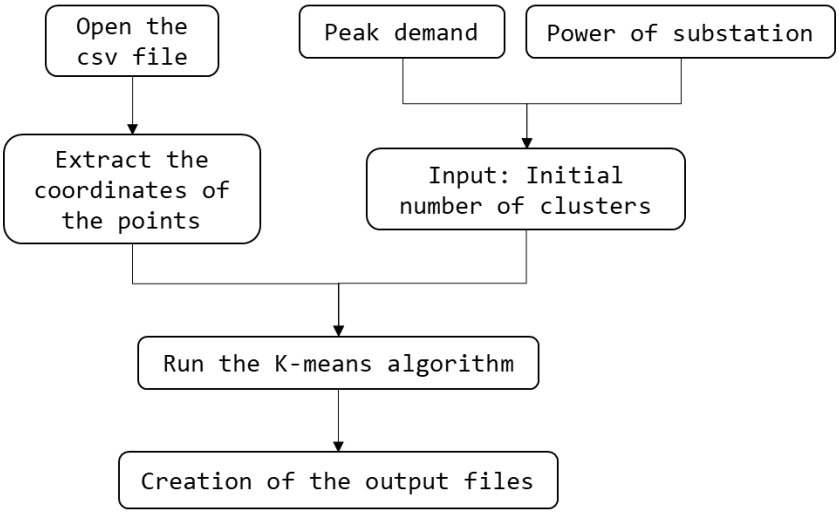

Figure 28. Structure of the normal k-means algorithm with no loop

Simple loop version

This second version of the code takes the input file with the points of a certain cluster and their coordinates, asks the number of sub-clusters k in which the cluster wants to be divided as an input and applies the k-means algorithm. However, this time, the distance between each point and the centre of its cluster is calculated and evaluated to see if it exceeds the threshold imposed. If not, it finishes and creates two output files:

- One for the points with the coordinates and the number of their cluster.
- Other for the centres and their coordinates.

If the threshold is exceeded, through a loop, it adds a new cluster and runs the k-means algorithm again for k+1 clusters and evaluates the distances again until it arrives to a solution where all the distances are under the limit. Then, the two same output files are created.

The structure of this code is presented in Figure 29:

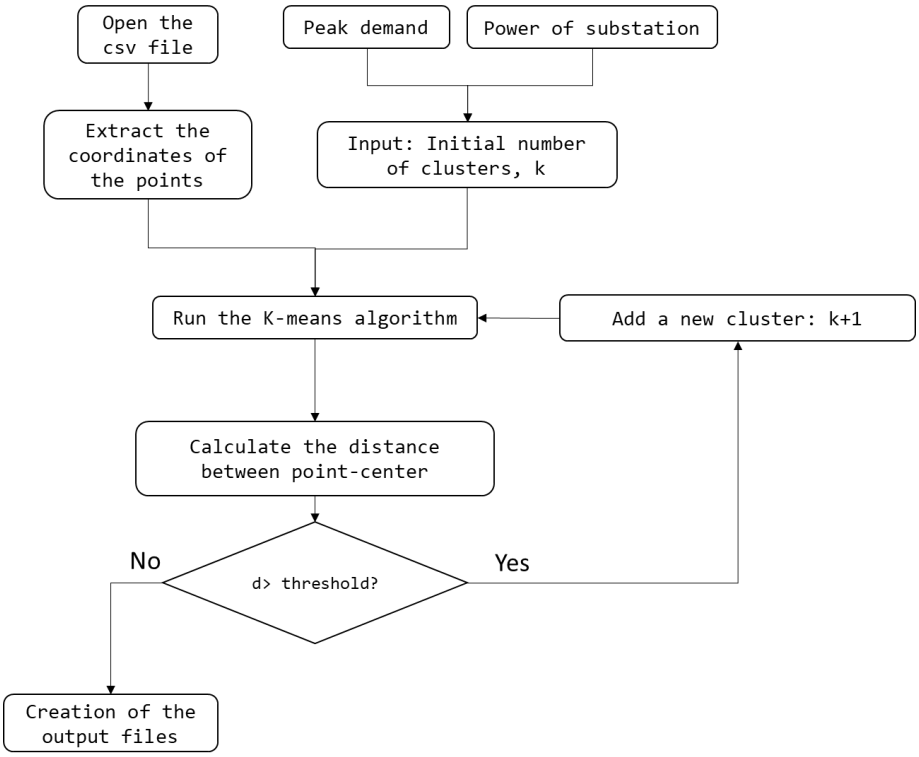

Figure 29. Structure of the normal k-means algorithm with a simple loop

Complex loop version

This third version of the code takes the input file with the points of a certain cluster and their coordinates, asks the number of sub-clusters *k* in which the cluster wants to be divided as an input and applies the k-means algorithm. Then, the distance between each point and the centre of its cluster is calculated and evaluated to see if it exceeds the threshold imposed. If not, the procedure finishes and creates two output files:

- One for the points with the coordinates and the number of their cluster.
- Other for the centres and their coordinates.

If, however, some points of a cluster exceed the threshold distance, the k-means algorithm is run only on that cluster which is divided in two parts. Then the procedure continues evaluating the distances and dividing in two the rest of the clusters in which some point exceeds the distance threshold. After a first round, it recalculates the new centres, computes the new distances between the points and the new centres and studies if there are distances higher than the limit. If yes, it goes through the loop again dividing in two only the clusters that do not meet the distance constraint, and if not, it exits the loop and creates the two output files.

The structure of this code is presented in Figure 30:

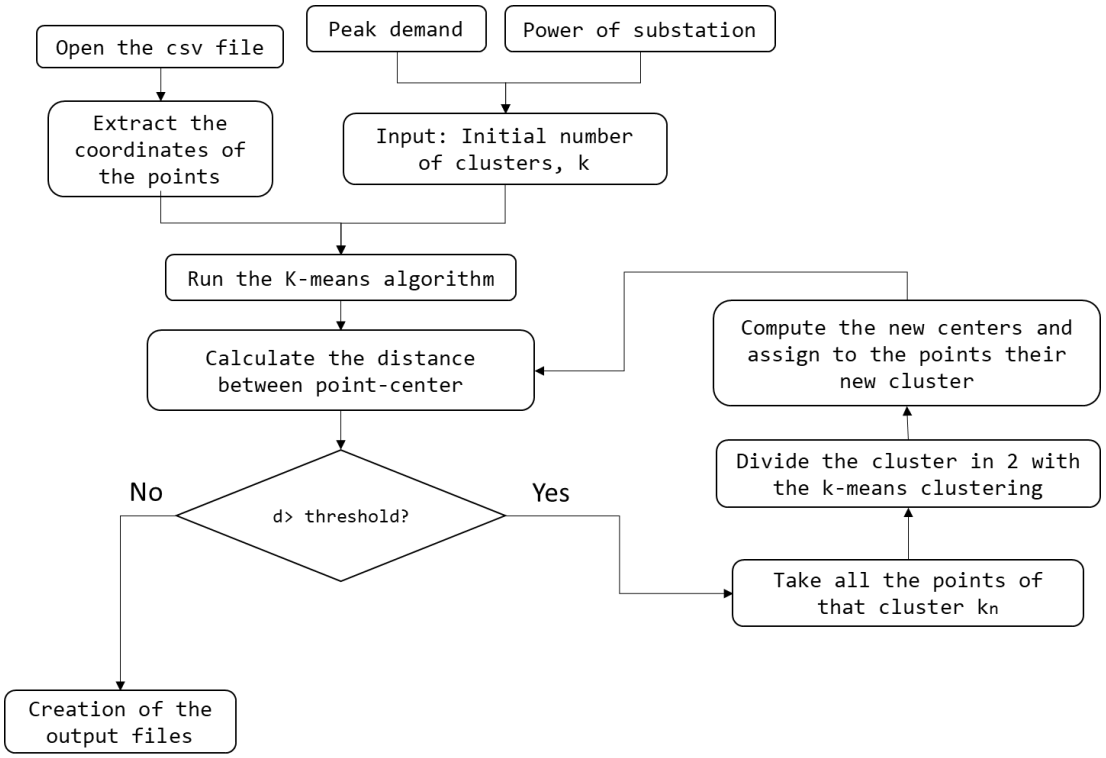

Figure 30. Structure of the normal k-means algorithm with a complex loop

WEIGHTED K-MEANS

The "weighted" k-means problem is a natural extension of the k-means problem that allows to include some more information, namely, a set of weights associated with the data points. These might represent a measure of importance, a frequency count, or some other information. The intent is that a point with a weight of 5 is twice as "important" as a point with a weight of 2.5, for instance.

In the weighted k-means problem, a set of N points $X(I)$ in M-dimensions are given, and a corresponding set of nonnegative weights $W(I)$. The goal is to arrange the points into K clusters, with each cluster having a representative point $Z(J)$, usually chosen as the weighted centroid of the points in the cluster:

$$Z(J) = \frac{\sum_{all\ X(I) in\ cluster\ J} W(I) \cdot X(I)}{\sum_{all\ X(I) in\ cluster\ J} W(I)}$$

The weighted energy of cluster J is:

$$E(J) = \sum_{all\ X(I) in\ cluster\ J} W(I) \cdot \|X(I) - Z(J)\|^2$$

In this case, the weight $W(I)$ is the power associated to each point. This will be obtained with the population of the point and the power per capita.

The steps of the weighted k-means are the following:

- Declaring the number of clusters, k.

- Creating a list *points* with the coordinates of the points under study and their population, extracted from a csv file.
- Creating the list *centres*, with the function *equally_spaced_initial_clusters*. This function initializes the coordinates of the centres by taking the coordinates of all the points, calculating the mean value of the *y*-axis and assigning this value to all the *y* coordinates of the centres; and for the x-axis, it takes the maximum and minimum value of the *x* of the points, divides the range in *k-1* equally parts and assign the *x* value of the centres from the minimum to the maximum value of *x* separated the step distance.

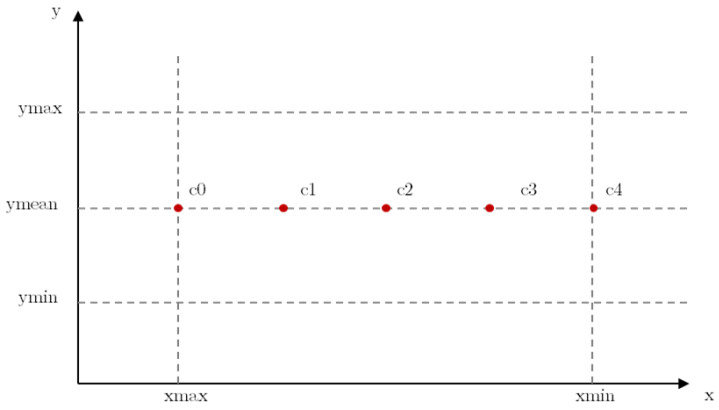

Figure 31. Example of the equally_spaced_initial_clusters for k=5

- Calling the function called *data_weighted_kmeans (points, centres, k)*, where the *points* are the coordinates of the points of the area under study, the *centres* are the coordinates of the centres initialized in the previous step and *k* is the number of clusters in which the area wants to be divided.
  Inside this function, first, each point is assigned to the closest centre and for each centre two columns, one with the number of points and other with the total population of the cluster, are added. Secondly, the mean value of the points of each cluster is calculated to obtain the new centre coordinates. However, these new coordinates are going to be computed not only considering the physical coordinates but also the population of that points (the weight *W(I)*), multiplying the coordinates of each point by their population weight. Then, the points are evaluated again to find the nearest cluster (see Equation of *Z(J)*).

Weights assignment

In order to test the performance of the weighted k-means, different weights are assigned to the points.

The criterion to assign different weight to some points is based on the hypothesis of higher power demand growth for those kinds of areas.

For instance, some features considered when assigning more weight to the points are:

- Distance to the centre of the village: usually people tend to live closer to the centre of the villages, so more importance (more weight) is given to these areas.
- Size of the roof of the houses: the bigger the size of the roof, usually more people are living in that building and/or higher is the power demand, so more weight is given to that buildings.
- Density distance between buildings: areas with less distance between buildings are usually denser the areas in terms of population, so more weight is given to these areas.
- Distance to the road: usually in developing countries people tend to establish around the roads, and together with the fact that electric posts go along roads, these areas are given higher weights.

No loop version

This first version of the code takes the input file with the points of a certain cluster and their coordinates, power and population, and asks for the number of sub-clusters *k* in which the big cluster wants to be divided as an input.

Then it computes the *equally_spaced_initial_clusters* and the *data_weighted_kmeans,* which returns the coordinates of the points, the centres and the number of iterations done until it has reached the final solution.

When it finishes, it creates two output files:

- One for the points with the coordinates, the number of their cluster, the power and the population.
- Other for the centres with their coordinates, number of points inside that cluster and the population.

The flow chart of this code is presented in Figure 32:

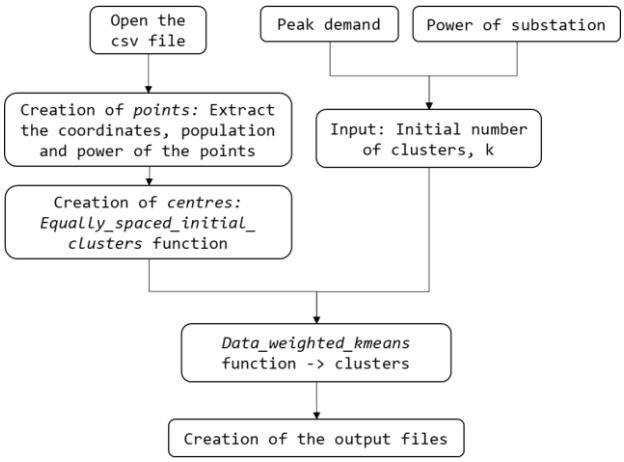

Figure 32. Structure of the weighted k-means algorithm with no loop

Simple loop version

This second version of the code takes the input file with the points of a certain cluster and their coordinates, power and population, and asks for the number of sub-clusters *k* in which the big cluster wants to be divided as an input.

Then it computes the *equally_spaced_initial_clusters* and the *data_weighted_kmeans,* which returns the coordinates of the points, the centres and the number of iterations done until it has reached the final solution.

However, this time, the distance between each point and the centre of its cluster is calculated and evaluated to see if it exceeds the threshold imposed. If not, it finishes and create two output files:

- One for the points with the coordinates, the number of their cluster, the power and the population.
- Other for the centres with their coordinates, number of points inside that cluster and the population.

If the threshold is exceeded, through a loop it adds a new cluster and runs the weighted k-means algorithm through the *equally_spaced_initial_clusters* and *data_weighted_kmeans* again for k+1 clusters and evaluates the distances again until it arrives to a solution where all the distances are under the limit. Then, the two same output files are created.

The structure of this code is presented in Figure 33:

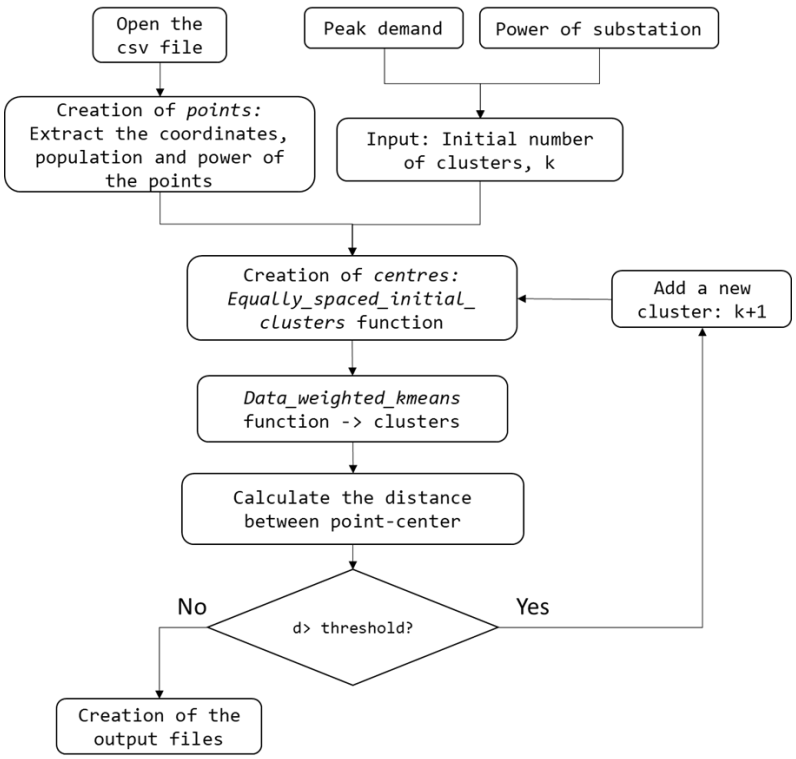

Figure 33. Structure of the weighted k-means algorithm with a simple loop

Complex loop version

This third version of the code takes the input file with the points of a certain cluster and their coordinates, power and population, and asks for the number of sub-clusters *k* in which the big cluster wants to be divided as an input.

Then it computes the *equally_spaced_initial_clusters* and the *data_weighted_kmeans,* which returns the coordinates of the points, the centres and the number of iterations done until it has reached the final solution.

Then, the distance between each point and the centre of its cluster is calculated and evaluated to see if it exceeds the threshold imposed. If not, it finishes and create two output files:

- One for the points with the coordinates, the number of their cluster, the power and the population.
- Other for the centres with their coordinates, number of points inside that cluster and the population.

If yes, through a loop, the code goes evaluating all the distances. When it finds the threshold is exceeded, it takes all the points of the cluster of that point that does not meet the distance constraint and computes the *equally_spaced_initial_clusters* and the *data_weighted_kmeans* only to that cluster dividing the group in two. Then, continues evaluating the distances and dividing in two the rest of the clusters in which some point exceeds the distance threshold. After a first round, it recalculates the new centres, computes the new distances between the points and the new centres and studies if there are distances higher than the limit. If yes, it goes through the loop again dividing in two only the clusters that do not meet the distance constraint, and if not, it exits the loop and creates the two same output files.

The structure of this code is presented in Figure 34:

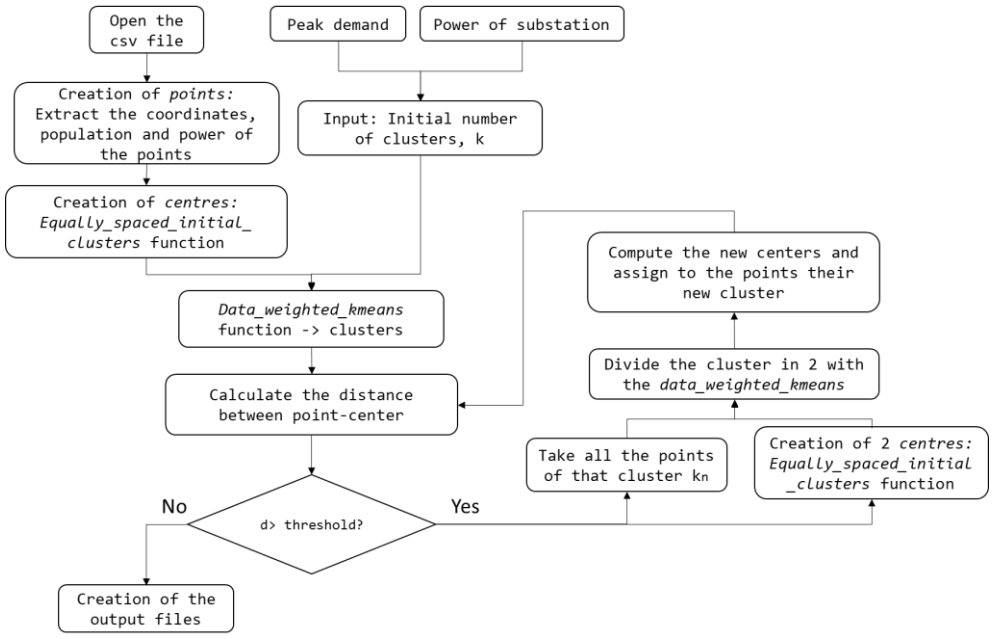

Figure 34. Structure of the weighted k-means algorithm with a complex loop

6.4 LUKES ALGORITHM

GRAPH THEORY

The Graph Theory has been implemented in this study taking the populated points as the nodes of the graph and creating a minimum spanning tree, with the edges as the representation of the low voltage lines.

The data is a grid, where each cell/point has a population density associated. The idea is to:

- Connect all the points. The points are the nodes and the lines are the edges. Each edge will have a weight. This weight will be the length of the line. In future steps, this length will be used to calculate the cost of the line, with a cost per metre of line, c[€/m].
- Once all the points are connected, the objective is to have the minimum cost. As the length and the cost are directly proportional, the longest edges will be cut off, with the MST function.
- The nodes will have also attributes. This means, they will have information associated to them. In this case, the attributes set to them will be the power and population of each node.

To design the MST of the input data, the Python module *NetworkX* has been used. *NetworkX* is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

The module has different functions for the graph itself, the nodes, the edges or the attributes, for example. Among the utilities that exists, some of them are:

- Graph: calculate the degree of a node, the density of the graph or copy the graph.
- Nodes: copy the nodes, obtain the number of nodes or obtain the neighbours of a node.
- Edges: obtain a list of the edges or obtain the number of edges.
- Attributes: set node attributes, set edges attributes or get the edges attributes of a graph.

Once the MST is created, a graph partitioning algorithm is applied to divide the grid in clusters, which will represent an area supplied by a secondary substation.

This graph partitioning algorithm is LUKES. It has been chosen as its performance perfectly fits with the characteristics of the data available and objective pursued. To understand this, the description of the properties of the graph are presented with how the code is implemented:

- Nodes (V): represent the populated points.
- Weight of the nodes (w): it is the power associated to the people of the node. The power is calculated multiplying the number of people by the average power per capita.

- Weight constraint (W): it is the maximum power a cluster could have. This means the maximum power of the transformer, as a cluster represents an area supply by a secondary substation.
- Edges (E): represent the low voltage lines. They connect populated points.
- The weight of the edges (v): it is the inverse of the distance between points/nodes. As the objective is to minimize costs, and costs are directly proportional to the length of the lines, if the distance between nodes is taken as the weight of the edges, the shortest distances are going to be cut instead of the longest. It is desirable to keep the shortest distances as is where the low voltages lines are going to be potentially constructed.

THE CODE

The core of the code is a function called *lukes_partitioning*.

Lukes_partitioning (G, max_size: int, node_weight = None, edge_weight = None)

This function is called introducing:

- G: the graph.
- Max_size (int): the maximum weight a partition can have in terms of sum of the node_weight for all the nodes in the partition.
- Edge_weight (key): edge data key to use as weight. If None, the weights are set to one.
- Node_weight (key): Node data key to use as weight. If None, the weights are all set to one. The data must be int.

The function returns a list of set of nodes representing the clusters of the partition.

The function works as follow:

i. It confirms if G is a tree and chooses the node root among the list of the nodes. Then it returns an oriented tree constructed from a depth-first search from source.
ii. It makes a copy of the original graph to not lose the information.
iii. It checks if all the node's weights are integers.
iv. Definition of the functions of subroutines:
    a. Function to look for the leaves' nodes.
    b. Function to look for the parents of that leaves nodes.
    c. Functions to calculate the value of the cluster, the value of the partition and the weight of the cluster.
    d. Function to concatenate or merge different partitions.
v. Initialization: it sets the leaves' nodes and save the other nodes.
vi. Core algorithm: evaluation of the nodes and production of the partitions by LUKES algorithm [52].

Once the function calling the LUKES algorithm is understood, the code implemented is:

1. Open a csv file with the coordinates, the population and the power of each point. Data presented as a DataFrame.
2. Compute the distance matrix between all the points.
3. Create a complete graph with all the connections between the points.
4. Create the Minimum Spanning Tree through a function of the module *scipy.sparse.csgraph* of Python.
5. Define a distance matrix with the MST. Compute an *Adj_matrix_inv* with the inverse of the distances.
6. Create a tree type graph G, with edge weight the inverse of the distances and the node weight the power of the points.
7. Call the function *lukes_partitioning* and obtain a list with the final clusters represented by their nodes.
8. Add a new column to the DataFrame with the number of the cluster.
9. Create an output csv file with the coordinates, population, power and number of the cluster of each point.

This structure of the algorithm is represented in Figure 35.

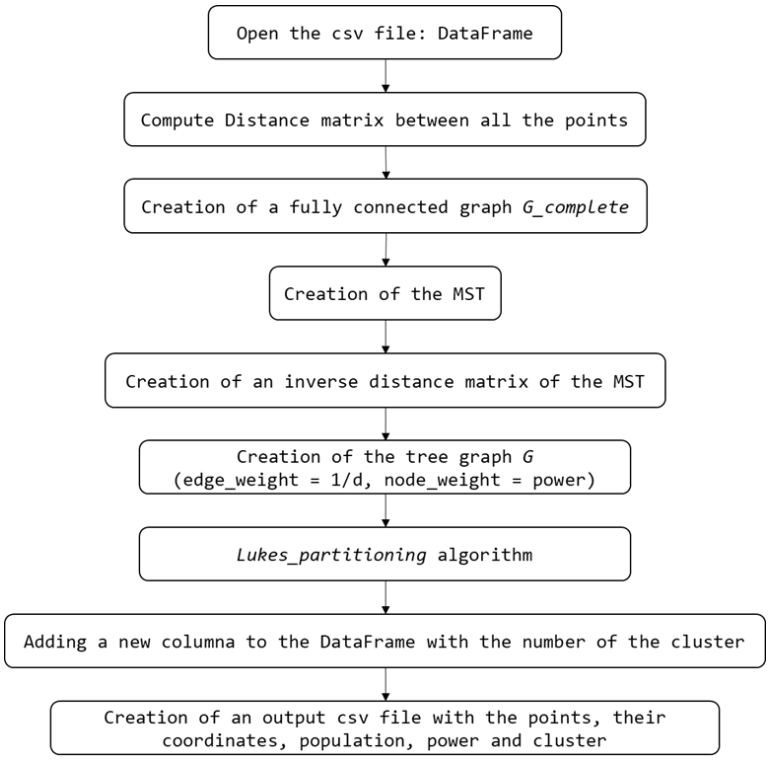

Figure 35. Structure of the implemented code with LUKES algorithm

Chapter 7

CASE STUDIES

Two case studies have been conducted to test the siting substations procedure. The first one is conducted in Namanjavira, a rural administrative post in Zambezia, Mozambique. The second case study is conducted in Omereque, a municipality of Cochabamba, Bolivia.

7.1 NAMANJAVIRA

CONTEXT OF THE COUNTRY AND AREA UNDER STUDY

Mozambique is a country in Southeast Africa bordered by Zimbabwe, Swaziland, South Africa, Tanzania, Malawi, Zambia, and the Indian Ocean. It has an area of 801,590 km^2 and a population of 30,366,036 inhabitants (2019), with a 51.44% of female population and a growth rate of 2.5% per year. In 2019, 62.2% of the population lived and worked in rural areas.

The country is well endowed with natural resources: arable land, forests, fish resources, water, mineral resources and solar energy. Its economy has recorded an average annual increase of around 7.5% from 2005 to 2015, without however managing to face deep development challenges, since the rapid macroeconomic growth was not matched by a significant reduction in the poverty of its population: social inequality is very pronounced (Gini coefficient = 54, [55]) and there is a substantial absence of an adaptation plan of the consumption of natural resources, in terms of efficiency and conservation.

Mozambique is one of the poorest countries of the world, with a GDP per capita of 499 US$ in 2018 and about 46.1% of the population living below the poverty line in 2015. The Mozambican population is characterized by an extreme and systematic fragility: the majority lives in a condition of high vulnerability and chronic crisis caused by systemic factors such as poor primary sector productivity, strong population growth (+ 43% in the last decade, + 80% in the last twenty years), the persistence of strong socio-economic tensions (which are exacerbated with increasing inequalities), the depletion of natural resources and the absence of other resources to ensure inclusive and equitable sustainable economic growth.

The country is divided in 10 provinces: Niassa, Cabo Delgado, Nampula, Tete, Zambezia, Manica, Sofala, Gazam Inhambane, Maputo; and one capital city with province status, Maputo City.

Figure 36. Provinces of Mozambique

The north-central provinces of Nampula and Zambezia are the most populous regions of Mozambique and account for 45% of the total population.

