ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

GENERATION OF RANDOM ELECTRICITY TRANSMISSION NETWORKS

Autor: Rafael Espejo González
Director: Sara Lumbreras Sancho
Andrés Ramos Galán

Madrid
Mayo 2016

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Rafael Espejo González DECLARA ser el titular de los derechos de propiedad intelectual de la obra: 'Generation of Random Electricity Transmission Networks', que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente).*

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a treinta de mayo de dos mil dieciséis.

ACEPTA

Fdo. Rafael Espejo González.

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

GENERATION OF RANDOM ELECTRICITY TRANSMISSION NETWORKS

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2015/2016 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Rafael Espejo González Fecha: 30/05/2016

Autorizada la entrega del proyecto

LOS DIRECTORES DEL PROYECTO

Fdo.: Sara Lumbreras Sancho Fecha: 30/05/2016

Fdo.: Andrés Ramos Galán Fecha: 30/05/2016

Vº Bº del Coordinador de Proyectos

Fdo.: Jaime de Rábago Marín Fecha: 30/0572016

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

GENERATION OF RANDOM ELECTRICITY TRANSMISSION NETWORKS

Autor: Rafael Espejo González
Director: Sara Lumbreras Sancho
Andrés Ramos Galán

Madrid
Mayo 2016

GENERACIÓN ALEATORIA DE REDES ELÉCTRICAS DE TRANSPORTE

Autor: Espejo González, Rafael.

Director: Lumbreras Sancho, Sara.

Ramos Galán, Andrés.

Entidad Colaboradora: IIT – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

1. INTRODUCCIÓN

El problema de la expansión de la red eléctrica de transporte (**TEP** por sus siglas en inglés), determina las nuevas líneas eléctricas que hay que añadir al sistema existente y cuándo. Con la instalación de nuevas líneas se busca poder satisfacer el incremento de demanda estimado para el futuro a la vez que se minimiza la suma del coste de operación y de inversión del sistema. Se trata de un problema combinatorio estocástico cuya complejidad se ha incrementado con la liberalización del mercado eléctrico y la introducción de plantas de producción de energía renovable.

Para resolver el TEP se han utilizado tanto métodos de resolución clásicos, basados en programación matemática, como no clásicos, destacándose los metaheurísticos. Entre estos últimos destacan los **Algoritmos Genéticos** (GA). El GA se presenta como una herramienta que permite obtener soluciones de alta calidad con un bajo coste computacional. Los recursos computacionales requeridos en este tipo de problemas suponen la principal restricción a la hora de seleccionar el algoritmo de resolución. Esta es la principal desventaja que presenta la **descomposición de Benders** (BD), la cual permite obtener resultados con una garantía de optimalidad. Ambos algoritmos se encuentran implantados para su utilización en el modelo TEPES, desarrollado en el IIT.

Una vez que la efectividad del GA y BD ha sido probada, se busca determinar cuándo es mejor utilizar un algoritmo u otro. Esta decisión está basada en coste computacional y bondad de la solución obtenida.

La referencia (Smith-Miles, Baatar, Wreford, & Lewis, 2014) propone un modelo teórico que permite discernir qué algoritmo utilizar para resolver un problema de optimización, basándose en que la bondad de un algoritmo está relacionada con las características que definen el problema. Para aplicar esta metodología a TEP, es necesario trabajar con un número de instancias elevado, es decir, con muchos casos particulares, para poder extraer conclusiones generales. Al existir pocos sistemas eléctricos con sus datos disponibles, es necesario generar instancias artificialmente, esto es, **redes eléctricas de transporte aleatorias** cuyas características sean similares a posibles redes reales existentes.

El problema de la generación aleatoria de redes eléctricas no ha sido abordado hasta ahora. Las leyes físicas que rigen este tipo de redes hacen que los modelos de generación usados en redes sociales u otro tipo de redes complejas no se puedan utilizar para los sistemas eléctricos. Además, las redes eléctricas son redes espacialmente embebidas, esto significa que la disposición geográfica de sus elementos es esencial para comprender el funcionamiento del sistema, lo cual plantea problemas adicionales. Los algoritmos utilizados en redes sociales no consideran tampoco esa característica.

Este proyecto ha partido de un modelo para redes espacialmente embebidas propuesto inicialmente en la referencia de sus directores (Patania et al. 2015), denominado modelo **Epsilon-disc.**

Este modelo epsilon-disc todavía presenta alguna deficiencia para obtener resultados similares a las redes eléctricas reales. Por ejemplo, no permite ajustarse a las distribuciones estadísticas de grado de conectividad que describen a estas redes. El modelo tampoco tiene en cuenta, de momento, la información económica del problema, relacionada con el coste variable de generación o fijo de desarrollo de la red.

A la vista de estos inconvenientes, el principal objetivo del proyecto se centra en el desarrollo de un nuevo modelo que sea capaz de generar redes eléctricas de transporte aleatorias que emulen a las redes reales solventando los inconvenientes previos.

2. METODOLOGÍA

El nuevo modelo, denominado **trade-off model**, consta de dos pasos: **casación eficiente de la demanda** (1) y **ajuste de la distribución de grado** (2).

En el primer paso, el modelo representa las etapas iniciales de desarrollo de un red eléctrica donde cada nodo de demanda busca la central de generación capaz de suministrar su energía demandada al mínimo precio.

El coste de suministro tiene en cuenta, el coste de instalación de la línea y el coste de la energía suministrada. El coste de instalación de una línea varía linealmente con la distancia entre nodos. El modelo elige la opción más barata entre conexión directa entre nodos o la conexión de éste a la red existente. Cuando dos nodos no son conectados directamente, se puede necesitar más de una línea para satisfacer la demanda.

El método busca el camino más corto entre los nodos. Al no estar inicialmente los nodos conectados entre ellos, es necesario definir un grafo auxiliar en el que los nodos estén unidos con posibles conexiones. En este grafo, el peso de los arcos será la distancia entre dos nodos.

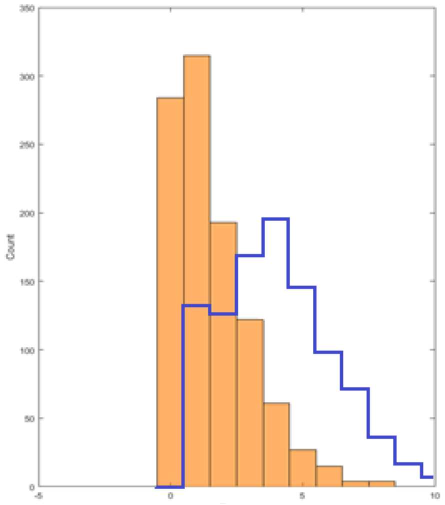

Figura-1. Distribución de grado deseada y distribución obtenida tras la primera etapa del trade-off.

Para evitar un alto coste computacional, se define un grafo donde la probabilidad de que dos nodos estén conectados está inversamente relacionada con la distancia entre ellos. Para ello se utiliza el modelo Epsilon-disc.

El segundo paso se basa en la instalación de nuevas líneas con el objetivo de conseguir una función de distribución de grado definida.

El modelo analiza las diferentes uniones entre nodos para elegir qué líneas se añaden al sistema. Se establecen limitaciones relacionadas con longitudes de líneas máximas para distintas tecnologías.

Durante su desarrollo, el algoritmo analiza el grado (número de conexiones) de los nodos candidatos a ser unidos, viendo si la unión de éstos contribuye a alcanzar la distribución de grado objetivo (la que se toma como modelo de una red real). En caso de que la unión facilite alcanzar esa distribución, será asignada una probabilidad de unión estrictamente positiva. Si la instalación de esa línea no contribuye a alcanzar la función objetivo, la probabilidad de unir esos dos nudos será cero.

Al tratarse de redes espacialmente embebidas, se asume que la probabilidad de unir dos nodos está directamente relacionada con la longitud de la línea. Al considerar distancias se está tomando de manera implícita el coste de instalación de la misma.

Si tanto la probabilidad de instalar una línea como la de que exista una línea de esa longitud son positivas, el algoritmo decidirá la instalación aleatoriamente.

3. RESULTADOS

Para comprobar la validez del modelo se generará una red aleatoria que se comparará posteriormente con una red eléctrica real. La red real es la resultante de la unión de las redes de 14 países europeos. Las características que se evalúan son: grado, centralidad, longitudes de línea, camino más corto y coeficiente de apuntamiento.