The area under study is Namanjavira, a *Posto Administrativo* inside the Mocuba District which belongs to the province of Zambezia. The reason behind choosing this area is to continue with the study started by Carnovali and Edeme [24] last year in their thesis work. They chose this area for the following reasons:

- One of the authors was working on an electrification project in this area (the Italian Cooperation for Development Agency (AICS)'s ILUMINA Program implemented by the Italian NGO COSV).
- Being GISEle designed to deal with both greenfield and brownfield areas, in order to effectively guarantee the validity of the case study's results, all data had to be most possible accurate. Unfortunately, the dataset related to the existing distribution network path in the country is not available. For this reason, Namanjavira had the most suitable configuration because despite being completely unelectrified, the EDM's planned transmission network has two lines crossing its borders.
- Being the uncertainty in terms of computational burden of the final model relatively high, concentrating on one smaller area gave the possibility of testing all algorithms' outputs, despite having a 200m spatial resolution, quite high compared to other tools that generally do not go below 1km.

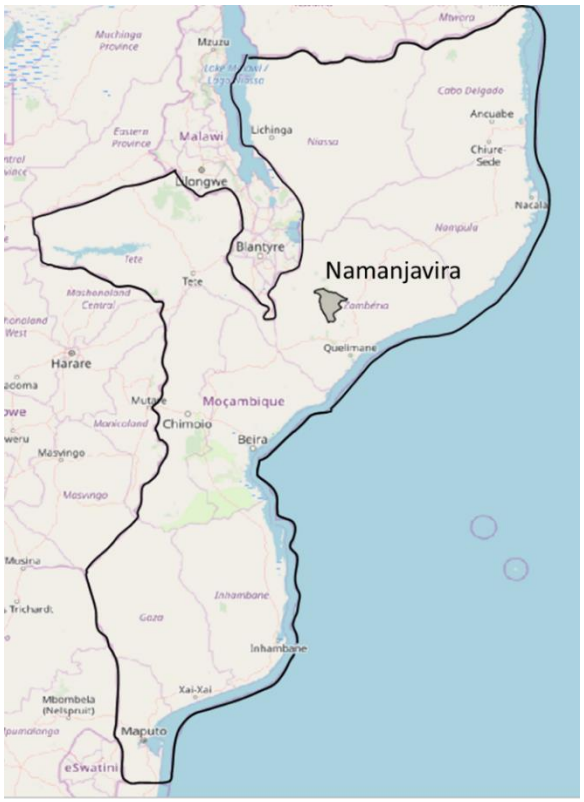

Figure 37. Map of Mozambique with Namanjavira highlighted

ENERGY SECTOR OF MOZAMBIQUE

At 187 gigawatts, Mozambique has the largest power generation potential in Southern Africa from untapped coal, hydro, gas, wind and solar resources. Hydropower currently accounts about 81% of installed capacity. However, natural gas and renewable energy sources occupy a growing share of Mozambique's energy mix. Despite the outsized potential, only 29% of the population has access to electricity, living 15% in rural areas and 57% in urban areas, due to limited transmission and distribution networks and unfavourable market conditions for new generation. The industrial and commercial segments are expected to drive demand growth, as residential consumers struggle with the existing highly subsidized tariffs.

According to [16], the national grid is largely managed by state-owned utility *Electricidade de Moçambique* (EDM). A small proportion of the lines are owned by *Hidroeléctrica de Cahora Bassa* (HCB), the operator of the *Cahora Bassa* hydroelectric plant, and by Mozambique Transmission Company (MOTRACO), which supplies power to the Mozal aluminium smelter owned by BHP Billiton. The smelter is not supplied by EDM; it imports power directly from ESKOM SA. Additionally, the Government established the FUNAE (Fundo de Energia), an administratively and financially autonomous public institution with the role of supporting and developing the management of energy resources, being responsible for the off-grid electrification field, to be complementary to EDM, the National Grid operator.

The transmission system is composed by three separate systems, northern, central and southern, although the northern and central systems have some interconnection. These three regions are [16]:

- The northern region has a 220 kV transmission system covering about 1,000 km from the Songo substation to Nampula and continuing at 110 kV to the town of Nacala. A separate 220 kV system (operated at 110 kV) extends from Tete, linking with the central region at Chibata.
- The central region has a 110 kV system linking the hydroelectric power stations at Chicamba and Mavuzi with the load centres in the Beira-Manica corridor.
- The southern region comprises a 110 kV network extending from Maputo to XaiXai, Chokwe and Inhambane, together with a 275 km single-circuit line from Maputo to Komatipoort, where it connects with the system operated by South African utility ESKOM.

The government intends to further expand the transmission grid until 2025, particularly from the resource-rich north to the capital Maputo in the south, via the creation of a backbone transmission project consisting of one 400 kV HVAC line and one 800 kV or 500 kV HVDC line, effectively creating a grid to service the country's major consumption zones and connect to the South African market. The line is anticipated to carry electricity generated from the Mphanda Nkuwa and Cahora Bassa extension generation projects as well as provide power to the SAPP, at an estimated cost of USD 2 billion [24].

In 2017, the length of the MV lines was 17,580 km with a power of 2378 MVA. EDM is carrying out numerous projects to develop distribution systems. These projects consist of 66 kV lines, 110/66 kV substations, distribution networks with 33 kV medium voltage lines, low voltage lines, 110/33 kV (10 MVA) and 66/33 kV (10MVA) substations, service connections, consultancy services for preparation of detailed design and supervision of works and project audit [17]. Precisely, they are:

- Construction of 64 km of fully operational 66 kV transmission lines.
- Construction of 360 km of fully operational 33 kV medium voltage line.
- Construction of 105 km of fully operational low voltage lines.
- Setting up of one fully operational 110/33 kV (10 MVA) substation.
- Setting up of four fully operational 66/33 kV (10 MVA) substations.
- Connection of 3000 consumers.
- Consultancy services recruited and services delivered.
- Project audit carried out.

The tariff structure is divided in social, household, farming, general tariff and a flat rate. The price per kWh is different depending on the number of kWh consumed. EDM connection fees do not recover the full cost of connecting new customers [17].

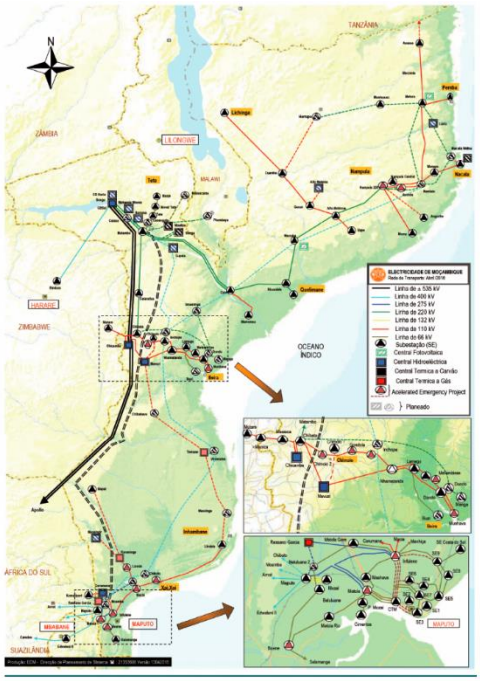

Figure 38. Transmission and distribution power system in Mozambique

The large distances between generation and consumption, dependency on single lines as well as large parts of the country not covered are major challenges for electricity supply and electrification. Grid breakdowns have led to widespread electricity outages, due to a lack of resilience on the system. This fragility has been evidenced by the loss of supply following the floods of January 2015, and by over 59 hours of transmission interruptions in 2013. The average interruption time increased from 30 minutes in 2009 to 68 minutes in 2013. In 2015, floods have damaged the main transmission line connecting the northern part of the country, which led to 4 weeks of outages for some consumers.

The DSO indicators from Chapter 2 for Mozambique are exposed in Table 6.

INPUT DATA

As mentioned before, this project suggests a two-step clustering procedure to site secondary substations.

For the first step, the DBSCAN algorithm is implemented. The input data is the population data with its coordinates.

For the second step, the input data used to test the algorithms to site secondary substation are:

- The population data divided in the clusters obtained through the DBSCAN clustering algorithm.
- The power per capita consumed in that cluster.
- The distance constraint.
- The power constraint (W).

Population

The position of the points of the input data for the DBSCAN clustering step is presented in Figure 39.

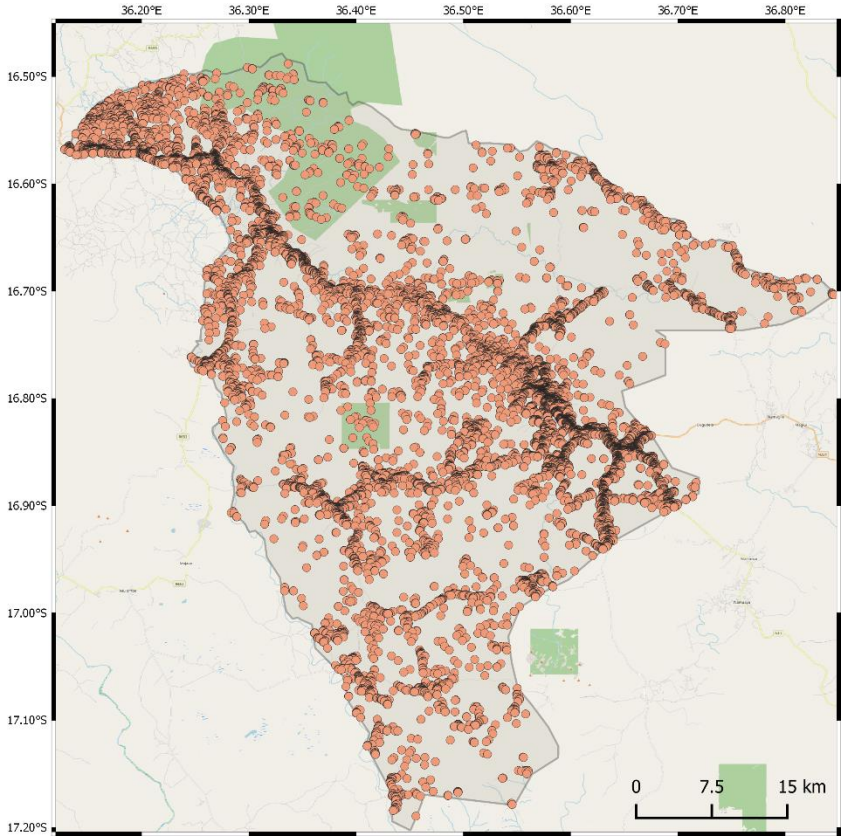

Figure 39. Input data for the first clustering step in Namanjavira

Power

Taking the results of the load profiles obtained by [24], the maximum power per capita is approximately 0.025 kWh/person a day, that means 9,13 kWh/capita per year. It is a very low value, realistic due to the rural area and low average income of the population.

Distance constraint

The maximum LV line length from transformer to the end of LV line is 1.1 km [18]. These long lines lead to losses. However, for the distance constraint, a rounded value of 1 km (1000 metres) has been chosen to be implemented in the codes [30].

Power constraint

The maximum power per cluster is the power a transformer can supplied. Each cluster will in fact be supplied by a MV/LV transformer. Due to the rural area, the maximum power chosen as constraint is W=50 kW.

7.2 OMEREQUE

CONTEXT OF THE COUNTRY AND AREA UNDER STUDY

Bolivia is a landlocked country located in western-central South America bordered to the north and east by Brazil, to the southeast by Paraguay, to the south by Argentina, to the southwest by Chile, and to the northwest by Peru. The capital is Sucre, while the seat of government and financial centre is located in La Paz.

It has an area of 1,098,581 km^2 and a population of 11,513,100 people [56], with a 50 % of female population and a growth rate of 1.41% per year [56]. In 2019, 31% of the population lived and worked in rural areas.

The country's population is multi-ethnic, including Amerindians, Mestizos, Europeans, Asians and Africans. Spanish is the official and predominant language, although 36 indigenous languages also have official status, of which the most commonly spoken are Guarani, Aymara and Quechua languages.

In Bolivia 5.8% of the population live under 1.90$/day, 11.8% under 3.20$/day and 24.7% under 5.5'$/day. The GDP per capita was 3,823US$ in 2019. Despite a series of mostly political setbacks, between 2006 and 2009 the Morales administration has spurred growth higher than at any point in the preceding 30 years. The growth was accompanied by a moderate decrease in inequality, with a GINI coefficient = 42 in 2017 [55].

A major blow to the Bolivian economy came with a drastic fall in the price of tin during the early 1980s, which impacted one of Bolivia's main sources of income and one of its major mining industries. Since 1985, the government of Bolivia has implemented a far-reaching program of macroeconomic stabilization and structural reform aimed at maintaining price stability, creating conditions for sustained growth, and alleviating scarcity. A major reform of the customs service has significantly improved transparency in this area. Parallel legislative reforms have locked into place market-liberal policies, especially in the hydrocarbon and telecommunication sectors, that have encouraged private investment.

The sovereign state of Bolivia is a constitutionally unitary state, divided into nine departments: Pando, La Paz, Beni, Oruro, Cochabamba, Santa Cruz, Potosí, Chuquisaca and Tarija:

Figure 40. Territorial division of Bolivia

The department of Cochabamba is the third most populated with 1,758,143 inhabitants in 2012. It is divided in 16 provinces which are further subdivided into 47 municipalities. The area under study is located in the municipality of Omereque in the province of Campero. Omereque has an area of 880 km² and a population of 5,643 inhabitants.

The reason to have chosen this area is due to the availability of real data thanks to a project the *Politecnico* is realizing in collaboration with the Spanish NGO *Luces Nuevas.*

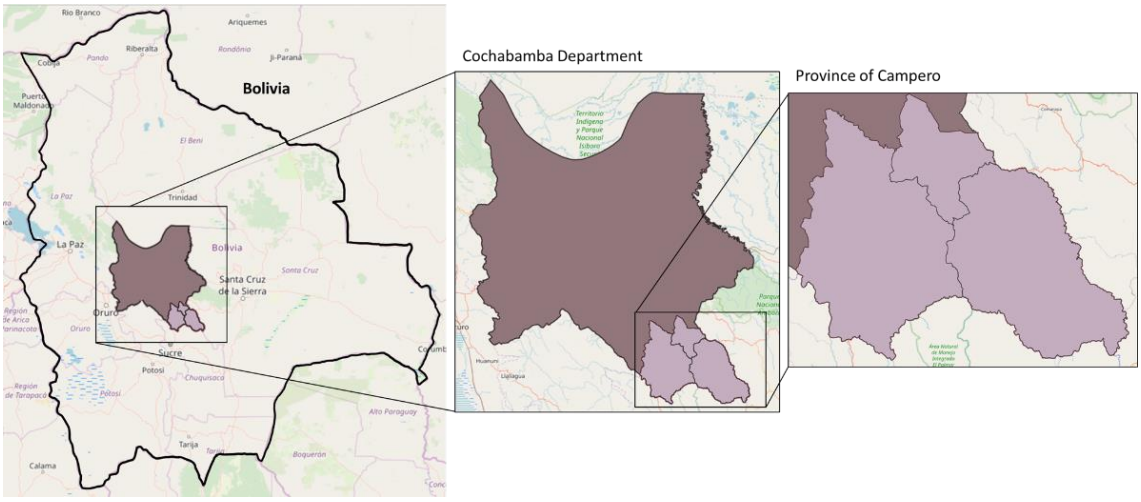

Figure 41. Map of Bolivia with the province of Campero highlighted

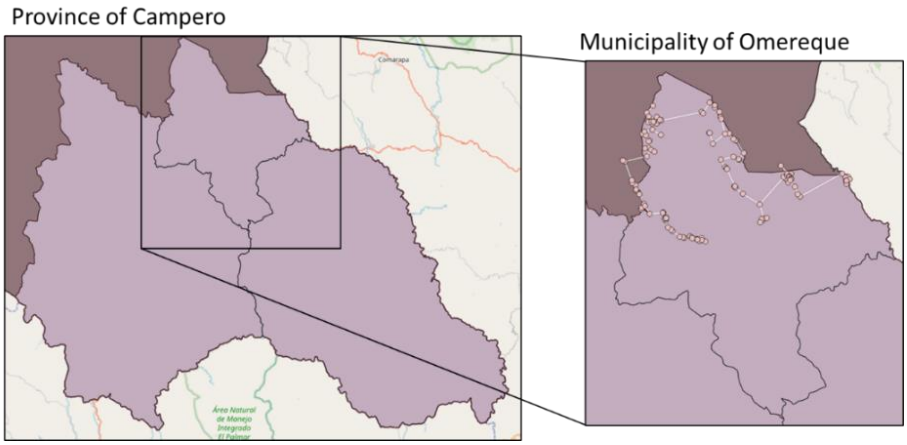

Figure 42. Area under study in the Municipality of Omereque

ENERGY SECTOR

Bolivia is one of the poorest countries of Latin America. While urban areas such as La Paz and Santa Cruz are modern cities with a relatively good supply of modern energy services, most Bolivia's rural areas are still experiencing a lack of most basic services, including reliable and affordable access to electricity and improved biomass cookstoves. The electrification ratio of Bolivia in 2017 was 91,8%, being 99% in urban areas while in rural areas is less than the 75%.

Furthermore, 43% of the rural population uses biomass fuels for their daily cooking and heating needs. Despite the advances made in recent years to reduce rural poverty, there are still regions which have little access to markets, basic public services, and energy. Because of the difficult topography of Bolivia, communication and transport are a challenge in general and there are regions that are completely isolated during the rainy season. This situation is gradually worsening due to the impacts of climate change in Bolivia.

Electricity is nearly exclusively generated by private companies from hydropower (36,3%) and thermal power plants mainly based on gas (59,7%). The total installed capacity is 1645 MW [57], the installed capacity connected to the National Grid System (SIN) in 2011 was 1.31 GW [57] and the contribution of other renewable sources than hydropower is almost negligible. 85% of the electricity were produced in the *Sistema Interconectado Nacional* (SIN - National Grid System), while 15% were produced in isolated systems (mainly diesel-driven generators). The demand for electricity rose in the last 20 years dramatically and led into a series of outages and unsatisfied demand. There was a significant rise in the household sector.

The Bolivian electricity market is strictly divided into three fields: generation, transmission, distribution. One company is not allowed to work in more than one of this fields. However, there is an exception for off-grid systems. Most of the electricity companies have been nationalized.

All mayor cities -except Tarija and Trinidad- are connected to the national grid. A line to connect Tarija is under construction.

Currently, there are two transmission companies in the SIN, *Transportadora de Electricidad* (TDE) and ISA Bolivia which runs 53% of the transmission network in Bolivia.

In Bolivia, the seven existing distribution companies enjoy a geographic monopoly in their concession areas. The largest company (in terms of kWh sold) is *Electropaz*, majority-owned by Spain's Iberdrola. The second place is occupied by the *Empresa de Luz* y *Fuerza Eléctrica Cochabamba (ELFEC)*, which was owned by the American PPL Global until 2007; followed by the Rural Electrification Cooperative (CRE), which operates in the Department of Santa Cruz.

The departments of Beni, Pando and Tarija and the eastern region of Santa Cruz are not integrated in the SIN. As a result, there are vertically integrated operators that provide the service. The most important operators are:

- SETAR (*Servicios Eléctricos Tarija, S.A.*): 44 MW, serves 56,885 clients
- ENDE (*Empresa Nacional de Electricidad*): 16.65 MW, serves 16,650 clients
- CRE (*Cooperativa Regional de Electricidad*): 14.53 MW, serves 4,940 clients.

In some cases, especially in the high plateau, cooperatives and community organizations access the distribution companies' network and sell electricity to small rural communities. Sometimes, those are organized enterprises that provide the service to middle-size towns, but in most cases, they are small organisations that serve family communities. This situation faces a legal vacuum since the consumers benefiting from these schemes, who do not consume the minimum power legally established, cannot be considered as regulated ones. In addition, these consumers are localized outside the distribution companies' concession areas, so they cannot receive the companies' service. In practice, the distribution companies are reselling electricity to the mentioned organizations outside the legal framework. Accurate information on the number of organizations that operate in rural areas does not exist. However, there are approximately three in La Paz, twenty in Oruro and three in Potosi.

The Bolivian government's efforts to improve delivery of energy services to the poor have been quite intensive in recent years. First, the broad energy sector reform programme that comprised among others the privatisation of state utilities, was implemented in the mid-1990s. The reform improved the overall performance of the electricity sector and achieved important coverage gains in urban areas, connecting and providing access to the grid for about 98% of the urban population. The access rate in rural areas, however, has grown from 13.7% in 1997 to 46,6% in 2010.

In 2002, the government of Bolivia developed an ambitious rural electrification plan (*PLABER – Plan Bolivia de Electrificación Rural*) to increase access to electricity in rural areas from 25% to 45% within five years. However, implementation of the plan has been slow due to the ongoing political and economic crisis.

The transmission lines in Bolivia in 2015, according to ENDE, operate at 250, 115 and 69 kV.

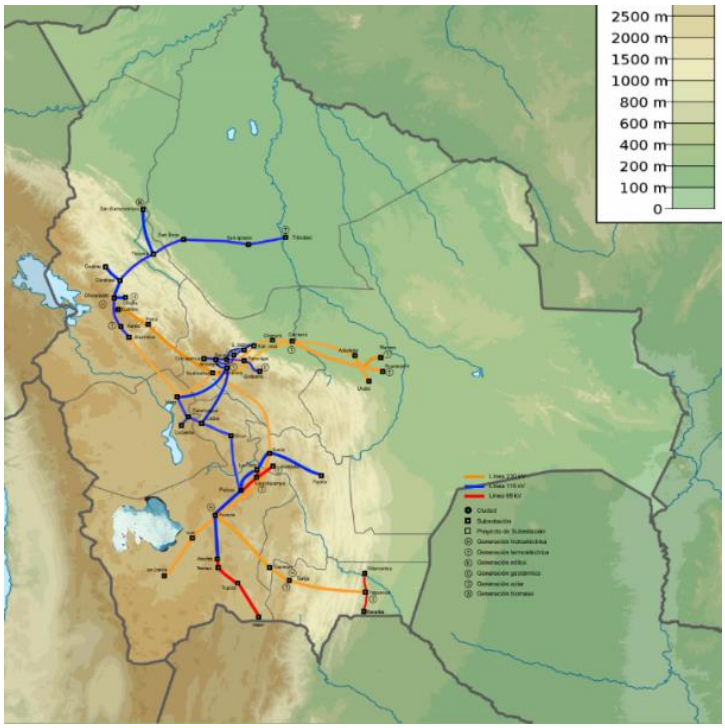

Figure 43. Transmission system in Bolivia

According to [58] and DELAPAZ *Distribuidora de Electricidad de la Paz*, the MV lines in Bolivia operate at 44, 24.9 and 19.9 kV and the LV lines of 220 V. The typical transformer capacity is between 25 and 15 kVA.

AT the Geo Portal VMEEA [59], information about the transmission and distribution system of Bolivia can be found, as it is shown in Figure 44:

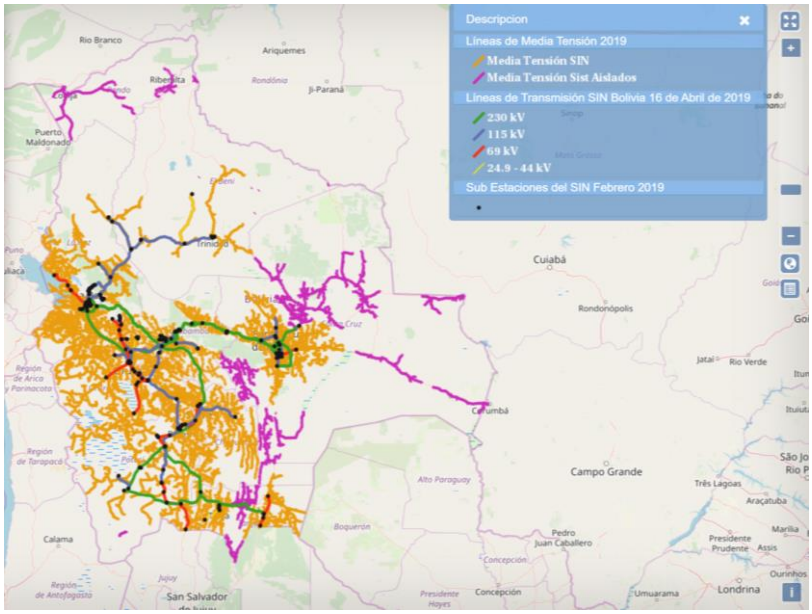

Figure 44. Transmission and distribution system in Bolivia [59]

The estimated DSO indicators in Bolivia according to literature are presented in the following table:

Table 16. DSO indicators of Bolivia

ID	DSOs indicators	Value
1	LV circuit length per LV consumer	500 -1000 m
2	Number of LV consumers per MV/LV substation	16 -26
3	MV/LV substation capacity per LV consumer	0.958 kW
4	Number of MV supply points per HV/MV substation	14 -24
5	Typical transformation capacity of MV/LV secondary substations in rural areas	15, 25 kVA

INPUT DATA

As mentioned before, this project suggests a two-step clustering procedure to site secondary substations. However, due to the sparsity of population and the limited dimension of the area under study, the first step, DBSCAN clustering, is not performed and the optimal site of secondary substations over an area in Omereque (Bolivia) with data provided by the NGO *Luces Nuevas* is directly found.

The input data used for the second step of the procedure to test the algorithms to site secondary substation are:

- The number and position of the households. This would be the output from the DBSCAN step.
- The power per capita consumed in that cluster.
- The distance constraint.
- The power constraint (W).

Population

The position of the households is presented in Figure 45. An assumption of 5 people per household in Bolivia has been adopted. The number of points is 139 and the total population 695 people.

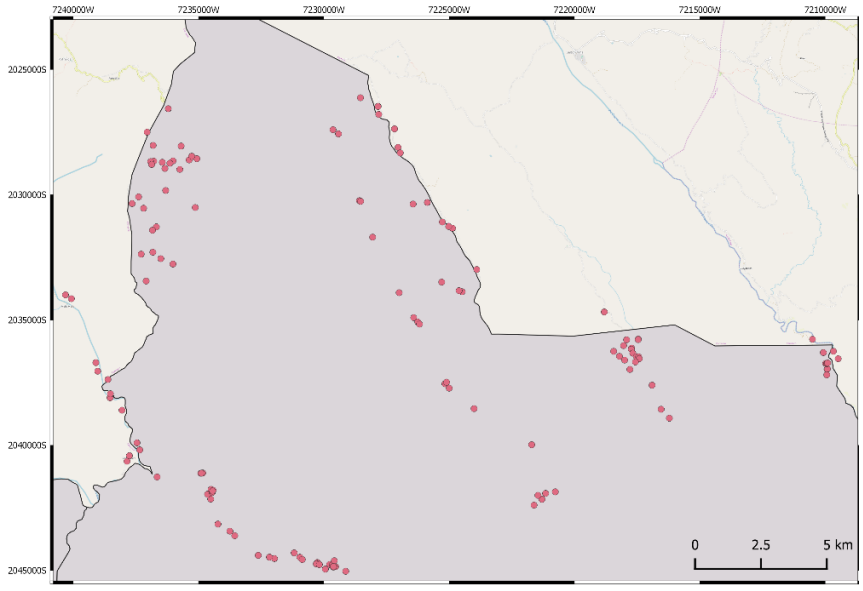

Figure 45. Position of the households in the study area in Omereque

Power

According to the study in Omereque, the power per capita in these households is 0.4 kW. Considering a number of 5 people per household, this leads to a power of 2 kW per household.

According to data from Enel, in Italy the power per household is 3 kW. In this rural area of Bolivia, the power is expected to be lower than in Italy, and this hypothesis is satisfied.

Distance constraint

According to official documents, the maximum LV line length from transformer to the end of LV line in this area is 600 metres. So, a distance constraint of 600 m is adopted in the study case.

Power constraint

The maximum power per cluster is the power a transformer can supply. As this data is smaller than the one of Namanjavira, several values for the power constraint have been adopted in order to better analyse the performance of LUKES. Due to the rural area, the maximum power chosen as constraint are W=10, W=20 and W=25 kW.