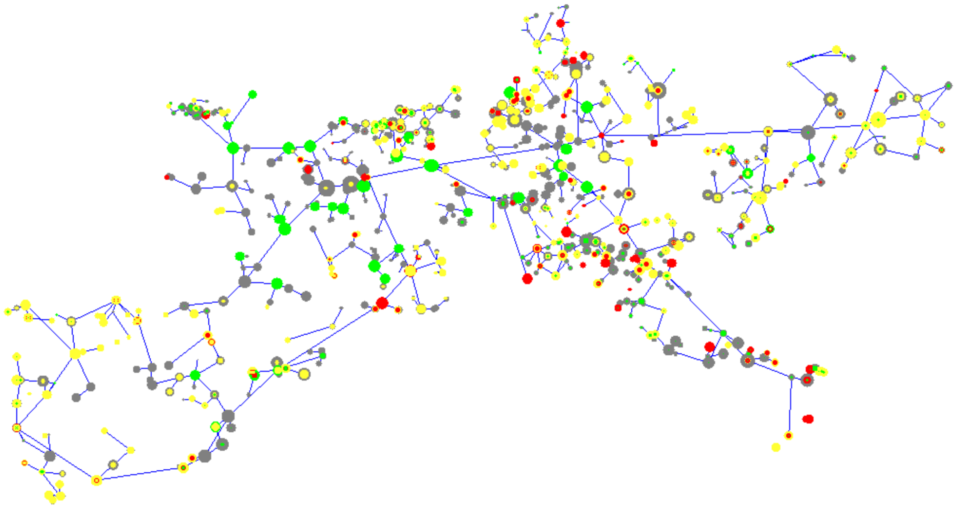

Figura-2. Ejemplo de red generada en la primera etapa del modelo trade-off.

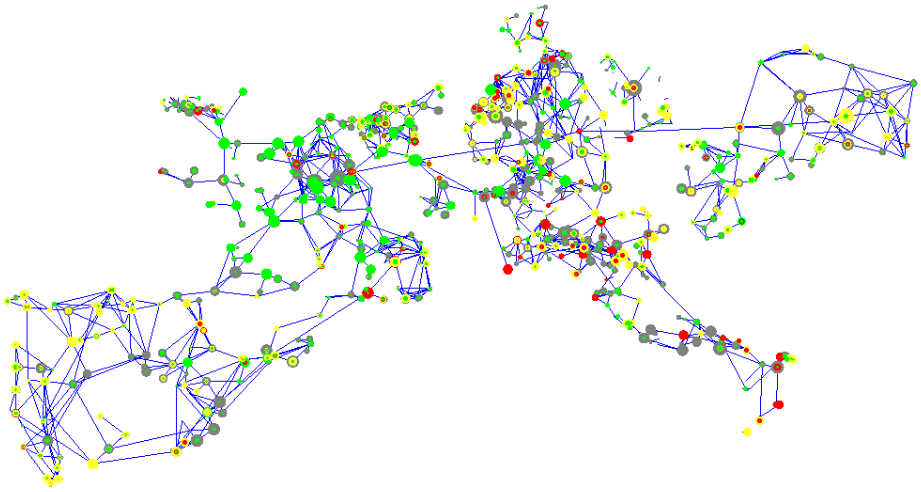

Figura-3. Ejemplo de red generada tras la segunda etapa del modelo trade-off.

La figura 4 muestra la distribución de grado para la red real y artificial. Ambas distribuciones presentan un grado de similitud elevado.

En el caso de la centralidad, ambas distribuciones son similares, en ambos casos el valor más frecuente es cero. Esto refleja que la mayor parte de los nodos no se encuentran camino más corto entre otros dos.

Si se compara la longitud de líneas, se observa que ambas distribuciones son muy similares. En este caso tanto la media como la desviación típica tienen valores similares.

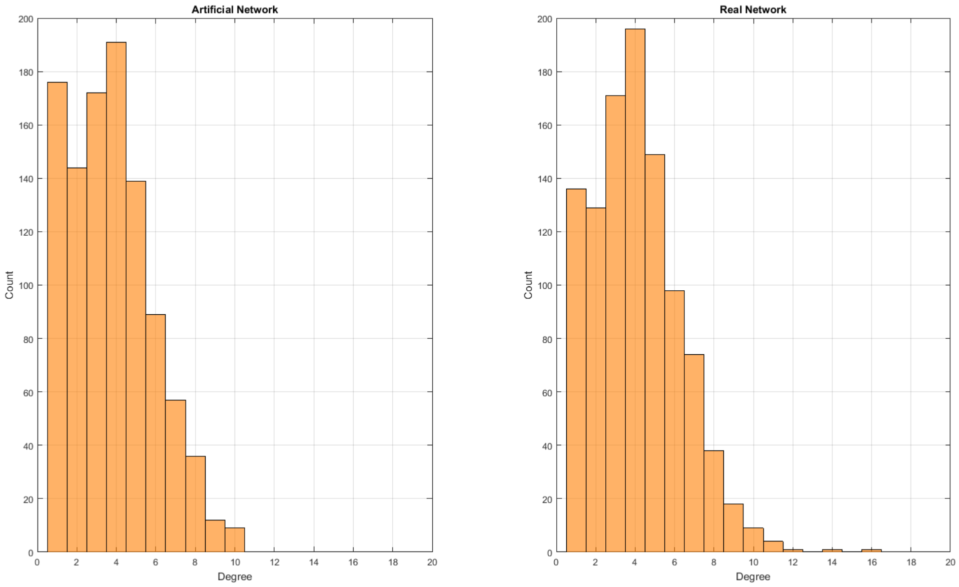

Figura-4. Distribución de grado de la red artificial (izquierda) y la red real (derecha)

Se puede observar que en la red artificial las distancias entre nodos son menores que en la red real.

El coeficiente de agrupamiento es la medida estadística que más difiere. Su valor medio es el doble en el caso de la red real. Los vecinos de un nodo están más unidos entre ellos.

CONCLUSIONES

El Trade-off Model se basa en un modelo de unión preferente que permite ajustar la distribución de grado a una función definida. Para ello añade líneas en el sistema de manera que los nodos de bajo grado vean incrementado su grado, mientras que los de grado elevado ya no ven añadidas nuevas líneas.

La primera conclusión es que el modelo se puede utilizar para crear redes que sean similares a las existentes. Las distribuciones de grado son similares. Ambas redes son muy parecidas desde el punto de vista de centralidad, longitud de líneas y camino más corto de la red. La mayor diferencia se encuentra en el coeficiente de agrupamiento. Se considera que esta diferencia es debida a la forma en que la red está mallada. Si se observa la red real, no se puede encontrar un patrón seguido durante el mallado del sistema. Dependiendo de si la red está mallada uniformemente, o sólo se encuentran malladas partes concretas de la misma, el coeficiente de agrupamiento será similar o no. Esto puede justificar el hecho de la pequeña diferencia en la distribución de caminos más cortos.

El modelo también puede generar redes con diferentes propiedades dependiendo de la zona. En el caso estudiado, la red artificial ha sido la unión de las 14 redes nacionales que forman la red real. Se pueden generar redes aleatorias completamente independientes que se pueden unir posteriormente. De esta manera se puede solventar el problema de redes en el que el mallado no se encuentra distribuido uniformemente.

Este proyecto no trata de generar redes que sean exactamente iguales a las redes reales. Se crean redes artificiales cuyos descriptores estadísticos son similares a los de redes reales. En este proyecto sólo se han considerado las características más relevantes de una red, pudiéndose utilizar para generar redes aleatorias que permitan su uso en el estudio del problema de expansión de la red eléctrica de transporte.

4. REFERENCIAS

Smith-Miles, K., Baatar, D., Wreford, B., & Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research, 45*, 12-24.

Patania, A., Young, J., Lumbreras, S., Pereda, M., Bertazzi, I., Citron, D., & Haraguchi, M. (2015) Part I: Generating random networks that are consistent with power transmission.

GENERATION OF RANDOM ELECTRICITY TRANSMISSION NETWORKS

1. INTRODUCTION

Transmission Expansion Planning (**TEP**) tries to solve the problem of what lines should be added to the existing network and when, to satisfy forecasted demand in the long-term and minimizing operation and investment costs. This is a combinatorial stochastic problem that has become more complicated due to market liberalizations and the integration of renewable energy sources.

As for other optimization problems, to solve the TEP problem classical and non-classical methods have been applied. Researchers find in metaheuristic methods, as **Genetic Algorithm** (GA), an attractive tool to generate solutions to the TEP problem. They give high-quality solutions with lower computational resources. Computational requirements are the main problem for TEP due to the size of the instances, where a huge number of variables and constraints are considered. This is also the main weakness of **Benders' Decomposition** (BD) in some cases, while this algorithm is able to provide solutions with any desired level of accuracy. Both algorithms are implemented in the model TEPES developed by IIT.

Once the effectiveness of GA and BD for the TEP problem have been tested; it would be interesting to find in which instances it is better to use one algorithm or another. It is based on computational cost and solution goodness.

Reference (Smith-Miles, Baatar, Wreford, & Lewis, 2014) proposes a framework to evaluate different algorithms for optimization problems. The theoretical framework is based on the concept that the *hardness* of an algorithm is related to the features of the problem. To apply it to the TEP problem, it is necessary to work with a high number of instances, real and random ones. There is a necessity to generate **random transmission networks** that should be consistent with the real ones.