Chapter 8

RESULTS

This chapter presents the results obtained in the areas of Namanjavira and Omereque after running the k-means and LUKES algorithms in order to place secondary substations and the comparison of the performance of both algorithms.

8.1 NAMANJAVIRA

The objective of this work is to create and test a population clustering procedure which could allow to find the optimal location of secondary substations and assign one substation to each cluster. The way to divide the cluster is to place the minimum number of substations per sub-cluster meeting a distance constraint between loads and substations while not overcoming the maximum load the substation can provide.

DBSCAN CLUSTERING

The clusters obtained from the input data (Figure 39) of the study of Carnovali and Edeme with the DBSCAN clustering algorithm and a grid distance of 30 metres are presented in Figure 46. As can be noticed, not all the points have been assigned to a cluster. They have been considered as outliers, meaning not chosen for electrification.

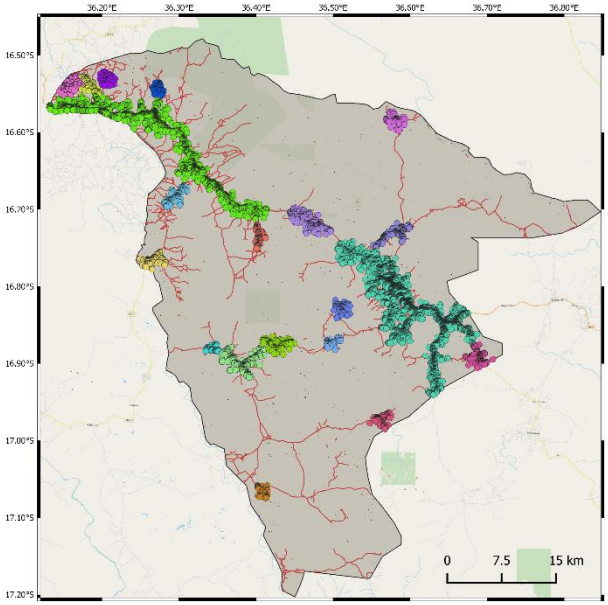

Figure 46. Output of the first step with DBSCAN in Namanjavira

The cluster chosen as input data for the second step of the procedure is cluster 10, which is reported in Figure 47, where the red lines are the roads of the area:

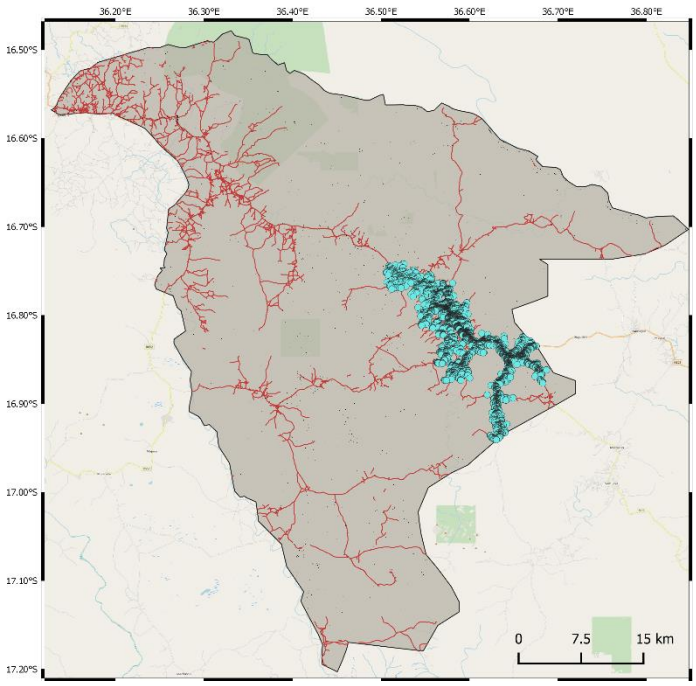

Figure 47. Input data for the second step in Namanjavira

The input data has a population of 4 people per point, 4,729 points and a total population of 18,916 people.

WEIGHTING PROCESS

For the weighted k-means different weights have been assigned to the points of cluster 10 according to the power. The criterion of giving more weight to some areas is based on the hypothesis of higher power demand growth for these areas. As mentioned before in section 6.3, different features have been considered.

In particular, for the area of this case study, higher power per capita, and hence weight has been assigned to:

i. Areas with higher population density, extracted from QGIS with the population layer. The population is represented with black dots.
ii. Areas where, according to data collected from OpenStreetMaps, there are recognized villages. The polygons are represented in green colour.
iii. Areas situated near the main road. The main road is represented with a yellow dashed line.

The area under study is presented in Figure 48:

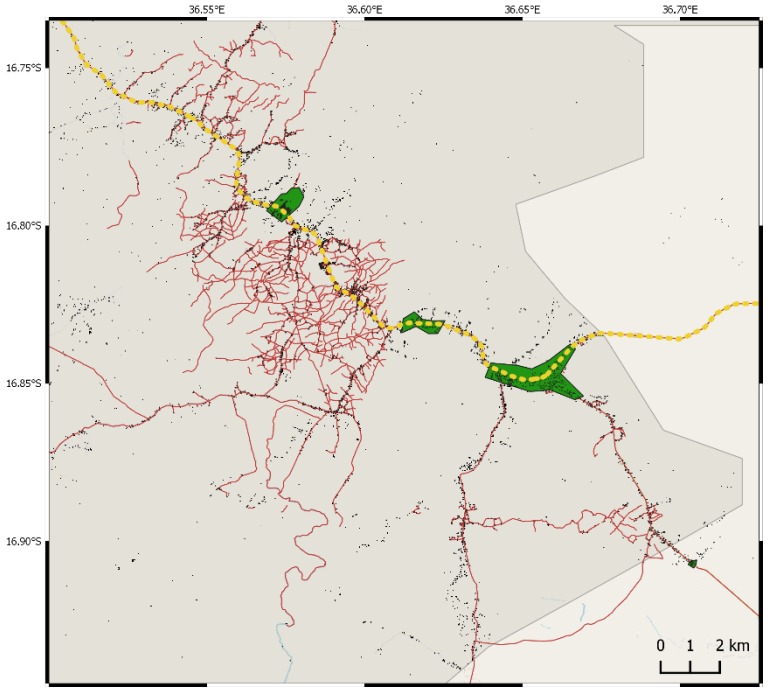

Figure 48. Potential areas with higher power per capita in Namanjavira

Figure 49 shows areas where a higher peak power per capita has been assigned. The legend colours are:

- Red: high population density areas;
- Purple: big villages;
- Blue: area situated near the main road.

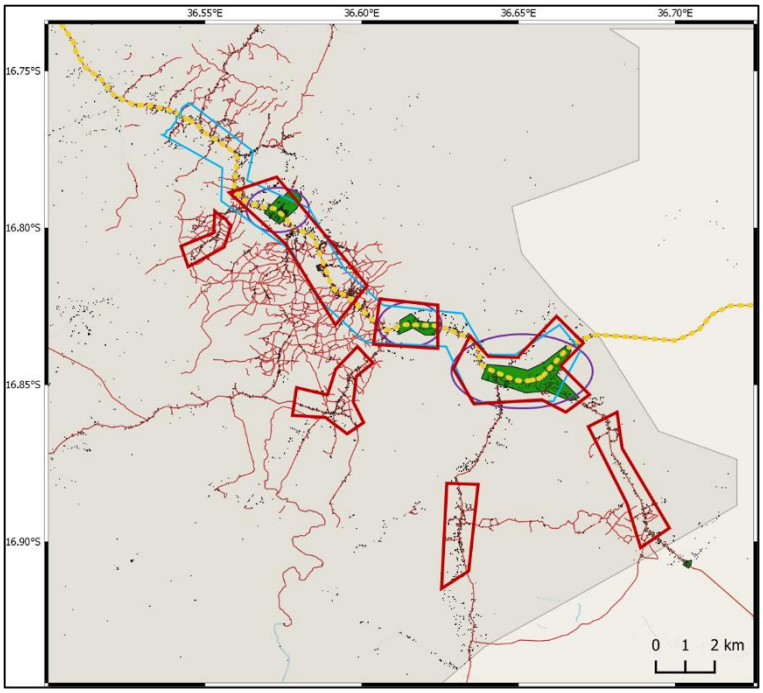

Figure 49. Areas with higher power per capita in Namanjavira

The weights (power per capita) given to the points are:

- 0.025: Power per capita according to [24]. Points outside the areas (i), (ii) and (iii) will remain with this weight.
- 0.05: weight given to points inside only one type of area (i), (ii) or (iii).
- 0.1: weight given to points inside two areas at the same time.
- 0.2: weight given to points inside the three areas (i), (ii) and (iii) at the same time.

As the population per point is 4, the power of the points is 0.1, 0.2, 0.4 and 0.8 kWh per point. Applying this criterion, the weights assigned to the points are presented in Figure 50.

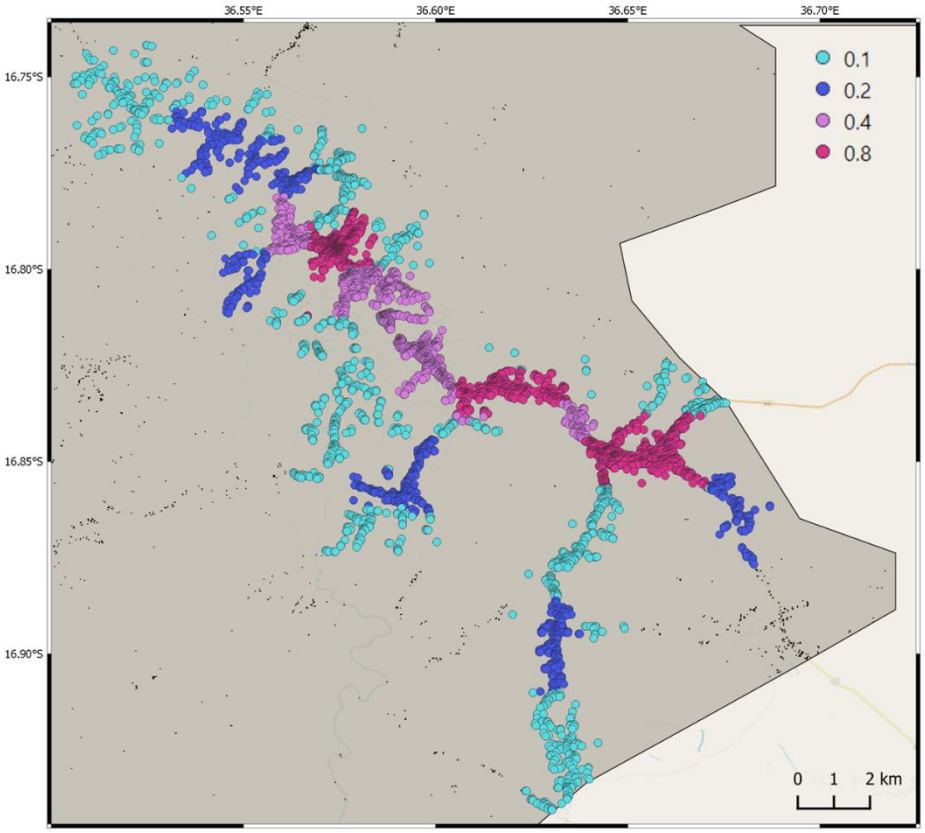

Figure 50. Points classified according to the new weights in Namanjavira

K-MEANS ALGORITHM

In Chapter 6 this algorithm and its different versions have been explained. To compare the versions, the properties evaluated are:

- The final number of clusters, k_{final}.
- The running time, t_{run}, in seconds.
- The maximum distance between load and substation inside the cluster, d_{max}, in metres, considering a distance constraint of 1000 metres.

After the weighting process, the peak power obtained is 1,860 kW. To be comparable, the normal and weighted versions of k-means should have the same peak power and number of input clusters.

The number of clusters' input is 37 clusters considering the peak demand is 1,860 kW and a small transformer of 50 kW of power as this is a rural area of a developing country:

$$k_{input} = \frac{1860}{50} \cong 37$$

The results for the different versions of the algorithm are presented in Table 17:

Table 17. Results of k-means in Namanjavira area

Version	Characteristics	Algorithm	
		Normal	Weighted
No loop	k_input	37	37
	k_final	-	-
	t_run (s)	3.76	230.96
	d_max (m)	1894.98	2611.13
Simple	k_input	37	500
	k_final	143	-
	t_run (s)	890.84	723.51
	d_max (m)	994.16	1,592.62
Complex	k_input	37	37
	k_final	84	98
	t_run (s)	6865.91	1426.71
	d_max (m)	998.68	990.43

The version with no loop is the fastest. The problem is that the number of clusters is a fixed input and the distance constraint is not met.

With the simple loop the distance constraint is met. However, the final number of clusters is not optimized, as it does not divide the big cluster and leave the ones that already meet the constraint, but it divides all the points again and again until de d_{max} is under 1000 m. For the weighted k-means the algorithm does not even converge to a comparable value, so the solution has been discarded.

Although the version with the complex loop is the one that takes more time to run, it is the most optimized version, as while meeting the distance constraint it returns the lower number of clusters.

In the no loop version, the normal k-means is much faster than the weighted, specifically 60 times. Whereas, in the complex loop version, the weighted k-means is faster than the normal one by 5 times.

To see the results, the output files are uploaded into the software QGIS. The centres of the clusters are represented with white dots, and the different clusters, meaning the populated points assigned to each specific substation, are represented with different colours.

Figure 51 shows the 37 clusters obtained with the normal k-means and no loop.

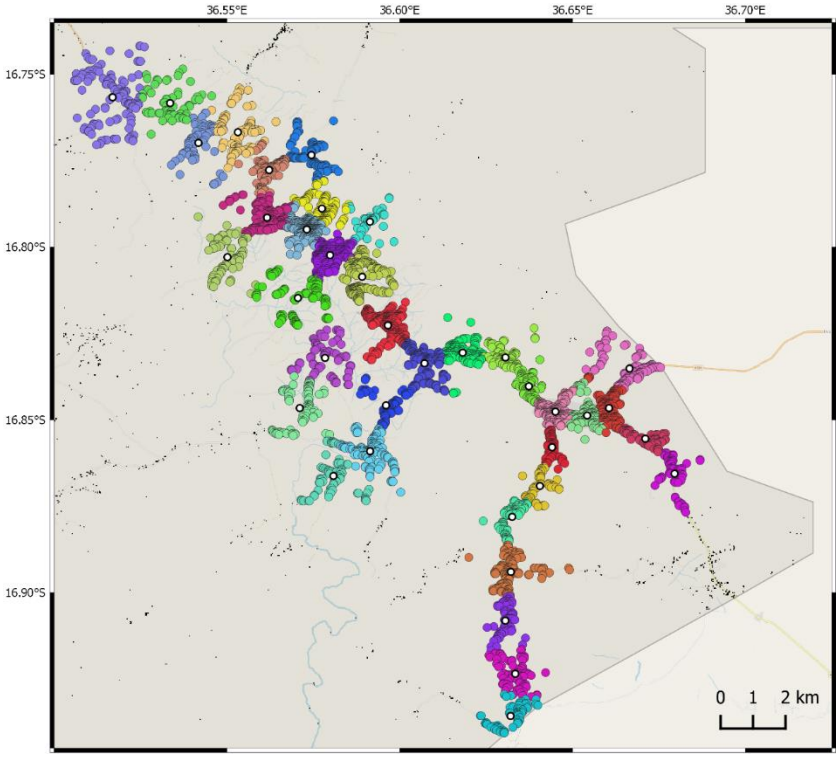

Figure 51. Clusters for normal k-means with no loop in Namanjavira

Figure 52 presents the 37 clusters obtained with the weighted k-means and no loop.

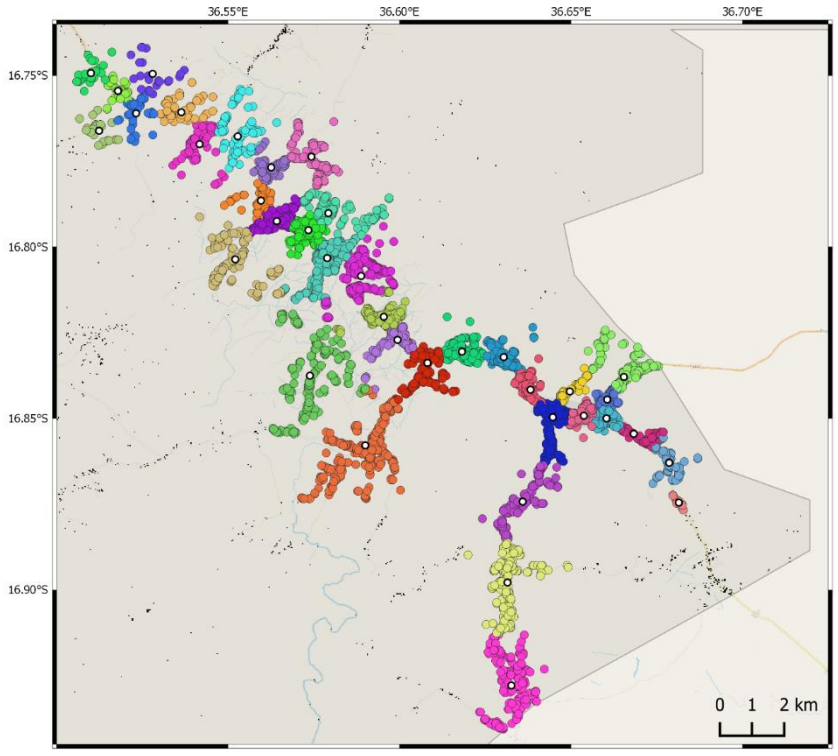

Figure 52. Clusters for weighted k-means with no loop in Namanjavira

As can be noticed, the solution differs from the one obtained with the normal k-means. This is due to the different weights of the points. More centres are assigned to the zones with higher weights, creating bigger clusters in areas with lower power.

In Figure 53, a comparison of the location of the centres with normal k-means and the weighted version is presented. The blue centres are from the normal version and the pink ones from the weighted k-means. Near the road (yellow dashed line) and the villages (green polygons) the number of pink dots is higher than the blue ones. The reason behind is that the size of the clusters with normal k-means is more homogeneous while the power of the clusters with weighted k-means is more homogeneous.

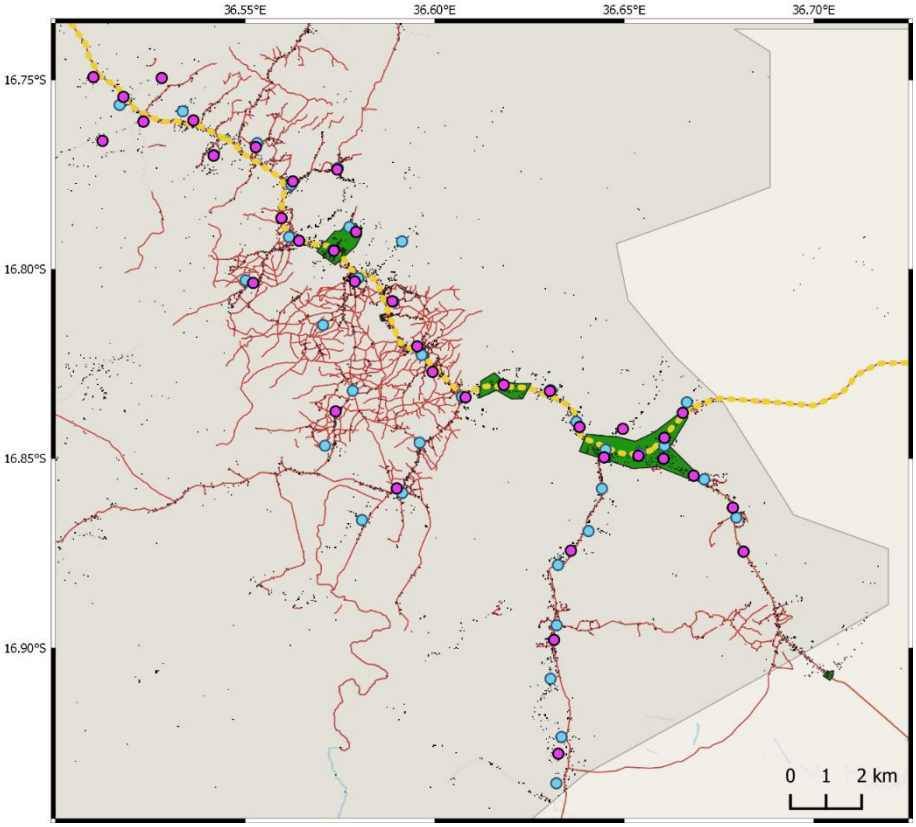

Figure 53. Normal vs. weighted k-means with no loop

Figure 54 presents the 143 clusters obtained with the normal k-means and simple loop.

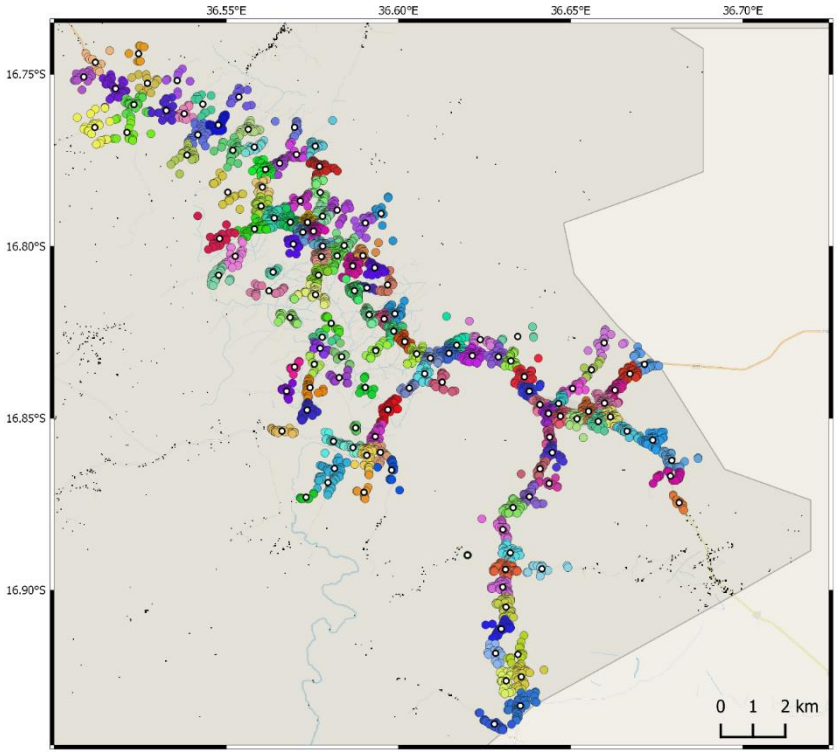

Figure 54. Cluster for normal k-means with simple loop in Namanjavira

As mentioned before, the weighted k-means with simple loop has been discarded due to lack of convergence in a reasonable time. However, it has been run with a fixed input to see which was the solution.

Figure 55 presents the solution for the weighted k-means and simple loop with a fixed number of clusters of k=500. The distance constraint was not met (d_{max} = 1,592.62 m). There are some very big clusters and other smalls, as the loop starts again and again from scratch when adding a new cluster. As can be seen, the centres are all along the main road, the areas with higher weights.

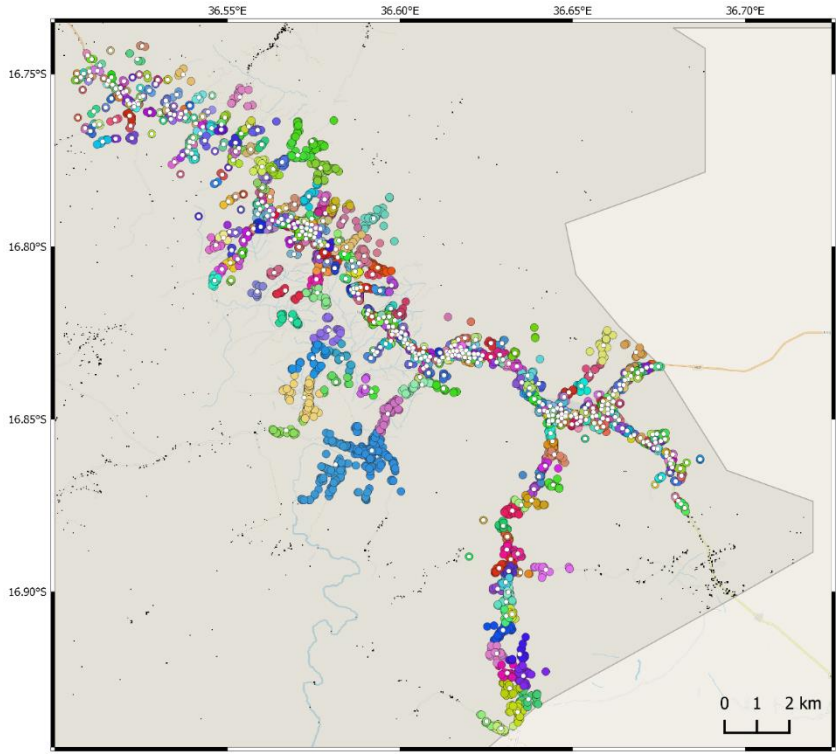

Figure 55. Clusters with k=500 and weighted k-means and simple loop in Namanjavira

Figure 56 presents the 84 clusters obtained with the normal k-means and complex loop.

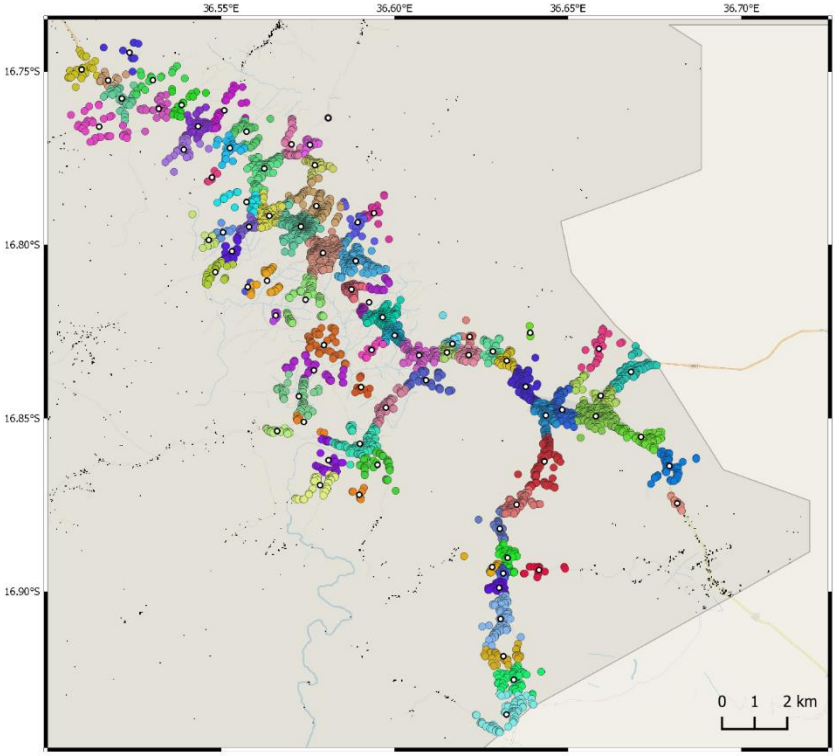

Figure 56. Clusters for normal k-means with complex loop in Namanjavira

Figure 57 presents the 98 clusters obtained with the weighted k-means and complex loop.