The generation of random networks has not been studied in depth in the case of power grids. The physical laws that govern power networks imply that the network-generation techniques used in social or other types of networks cannot be used in this context. Furthermore, power grids are a spatially embedded network. This means that the geographical placement of its elements is essential to understand the behaviour of the system. This makes it more difficult to apply already existing techniques that do not take this characteristic into account.

This work has implemented a state-of-the-art method for generating spatially embedded networks applied to the power grid (Patania et al. 2015), the **Epsilon-disc model**, proposed by the supervisors of this project.

This project has identified the flaws of the Epsilon-disc model in terms of not fitting the descriptive distributions of the network and not taking into account economic information such as generation cost in the development of the network.

The main objective of this project is to develop a new model that will be able to generate random power systems that emulate real ones.

2. METHODOLOGY

The new model, which has been named **Trade-off** model, can be described in two steps: **meeting demand efficiently** (1) and **fitting a degree distribution** (2).

The first step of the algorithm simulates a trade-off between demand (customers) and generators (suppliers). Each sink node will try to find the generator that is able to supply their energy needs at the lowest cost.

The cost of linking two nodes considers the cost of line installation and the cost of generation. The cost associated to installing a line is linear with the length of the line, modelled as the Euclidean distance between two nodes. The model chooses the lowest-cost option between direct connections between two nodes or connecting them through already existing lines. When nodes are not connected directly, more than one line could be required.

The algorithm will look for the shortest path between nodes. As nodes are not connected yet, it is necessary to define an imaginary graph. In this network, the weight of edges will be the geographical distance, if there is no line or '1' if there is a line connecting two nodes.

To avoid high computational requirements, the imaginary graph cannot be a network where all nodes are linked among them. It is necessary to define a graph where it is more probable to find a connection between closed nodes than between those which are located far away. This is performed with the epsilon-disc model.

The second step can be summarized as adding new lines smartly to obtain a defined distribution.

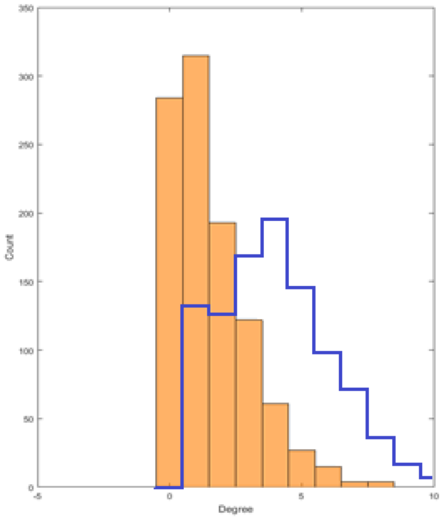

Figure-1. Target degree distribution and degree distribution after step 1 in the trade-off model.

The algorithm will analyse all the possible combinations of nodes to decide if a line is added to the system. Constraints related to distance can be introduced to avoid studying lines that are too long.

The process will analyse the current degree of the two candidate nodes to be linked. If the line contributes to reach the target distribution, it is assigned a positive probability of being installed. If that connection will not contribute to the fitting process, the probability of linking those nodes will be zero.

As power networks are spatially embedded, we assume that the probability of linking two nodes is correlated to the distance between them. We will condition the connection between two nodes not only to its contribution to the fitting process but also to a fitness function based on distance. By considering distance we are taking into account the cost of installing a line. The fitness function can be defined as the probability of installing a line of a specific length.

If a line contributes to fitting the degree distribution and has a positive probability to be installed based on length of line, the algorithm will decide whether to install it or not based on generating a random number.

3. RESULTS

A real case of the European network will be used to test the algorithm explained. The real network used is a combination of 14 European countries. It is evaluated: degree, betweenness centrality, length of lines, shortest-path and clustering coefficient.

Figure 4 shows the degree distribution for the real and artificial networks. The degree of accuracy is high, both histograms are quite similar. This similarity is reflected in the results that are shown in the table 1.

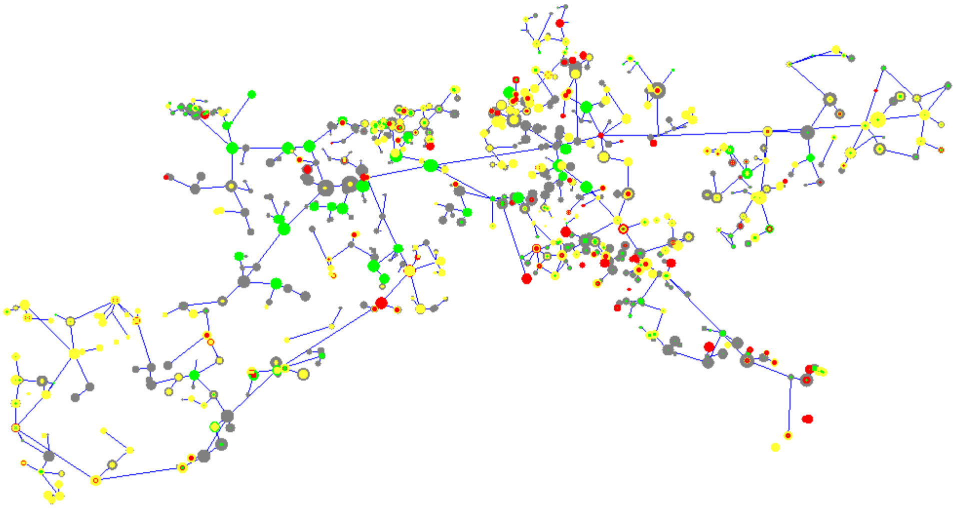

Figure-2. Example of network generated after first step.

In the case of betweenness centrality, in both cases the most frequent value is zero. Most part of the nodes are not in the shortest path between other two.

Comparing length of lines, we observe that the shape of both functions is really similar. In this case, mean and standard deviation have identical values.

If we compare the real and artificial networks we can observe that in the artificial one distances are shorter than in the real one. In the real network usually there are two edges more than in the artificial one in the shortest path length. Standard deviation is also lower for the artificial network.

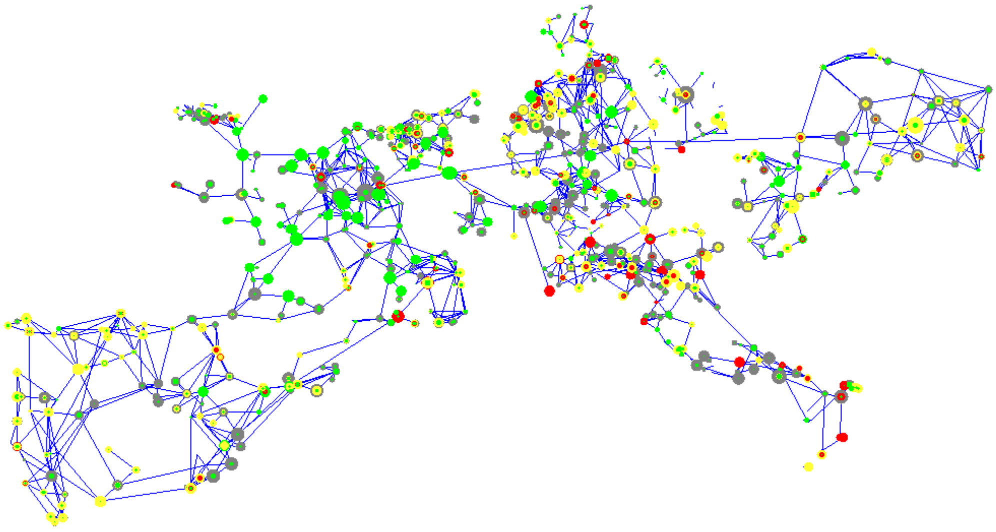

Figure-3. Example of network generated after second step.

The clustering coefficient is the most different metric among the ones we have analyzed. The mean value is the double in the real network. Neighbors of a node are more linked in that case.

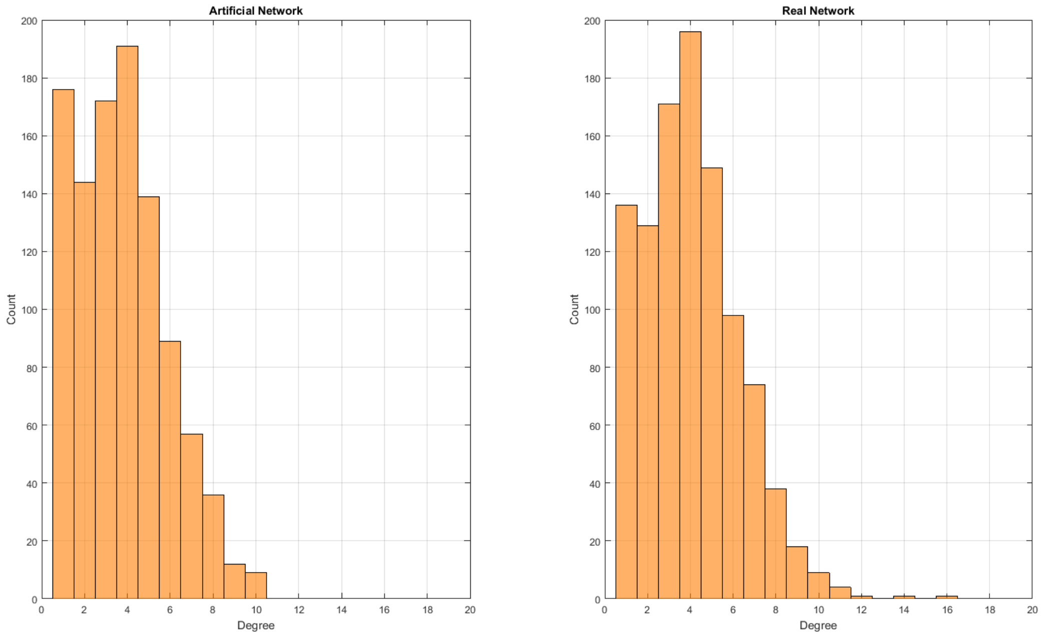

Figure-4. Degree distribution for artificial (left) and real (right) European networks.

4. CONCLUSIONS

The Trade-off Model is based on a preferential attachment model that effectively shifts low-degree nodes towards the high degree region by adding edges smartly .The results proved that the algorithm is able to fit a general degree distribution that is taken from input data.

Our first conclusion is that the algorithm can be used to create networks that are similar to existing ones. Degree distributions are almost identical. They are also similar from the point of view of betweenness centrality, length of lines and shortest path length. There was a larger difference in the case of the clustering coefficient. We think that this difference is due to the way the grid is meshed. If we see the real network there is not a pattern in the mesh of the network. Depending on whether the mesh is spread across the space or is concentrated in specific areas, we will have more or less similar networks. This can justify the fact that in the artificial network the shortest path length between nodes is lower.

We also concluded that we can create networks that have different properties depending on the area. We applied this to Europe, where the network is a set of 14 small different networks. We can create completely independent random networks that can be linked, resulting a network with clear differenced areas. This could be a way to avoid the problem of some networks in which the degree of meshness depends on the zone.

It is clear that we are not going to create networks that are exactly equal to the one that we are using as a template. That is not the objective of the project. We try to create artificial networks that are consistent with the real ones. There is not only one pattern that have been follow in the generation of the existing network. There are a lot of properties that should be considered. We have only considered the ones we think have more relevant in the process of network design. They are good to recreate existing ones, and to be used in studies about the expansion of existing networks.

5. REFERENCES

Smith-Miles, K., Baatar, D., Wreford, B., & Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research, 45*, 12-24.

Patania, A., Young, J., Lumbreras, S., Pereda, M., Bertazzi, I., Citron, D., & Haraguchi, M. (2015). Part I: Generating random networks that are consistent with power transmission.

Contents

Figures

Tables

1. INTRODUCTION

Author: Chad Lewis

In late 1880s and early 1890s, the War of Currents between Edison and Westinghouse (who bought Tesla's patents) made the beginning of power transmission systems a rocky start.

In 1882 Edison established the first power generation plant in Manhattan, New York, which was able to supply energy for 400 incandescent lamps. This was a 110 V direct-current (DC) power system. With such a low voltage, generation plants had to be close to demand because of power losses.

With the first alternating current (AC) transformer in Europe, Westinghouse noticed that large power plants could be built far from the demand using AC and transformers. With this combination, losses could be reduced. The first AC power system was installed in Massachusetts in 1886.

DC and AC power systems were installed in different cities in North America while both businessmen were involved in an embarrassing struggle. Topsy, an elephant, was electrocuted as Harold Brown (one of Edison's employees) tried to demonstrate the risks of AC. In 1893 at the Chicago World Fair, Westinghouse developed an AC grid with 100,000 electric lights. This system, cheaper than Edison's DC proposal, led to the final AC supremacy we see today.

The power system has since then evolved from those small city grids to become the largest engineering system developed by humankind.

1.1. The Power System

In an electric system, generation plants are connected to demand through the transmission and distribution grids. The transmission system plays a key role to ensure the reliability and effectiveness of the electric system.

Spain has more than 42.000 km of high voltage lines[1] with a capacity of 84.539 MVA. It is connected to France, Portugal and Morocco, being part of an international network. Although almost the entire network is built with AC technology, one of the connections with France is DC and other to the Balearic Islands. Transmission System Operators (TSO) are installing DC technology for long distances using high-voltage connections (HVDC). This technology requires HVDC converter stations to transform from AC to DC and back.

Power systems have to face new challenges as an increase in the degree of interconnection among countries or the integration of new renewable plants. These goals need new tools for Transmission Expansion Planning.

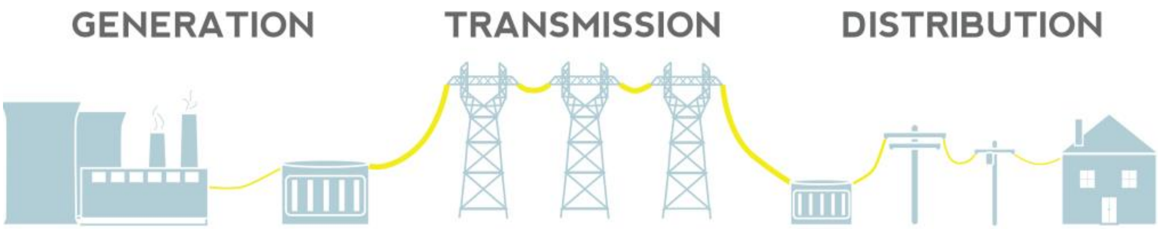

Figure 1-1. Power system structure. Source: Institute for Energy Research

1.2. Transmission Expansion Planning

Transmission Expansion Planning (TEP) tries to solve the problem of which lines should be added to the existing network and when, to satisfy forecasted demand in the long-term and minimizing operation and investment costs, including security of supply.

[1] Information available at Red Eléctrica de España, www.ree.es.

This is a combinatorial stochastic problem that has become even more complicated due to market liberalizations and integration of renewable energy resources, placed far from existing transmission lines.

'Desertec', 'MedGrid' or 'Beyond 2020' are examples of TEP projects. In these cases, the aim is to develop a large and well-connected network to encourage renewable generation and international exchanges.

New transmission lines are high-capital investments with long useful lives: the decision of which lines will be built will have an enormous impact on the whole system. For instance, the budget of ENTSO-e countries for the period 2012-2020 is over EUR 100 bn (Chaniotis, 2012). ENTSO-e is the European Network of Transmission System Operators. It represents 45 TSO of 35 countries in Europe, in the case of Spain the TSO is Red Eléctrica de España. The objective of ENTSO-e is to set-up a European energy market, supporting its functioning and helping to achieve the European climate and energy agenda[2].

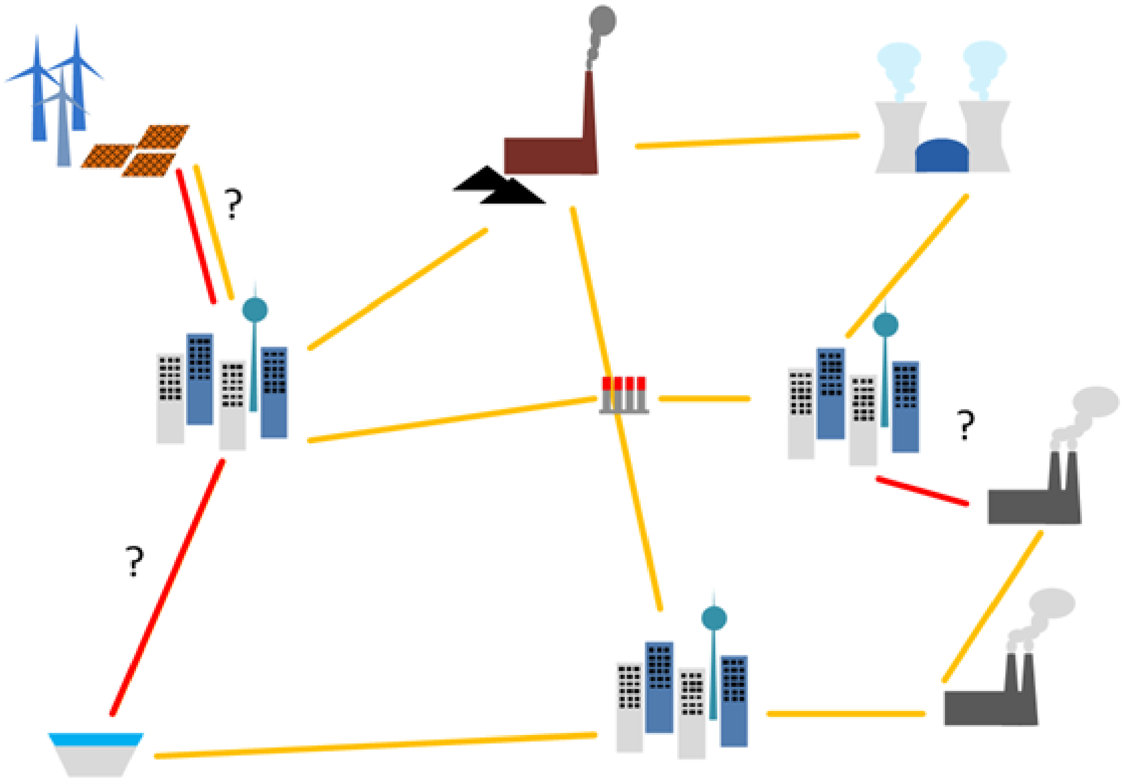

Figure 1-2. TEP PROBLEM Author: S. Lumbreras

As for other optimization problems, to solve the TEP optimization problem classical and non-classical methods have been applied. Figure 1-3 shows a classification of solution techniques.