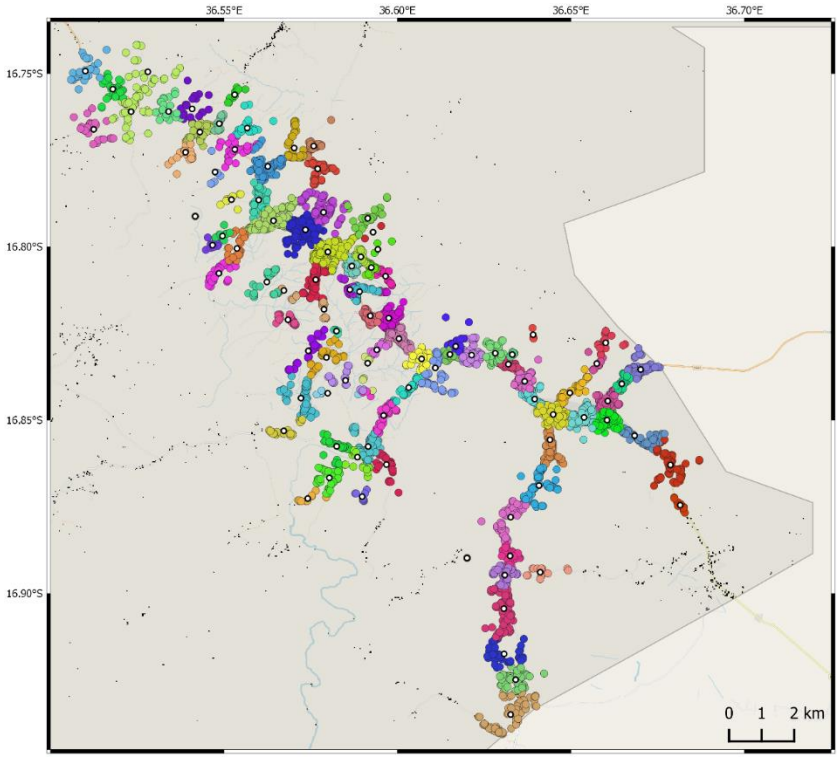

Figure 57. Cluster for weighted k-means with complex loop in Namanjavira

As with the version with no loop, the solutions obtained with the normal k-means and weighted k-means are different.

In Figure 58, a comparison of the location of the centres with normal k-means and the weighted version is presented. The blue centres are from the normal version and the red stars from the weighted k-means. Near the road (yellow dashed line) and the villages (green polygons) the number of red stars is higher than the blue dots. The blue dots are more homogeneously distributed as they do not consider the weights of the points. However, the difference, in this loop version, is less evident than before due to the higher number of clusters. The process of splitting the bigger clusters, in fact, leverages the two outputs.

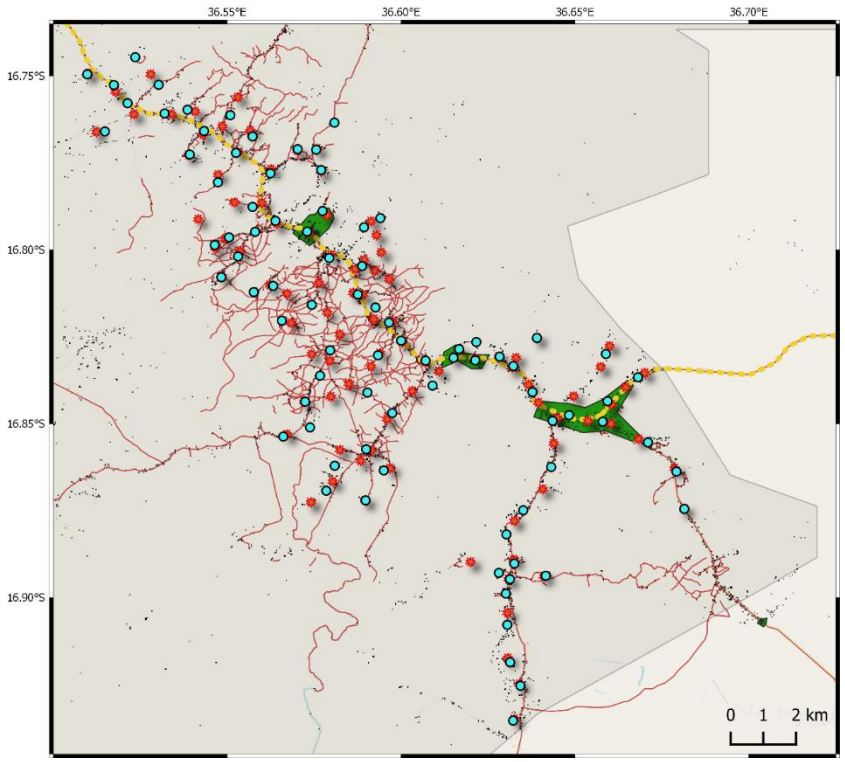

Figure 58. Normal vs. weighted k-means and complex loop

The detailed results of this algorithm are found in Annex A.

In Table 18, the average power, the standard deviation, the average size and the standard deviation of the size of the clusters obtained with the different versions of k-means are presented, where:

- NKM NL = Normal k-means and no loop
- WKM NL = Weighted k-means and no loop
- NKM CL = Normal k-means with complex loop
- WKM CL = Weighted k-means with complex loop

Making the hypothesis of round shape for the clusters, the maximum distance between the points and centres of the clusters has been taken as an approximation of the radius of the cluster in order to estimate the size of the clusters.

As mentioned before, the algorithm developed with k-means deals with the distance constraint but not with the power.

Table 18. Detailed results of k-means

ALGORITHM	NKM NL	WKM NL	NKM CL	WKM CL
Average power [kW]	50.46	50.46	22.23	19.05
St. deviation of the power	52.01	52.10	38.05	33.11
Average size [m]	1,219.66	1,269.06	682.84	626.94
St. deviation of the size	302.49	517.19	252.19	254.76

Figure 59 shows the probability density function of the power with the different version of k-means in the area of Namanjavira.

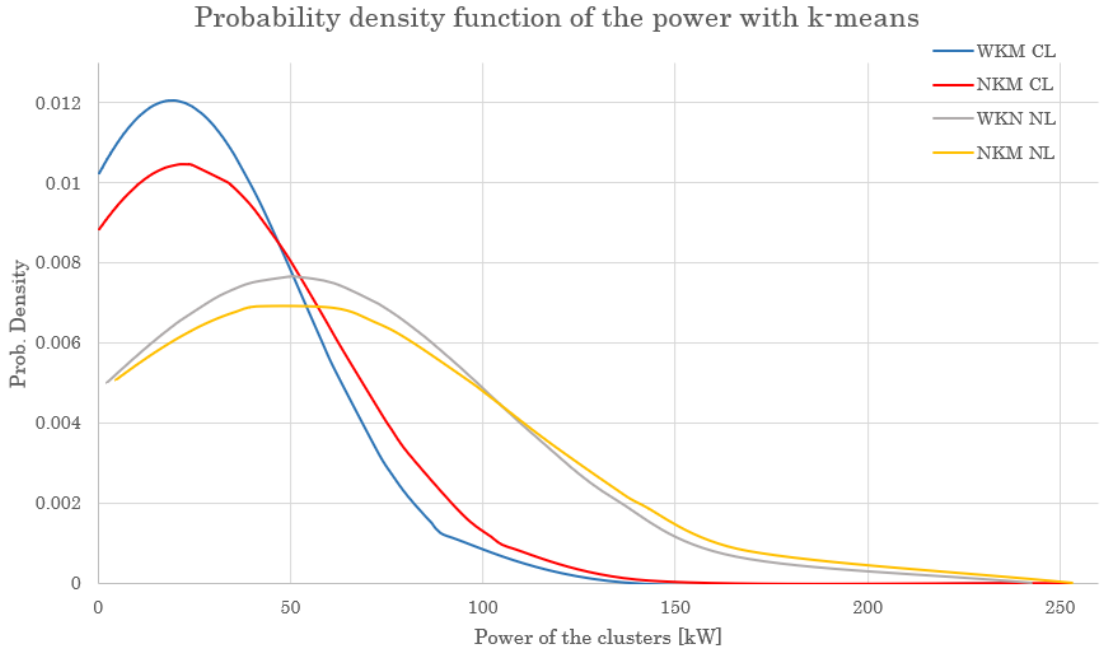

Figure 59. Probability density function of the power with k-means

The average power is lower with the version of the complex loop, as the number of clusters is higher.

The power is more homogeneous among the clusters in the version with complex loop than in the version with no loop, reflected in the lower values of the standard deviation.

In the case with no loop, as the distance constraint is not considered, the clusters are assigned with a fixed input and in the case of the weighted version they tend to be situated near the areas with higher weights. This results in small clusters in the areas with higher weights and bigger clusters in the areas of lower weights (see Figure 52). This means the higher the power per capita of the cluster the smaller the cluster is.

In the case of the complex loop, the initial size of the clusters is uneven with the weighted procedure. So, when it starts splitting the big ones, it results in lower power there and higher power in the others. However, in the normal version it splits all the cluster equally.

Comparing the weighted version with the normal one, the power is more homogeneous in the weighted version, reflected in the lower standard deviation, as the power per capita is considered in the algorithm.

In the complex loop version (blue and red lines), after dividing the clusters in order to meet the distance constraint, some clusters end having only one point, with a power between 0.1 and 0.8 kW/point. That is the reason the curve starts so close to the zero.

Figure 60 shows the probability density function of the size of the clusters with the different version of k-means in the area of Namanjavira.

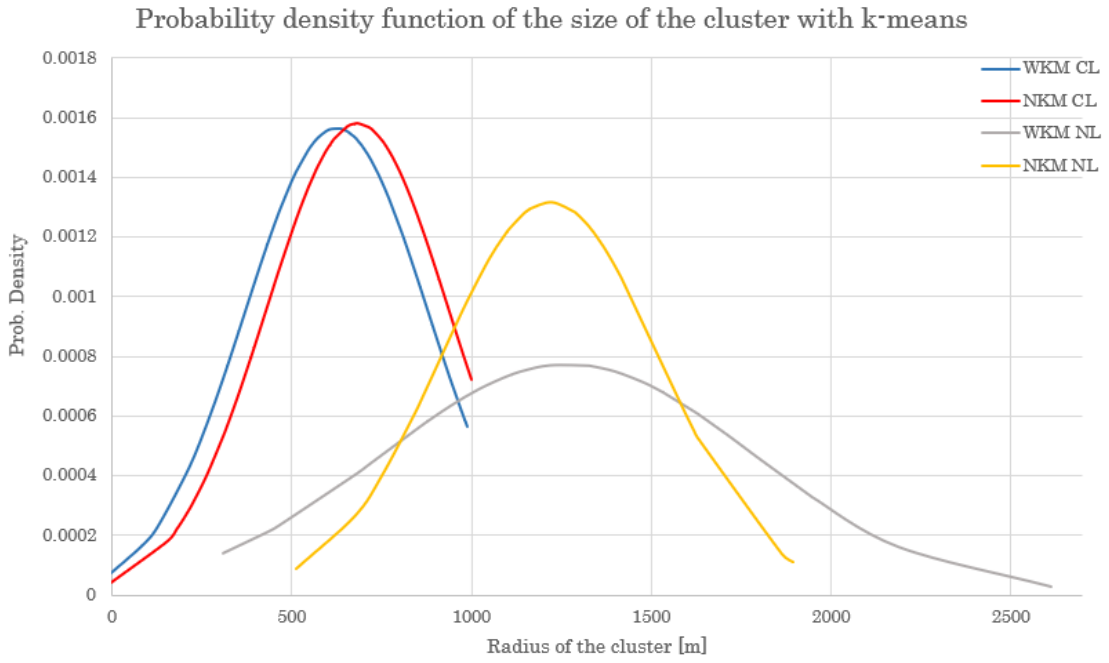

Figure 60. Probability density function of the size of the clusters with k-means

As can be seen, the distance constraint is not met with the no loop version, as the average size of the clusters is higher than the threshold imposed. With the complex loop versions, the maximum is in 1000 m, the distance constraint.

The average size is more homogeneous with the normal k-means and no loop than with the weighted k-means, as it divides the area in the input number of clusters not taking into account the power per capita of the points but only geographic position. This is reflected in a lower standard deviation of the size for the normal k-means. With the version of complex loop happens the same.

The reason behind having clusters with a radius of 0 metres is that after dividing the clusters in order to meet the distance constraint, some clusters only have one point, so the distance between the point and the centre is 0 as it is itself.

The k-means algorithm fails to be realistic. It does not consider the structure and construction of the energy system. This could lead to different problems, as assignment problems as the one shown in Figure 61, or the fact that the substations (centres) are sat in random places, not evaluating the characteristics of the terrain.

In Figure 61, which is a zoom of the total case study area of Namanjavira, some limitations of the algorithm are shown. The circled points belong to a cluster because the distance is lower to that centre than to another one. On another note, it is known the low voltage lines usually follows roads and streets as it is easier to construct them this way. The circled points are connected through a road to the other clusters, and not to their cluster. However, they belong to the cluster of the closer centre instead of following the road. This is a limit of the algorithm, requiring a final check from a human operator.

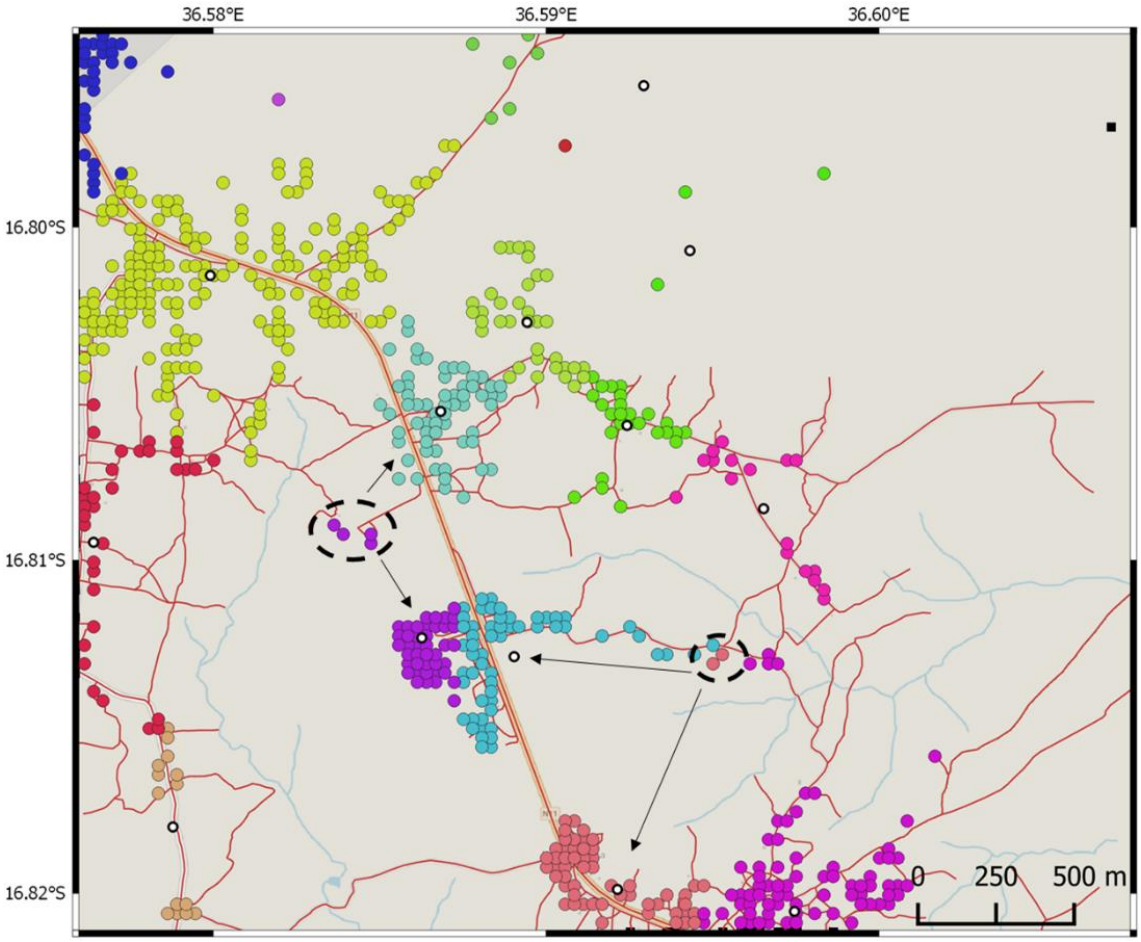

Figure 61. Connection problems that could arise with the k-means algorithm

LUKES ALGORITHM

In Chapter 6 this algorithm has been explained. To check the performance of the algorithm, the features evaluated are:

- The maximum power per cluster imposed.
- The final number of clusters, k.
- The running time, t_{run}.
- The maximum distance between load and substation inside the cluster, d_{max}.

As explained in section 5.2, LUKES algorithm works on a graph-type data. The input for the algorithm is the MST of the graph created from the points-population data. The MST obtained with Kruskal algorithm, connecting all the populated points is shown in Figure 62:

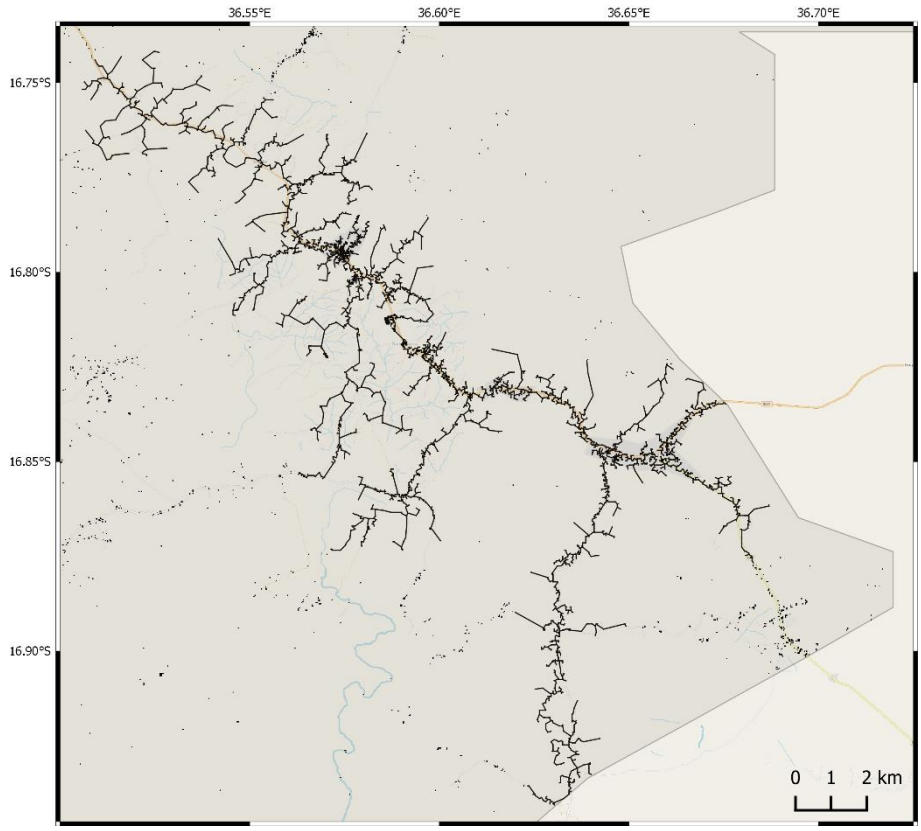

Figure 62. MST of the area of Namanjavira

The power attribute of the function *lukes_partitioning* has to be an integer value [52], so in order to work, a minimum value of 1 kW per point is needed. In the algorithm, the maximum size "W" represents the maximum power a cluster and hence a secondary substation could supply. The power is found as an attribute of the nodes of the graph obtained from the MST.

However, if the value of 0.025 kW/capita is taken, this would make a power per point of 0.1 kW/point. Not an integer. To solve this, everything has been multiplied by 10 in the algorithm, the power per capita and the power constraint, making the power per capita 0.25 and the power constraint 500 kW.

To evaluate the results, the code of *lukes_partitioning* has been run with the weighted data input. The modified data has a power per capita of 0.25, 0.5, 1 and 2 kW/person, leading to 1, 2, 4 or 8 kW per point.

The results are presented in Table 19:

Table 19. Results of LUKES algorithm in the area of Namanjavira

Limit power, W [kW]	Number of clusters, k	Running time, t_run [s]	Max radio, d_max [m]
500	50	87,060	2,400

To see the results, the output files are uploaded into the software QGIS. The different clusters are represented with different colours.

Figure 63 presents the 50 clusters obtained with LUKES W=500 and the weighted input data.

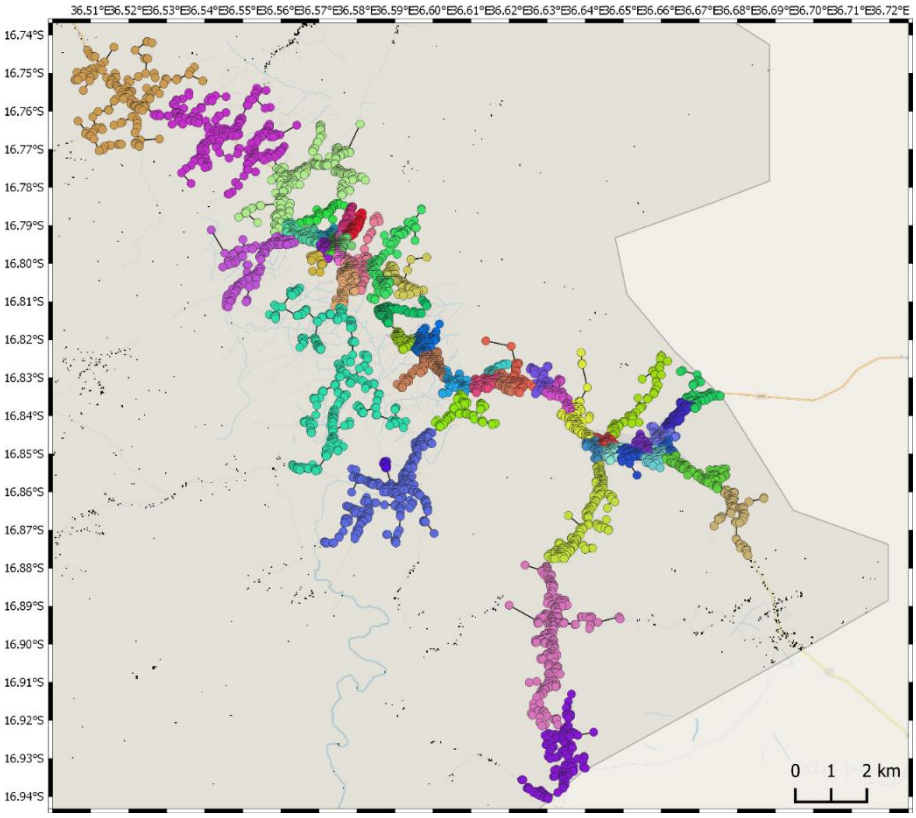

Figure 63. Clusters with LUKES, W=500 and original data in Namanjavira

In Figure 64, the areas with higher weights are around the main road (yellow dashed line) and the villages (dark red polygons). As can be noticed, in the areas where the weight is higher the clusters are smaller. This is because the power constraint is met with a lower number of points than in areas with lower power.

However, a disadvantage of the LUKES algorithm is that, although the power is controlled, control on the distance is lost. Some of the cluster are very big, because they keep growing until the power constraint is met.

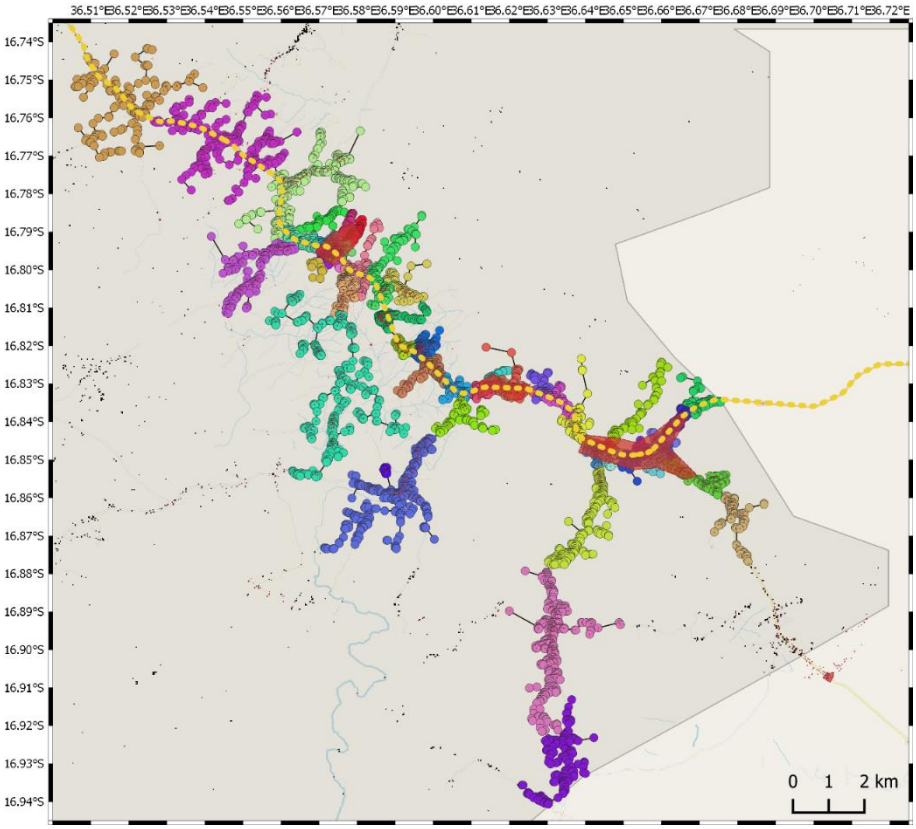

Figure 64. Areas with higher weights, smaller clusters

The detailed results of this algorithm are found in Annex A.

In Table 20, the average power and its standard deviation of the clusters obtained with LUKES are presented, compared to the values obtained with the weighted k-means with complex loop.

Table 20. Detailed results of LUKES and weighted k-means with complex loop

ALGORITHM	**LUKES W=500**	**WKM CL**
Average power [kW]	37.34	19.05
St. deviation of the power	13.24	33.11

Figure 65 presents the probability density function of the power obtained with LUKES algorithm in the area of Namanjavira. As can be seen, there is a maximum power of 50 kW due to the power constraint imposed.

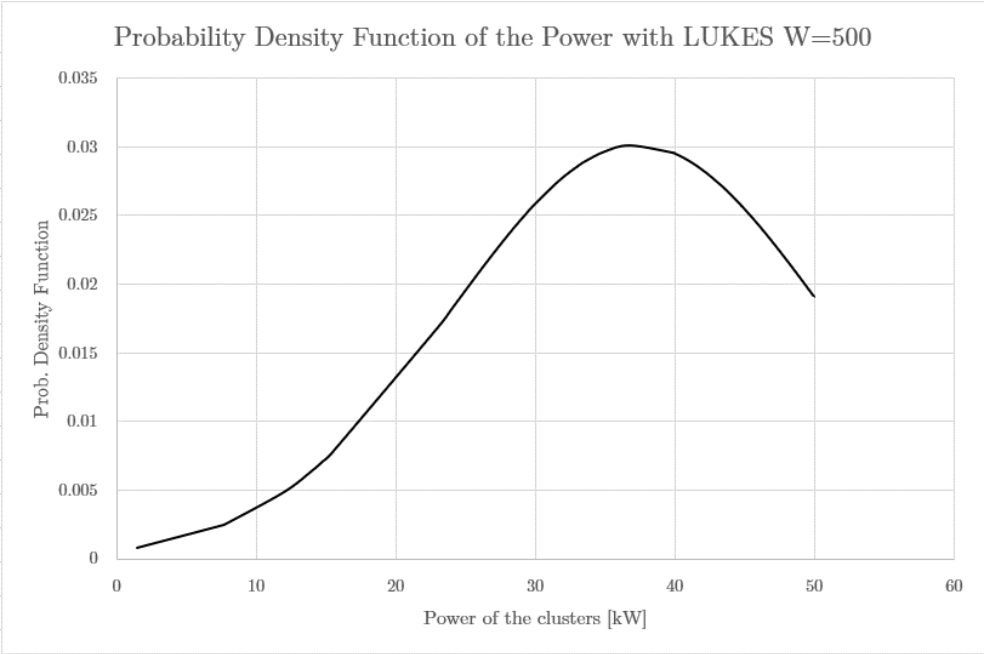

Figure 65. PDF for the Power with LUKES W=500 in Namanjavira

To better understand how LUKES deals with the power compared to the k-means, in Figure 66 the probability density function of the power with LUKES and with the weighted k-means with complex loop is presented.

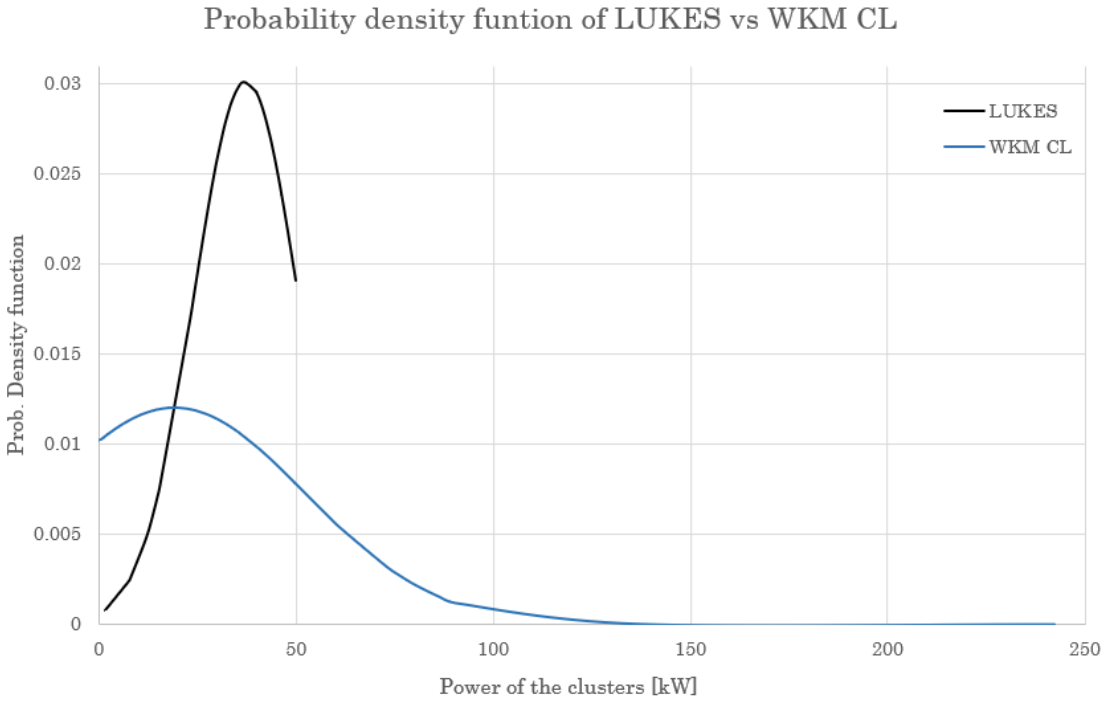

Figure 66. Prob. density function of LUKES vs WKM CL

The average power of LUKES is higher due to lower number of clusters. Also, the power is more homogenously distributed, reflected in a lower value of the standard deviation. The problem of LUKES is that it does not deal with the distance between loads and substations.

COMPARISON OF THE RESULTS OF THE ALGORITHMS

To better compare the results, a resume of the data obtained with the k-means with complex loop and LUKES algorithms in terms of performance are presented in Table 21.

Table 21. Performance comparison of the algorithms for Namanjavira

ID	Algorithm characteristics	Normal k-means	Weighted k-means	LUKES W=500
1	Number of clusters input	37	37	-
2	Final number of clusters	84	98	50
3	Running time [s]	6,865.9	1,426.71	87,060
4	Maximum distance between load and substation [m]	998.68	990.43	2,400
5	Average power [kW]	22.23	19.05	37.34

1. LUKES does not need the input of number of clusters while k-means needs it.
2. The final number of clusters is higher with the k-means as it looks to meet the distance constraint and the houses are scattered. This is supported with the low values of average power for the k-means.
3. K-means is much faster than LUKES, having LUKES a running time between 13 and 61 times bigger. The reason is the time complexity of LUKES is proportional to the number of nodes and this is very high (4,729).
4. The maximum distance with k-means is controlled, however, with LUKES the distance constraint is not implemented and hence not met.

In conclusion, related to k-means, the power is more homogeneously distributed among the clusters in the weighted version, while the size is more homogeneous in the normal version. Meeting the distance constraint in the complex loop both versions, it is better to have the power better distributed, so the weighted k-means with complex loop is the best version.

Related to LUKES, the power is even better distributed compared to the weighted k-means with complex loo. However, the size of the clusters is totally out of control and there is a need to introduce a distance constraint in this algorithm.

To compare the solution with reality, the 5 DSO indicators from Chapter 2 have been selected. In the following table this information is shown and compare with the Mozambican values.

The values of Mozambique related to literature are more representative of urban areas, as data related to these zones is more accessible.

Table 22. DSO Indicators to compare the algorithms and reality in Namanjavira

ID	Indicators	Normal k-means	Weighted k-means	LUKES W=500	Mozambique (Literature)
1	LV circuit length per LV consumer	1,000	1,000	2,400	22 m
2	Number of LV consumer per MV/LV substation	225	193	379	110
3	MV/LV substation capacity per LV consumer	0.22	0.26	0.13	1.98
4	Number of MV supply points per HV/MV substation	84	98	50	157
5	Typical transformation capacity of MV/LV secondary substations in rural areas	50 kW	50 kW	50 kW	176 kW*

*Value calculated with a power factor=0.8

The conclusions extracted are:

1. The LV circuit length is higher, which makes sense due to the lower population density in rural areas.
2. The number of LV consumers per substation is also higher for the same reason of population density along with the lower power demand in rural areas.
3. The capacity of the MV supply point is the typical transformation capacity of the substations in rural areas [ID 5]. This value should be lower in rural areas, as usually smaller transformers are used, and the power demand is lower too.
4. The number of substations is usually lower in rural areas due to lower power demand and lower population density. Also, the value obtained in the case study is lower because in this study case only the MV/LV substations have been considered as MV supply points, neglecting the MV consumers.
5. The value of the capacity of the transformers is the one used for the case study. The capacity of the transformers is lower for rural areas due to lower power demand, lower simultaneity factor and lower population density.

8.2 OMEREQUE

The analysis in Omereque is the same as the one done in Namanjavira, with the same objectives, comparison of the algorithm's performance and indicators.

WEIGHTING PROCESS

For the weighted k-means different weights are assigned to the points. As mentioned before in section 6.3, different features have been considered.

In particular, for this area, more weight has been given to:

i. Areas where, according to data collected from OpenStreetMaps, settlements already exist, so population density is higher. These areas are the big yellow polygons. Some of the small ones are water reserves, so they are not considered.
ii. Areas situated near the road. The main road is represented with a yellow line and secondary roads with a grey line.

The area under study is presented in Figure 67:

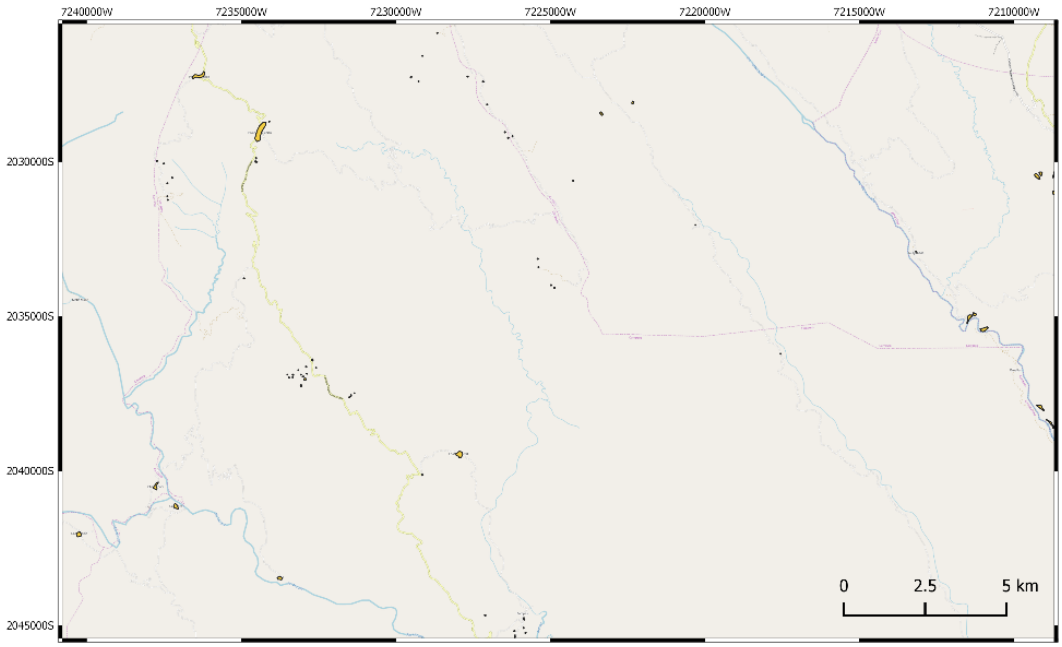

Figure 67. Features considered to give different weights to the points of the input file in Omereque, Bolivia

Inside the area, the areas highlighted are the chosen areas where a higher peak power per capita has been assigned. The red dots are the houses. The legend colours are:

- Red: areas around established settlements.
- Green: area situated near the road.

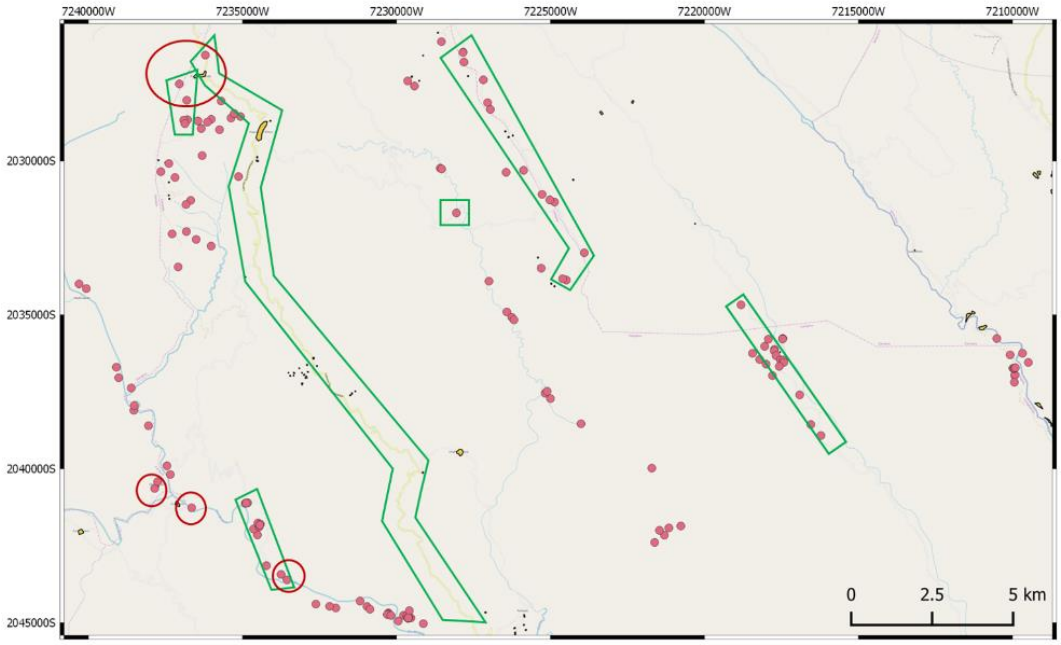

Figure 68. Marked areas according to the features considered to give different weights in Omereque

The weights (power per capita) given to the points are:

- 0.2: power per capita for the points outside the areas (i) and (ii).
- 0.4: weight given to points inside only one type of area (i) or (ii). It is the power per capita estimated by the *Politecnico* study.
- 0.8: weight given to points inside the two areas at the same time.

As the population per point is 5, the power of the points is 1, 2 and 4 kWh per point. Applying this criterion, the weights assigned to the points are presented in Figure 69:

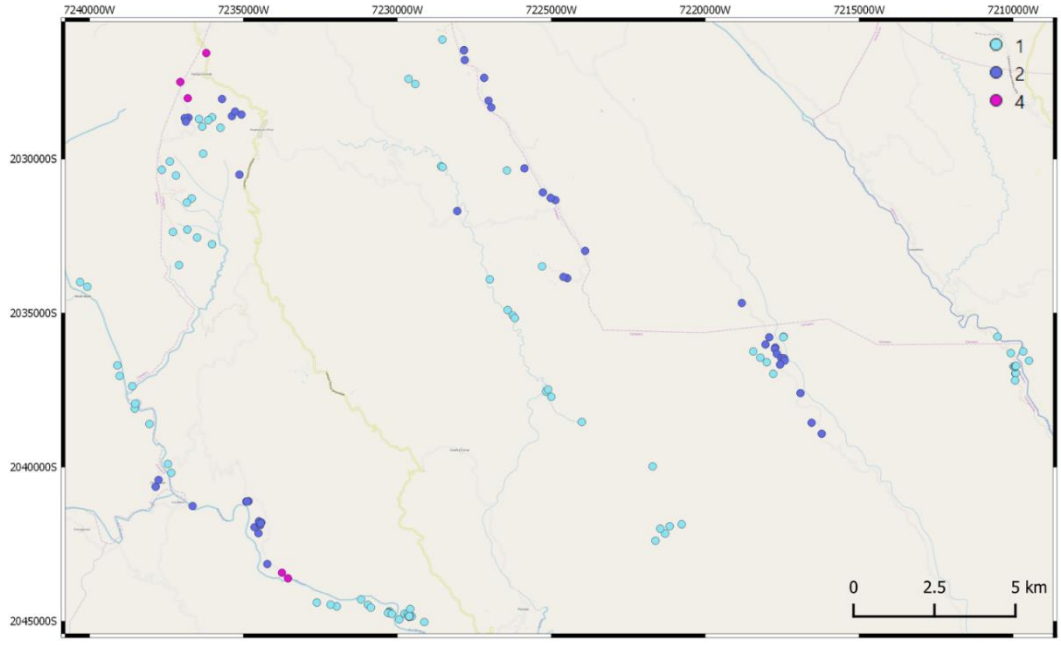

Figure 69. Points classified according to the new weights in Omereque

K-MEANS ALGORITHM

In chapter 6 this algorithm and its different versions have been explained. To compare the versions, the properties evaluated are:

- If the number of initial clusters, k_{input}, is fixed.
- The final number of clusters, k_{final}.
- The running time, t_{run}, in seconds.
- The maximum distance between load and substation inside the cluster, d_{max}, in metres, considering a distance constraint of 600 metres.

After the weighting process, the peak power obtained is 203 kW. To be comparable, the normal and weighted versions of k-means should have the same peak power and number of input clusters.

The input is 10 clusters considering the peak demand is 1,860 kW and a small transformer of 20 kW of power as this is a rural area of a developing country:

$$k_{input} = \frac{203}{20} \cong 10$$

The results for the different versions of the algorithm are presented in Table 23:

Table 23. Results of k-means for the area of Omereque, Bolivia

Version	Characteristics	Algorithm	
		Normal	Weighted
No loop	k_input	10	10
	k_final	-	-
	t_run (s)	0.20	0.64
	d_max (m)	3,307.43	4,450.09
Simple	k_input	10	10
	k_final	51	96
	t_run (s)	21.21	111.06
	d_max (m)	530.58	485.5
Complex	k_input	10	10
	k_final	44	49
	t_run (s)	29.95	5.34
	d_max (m)	598.80	599.92

The running times are lower than in Namanjavira, because here there are only 139 points, while in the other case study were 4729 points.

As before, the version with no loop is the fastest. The problem is the number of clusters is a fixed input and the distance constraint is not met.

With the simple loop the distance constraint is met. However, the final number of clusters is not optimized, being almost the double as the solution with the complex

loop, as it does not focus on the clusters that do not meet the constraint, but it divides all the points again and again until de d_{max} is under 600 m.

With the complex loop, the weighted version is faster than the normal one. However, with the simple loop is faster the normal k-means.

In the no loop version, the normal k-means is faster than the weighted, specifically 3 times. Whereas, in the complex loop version, the weighted k-means is faster than the normal by 5.5 times.

Compared to Namanjavira, the final number of clusters is higher in terms of points per clusters. This is due to the nature of the data in Omereque, where the points are very scattered.

To see the results, the output files are uploaded into the software QGIS. The centres of the clusters are represented with white dots, and the different clusters, meaning the areas supplied by a secondary substation, with different colours.

Figure 70 shows the 10 clusters obtained with the normal k-means and no loop.

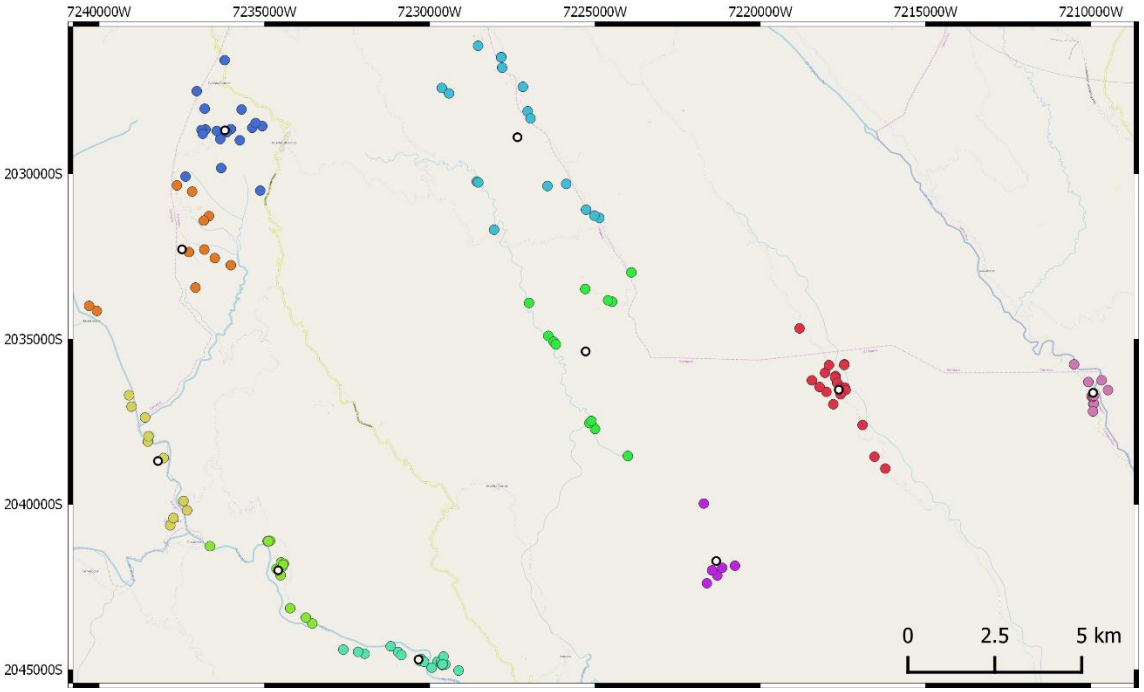

Figure 70. Cluster for normal k-means with no loop in Omereque

Figure 71 presents the 10 clusters obtained with the weighted k-means and no loop.

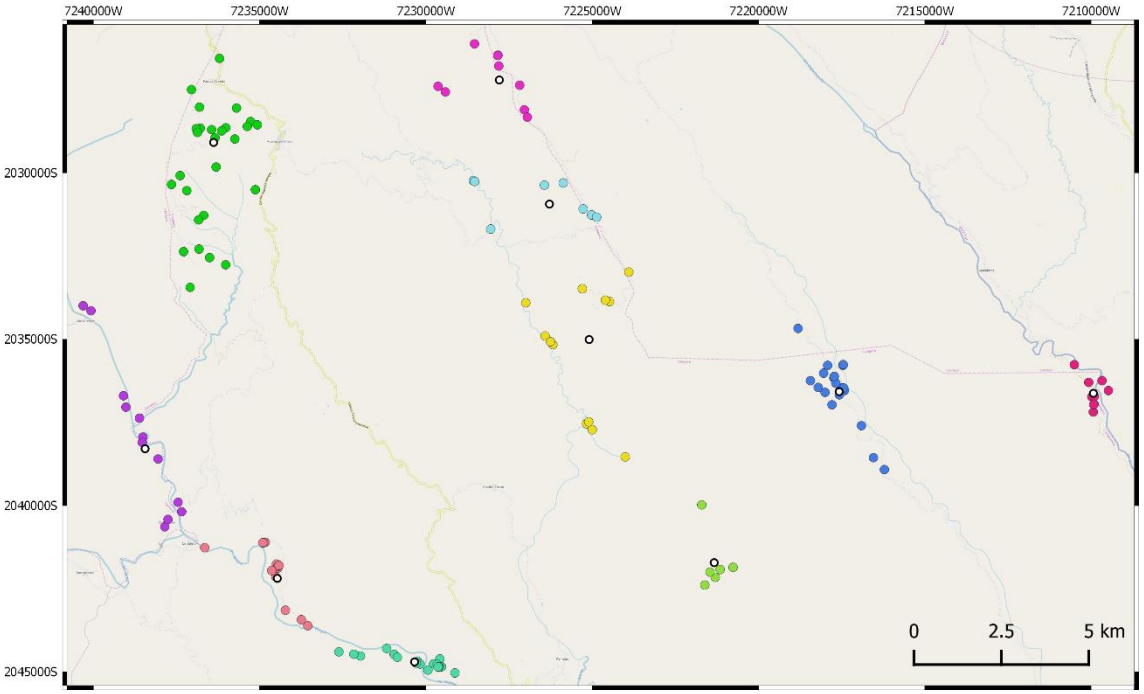

Figure 71. Clusters for weighted k-means with no loop in Omereque

As can be noticed, the solution differs from the one obtained with the normal k-means. This is due to the different weights of the points. More centres are assigned to the zones with higher weights, creating bigger clusters in areas with lower power.

In Figure 72, a comparison of the location of the centres with normal k-means and the weighted version is presented. The white dots are the centres from the normal version and the red stars the centres from the weighted k-means. The blue and purple dots are the houses with different weights:

- Light blue: 1
- Dark blue: 2
- Purple: 4

As can be noticed, the red stars tend to be closer to the more weighted area than the white dots.

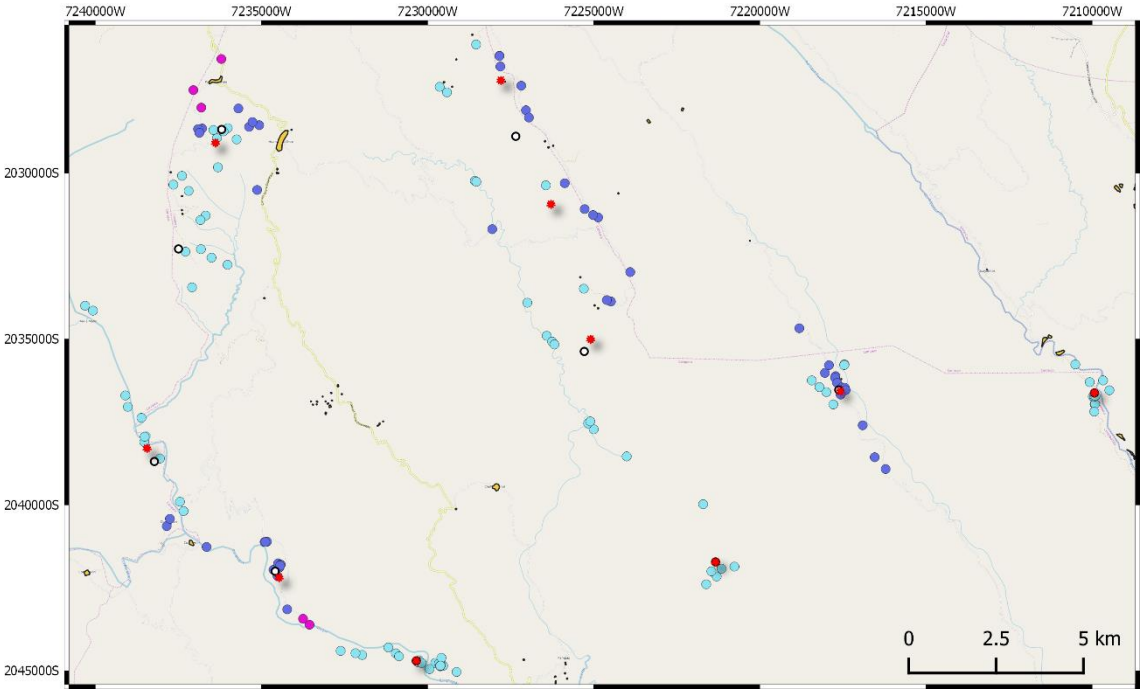

Figure 72. Normal vs. weighted k-means with no loop in Omereque

Figure 73 presents the 51 clusters obtained with the normal k-means and simple loop.

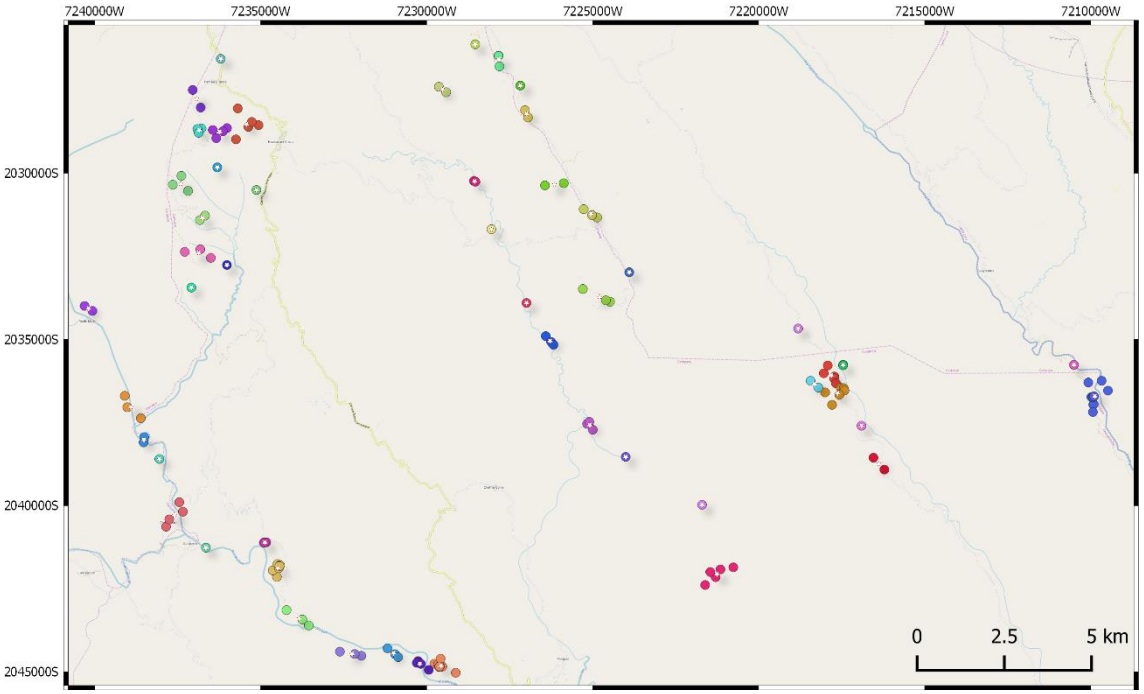

Figure 73. Cluster for normal k-means with simple loop in Omereque

Figure 74 presents the 96 clusters obtained with the weighted k-means and simple loop.