Classical techniques have been applied with the TEP problem as Linear Programming (LP), although it assumes important simplifications its low computational

[2] Who is ENTSO-e, https://www.entsoe.eu/about-entso-e/

effort requirements makes it a practical tool to solve large systems. Linear Programming (LP) assumes a DC power flow model and ignores the discrete nature of investment variables. Quadratic Programming is used to take into account losses from the DC model. With Mixed-Integer Programming (MIP) the binary nature of investment decision variables is considered. Non Linear Programming (NLP) and Mixed-Integer Non Linear Programming (MINLP) are used in when modelling problems with AC power flow (Lumbreras, 2014).

When uncertainties are added to the decision process, decomposition techniques, as Benders Decomposition (BD), have an important role to solve the TEP problem (Lumbreras, 2012).

Researchers find in metaheuristic methods, as Genetic Algorithm (GA), an attractive tool to generate solutions to the TEP problem. They give high-quality solutions with lower computational resources. Computational requirements are the main problem for TEP due to the size of the instances, where a huge number of variables and constraints are considered.

<table>
<tr><th colspan="4">CLASSICAL METHDOS</th><th colspan="2">NON-CLASSICAL METHODS</th></tr>
<tr><td colspan="2">MATHEMATICAL PROGRAMMING</td><td rowspan="4">STOCHASTIC DECOMPOSITION</td><td rowspan="4">EQUILIBRIUM FORMULATION</td><td rowspan="3">SENSITIVITY ANALYSIS</td><td rowspan="4">METAHEURISTICS</td></tr>
<tr><td>LP</td><td rowspan="2">NLP</td></tr>
<tr><td>QP</td></tr>
<tr><td>MIP</td><td>MINLP</td><td>EXPERT SYSTEMS</td></tr>
</table>

Figure 1-3. Solution techniques for TEP problem.

1.2.1. *Benders' Decomposition*

Benders' Decomposition (BD) is one of the most applied algorithms in the Stochastic Optimization domain. Developed by the mathematician J.F. Benders, BD is a multistage optimization algorithm that divides the problem into multiple smaller problems, solving them iteratively. The first step is to solve the master problem giving a value to a subset of variables, the rest of variables are valued in the sub-problem. In a second stage, the sub-problem studies the optimality of the solution proposed by the master problem and the gives feedback to the master problem, adding new constraints (cuts). Figure 1-4 shows the flow chart followed in Benders' decomposition.

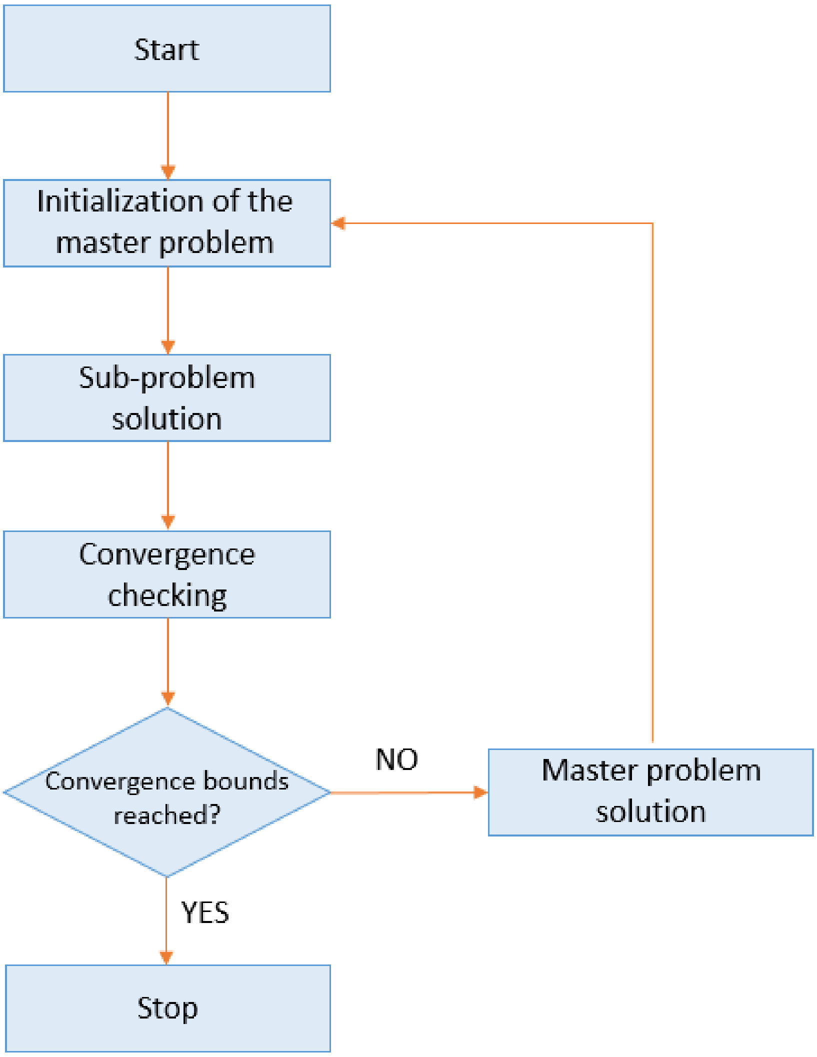

Figure 1-4. Benders' Decomposition Flow Chart

1.2.2. *Genetic Algorithms*

GA are evolutionary algorithms that generate solutions for an optimization problem based on natural evolution theory. It was proposed by J.H. Holland in the seventies.

The algorithm starts with an initial population of candidate solutions. Each candidate solution has a set of properties (chromosomes) which can be mutated and altered.

The evolution starts from a population of randomly generated individuals, and is an iterative process. In each generation the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem including some penalty term associated to the evaluation of the feasibility of the solution. The fittest individuals are selected from the current population, and each individual's genome is modified.

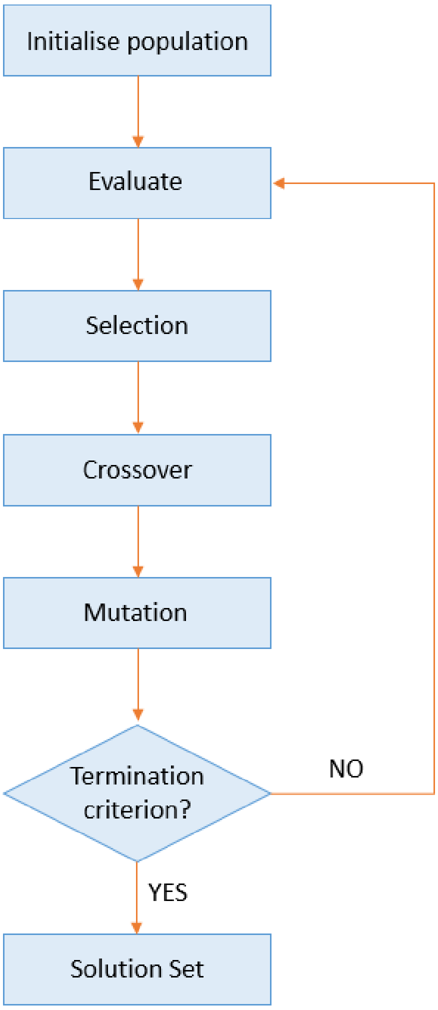

Figure 1-5. Genetic Algorithm flow chart.

The algorithm terminates when a satisfactory fitness level has been reached for the population or after a certain number of iterations.

The process that follow the algorithm is shown in the Figure 1-5.

1.2.3. *TEPES, Transmission Expansion Problem for an Energy System*

The Institute for Research in Technology of ICAI (IIT) developed a model to solve the TEP problem, TEPES (Transmission Expansion Problem for an Energy System), based on BD. This model evaluates the future network's needs in a tactical level, supporting the decision of future investments for the long term.