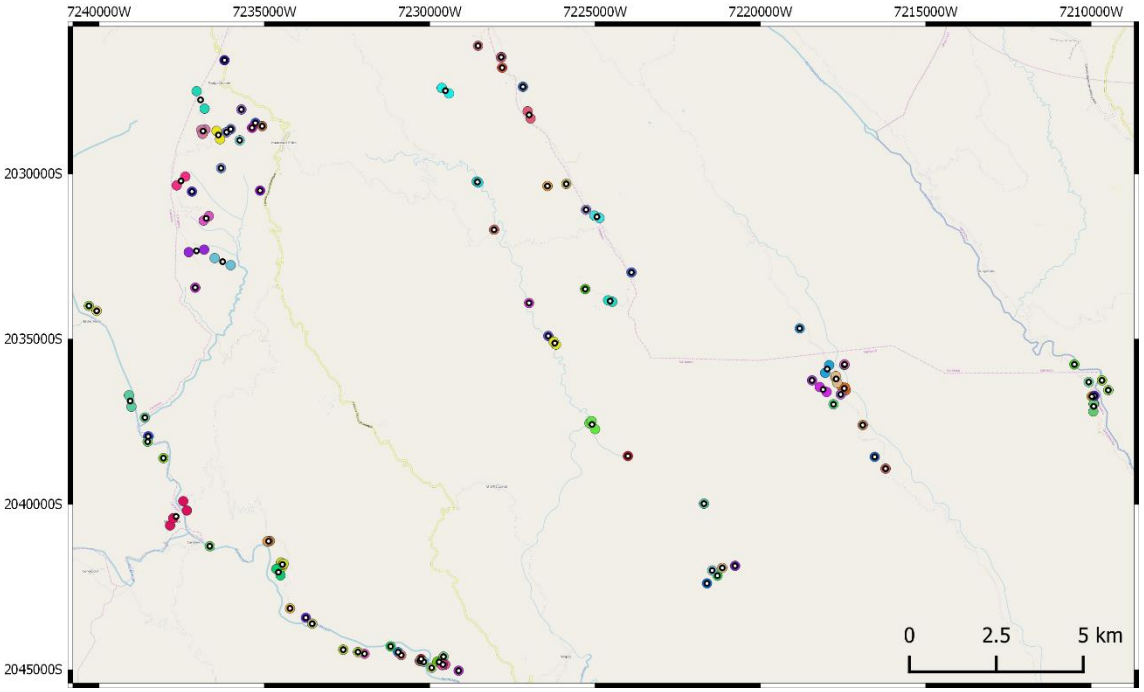

Figure 74. Cluster for weighted k-means with simple loop in Omereque

As can be noticed, the centres from the weighted version tend to be closer to the more weighted area than the centres of the normal k-means.

The number of centres is so high that the number of points per cluster is very low.

Figure 75 shows the 44 clusters obtained with the normal k-means and complex loop.

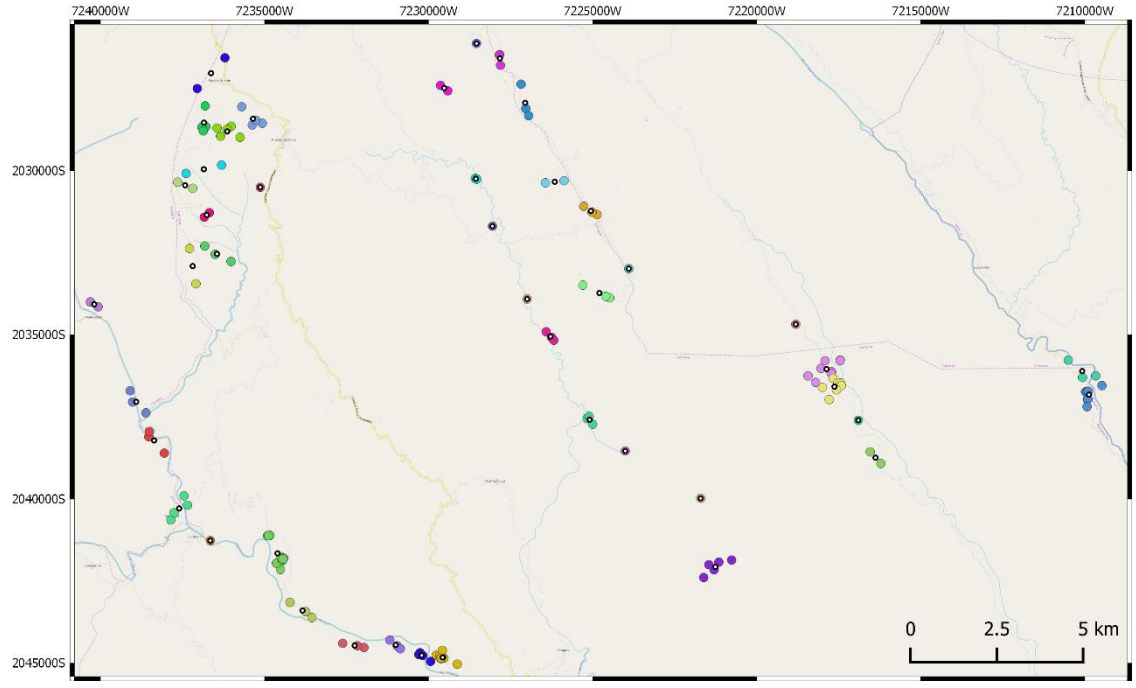

Figure 75. Clusters for normal k-means with complex loop in Omereque

Figure 76 shows the 49 clusters obtained with the weighted k-means and complex loop.

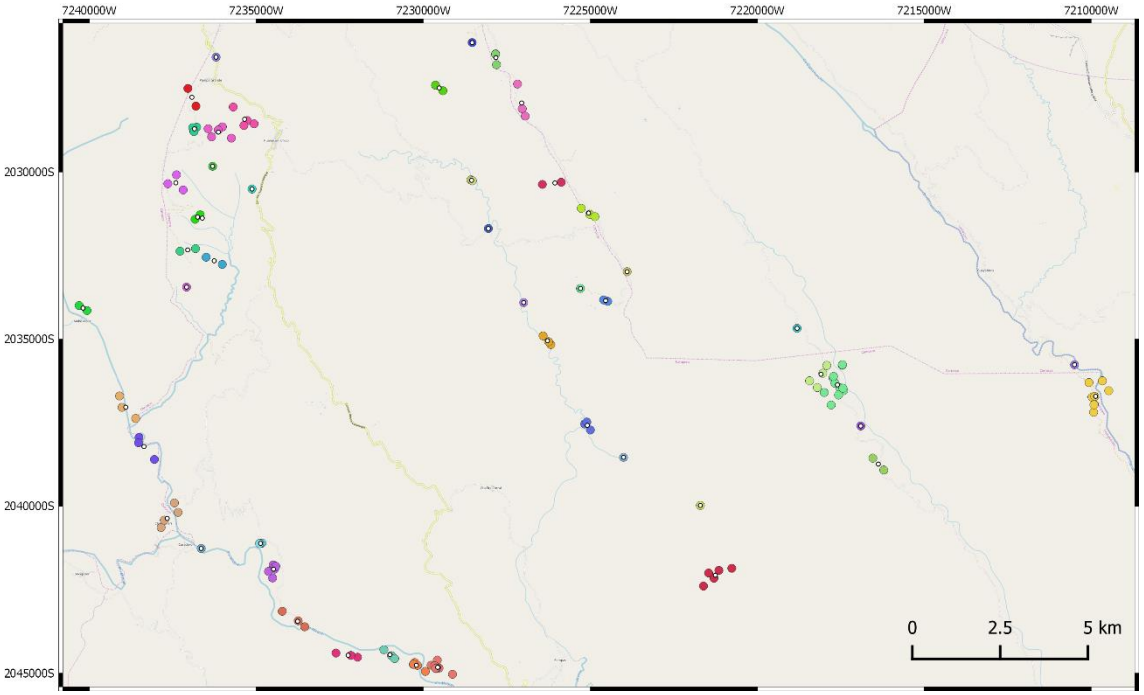

Figure 76. Cluster for weighted k-means with complex loop in Omereque

As with the version with no loop, the solutions obtained with the normal k-means and weighted k-means are different.

In Figure 77, a comparison of the location of the centres with normal k-means and the weighted version is presented. The black dots are from the normal version and the red stars from the weighted k-means. The blue and purple dots are the houses with different weights:

- Light blue: 1
- Dark blue: 2
- Purple: 4

As can be noticed, the red stars tend to be closer to the more weighted area than the black dots.

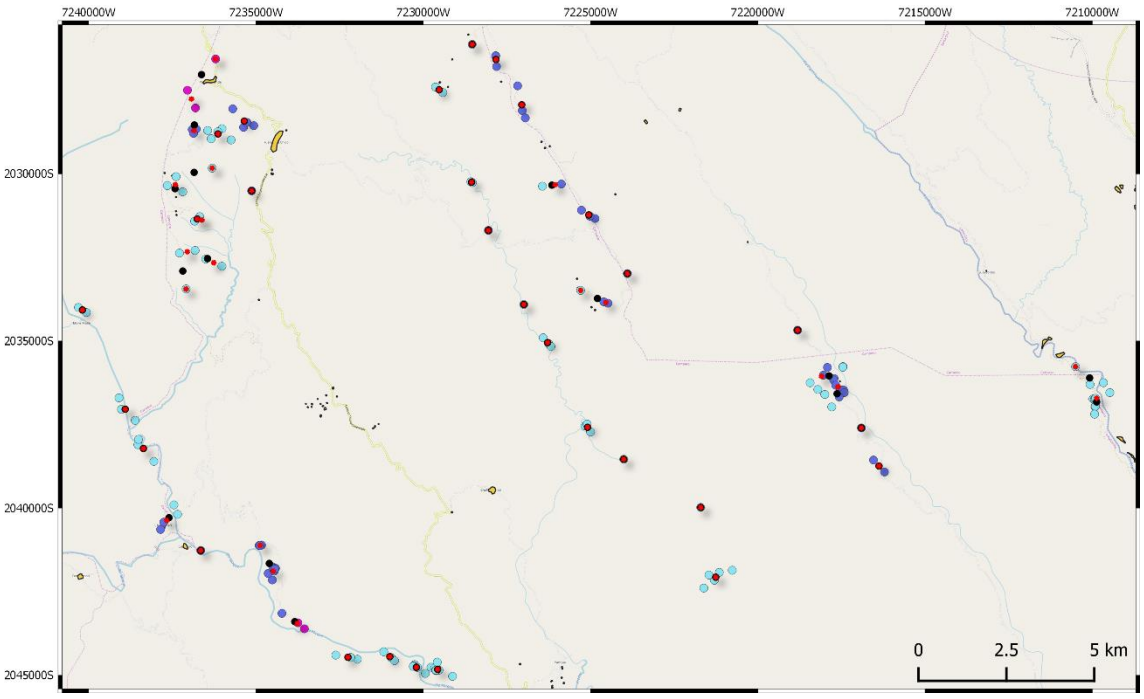

Figure 77. Comparison of the centre's location between normal and weighted k-means with complex loop

The detailed results of this algorithm are found in Annex B.

In Table 24, the average power, the standard deviation, the average size and the standard deviation of the size of the clusters obtained with the different versions of k-means are presented, where:

- NKM NL = Normal k-means and no loop
- WKM NL = Weighted k-means and no loop
- NKM CL = Normal k-means with complex loop
- WKM CL = Weighted k-means with complex loop

Making the hypothesis of round shape for the clusters, the maximum distance between the points and centres of the clusters has been taken as an approximation of the radius of the cluster in order to estimate the size of the clusters.

As mentioned before, the algorithm developed with k-means deals with the distance constraint but not with the power.

Table 24. Detailed results of k-means

ALGORITHM	NKM NL	WKM NL	NKM CL	WKM CL
Average power [kW]	20.30	20.30	4.61	4.14
St. deviation of the power	10.67	11.91	3.34	3.54
Average size [m]	2,567.36	2,586.33	286.59	202.45
St. deviation of the size	635.33	1,121.78	211.33	195.23

Figure 78 presents the probability density function of the power obtained with the k-means algorithm in the area of Omereque.

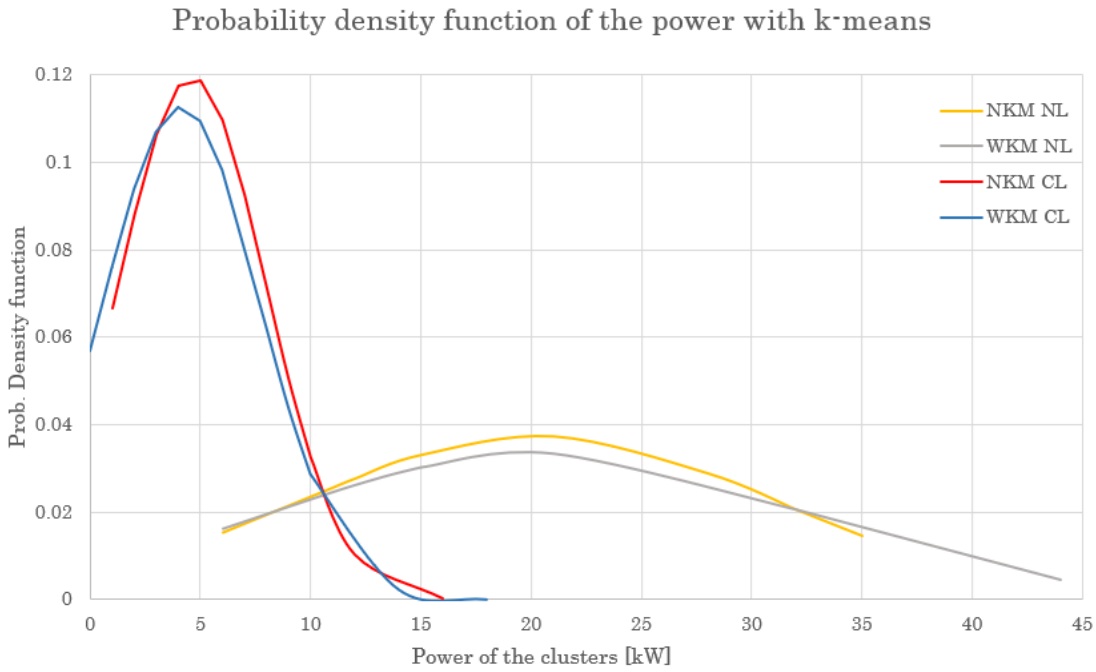

Figure 78. Prob. density function of the power with k-means

As in Namanjavira, the average power is lower with the version of the complex loop, as the number of clusters is higher. The maximum power with the complex loop is under 20 kW, the hypothetical value chosen for the power of the substations, while in the no loop version, the maximum power is almost the double. Although there is no power constraint in this algorithm, the power is controlled indirectly with the size of the clusters: the smaller the cluster, usually the lower the power of it, because less points are inside due to the scattered nature of this data.

The power is more homogeneous among the clusters in the version with complex loop than in the version with no loop, reflected in the lower values of the standard deviation and Figure 78.

In contrast to the case of Namanjavira, in the case of the complex loop, the power is more homogeneous in the normal version than in the weighted. This is due to the scattered points and that less points have higher weight (power per capita), so the solutions are similar.

In the weighted complex loop version (blue line), after dividing the clusters in order to meet the distance constraint, some clusters end having only one point, with a power between 1 and 4 kW/point. That is the reason the curve starts so close to the zero.

Figure 79 presents the probability density function of the size of the clusters obtained with the k-means algorithm in the area of Omereque.

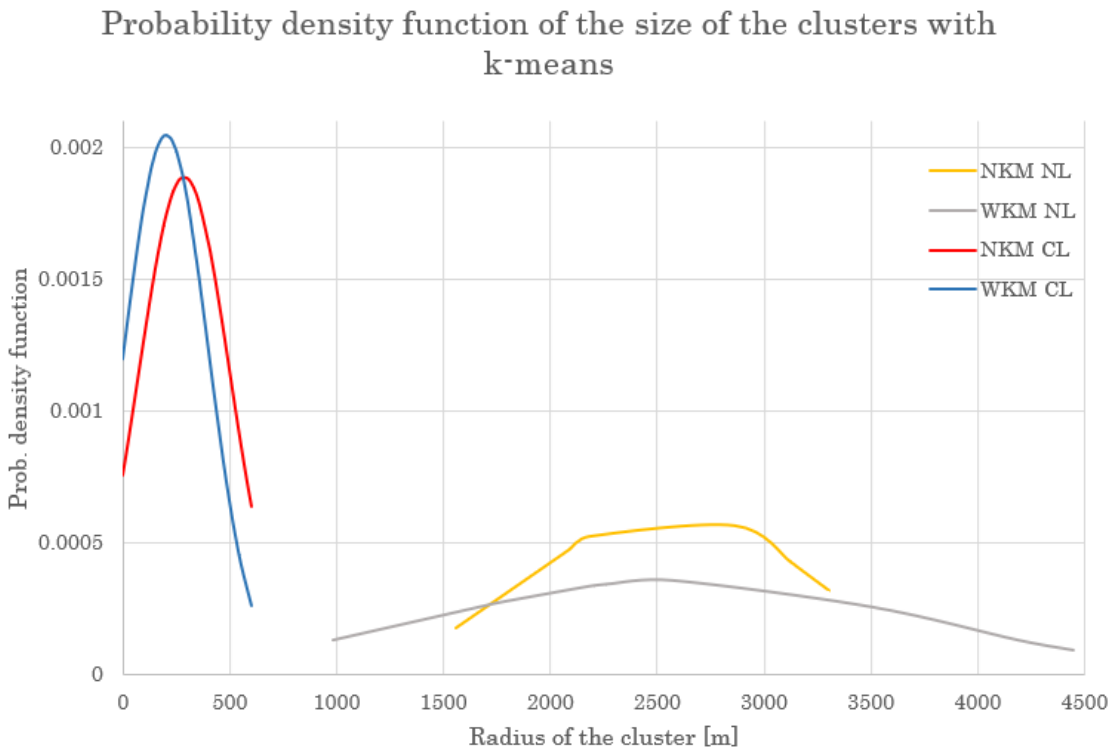

Figure 79. Prob. density function of the size of the clusters with k-means

The distance constraint in this case is 600 m. As can be seen, is the maximum distance with the complex loop version, while in the no loop version the distance is much higher. This is due to the big area under study and that the points are very scattered.

The average size is more homogeneous with the complex loop version than in the no loop version, reflected in a lower value of the standard deviation. Comparing the normal and the weighted versions, the weighted one is more homogeneous in terms of size with a lower standard deviation.

As mentioned before, in the complex loop version, the number of clusters is very high with respect to the total number of points due to the scattered nature of theses ones. This is the reason behind having clusters with a radius of 0 metres. After dividing the clusters in order to meet the distance constraint, some clusters only have one point, so the distance between the point and the centre is 0 as it is itself.

In the case with no loop, as the distance constraint is not considered, the clusters are assigned with a fixed input and in the case of the weighted version they tend to be situated near the areas with higher weights. This results in small clusters in the areas with higher weights and bigger clusters in the areas of lower weights (see Figure 71). Because of this, the size of weighted version is uneven, while in the normal version, as it only considers geographic position, the size is more homogeneous. This is reflected in the lower value of the average distance and standard deviation of the normal version in contrast to the weighted one.

LUKES ALGORITHM

In Chapter 6 this algorithm has been explained. To check the performance of the algorithm, the features evaluated are:

- The maximum power per cluster imposed.
- The final number of clusters, k.
- The running time, t_{run}.
- The maximum distance between load and substation inside the cluster, d_{max}.

As explained in section 5.2, LUKES algorithm works on a graph-type data. The input for the algorithm is the MST of the graph created from the points-population data. The MST, obtained with the Kruskal algorithm connecting all the populated points is shown in Figure 80.

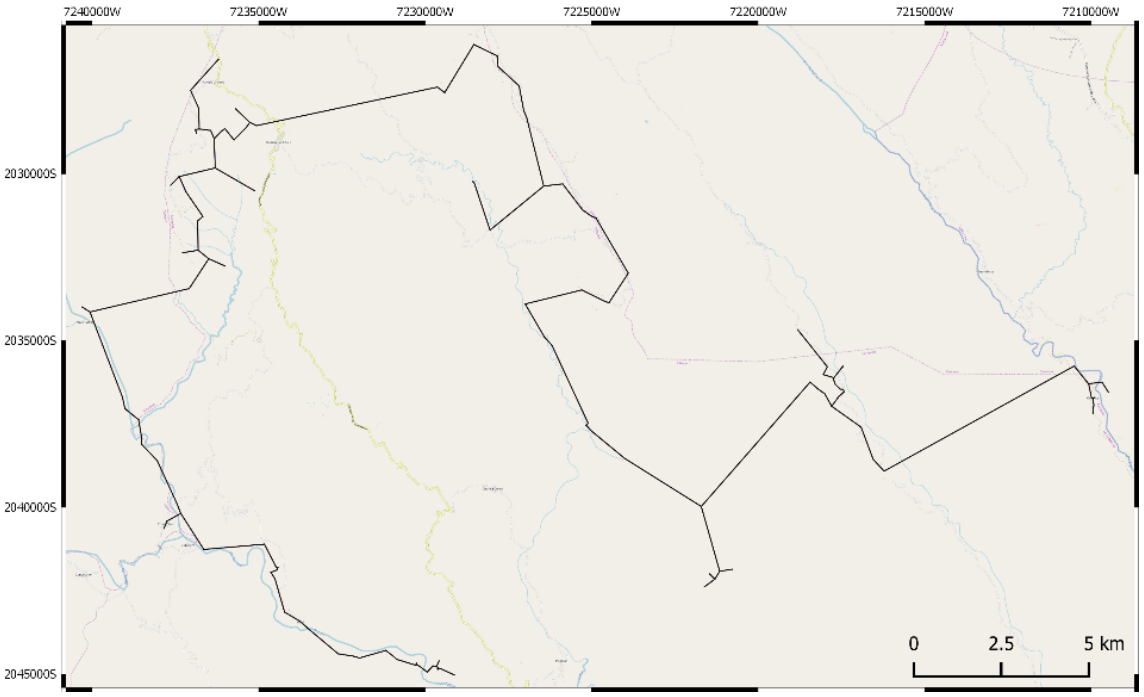

Figure 80. MST of the data in Omereque

The power attribute of the function *lukes_partitioning* has to be an integer value [52], so in order to work, it is needed a minimum value of 1 kW per point. The "max_size" of the algorithm, "W", represents the maximum power a cluster and hence a secondary substation could supply. The power is found as an attribute of the nodes of the graph obtained from the MST.

In this case study the power per capita has been set to 0.2, 0.4 and 0.8 kW per capita. The power per point, with 5 people per household, is 1, 2 and 4 kW/point. As they are integer values the code works.

To evaluate the results, the code of *lukes_partitioning* has been run with the weighted data input and different power constraints:

- W=10: simulating a small transformer for rural areas of developing countries.
- W=20: to compare the results with k-means.

- W=50: to use a bigger transformer and see how the power affects the size of the clusters.

The results are presented in Table 25:

Table 25. Results of LUKES algorithm in Omereque

Limit power, W [kW]	Number of clusters, k	Running time, t_run [s]	Max radio, d_max [m]
10	28	14.4	2,700
20	14	17.72	4,800
50	5	22.17	8,400

The lower the power of the transformer, the higher the number of clusters. In rural areas, the capacity of the transformers tends to be smaller. However, the population density tends to be high. This means that is difficult to meet power and distance constraints at the same time. This case study in Omereque reflects this problem, as the houses are very scattered and with low power demand per household.

A disadvantage of the LUKES algorithm is that, although the power is controlled, control on the distance is lost. However, the lower the value of W, the higher the number of clusters and the smaller the size of the clusters.

To see the results, the output files are uploaded into the software QGIS. The different clusters are represented with different colours, meaning groups of points supplied by a secondary substation.

Figure 81 presents the 28 clusters obtained with LUKES algorithm in the area of Omereque.

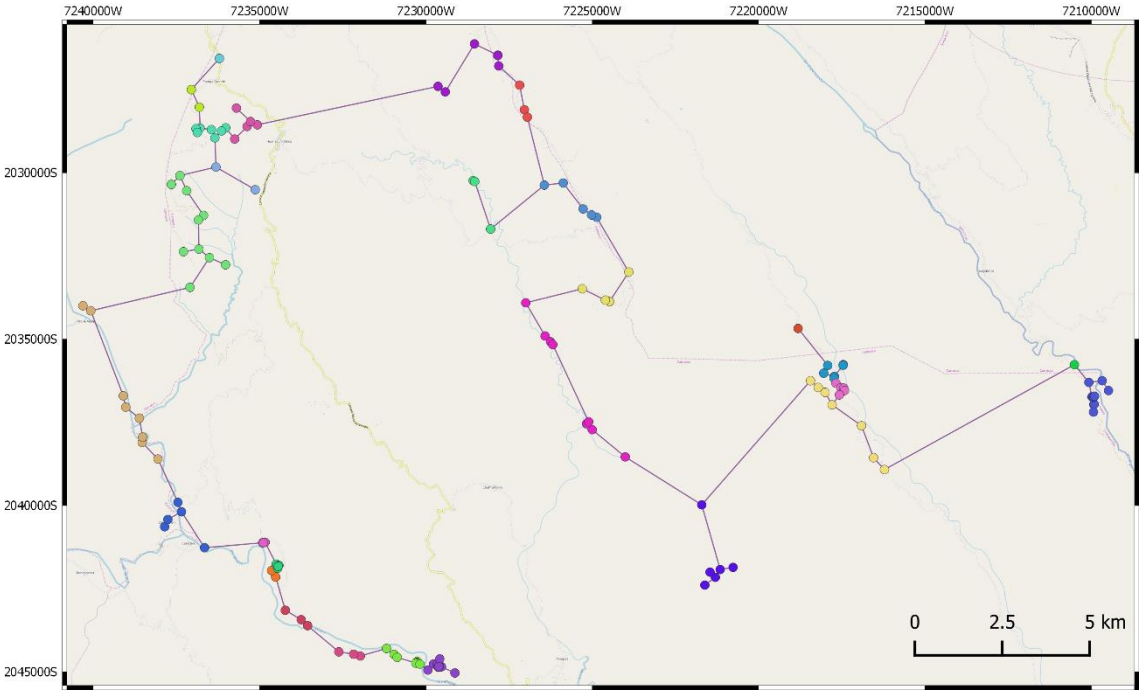

Figure 81. Clusters with LUKES algorithm W=10 in Omereque

Figure 82 shows the 14 clusters obtained with LUKES algorithm and a power constraint of 20 kW.

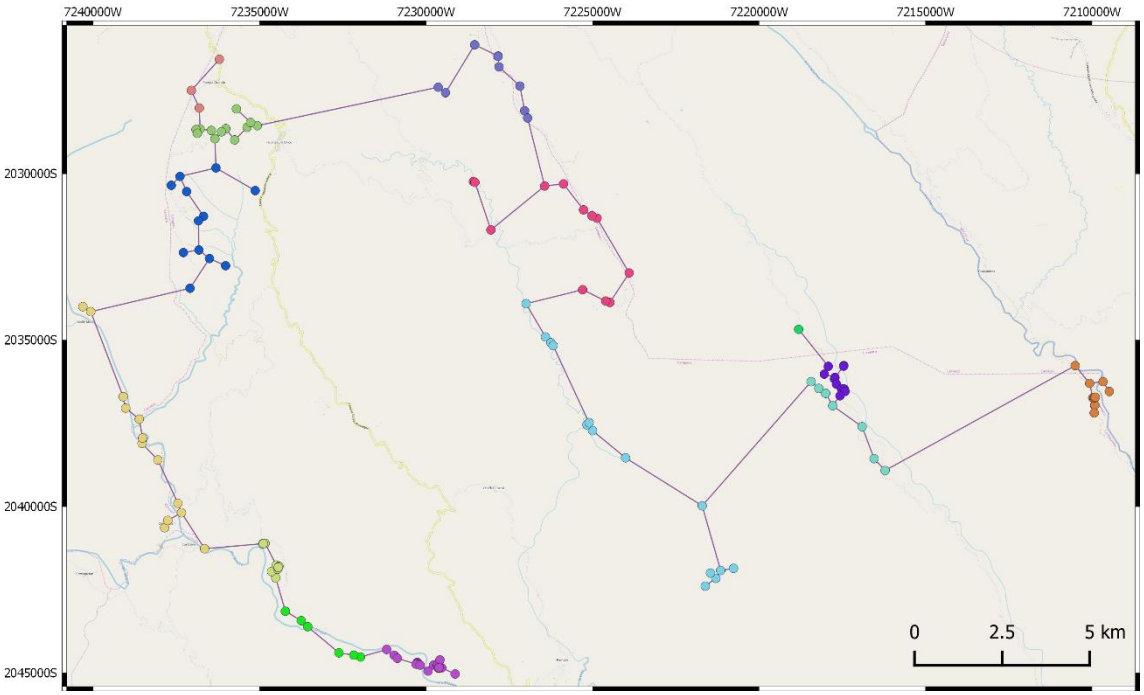

Figure 82.Clusters with LUKES algorithm W=20 in Omereque

Figure 83 shows the 5 clusters obtained with LUKES algorithm and a power constraint of 50 kW.

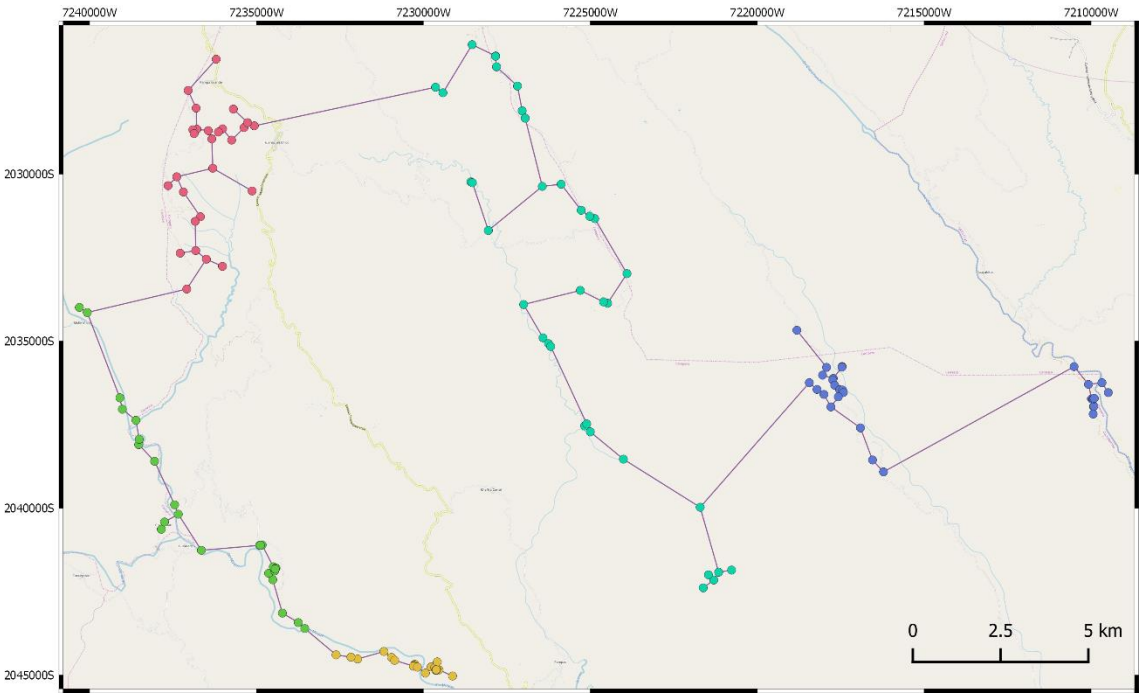

Figure 83.Clusters with LUKES algorithm W=50 in Omereque

The detailed results of this algorithm are found in Annex B.

In Table 26, the average power and its standard deviation of the clusters obtained with LUKES are presented.

Table 26. Detailed results of LUKES and WKM CL

ALGORITHM	**LUKES W=10**	**LUKES W=20**	**LUKES W=50**	**WKM CL**
Average power [kW]	7.25	14.50	40.60	4.14
St. deviation of the power	2.85	5.00	11.19	3.54

Figure 84 presents the probability density function of the power obtained with the different version of LUKES algorithm in the area of Omereque.

As can be seen, the power constraint is met is every case, with a maximum power of 10 kW, 20 kW or 50 kW, depending on the case.

As mentioned before, the average power is lower when the power constraint is lower due to the higher number of final clusters obtained.

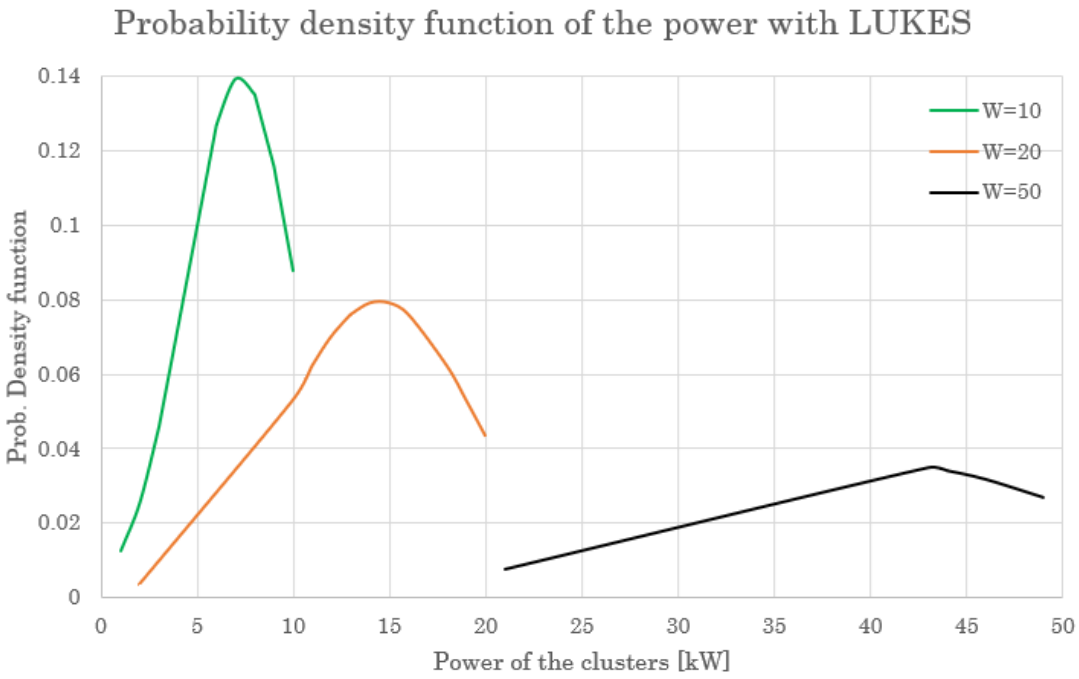

Figure 84. Prob. density function of the power with LUKES

Comparing the LUKES version of W=20 and the weighted k-means with complex loop in Figure 85, the conclusion extracted is, that although the power seems more homogeneously distributed due the lower value of the standard deviation, the power with LUKES gives better results. This means it has more clusters with higher power tending towards the value of 20 kW, the power of the substations. The reason behind is the average power is higher and less clusters tend to zero power.

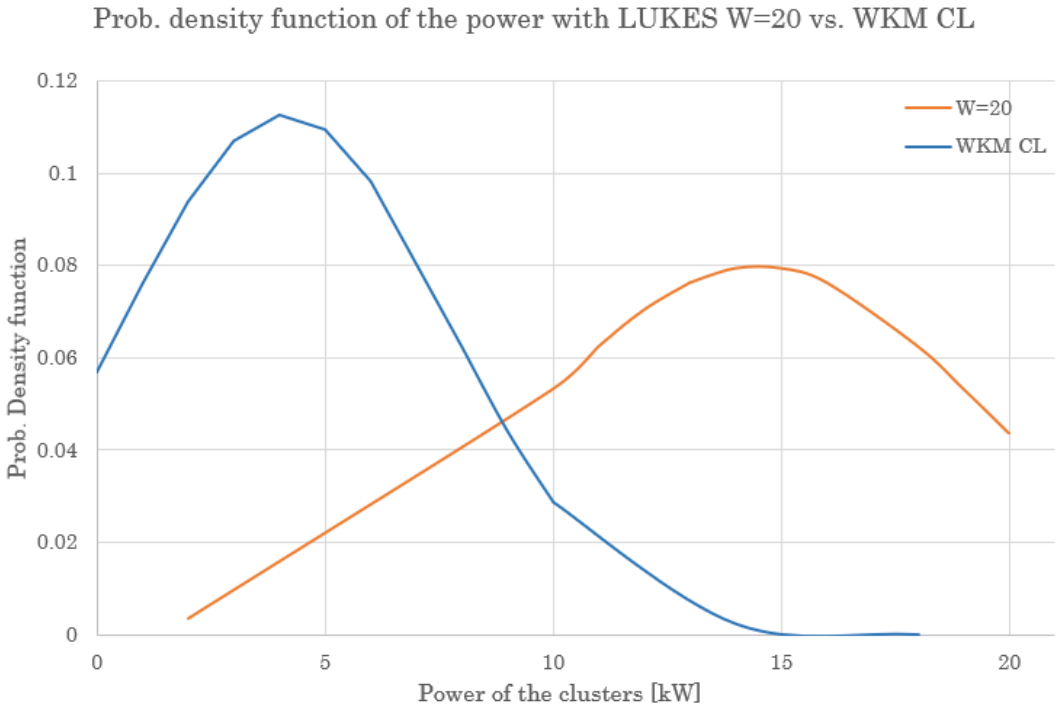

Figure 85. Prob. density function of the power with LUKES W=20 vs. WKM CL

Besides the power constraint, LUKES algorithm assigns the points to the clusters looking for cutting the longest edges in order to have the minimum cost. Also, points are in the same cluster only if they are connected through an edge. This makes that when evaluating a branch, if the power constraint is met, the rest of the points are assigned to a new cluster that could be smaller. All these reasons lead to clusters of different sizes

In Figure 86 this problem is depicted. In the case of W=20 in Omereque, the circled point is a cluster itself because that branch has already reached the maximum value of 20 kW. So, the green point is assigned to a new cluster. And, as it is not connected to any other points, it forms a cluster itself.

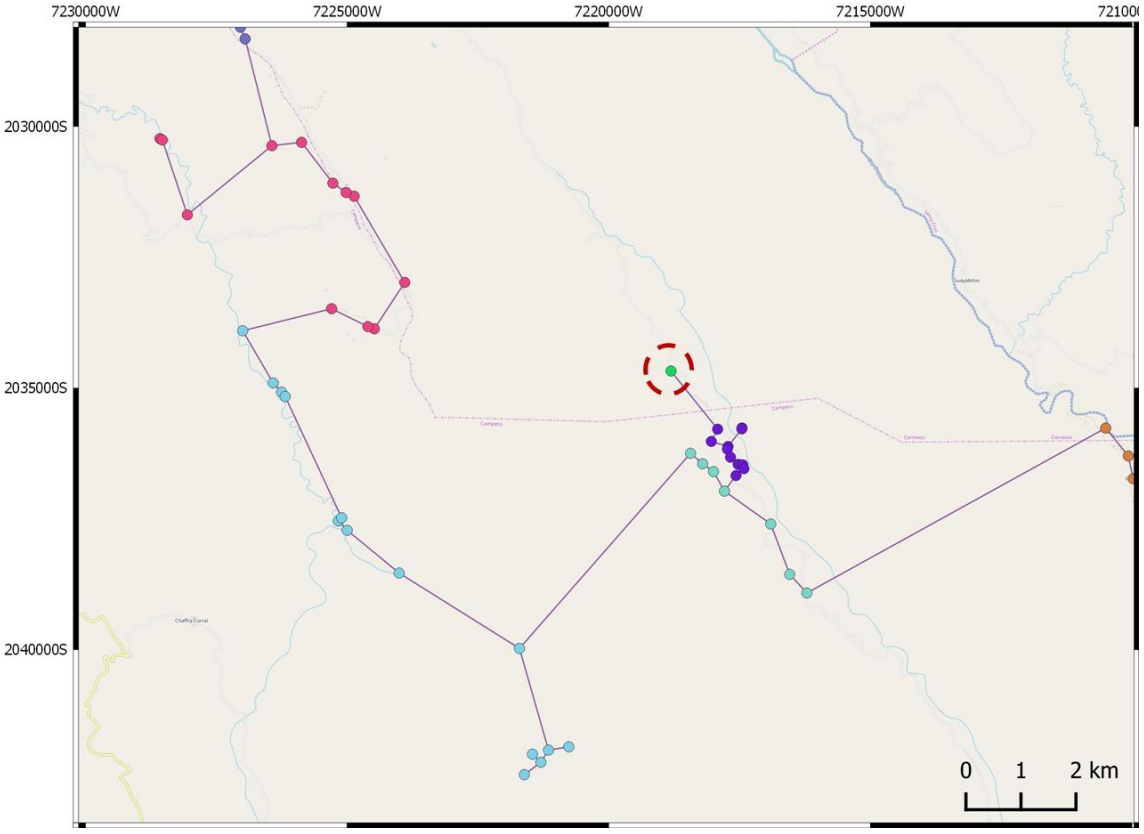

Figure 86. Assignment problems with LUKES

COMPARISON OF THE RESULTS OF THE ALGORITHMS

To better compare the results, a resume of the data obtained with the k-means and LUKES algorithms in terms of performance are presented in Table 27. The LUKES algorithm has been run with the data with the new power per capita, to see the behaviour of the algorithm with points of different weight.

Table 27. Performance comparison of the algorithms in Omereque

ID	Algorithm characteristics	Normal k-means	Weighted k-means	LUKES W=10	LUKES W=20	LUKES W=50
1	Number of clusters input	10	10	-	-	-
2	Final number of clusters	44	49	28	14	5
3	Running time [s]	29.95	5.34	14.4	17.72	22.17
4	Maximum distance between load and substation [m]	598.8	599.92	2,700	4,800	8,400
5	Average power [kW]	4.61	4.17	7.25	14.5	40.6

1. LUKES does not need and input of number of clusters while k-means yes.
2. The final number of clusters is higher with the k-means as it looks to meet the distance constraint and in this case the houses are very scattered. This is supported with the low values of average power per cluster in the k-means.
3. Time complexity is similar, although k-means is usually faster than LUKES. However, the time complexity of LUKES is proportional to the number of nodes. And as the number of nodes in this case is small, the running times are similar.
4. The maximum distance with k-means is controlled, however, with LUKES the distance constraint is not implemented and hence not met.

In conclusion, in contrast to the case study of Namanjavira, in this one, with the k-means, the power is more homogeneously distributed among the clusters in the normal version, while the size is more homogeneous in the weighted one. This is due to the nature of the data: a small sample of points, which are very scattered, and they do not present a great change in their weights as in Namanjavira.

Related to LUKES, the power is better distributed despite the higher value of the standard deviation compared to the weighted k-means and complex loop. This is reflected in more clusters with higher power tending towards the value of 20 kW, the power of the substations. As in Namanjavira, the size of the clusters is totally out of control and there is a need to introduce a distance constraint in this algorithm.

To compare the solution with reality, the 5 DSO indicators from Chapter 2 have been selected. In Table 28, this information is shown and compare with the corresponding values of Bolivia.

Table 28. DSO Indicators to compare the algorithms and reality in Bolivia

ID	Indicators	Normal k-means	Weighted k-means	LUKES W=10	LUKES W=20	LUKES W=50	Bolivia (Literature)
1	LV circuit length per LV consumer	599 m	600 m	2700 m	4800 m	8400 m	500 -1000 m
2	Number of LV consumer per MV/LV substation	16	14	25	50	139	16-26
3	MV/LV substation capacity per LV consumer	1.27	1.41	0.40	0.40	0.36	0.958
4	Number of MV supply points per HV/MV substation	44	49	28	14	5	14 - 24
5	Typical transformation capacity of MV/LV secondary substations in rural areas	20 kW	20 kW	10 kW	20 kW	50 kW	12, 20 kW*

*Value calculated with a power factor=0.8

Considering the values of Bolivia:

1. When the power constraint is met, the LV circuit length is higher, which makes sense due to the lower population density in rural areas.
2. The number of LV consumers per substation suits with the values from Literature when the power is similar (10-20 kW).
3. The capacity of the MV supply point is the typical transformation capacity of the substations in rural areas [ID 5]. This value should be lower in rural areas, as usually smaller transformers are used, and the power demand is lower too. With k-means is higher due to the higher number of clusters. The systems would be highly oversized.
4. The number of substations is higher with k-means due to the distance constraint and the low population density. However, with the power constraint (LUKES) the values are aligned with literature.
5. The value of the capacity of the transformers is the one used for the case study. The capacity is usually lower in rural areas due to lower power demand, lower simultaneity factor and lower population density.

Chapter 9

CONCLUSIONS

In 2019, slightly less than 1 billion people lacked access to electricity in the world. Approximately, 50% of them are found just in Sub-Saharan Africa and 87% live in rural areas. In a global framework where the goal of the 7th SDG is to ensure access to affordable, reliable, sustainable and modern energy for all, rural electrification planning is needed.

Numerous attempts have been made to address the problem of rural electrification planning, and many tools have been developed to look for a solution. Related to the setting up of rural electricity infrastructure, the power system planning is one of the important things for obtaining the economic operation of the power system, and specifically in the distribution system. Most of the researches are focused on the MV level and less attention has been devoted to LV networks [7].

The LV lines account for 60% of the length of the voltage lines and the losses in LV network are approximately 50% of the total losses of the power system. So, to consider the distribution system in the rural electrification planning is important.

This thesis work is centred on access to electricity and focuses on providing connectivity to households by studying the siting of secondary substations.

The objective was to design a procedure able to evaluate the optimal location of secondary substations from a topological perspective considering the population distribution given by georeferenced data. To do so, it is required to detect the populated areas by clustering the population. A two-step procedure has been developed.

For the first step the DBSCAN algorithm has been chosen because it is written in Python, its reasonable low running times, low number of inputs and because it had been already tested in the area on Namanjavira.

This first step consists of clustering densely populated areas with the DBSCAN algorithm in order to find the more dense areas in terms of inhabitants, and the second step consists of applying another clustering algorithm to the groups obtained with DBSCAN in order to divide the population in areas supplied by a secondary substation.

For the second step, two different algorithms have been studied in order to see different advantages and disadvantages in the siting of secondary substations and choose the one which returns best results. These two algorithms are k-means, based on partition, and LUKES, based on graph theory.

To evaluate the performance of the algorithms, two case studies have been analysed: one in Namanjavira (Mozambique) and the other in Omereque (Bolivia).

After conducting the case studies, both pros and cons are found for the algorithms.

K-MEANS

Advantages:

- The algorithm is fast, with low running times.
- A distance constraint is implemented and reached.
- It is easy to implement as a first approach to the problem.
- The weighted k-means divides efficiently the area considering the power of the points.

Disadvantages:

- There is no power constraint for the clusters and the power among the clusters is uneven.
- There is need of the input of the number of clusters. And together with the fact that the final output is a local optimal, not global, this could lead to different solutions when run different times.
- The algorithm does not consider geographic information, so a substation or (centre) can be placed anywhere within the planned area, in random places, not evaluating the characteristics of the terrain.
- The algorithm fails to be realistic. It does not consider the structure and construction of the energy system and this leads to different problems, as assignment problems (Figure 61).

LUKES

Advantages:

- A power constraint is implemented and reached.
- The substations would be place on the edges, a better approximation than the centres of the k-means.
- There is no need to provide the number of clusters.
- The points of the cluster are always connected through the edges of the MST, so there are no assignment problems as the ones of k-means.

Disadvantages:

- The algorithm is very slow, with high running times due to the proportionality to the power constraint and number of nodes of the time complexity (W^2n)
- There is no distance constraint for the clusters.
- The power of the nodes has to be an integer value.
- Besides the power constraint, the assignment process of the algorithm works including the points in the clusters by cutting the longest edges in order to have the minimum cost. Also, points are in the same cluster only if they are

connected through an edge. This makes that when evaluating a branch, if the power constraint is met, the rest of the points are assigned to a new cluster that could be smaller. All these reasons lead to clusters of different sizes.

In a nutshell, although distance constraints are not considered in LUKES and power constraints are not considered in k-means, interesting results have been obtained. The power among the clusters in LUKES is controlled and in k-means although the power is uneven among clusters, is controlled indirectly through the distance constraint, leading to less power in smaller clusters.

A combination of both constraints will lead to better and more real results, although as seen in the case of Omereque, in areas with low power demand and scattered population is difficult to meet power and distance constraints at the same time.

Regarding the comparison of the DSO's indicators, k-means returns best results in terms of length of the lines, while LUKES fits better with literature in terms of power. This is due to the different constraint imposed in both algorithms.

As there is room for improvement in this procedure, the future steps of this project are:

- Optimization of the k-means and LUKES algorithms in order to implement both distance and power constraints at the same time. For the moment k-means deals with distance and LUKES with power.
- Try different optimization options for the distance constraint loop to avoid having very small clusters of only one point. For example, not splitting the cluster in 2 when it does not meet the constraint but in 3 for example, if the maximum distance between point and centre is higher than 2000 m.
- Look for the centre of the clusters with LUKES algorithm or a place to sit the secondary substation. This step has not been implemented yet. The idea is to calculate the centres with betweenness centrality. In Python there is a function which calculates this directly. The location returned is situated on an edge of that cluster.
- For the power constraint consider also the oversizing of the substations, in order to meet the demand and foresee the power demand growth when a place is electrified.
- Applying the function *lukes_partitioning*, sometimes very small clusters are obtained (Figure 86). So, another function with this algorithm has been developed by a research group in Switzerland (*Swiss Centre for Competence in Energy Research on the Future Swiss Electrical Infrastructure (SCCER-FURIES)*) where besides applying LUKES algorithm, then it evaluates reallocation of loads creating a compatibility graph. Once the clustering is complete, grid sizing can be executed. The substations are placed on edges that maximize betweenness centrality. This new function is called *smart_cluster*. However, it has not been implemented yet in this project work. It could be a future development of the application of LUKES to this problem of rural electrification.

In line with the procedure approach, an improvement for future works would be also to evaluate other algorithms considered in Chapter 5, to compare the performance with the DBSCAN, the k-means or the LUKES, and to conduct an economic analysis.

The results obtained are promising as they already fit with literature. And, as soon as both constraints are included in the procedure, better and more realistic results are going to be obtained.

This project tried to contribute to this duty of rural electrification planning framed in the 7th SDG. Because having access to electricity brings benefits as increasing incomes, improving education, increasing productivity in business and agriculture, improving health, increasing security and lowering environmental contamination, among others. Access to electricity is a prerequisite for a society if it wants to move out of subsistence.

The SDGs not only reaffirm the commitment to end poverty but also to build a more sustainable, safer and more prosperous planet for all humanity.

Annex A

RESULTS OF NAMANJAVIRA

K-MEANS

Table 29. Results for normal k-means and no loop

Cluster	Points	Pop	Power	Pop density	Power per capita	Max Distance
0	171	684	100.70	4	0.1472	1621.11
1	71	284	13.70	4	0.0482	1407.98
2	131	524	104.80	4	0.2000	1146.78
3	84	336	13.50	4	0.0402	513.90
4	164	656	61.50	4	0.0937	1628.39
5	244	976	97.00	4	0.0994	1302.31
6	86	344	9.90	4	0.0288	1065.20
7	135	540	13.50	4	0.0250	1322.88
8	146	584	27.80	4	0.0476	1417.48
9	99	396	10.10	4	0.0255	1164.87
10	94	376	71.90	4	0.1912	948.02
11	62	248	6.20	4	0.0250	1040.73
12	74	296	14.80	4	0.0500	1072.83
13	293	1172	117.20	4	0.1000	1366.68
14	120	480	20.70	4	0.0431	1393.38
15	135	540	32.40	4	0.0600	1299.88
16	320	1280	253.20	4	0.1978	1863.85
17	90	360	9.00	4	0.0250	841.64
18	88	352	8.80	4	0.0250	751.99
19	111	444	25.40	4	0.0572	1411.71
20	69	276	10.10	4	0.0366	1099.69
21	163	652	30.00	4	0.0460	946.37
22	126	504	13.20	4	0.0262	1287.43
23	84	336	16.30	4	0.0485	1105.98
24	235	940	104.40	4	0.1111	1292.64
25	47	188	4.70	4	0.0250	1344.40
26	85	340	8.50	4	0.0250	1159.56
27	175	700	134.40	4	0.1920	943.83
28	181	724	141.30	4	0.1952	1224.22
29	69	276	6.90	4	0.0250	676.47
30	98	392	41.30	4	0.1054	1376.83
31	80	320	29.00	4	0.0906	869.02
32	75	300	12.70	4	0.0423	1582.29
33	134	536	80.60	4	0.1504	1894.99
34	97	388	16.80	4	0.0433	1418.05
35	211	844	168.80	4	0.2000	987.36
36	82	328	36.10	4	0.1101	1336.58
TOTAL	4729	18916	1867.20			45127.30
AVERAGE		511.24	50.46			1219.66
DEVIATION		259.70	57.00			302.49

Table 30. Results for the weighted k-means no loop

Cluster	Points	Pop	Power	Pop density	Average power per capita	Max Distance
0	28	112	2.8	4	0.0250	943.89
1	24	96	2.4	4	0.0250	722.94
2	38	152	3.8	4	0.0250	681.36
3	43	172	4.3	4	0.0250	918.22
4	24	96	2.4	4	0.0250	989.47
5	53	212	9.1	4	0.0429	1099.21
6	66	264	12.7	4	0.0481	1399.86
7	96	384	16.6	4	0.0432	1532.61
8	117	468	20.1	4	0.0429	1800.42
9	96	384	19.4	4	0.0505	859.04
10	131	524	13.7	4	0.0261	1334.82
11	54	216	18.9	4	0.0875	1148.82
12	146	584	61.4	4	0.1051	976.61
13	303	1212	242.4	4	0.2000	943.32
14	164	656	84.3	4	0.1285	1948.67
15	279	1116	113.8	4	0.1020	1715.67
16	259	1036	97.6	4	0.0942	1751.66
17	214	856	21.4	4	0.0250	2186.97
18	202	808	79.3	4	0.0981	1515.45
19	132	528	51.6	4	0.0977	1750.65
20	300	1200	49.7	4	0.0414	2611.13
21	163	652	92.8	4	0.1423	1637.55
22	172	688	134.1	4	0.1949	1219.85
23	97	388	73.1	4	0.1884	1325.05
24	219	876	40.2	4	0.0459	1945.02
25	168	672	16.8	4	0.0250	1711.19
26	154	616	15.4	4	0.0250	1534.08
27	107	428	49.7	4	0.1161	1035.45
28	249	996	165.6	4	0.1663	1475.54
29	43	172	30.2	4	0.1756	878.24
30	108	432	86.4	4	0.2000	713.71
31	91	364	72.8	4	0.2000	434.56
32	137	548	34.7	4	0.0633	1623.22
33	95	380	76	4	0.2000	463.31
34	73	292	34.9	4	0.1195	920.65
35	12	48	2.4	4	0.0500	310.59
36	72	288	14.4	4	0.0500	896.72
TOTAL	4729	18916	1867.2			46955.51
AVERAGE		511.24	50.46			1269.07
DEVIATION		325.44	52.10			517.19

Table 31. Results for the normal k-means with the complex loop

Cluster	Points	Pop	Power	Pop Density	Power per capita	Max Distance
0	112	448	76.0	4	0.1696	614.51
1	105	420	24.2	4	0.0576	774.95
2	62	248	12.4	4	0.0500	864.53
3	102	408	17.2	4	0.0422	947.31
4	30	120	3.0	4	0.0250	598.28
5	94	376	37.6	4	0.1000	392.80
6	35	140	3.5	4	0.0250	944.27
7	37	148	3.7	4	0.0250	866.58
8	323	1292	252.7	4	0.1956	948.03
9	179	716	143.2	4	0.2000	831.03
10	60	240	11.2	4	0.0467	830.10
11	118	472	94.4	4	0.2000	895.73
12	52	208	10.4	4	0.0500	514.17
13	221	884	102.1	4	0.1155	845.60
14	94	376	38.0	4	0.1011	751.25
15	42	168	33.6	4	0.2000	466.69
16	67	268	6.7	4	0.0250	776.42
17	39	156	7.8	4	0.0500	654.86
18	114	456	34.5	4	0.0757	835.51
19	36	144	3.6	4	0.0250	387.14
20	119	476	23.4	4	0.0492	825.84
21	29	116	5.8	4	0.0500	591.84
22	27	108	2.7	4	0.0250	794.63
23	75	300	7.5	4	0.0250	722.90
24	45	180	35.3	4	0.1961	959.99
25	91	364	35.8	4	0.0984	442.09
26	72	288	7.2	4	0.0250	946.17
27	82	328	13.3	4	0.0405	966.98
28	48	192	4.8	4	0.0250	910.39
29	133	532	80.5	4	0.1513	909.07
30	152	608	58.4	4	0.0961	963.19
31	105	420	47.7	4	0.1136	972.69
32	77	308	14.3	4	0.0464	864.28
33	27	108	2.7	4	0.0250	617.48
34	34	136	6.8	4	0.0500	595.73
35	83	332	36.9	4	0.1111	778.80
36	21	84	3.5	4	0.0417	839.58
37	51	204	20.6	4	0.1010	847.47
38	12	48	2.4	4	0.0500	310.59
39	57	228	5.7	4	0.0250	876.61
40	159	636	63.6	4	0.1000	781.50
41	44	176	4.4	4	0.0250	720.62
42	39	156	5.1	4	0.0327	716.11
43	97	388	74.1	4	0.1910	908.49
44	15	60	1.5	4	0.0250	356.26