Based on a DC model flow, the objective function of the model is to minimize investment and operational costs. A cost related to reliability of the system is also taken into consideration. The master problem proposes investing in new lines and the sub-problem gives it feedback with the operation cost, and unserved power due to transmission contingencies. BD is the main algorithm used in classical method, nevertheless, depending on the size of the problem, the number of lines, the integrality conditions or the number of cuts added can make the resolution of the problem slow.

A Genetic Algorithm model was coded for TEPES in reference (Duro, 2014)

The TEPES software cannot decide which algorithm should choose depending on the instance that will be solved. The user decides when to use each one, thinking on computational time and quality of results.

1.2.4. *Choosing the best algorithm*

Once the effectiveness of GA and BD for the TEP problem have been tested; the minimization of time spent and computational resources required make necessary to find in which instances it is better to use one algorithm or another.

Reference (Smith-Miles, Baatar, Wreford, & Lewis, 2014) proposes a framework to evaluate different algorithms for optimization problems. It has been tested for the 'Traveling Salesman Problem' (TSP), where two variants of the Lin-Kernigan heuristic (proposed for the TSP problem more than thirty years ago) were tested. The theoretical framework is based on that the hardness of an algorithm is related to the features of the problem. It can be described in five steps.

Summarising Instances

Each instance, where the algorithm will be tested, has some properties that are correlated with the difficulty of solving the optimization problem with a specific algorithm. Not only generic properties like the number of variables or number of constraints, but also specific properties such as the number of candidate lines, number of sinks, or average distance between a pair of nodes in the case of TEP problem.

These properties can be represented in the instance space ($\mathbb{R}^n$). It is assumed that two instances that are closely located, are similar and will have a similar behaviour (except if they define a boundary between regions).

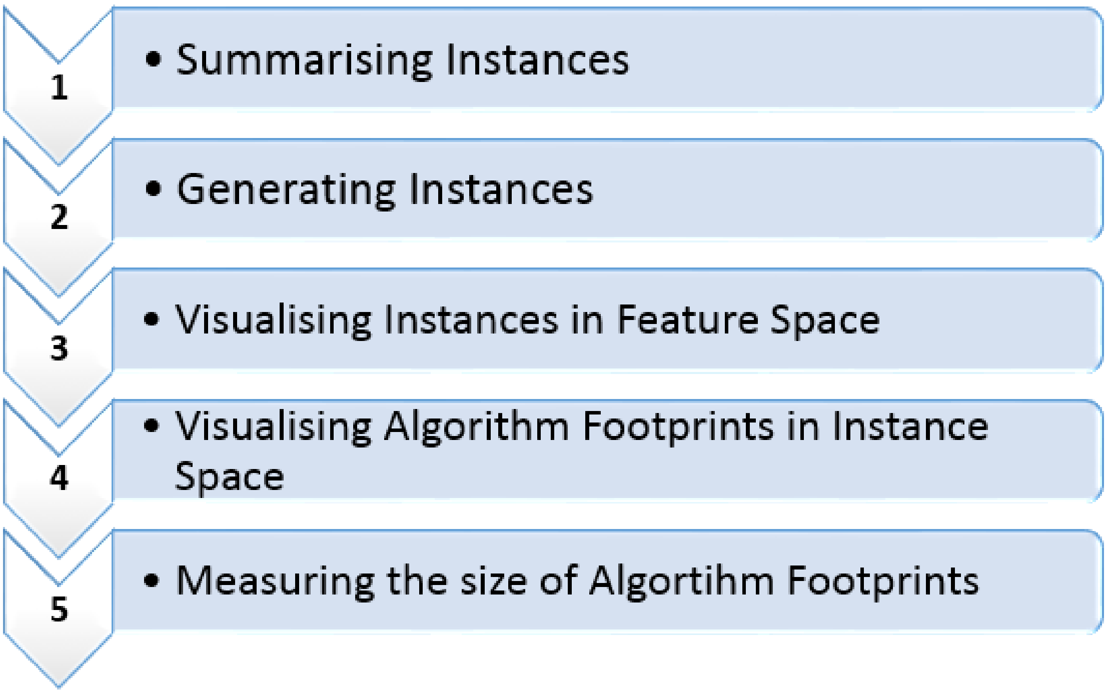
1
• Summarising Instances
2
• Generating Instances
3
• Visualising Instances in Feature Space
4
• Visualising Algorithm Footprints in Instance Space
5
• Measuring the size of Algortihm Footprints

Figure 1-6. Steps followed in classification of optimization algorithms.

Generating Instances

To assess the performance of both algorithms and to establish a relationship between the features of the instances and the future behaviour, a large number of instances is required. It is necessary that they will be well spread across the feature space, for statistical generalization.

The author proposes the idea of using evolutionary algorithms to evolve instances from the existing ones. In this way you can ensure that you will have instances that the algorithm will solve 'goodly' and 'badly'. The definition of goodness and badness used when an algorithm solved a case is defined below.

The number of real instances that are available in the case of power networks is very low. This is the case because there are very few power systems worldwide with publicly available data. Therefore, in order to apply the framework in (Smith-Miles, Baatar, Wreford, & Lewis, 2014), new instances should be generated.

The physical laws that govern the TEP problem make this step a critical one.

Visualising Instances in the Feature Space

Once there is a sufficient number of instances and their features have been described, they will be reduced from $\mathbb{R}^{n\cdot m}$ (m instances, n measurable features for each instance) to $\mathbb{R}^2$, or $\mathbb{R}^3$. Not all the measurable features have the same weight when an instance is defined. With this reduction a couple of new variables (combination of the most important ones) will describe each instance. This can be achieved using Principal Components Analysis (PCA). This will allow visualizing them in 2-D or 3-D space. With this reduction, the instances space is obtained.

An analysis of PCA can be also useful to understand better the TEP problem supporting other research projects.

Visualising the Algorithm Footprint in the Instance Space

To compare the performance of the algorithm, both methods will be evaluated for all the instances. First of all, it is necessary to define when algorithms have a good or bad performance, it will be based on solution time criterion and the accuracy of the algorithm. An algorithm will be good to solve an instance if the resolution time is inside a predefined optimality gap.

After running all candidate algorithms, instances will be labelled as good or bad, making it easy to visualize the result in the instance space.

Figure 1.-7 displays this step for the case of the TSP problem. As we said two algorithms were tested, A and B. In the figure we can identify four areas, these areas correspond to:

- Algorithm A is good and B is bad.
- Algorithm B is good and A is bad.
- Algorithms A and B are good.
- Algorithms A and B are bad.

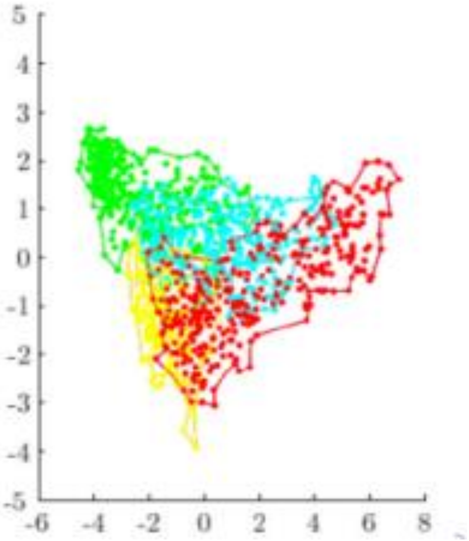

Figure 1-7. An example of two algorithm footprints Source: (Smith-Miles et al., 2014)

Measuring the size of Algorithm Footprints

In this step different regions will be defined in the instance space based on good and bad performance. Clustering classification allows us to establish boundaries in the instance space and develop a model that can evaluate which algorithm is better depending on the instance features. Each region will be related to a specific algorithm. The algorithm to solve a new instance will be chosen depending on where the new instance is located in this space.

In the previous figure it is easy to identify some areas where the decision of choosing an algorithm is clear (in the perimeter area). Nevertheless, in the centre the algorithm cannot provide an accurate answer, as boundaries of the clusters are not well defined. We could potentially define a new dimension to try to identify better limits between two regions.

It should be stressed that this framework has only recently been proposed for algorithms in general and therefore it has never been applied to TEP.

1.3. Objectives:

Lot of studies have been focused on TEP problem. While GA and BD's have been modelled and tested to solve this problem, it is not clear when they should be used. The necessity of using the most accurate algorithm make that to know when one is better than the other is the next step in the researching process. Computational requirements remarks the importance of that study. To answer the previous question we can use the theoretical framework proposed by K.S. Miles exposed above. To apply it to the TEP problem, it is necessary to work with a high number of instances, real and random ones. As we know the number of real networks is really low, we can assume there are no more than one network for country. Nevertheless if we are studying the network from an international point of view, as in the case of Europe, the number of real networks is reduced to a few of them. There is a need to generate new ones.

The generation of random network (that should be well spread across the feature space) has not been studied deeply and there are not good results. The physical laws that govern power networks, make that networks generation techniques used in social or other types of networks cannot be used for it. Furthermore power grids have a special consideration, they are embedded network. It means that the place where the elements of the grid are located are essential to understand the behaviour of the system.

The main objective of this project is to develop a model that will be able to generate random power systems which should emulate a real one. This model should be parametrical and should guarantee a certain degree of randomization for fixed inputs. For the same node location and features, the algorithm should be able to link them differently.

Once the algorithm is developed we will need to test if random networks could be used as real ones. A statistical approach will be used to know the degree of accuracy of the artificial networks.

Creating a huge number of instances we will be able to test them in the TEPES software, comparing results for the Genetic Algorithm and the Benders' Decomposition. This is not the scope of this project. Results can also be tested to know if the algorithm could be used in the generation of real network. Emerging countries are trying to find optimal power networks, nevertheless as in the TEP problem, the main problem are computational constraints.

2. NETWORK STUDY

A network or graph can be defined as a set of vertices (nodes) connected by edges (Figure 1-5). The internet or human societies are two particular instances of networks. In the case of the Internet, each computer can be seen as a node and edges represent data connections. Many objects of interest in the physical, biological, and social sciences can be thought of as networks. Using networks, the system is reduced to an abstract structure where the most relevant information is the pattern of connection between vertices. The pattern of connections among nodes will condition the behavior of the network. The transmission power system can be thought as a network.

Reference (Newman, 2010) is the best-known book about networks. In this section we will make a brief analysis of some network generation models based on this book.

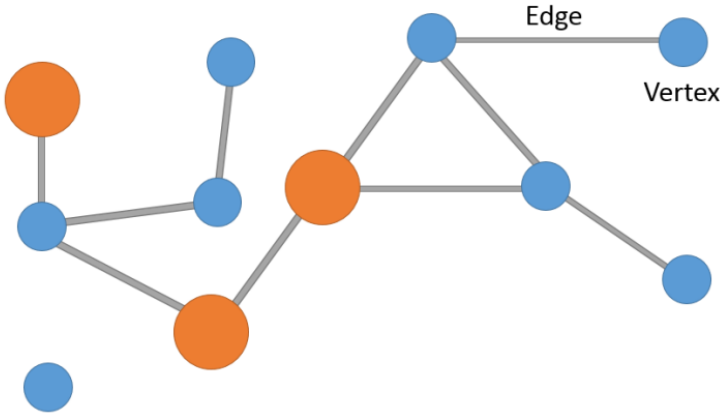

Figure 2-1. Undirected graph composed of 11 nodes and 10 edges.

2.1.1. *Network Generation*

In general, the structure of a realized network is the result of a series of processes where there is some sort of randomness involved.

All the algorithms described below can be used for all types of networks. Although the internet, social networks or the transmission grid can be seen as graphs, they have different properties that make them really different. For instance, while in the case of human networks it is not important where people are located, when we talk about power networks it is a fundamental feature, as they are spatially embedded networks.

This state of the art is basically divided in two parts. The first one is related to generative models in classical network theory, where they try to create an 'optimal' graph for a given number of nodes. The second part is based on complex networks where some new algorithms are proposed to obtain networks with specific features.

2.1.1.1. *Generative models in classical network theory*

First of all, it is necessary to define the minimum weight spanning tree of connected and undirected graphs. This tree is the set of edges that link all the vertices with the minimal total weighting for its edges (See Figure 2-2).

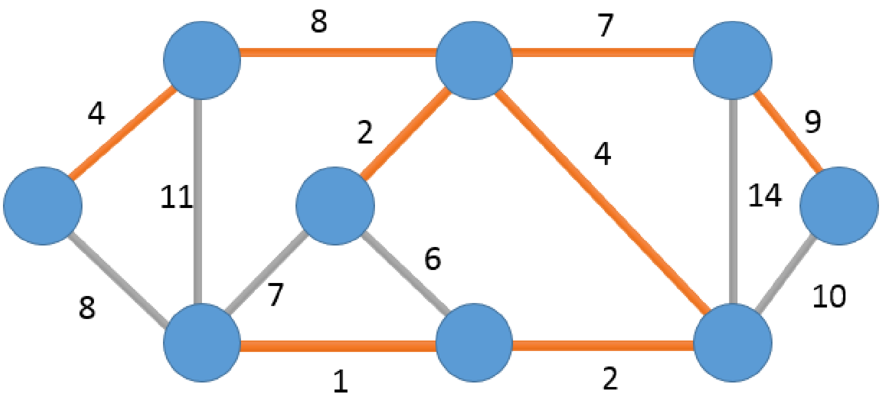

Figure 2-2. Minimum weight spanning tree

Two main algorithms to create a minimum weight spanning tree for a given graph are the following ones:

Prim's Algorithm:

This algorithm was proposed by Prim in 1957 (Prim, 1957). It tries to find the minimum weight spanning tree in three steps:

1. - Initialize the minimum weight spanning tree with one randomly chosen vertex.

2. - Find the edge of minimum weight that connects a new vertex with the current graph (in the first iteration the first node).

3. - Repeat step 2 until all the nodes are in included in the graph.

Figure 1-7 displays the process the algorithm follow for a set of nine nodes.

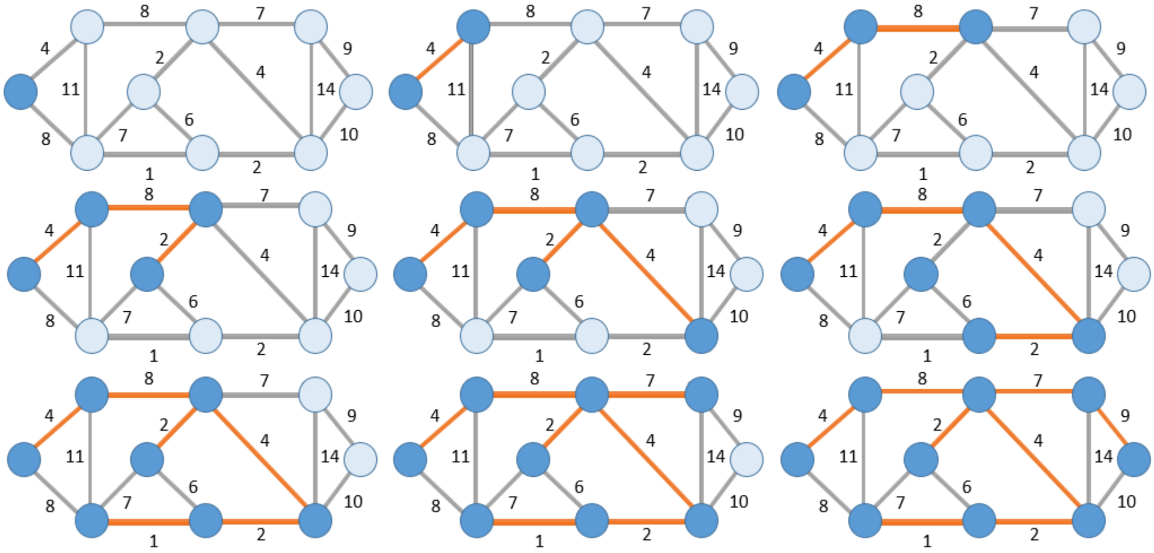

Figure 2-3. Prim’s Algorithm

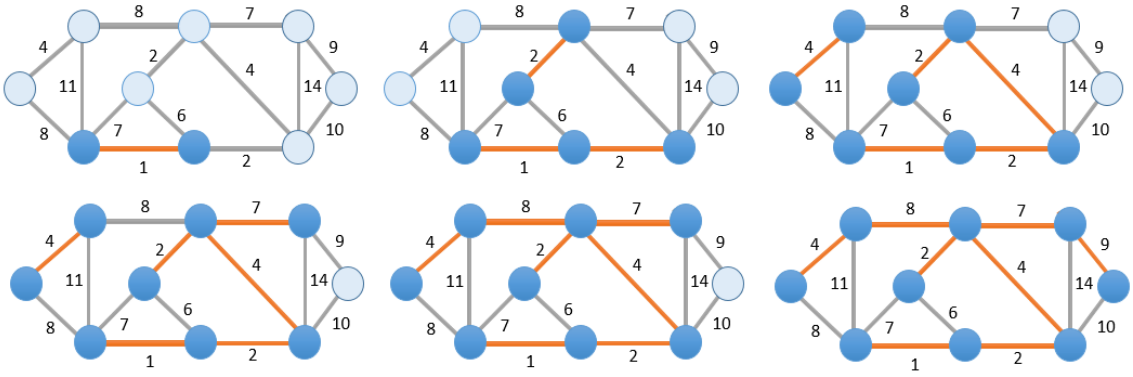

Figure 2-4. Kruskal’s Algorithm

Kruskal’s Algorithm:

This algorithm also looks for the set of edges which connect all the vertices of a graph creating a minimum-weight spanning tree (Kruskal, 1956). Steps:

1. - Initialize the minimum-weight spanning tree looking for the minimum weight edge. This edge and the two vertices, which are linked, are added to the new graph.