Table 32. Cont. Results for the normal k-means with the complex loop

45	133	532	106.4	4	0.2000	686.51
46	25	100	5.0	4	0.0500	181.50
47	33	132	11.7	4	0.0886	360.56
48	3	12	0.3	4	0.0250	217.11
49	8	32	1.0	4	0.0313	719.75
50	5	20	0.7	4	0.0350	223.67
51	27	108	2.7	4	0.0250	801.45
52	48	192	4.8	4	0.0250	567.67
53	45	180	6.8	4	0.0378	998.68
54	14	56	2.7	4	0.0482	969.19
55	20	80	2.0	4	0.0250	953.61
56	35	140	3.5	4	0.0250	714.75
57	12	48	6.8	4	0.1417	545.25
58	54	216	21.6	4	0.1000	925.03
59	38	152	3.8	4	0.0250	996.35
60	10	40	1.0	4	0.0250	197.79
61	20	80	2.0	4	0.0250	574.00
62	31	124	4.9	4	0.0395	933.16
63	18	72	1.8	4	0.0250	977.67
64	16	64	6.4	4	0.1000	396.09
65	9	36	0.9	4	0.0250	264.02
66	15	60	1.7	4	0.0283	804.94
67	41	164	4.7	4	0.0287	818.79
68	2	8	0.2	4	0.0250	157.43
69	51	204	40.8	4	0.2000	321.08
70	35	140	5.7	4	0.0407	838.85
71	30	120	4.8	4	0.0400	838.10
72	39	156	7.8	4	0.0500	347.00
73	32	128	10.1	4	0.0789	895.73
74	50	200	38.0	4	0.1900	399.51
75	23	92	2.3	4	0.0250	387.33
76	30	120	5.8	4	0.0483	772.36
77	27	108	5.4	4	0.0500	547.50
78	65	260	52.0	4	0.2000	330.68
79	33	132	5.4	4	0.0409	674.43
80	9	36	0.9	4	0.0250	808.29
81	1	4	0.1	4	0.0250	0.00
82	15	60	2.9	4	0.0483	933.13
83	11	44	2.0	4	0.0455	320.85
TOTAL	4729	18916	1867.2			57358.82
AVERAGE		225.19	22.23			682.84
DEVIATION		210.59	38.05			252.19

Table 33. Results for the weighted k-means with complex loop

Cluster	Points	Pop	Power	Pop density	Average power per capita	Max Distance
0	28	112	2.8	4	0.025	943.891
1	24	96	2.4	4	0.025	722.937
2	38	152	3.8	4	0.025	681.363
3	43	172	4.3	4	0.025	918.216
4	24	96	2.4	4	0.025	989.475
5	34	136	5.9	4	0.043	682.841
6	26	104	5	4	0.048	743.717
7	39	156	7.8	4	0.050	698.979
8	31	124	6.2	4	0.050	563.087
9	96	384	19.4	4	0.051	859.045
10	42	168	4.8	4	0.029	867.035
11	9	36	0.9	4	0.025	379.348
12	146	584	61.4	4	0.105	976.608
13	303	1212	242.4	4	0.200	943.315
14	124	496	80.3	4	0.162	816.242
15	10	40	3.1	4	0.078	570.147
16	14	56	1.4	4	0.025	329.166
17	28	112	2.8	4	0.025	674.660
18	5	20	0.5	4	0.025	95.880
19	8	32	2	4	0.063	889.819
20	10	40	1	4	0.025	571.671
21	29	116	5.6	4	0.048	717.861
22	51	204	40.8	4	0.200	321.077
23	41	164	32.8	4	0.200	466.894
24	99	396	17.6	4	0.044	940.737
25	38	152	3.8	4	0.025	773.552
26	91	364	9.1	4	0.025	906.880
27	57	228	22.5	4	0.099	808.730
28	78	312	28.8	4	0.092	870.520
29	43	172	30.2	4	0.176	878.242
30	108	432	86.4	4	0.200	713.707
31	91	364	72.8	4	0.200	434.557
32	9	36	0.9	4	0.025	450.591
33	95	380	76	4	0.200	463.312
34	73	292	34.9	4	0.120	920.653
35	12	48	2.4	4	0.050	310.588
36	72	288	14.4	4	0.050	896.722
37	19	76	3.2	4	0.042	919.229
38	34	136	6.8	4	0.050	461.059
39	20	80	4	4	0.050	323.216
40	30	120	3	4	0.025	676.559
41	53	212	5.3	4	0.025	565.000
42	45	180	18	4	0.100	611.123
43	40	160	4	4	0.025	903.667
44	206	824	90.3	4	0.110	892.250
45	61	244	24.4	4	0.100	317.998
46	30	120	3	4	0.025	376.853
47	124	496	49.6	4	0.100	861.655
48	112	448	44.8	4	0.100	549.405
49	82	328	16	4	0.049	772.178
50	58	232	34	4	0.147	983.498
51	75	300	57.9	4	0.193	575.745
52	43	172	32	4	0.186	432.920

Table 34. Cont. Results for the weighted k-means with complex loop

53	1	4	0.1	4	0.025	0.000
54	78	312	7.8	4	0.025	985.449
55	63	252	6.3	4	0.025	746.096
56	50	200	27.2	4	0.136	462.062
57	171	684	136.8	4	0.200	509.712
58	17	68	1.7	4	0.025	595.062
59	6	24	0.9	4	0.038	687.613
60	11	44	2.2	4	0.050	286.583
61	36	144	3.6	4	0.025	990.433
62	63	252	20.4	4	0.081	655.470
63	56	224	22.4	4	0.100	637.361
64	67	268	6.7	4	0.025	760.996
65	73	292	29.2	4	0.100	853.136
66	12	48	4.8	4	0.100	307.414
67	36	144	6.8	4	0.047	570.721
68	76	304	53.2	4	0.175	515.237
69	46	184	35.4	4	0.192	977.293
70	52	208	5.2	4	0.025	893.194
71	40	160	24.3	4	0.152	354.788
72	27	108	3.8	4	0.035	831.372
73	26	104	10.4	4	0.100	293.113
74	30	120	3	4	0.025	741.953
75	57	228	11	4	0.048	973.215
76	10	40	8	4	0.200	137.460
77	15	60	1.7	4	0.028	858.815
78	71	284	7.8	4	0.027	688.989
79	45	180	17.4	4	0.097	469.927
80	9	36	0.9	4	0.025	264.020
81	10	40	1	4	0.025	371.075
82	35	140	14	4	0.100	263.922
83	31	124	3.1	4	0.025	867.417
84	29	116	4.7	4	0.041	973.068
85	39	156	7.8	4	0.050	361.410
86	33	132	6.5	4	0.049	569.613
87	5	20	0.5	4	0.025	934.913
88	29	116	4.5	4	0.039	717.437
89	17	68	6.8	4	0.100	358.105
90	23	92	2.3	4	0.025	642.306
91	3	12	0.3	4	0.025	217.109
92	48	192	4.8	4	0.025	511.840
93	65	260	13	4	0.050	411.309
94	1	4	0.1	4	0.025	0.000
95	3	12	0.6	4	0.050	500.084
96	2	8	0.2	4	0.025	321.687
97	11	44	2.1	4	0.048	356.621
TOTAL	4729	18916	1867.2			61439.82
AVERAGE		193.02	19.05			626.94
DEVIATION		181.96	33.11			254.76

LUKES

Table 35. Results for LUKES with W=500

Cluster	Points	Pop density	Power per capita	Pop	Power
0	120	4	0.100	480	48
1	116	4	0.100	464	46.4
2	74	4	0.164	296	48.4
3	69	4	0.107	276	29.4
4	294	4	0.042	1176	49.2
5	9	4	0.042	36	1.5
6	62	4	0.200	248	49.6
7	33	4	0.200	132	26.4
8	46	4	0.200	184	36.8
9	55	4	0.184	220	40.5
10	42	4	0.200	168	33.6
11	69	4	0.162	276	44.8
12	96	4	0.122	384	46.8
13	51	4	0.200	204	40.8
14	62	4	0.200	248	49.6
15	50	4	0.200	200	40
16	51	4	0.200	204	40.8
17	55	4	0.200	220	44
18	29	4	0.200	116	23.2
19	62	4	0.200	248	49.6
20	72	4	0.171	288	49.2
21	72	4	0.027	288	7.9
22	99	4	0.126	396	49.7
23	74	4	0.050	296	14.8
24	53	4	0.200	212	42.4
25	200	4	0.051	800	41
26	279	4	0.041	1116	46.2
27	140	4	0.025	560	14
28	66	4	0.120	264	31.8
29	75	4	0.100	300	30
30	110	4	0.099	440	43.4
31	72	4	0.098	288	28.2
32	153	4	0.076	612	46.3
33	89	4	0.133	356	47.4
34	53	4	0.200	212	42.4
35	57	4	0.200	228	45.6
36	56	4	0.200	224	44.8
37	96	4	0.129	384	49.6
38	128	4	0.065	512	33.2
39	45	4	0.042	180	7.6
40	273	4	0.046	1092	50
41	224	4	0.045	896	39.9
42	154	4	0.025	616	15.4
43	61	4	0.200	244	48.8
44	15	4	0.200	60	12
45	52	4	0.200	208	41.6
46	58	4	0.200	232	46.4
47	30	4	0.200	120	24
48	123	4	0.100	492	49.2
49	305	4	0.029	1220	35
TOTAL	4729			18916	1867.2
AVERAGE				378.32	37.34
DEVIATION				286.40	13.24

Annex B

RESULTS OF OMEREQUE

K-MEANS

Table 36. Results for normal k-means and no loop

Cluster	Points	Pop density	Power per capita	Pop	Power	Max Distance
0	6	5	0.2000	30	6.00	3157.78
1	11	5	0.2000	55	11.00	2064.11
2	14	5	0.4571	70	32.00	3240.47
3	17	5	0.3294	85	28.00	2864.02
4	11	5	0.2000	55	11.00	3126.26
5	19	5	0.3368	95	32.00	2093.71
6	21	5	0.2000	105	21.00	2180.97
7	18	5	0.3889	90	35.00	2081.36
8	12	5	0.2500	60	15.00	3307.43
9	10	5	0.2400	50	12.00	1557.43
TOTAL	139			695	203	25673.56
AVERAGE				69.50	20.30	2567.36
DEVIATION				23.62	10.67	635.33

Table 37. Results for the weighted k-means and no loop

Cluster	Points	Pop density	Power per capita	Pop	Power	Max Distance
0	12	5	0.233	60	14	4450.09
1	27	5	0.326	135	44	4192.10
2	14	5	0.457	70	32	2250.38
3	21	5	0.200	105	21	2180.97
4	8	5	0.325	40	13	2279.57
5	12	5	0.250	60	15	3504.92
6	6	5	0.200	30	6	1689.80
7	19	5	0.337	95	32	2572.49
8	9	5	0.333	45	15	1760.43
9	11	5	0.200	55	11	982.51
TOTAL	139			695	203	25863.27
AVERAGE				69.5	20.3	2586.33
DEVIATION				32.70	11.91	1121.78

Table 38. Results for the normal k-means with the complex loop

Cluster	Points	Pop density	Power per capita	Pop	Power	Max Distance
0	8	5	0.2000	40	8.00	572.06
1	3	5	0.2000	15	3.00	470.99
2	2	5	0.6000	10	6.00	527.58
3	3	5	0.6667	15	10.00	351.50
4	3	5	0.2667	15	4.00	185.36
5	8	5	0.3000	40	12.00	457.99
6	2	5	0.4000	10	4.00	130.99
7	5	5	0.3600	25	9.00	507.71
8	10	5	0.2000	50	10.00	585.16
9	2	5	0.2000	10	2.00	97.83
10	1	5	0.2000	5	1.00	0.00
11	4	5	0.2000	20	4.00	410.12
12	4	5	0.3500	20	7.00	477.21
13	9	5	0.3556	45	16.00	465.79
14	3	5	0.2667	15	4.00	158.44
15	3	5	0.2667	15	4.00	514.24
16	1	5	0.4000	5	2.00	0.00
17	1	5	0.2000	5	1.00	0.00
18	3	5	0.4000	15	6.00	433.43
19	2	5	0.4000	10	4.00	132.94
20	1	5	0.4000	5	2.00	0.00
21	3	5	0.3333	15	5.00	425.24
22	1	5	0.4000	5	2.00	0.00
23	3	5	0.2000	15	3.00	222.89
24	1	5	0.2000	5	1.00	0.00
25	3	5	0.3333	15	5.00	551.32
26	1	5	0.2000	5	1.00	0.00
27	2	5	0.2000	10	2.00	518.57
28	2	5	0.2000	10	2.00	230.31
29	2	5	0.3000	10	3.00	598.80
30	6	5	0.2000	30	6.00	308.05
31	1	5	0.2000	5	1.00	0.00
32	3	5	0.2000	15	3.00	249.01
33	7	5	0.2000	35	7.00	404.12
34	1	5	0.2000	5	1.00	0.00
35	3	5	0.4000	15	6.00	202.87
36	2	5	0.6000	10	6.00	235.91
37	5	5	0.4000	25	10.00	409.96
38	2	5	0.4000	10	4.00	25.06
39	3	5	0.4000	15	6.00	530.59
40	3	5	0.2000	15	3.00	461.89
41	4	5	0.2000	20	4.00	485.18
42	2	5	0.2000	10	2.00	271.07
43	1	5	0.2000	5	1.00	0.00
TOTAL	139			695	203	12610.16
AVERAGE				15.80	4.61	286.59
DEVIATION				11.31	3.34	211.33

Table 39. Results for the weighted k-means with complex loop

Cluster	Points	Pop density	Power per capita	Pop	Power	Max Distance
0	2	5	0.2	10	2	132.94
1	0	5	0.0	0	0	0
2	1	5	0.4	5	2	0
3	3	5	0.2	15	3	351.50
4	2	5	0.2	10	2	25.06
5	1	5	0.2	5	1	0
6	1	5	0.2	5	1	1.52E-09
7	1	5	0.4	5	2	1.52E-09
8	2	5	0.2	10	2	130.99
9	1	5	0.2	5	1	1.391E-09
10	3	5	0.2	15	3	470.99
11	1	5	0.2	5	1	1.346E-09
12	7	5	0.4	35	14	251.13
13	6	5	0.2	30	6	308.05
14	2	5	0.3	10	3	361.42
15	3	5	0.2	15	3	158.44
16	5	5	0.2	25	5	507.71
17	1	5	0.4	5	2	0
18	3	5	0.4	15	6	202.87
19	10	5	0.2	50	10	481.82
20	3	5	0.2	15	3	425.24
21	3	5	0.4	15	6	39.53
22	3	5	0.2	15	3	222.89
23	1	5	0.4	5	2	0
24	1	5	0.2	5	1	0
25	4	5	0.3	20	6	391.37
26	1	5	0.2	5	1	0
27	4	5	0.3	20	6	485.50
28	2	5	0.8	10	8	276.30
29	3	5	0.7	15	10	520.12
30	9	5	0.2	45	9	465.79
31	3	5	0.4	15	6	249.01
32	1	5	0.2	5	1	0
33	2	5	0.4	10	4	230.31
34	3	5	0.4	15	6	551.32
35	3	5	0.2	15	3	185.36
36	3	5	0.2	15	3	291.57
37	1	5	0.4	5	2	1.52E-09
38	11	5	0.3	55	18	599.92
39	5	5	0.2	25	5	409.96
40	2	5	0.2	10	2	224.15
41	2	5	0.4	10	4	62.14
42	1	5	0.8	5	4	0
43	1	5	0.2	5	1	0
44	4	5	0.4	20	8	477.21
45	3	5	0.4	15	6	79.91
46	2	5	0.2	10	2	97.83
47	2	5	0.2	10	2	251.75
48	1	5	0.4	5	2	0
TOTAL	139			695	203	9920.10
AVERAGE				14.18	4.14	202.45
DEVIATION				11.70	3.54	195.23

LUKES

Table 40. Results of LUKES W=10

Cluster	Points	Pop	Power	Pop density	Power per capita
0	5	25	8	5	0.32
1	8	40	8	5	0.20
2	10	50	10	5	0.20
3	2	10	3	5	0.30
4	7	35	10	5	0.29
5	2	10	8	5	0.80
6	1	5	4	5	0.80
7	5	25	9	5	0.36
8	6	30	9	5	0.30
9	3	15	6	5	0.40
10	5	25	9	5	0.36
11	3	15	4	5	0.27
12	4	20	7	5	0.35
13	8	40	8	5	0.20
14	6	30	6	5	0.20
15	7	35	10	5	0.29
16	5	25	10	5	0.40
17	6	30	10	5	0.33
18	1	5	2	5	0.40
19	1	5	1	5	0.20
20	10	50	10	5	0.20
21	3	15	6	5	0.40
22	5	25	10	5	0.40
23	2	10	4	5	0.40
24	3	15	10	5	0.67
25	3	15	3	5	0.20
26	8	40	8	5	0.20
27	10	50	10	5	0.20
TOTAL	139	695	203		
AVERAGE		24.82	7.25		
DEVIATION		13.91	2.85		

Table 41. Results of LUKES W=20

Cluster	Points	Pop	Power	Pop density	Power per capita
0	7	35	10	5	0.286
1	14	70	14	5	0.200
2	12	60	20	5	0.333
3	9	45	15	5	0.333
4	12	60	19	5	0.317
5	3	15	12	5	0.800
6	12	60	13	5	0.217
7	13	65	16	5	0.246
8	10	50	20	5	0.400
9	6	30	13	5	0.433
10	18	90	18	5	0.200
11	11	55	20	5	0.364
12	1	5	2	5	0.400
13	11	55	11	5	0.200
TOTAL	139	695	203		
AVERAGE		49.64	14.50		
DEVIATION		22.31	5.00		

Table 42. Results of LUKES W=50

Cluster	Points	Pop	Power	Pop density	Power per capita
0	30	150	43	5	0.287
1	35	175	49	5	0.280
2	27	135	44	5	0.326
3	26	130	46	5	0.354
4	21	105	21	5	0.200
TOTAL	139	695	203		
AVERAGE		139.00	40.60		
DEVIATION		25.84	11.19		

Bibliography

[1] UNDG, "Background of the Sustainable Development Goals." [Online]. Available: https://www.undp.org/content/undp/en/home/sustainable-development-goals/background.html. [Accessed: 29-Mar-2020].

[2] H. Chandra Garg, M. & Joshi, "A Review on PV-RO Process: Solution to Drinking Water Scarcity due to High Salinity in Non-Electrified Rural Areas," *Sep. Sci. Technol.*, vol. 50, no. 8, pp. 1270–1283, 2015.

[3] Energypedia.info, "Rural Electrification Planning." [Online]. Available: https://energypedia.info/wiki/Rural_Electrification_Planning.

[4] M. Torero, "The impact of rural electrification: Challenges and ways forward," *Rev. Econ. Dev.*, vol. 23, no. HS, pp. 49–75, 2016.

[5] P. W. India, "Rural Electrification: Challenges and the way ahead," *energy Resour. Inst.*, 2015.

[6] Sarjiya, H. R. Ali, and R. B. A. Pardede, "Application of genetic algorithm for optimal sizing and placement of distribution transformers in PT PLN East Medan Indonesia," *AIP Conf. Proc.*, vol. 1755, 2016.

[7] M. R. Haghifam, M. Esmaeeli, A. Kazemi, and H. Shayanfar, "Optimal placement of the distribution substations to improve reliability under load growth," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2015-Septe, pp. 1–5, 2015.

[8] EURELECTRIC, "Power Statistics and Trends," 2013.

[9] JRC TECHNICAL REPORTS, *DISTRIBUTION SYSTEM OPERATORS From European Electricity Distribution Systems to*. 2016.

[10] G. Prettico, M. G. Flammini, N. Andreadou, S. Vitiello, G. Fulli, and M. Masera, *JRC Science for Policy Report; Distribution System Operators Observatory 2018*. 2019.

[11] Z. Ezor, "Power to the People: Rural Electrification in Uganda," *ISP Collect.*, p. 35, 2009.

[12] T. W. Bank, "Rural population (% of total population) | Data." [Online]. Available: https://data.worldbank.org/indicator/sp.rur.totl.zs. [Accessed: 30-Mar-2020].

[13] "Energy GIS Working Group." .

[14] F. Businge, J., Nataba, A., Atuhaire, A. and Nassaka, *UMEME Power. Transforming Uganda*. The Independent Publications Limited, 2018.

[15] USAID, "Mozambique. Power Africa Fact Sheet," 2019. [Online]. Available: https://www.usaid.gov/powerafrica/mozambique.

[16] GET.invest, "Mozambique. Energy Sector," 2018. [Online]. Available:

https://www.get-invest.eu/market-information/mozambique/energy-sector/.

[17] "Electricidade de Moçambique." [Online]. Available: https://www.edm.co.mz/en.

[18] A. D. Fund, "MOZAMBIQUE. ELECTRICITY II PROJECT," 2005.

[19] R. Amatya *et al.*, "Computer-aided electrification planning in developing countries : The Reference Electrification Model (REM)," *Univers. Energy Access Lab*, pp. 1–111, 2018.

[20] C. Mateo Domingo, T. Gómez San Román, Á. Sánchez-Miralles, J. P. Peco González, and A. Candela Martínez, "A reference network model for large-scale distribution planning with automatic street map generation," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 190–197, 2011.

[21] F. Kemausuor, E. Adkins, I. Adu-Poku, A. Brew-Hammond, and V. Modi, "Electrification planning using Network Planner tool: The case of Ghana," *Energy Sustain. Dev.*, vol. 19, no. 1, pp. 92–101, 2014.

[22] S. Watchueng, R. Jacob, and A. Frandji, "Planning tools and methodologies for rural electrification," *Rep. From Http//Www.Club-Er.Org/*, no. December, p. 56, 2010.

[23] R. Fronius, "Rural electrication planning software (LAPER)," pp. v5-20-v5-20, 2005.

[24] T. Edeme, D. and Carnovali, "GISele: an innovative GIS-based approach for electric networks routines.," Politecnico di Milano, 2019.

[25] S. K. Khator and L. C. Leung, "Power distribution planning: A review of models and issues," *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1151–1159, 1997.

[26] United States Department of Agriculture, "Design guide for rural substations," *Dep. Agric.*, vol. 4, no. June, p. 567, 2001.

[27] L. Yu *et al.*, "An efficient substation placement and sizing strategy based on GIS using semi-supervised learning," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 371–379, 2018.

[28] S. Najafi and R. Gholizadeh, "On optimal sizing, siting and timing of distribution substations," *EPDC 2013 - 18th Electr. Power Distrib. Netw. Conf.*, pp. 1–6, 2013.

[29] G. C. Cabrera-celi and P. F. Vasquez-miranda, "using Clustering and Shortest Path Algorithms," 2017.

[30] I. J. Hasan, C. K. Gan, M. Shamshiri, M. Ruddin, A. Ghani, and R. Bin Omar, "Optimum Feeder Routing and Distribution Substation Placement and Sizing using PSO and MST," *Indian J. Sci. Technol.*, vol. 7, no. 0974–5645, pp. 1682–1689, 2014.

[31] S. Wang, Z. Lu, S. Ge, and C. Wang, "An improved substation locating and sizing method based on the weighted voronoi diagram and the transportation model," *J. Appl. Math.*, vol. 2014, 2014.

[32] D. Xu and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, 2015.

[33] S. J. Swarndepp and S. Pandya, “An Overview of Partitioning Algorithms in Clustering Techniques,” *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 5, no. 6, pp. 1943–1946, 2016.

[34] H. Khanali and B. Vaziri, “A Survey on Clustering Algorithms for Partitioning Method,” *Int. J. Comput. Appl.*, vol. 155, no. 4, pp. 20–25, 2016.

[35] B. Jiang, J. Pei, Y. Tao, and X. Lin, “CLUSTERING UNCERTAIN DATA BASED ON PROBABILITY DISTRIBUTION SIMILARITY 1 Clustering Uncertain Data Based on Probability Distribution Similarity,” pp. 1–14, 2011.

[36] X. Xu, M. Ester, H. Kriegel, and J. Sander, “Published in the Proceedings of 14th International Conference on Data Engineering (ICDE ’ 98) A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases D-80538 München 3 . A Notion of Clusters Based on the Distance Distribution,” pp. 324–331, 1998.

[37] E. Hartuv and R. Shamir, “Clustering algorithm based on graph connectivity,” *Inf. Process. Lett.*, vol. 76, no. 4–6, pp. 175–181, 2000.

[38] M. Maier, U. Von Luxburg, and M. Hein, “Influence of graph construction on graph-based clustering measures,” *Adv. Neural Inf. Process. Syst. 21 - Proc. 2008 Conf.*, pp. 1025–1032, 2009.

[39] Y. Chen, “Fractal Modeling and Fractal Dimension Description of Urban Morphology,” pp. 1–28.

[40] P. Castillo, O. and Melin, “Hybrid Intelligent Systems for Time Series Prediction Using Neural Networks, Fuzzy Logic, and Fractal Theory.,” *IEEE Trans. Neural Networks.*, vol. 13, no. 6, pp. 1395–1408, 2002.

[41] T. Alqurashi and W. Wang, “Clustering ensemble method,” *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 6, pp. 1227–1246, 2019.

[42] B. Wu and Z. Shi, “A clustering algorithm based on swarm intelligence,” *2001 Int. Conf. Info-Tech Info-Net A Key to Better Life, ICII 2001 - Proc.*, vol. 3, pp. 58–66, 2001.

[43] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science (80-.).*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[44] M. Rouse, “Dashboard development and data visualization tools for effective BI.,” *Teach Target*, 2013.

[45] Techopedia, “What is Large Scale Data Analysis?” [Online]. Available: https://www.techopedia.com/definition/28987/large-scale-data-analysis.

[46] M. Parimala, D. Lopez, and N. C. Senthilkumar, “A Survey on Density Based Clustering Algorithms for Mining Large Spatial Databases,” vol. 31, pp. 59–66, 2011.

[47] E. Kolatch, “Clustering Algorithms for Spatial Databases: A Survey,” pp. 1–22, 2001.

[48] U. A. P. J. Bindiya M Varghese, “Spatial Clustering Algorithms - an Overview,” *Asian J. Comput. Sci. Inf. Technol.*, vol. 3, no. 1, 2013.

[49] L. Shah, H., Napanda, K. and D’mello, “Density Based Clustering Algorithms.,” *Int. J. Comput. Sci. Eng.*, vol. 3, no. 11, pp. 54–57, 2015.

[50] E. Schubert, “DBSCAN and OPTICS clustering,” *Vitavonni Blog*, 2012. [Online]. Available: https://www.vitavonni.de/blog/201211/2012110201-dbscan-and-optics-clustering.html.

[51] B. Klein, “Graphs in Pyhton,” *Python Advanced Course Topics.*, 2020. [Online]. Available: https://www.python-course.eu/graphs_python.php.

[52] J. A. Lukes, “Efficient Algorithm for the Partitioning of Trees.,” *IBM J. Res. Dev.*, vol. 18, no. 3, pp. 217–224, 1974.

[53] S. Saini and P. Rani, “A Survey on STING and CLIQUE Grid Based Clustering Methods,” *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 2015–2017, 2017.

[54] A. Topchy, A. K. Jain, and W. Punch, “Clustering ensembles: Models of consensus and weak partitions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, 2005.

[55] “GINI index (World Bank estimate) | Data.” [Online]. Available: https://data.worldbank.org/indicator/si.pov.gini. [Accessed: 31-Mar-2020].

[56] “Worldometer - real time world statistics,” 2019. .

[57] F. Molina Ortiz, “Balance Energético Nacional 2000 - 2010,” 2011.

[58] E. Corporación, “Memoria Anual 2018.”

[59] G. P. VMEEA, “Mapa del Sistema Eléctrico de Bolivia.” [Online]. Available: http://sigvmeea.minenergias.gob.bo/maps/60/view.