2. - Look for the next edge with the minimum weight. Add it to the new graph if at least one of the two nodes linked by the edge are not in the new graph.

3. - Repeat step two up to all vertices are add to the minimum weight spanning tree.

Figure 1-8 shows the steps followed to obtain the minimum weight spanning tree for a given set of nodes and edges.

The minimum-weight spanning tree can be used for network design. In the case of power networks, the TSO knows where nodes are located and the weight of edges can represent the cost of installing a line. Using this algorithm we can obtain a network, nevertheless two main limitations for power networks are identified. The first one is that not only the cost of a line should be considered when you create a power network, it also should take into account the cost of energy supplied (cost associated to generation). In addition, in a real power network nodes have redundant lines to ensure security of supply as we will see later.

2.1.1.2. *Generative models for Complex Networks:*

This part is based on reference (Newman, 2010).

These models try to reproduce the mechanisms by which networks are created. Their main objective is not always to find optimal networks but to generate networks that exhibit some given properties. Preferential Attachment and Vertices-Copying are two particular generative models for scale-free networks. Optimization models generate networks that are created to achieve a specific goal.

Scale-free networks are those which degree distribution follows a power law (at least in the tail). The degree of a graph is the number of edges attached to each node.

Preferential Attachment:

This is the best-known algorithm for networks with a power-law degree distributions (scale-free networks).

In the 1970s Price (PRICE, 1965) tried to explain why a network could have that distribution. The explanation was based in Herbert Simon’s idea of ‘rich-get-richer’ (the more money you have to invest the higher the return on your investment). Simon showed mathematically that the idea gives rise to a power-law distribution.

This idea adapted to network analysis was called by Price: cumulative advantage. Barabási and Albert in reference (Barabasi & Albert, 1999) developed this theory under the name of *Preferential Attachment*.

Barabási and Albert is a generative model for undirected graphs. When a vertex is added to the network, this will be linked to a set of candidates vertices based on a probabilistic function. The probability of linking to a node is proportional to the nodes’ degree (number of edges attached to a vertex). Barabási and Albert’s model established the number of connections for each new node added. The probability of a new edge attaches to a vertex is linear in the degree of the vertex (equation 1-1).

$$p_i = \frac{k_i}{\sum_j k_j} \qquad (2\text{-}1)$$

Where k_i is degree of node “i” (to which new node would be added), and $\sum_j k_j$ is the overall number of edges in the graph.

The networks generated with Barabási-Albert algorithm follow a power-law degree distribution (equation 1-2). The value of α is usually 3.

$$p_k \approx k^{-\alpha} \tag{2-2}$$

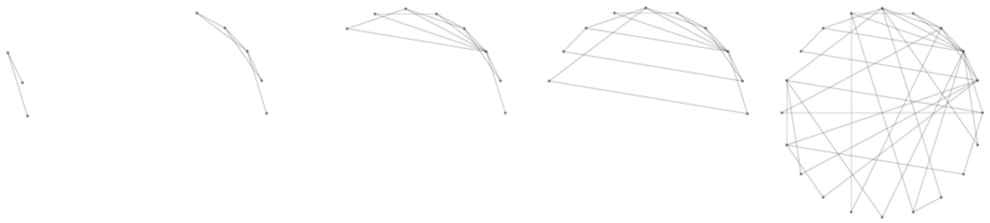

Figure 2-5. The steps of the growth of the network according to the Preferential Attachment model (m=m0=2). Author: Arpad Horvath.

Some relevant extensions to the Preferential Attachment model have been proposed.

Addition of extra edges:

There are certain types of networks where extra edges are added after the graph is created. There is still a power-law degree distribution. In power networks it can be used to ensure the reliability of the system, where redundant lines are added.

At each step, in addition to new edges linked with new vertices, some edges are added linking existing nodes. Both ends of edges are chosen based on degree. As in the general model, the number of edges for new vertices and for existing ones can be defined.

Removal of edges:

In the same way, while some networks add new edges, others remove them when they grow. The probability of losing an edge is proportional to the degree of each node. It can be understood like an inverse preferential attachment model.

Non-linear preferential attachment:

Barabási-Albert proposed a model in which the probability of attaching with an existing vertex is proportional to the degree. There may are some processes in which this relationship is non-linear. For instance, it can also depend on the network size.

Using a non-linear function, the outcome cannot follow a perfect power-law degree distribution.

Vertices of varying quality or attractiveness:

In the model described above, it is assumed that all vertices are equally likely to be linked by a new edge (based on degree). The quality or attractiveness of some nodes can be higher than the rest. When we refer to the power network, nodes will be characterized by size and technology in case of generators. It is clear that the higher the size of the node in terms of power, the higher its attractiveness to be linked.

Biaconi and Barabási (Bianconi & Barabási, 2001a; Bianconi & Barabási, 2001b) proposed a model where nodes could have different quality or fitness.

New nodes will be linked to a defined number of vertices as in the general model. Each node is assigned a fitness (a real number which values follow a given distribution). The probability of linking a new node to an existing one depends on both its degree and fitness.

As in any non-linear preferential attachment, the overall degree distribution may not have a power-law tail. This will depend on the fitness distribution that has been used.

Vertex-copying models:

When new nodes are added, instead of selecting another node to link, the new node will copy all the existing edges of the node selected. To avoid that only nodes which have connections can increase their degree, the copy only happens with a probability ‘p’. With a probability ‘1-p’, the new vertex is added based on degree (preferential attachment algorithm).

Depending on this probability, it can be shown that the network follows a power-law degree distribution.

Figure 2-6 shows an example of the algorithm when a vertex is copied.

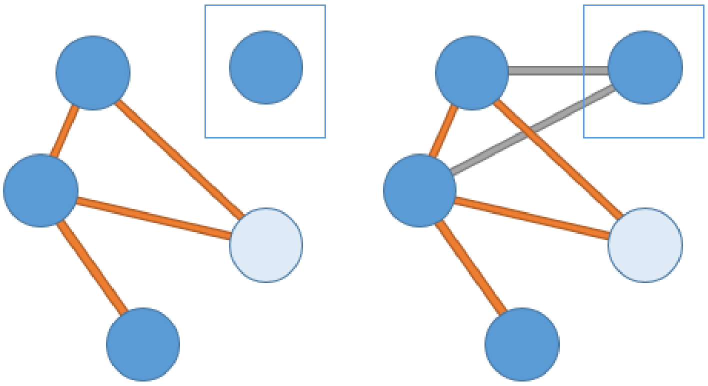

Figure 2-6. Example of vertex copying. Light blue node is chosen to be copied.

Network Optimization models:

Some networks (like a transportation network) have a specific goal to achieve. The design of the network can therefore be based on an optimization process, for instance to reduce the investment cost or the distance between nodes.

There are different models to generate this type of networks, one of the most simple is proposed in reference (i Cancho & Solé, 2003) by Ferrer I Cancho and Solé.

Before thinking on the goodness of this type of algorithms to replicate real grids, we should consider that power networks are not always designed based on optimization process. Nevertheless these algorithms can be really interesting when networks are being developed from scratch (which is known is some contexts as *greenfield* problems).

The small-world model:

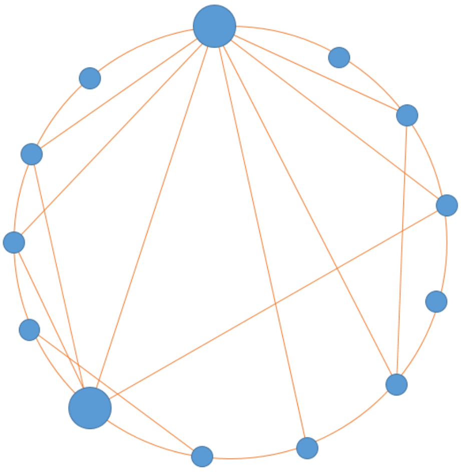

Figure 2-7. An example of small-world graph.

A small-world network is a graph in which although most nodes are not neighbors, the distance (number of edges between two vertices) among them is low. See Figure 2-7:

There are different mechanisms to build small-world networks, such the proposed by Watts-Strogatz or Barmpoutis. (Watts, Dodds, & Newman, 2002), (Barmpoutis & Murray, 2010).

These algorithms have been applied to generate different types of networks. Nevertheless, as we said in Section 1, this is not applicable to power networks. In the next chapter we will explain and implement one of those algorithms, the epsilon disc model which was proposed in reference (Patania et al., 2015).

3. GENERATING RANDOM POWER NETWORKS

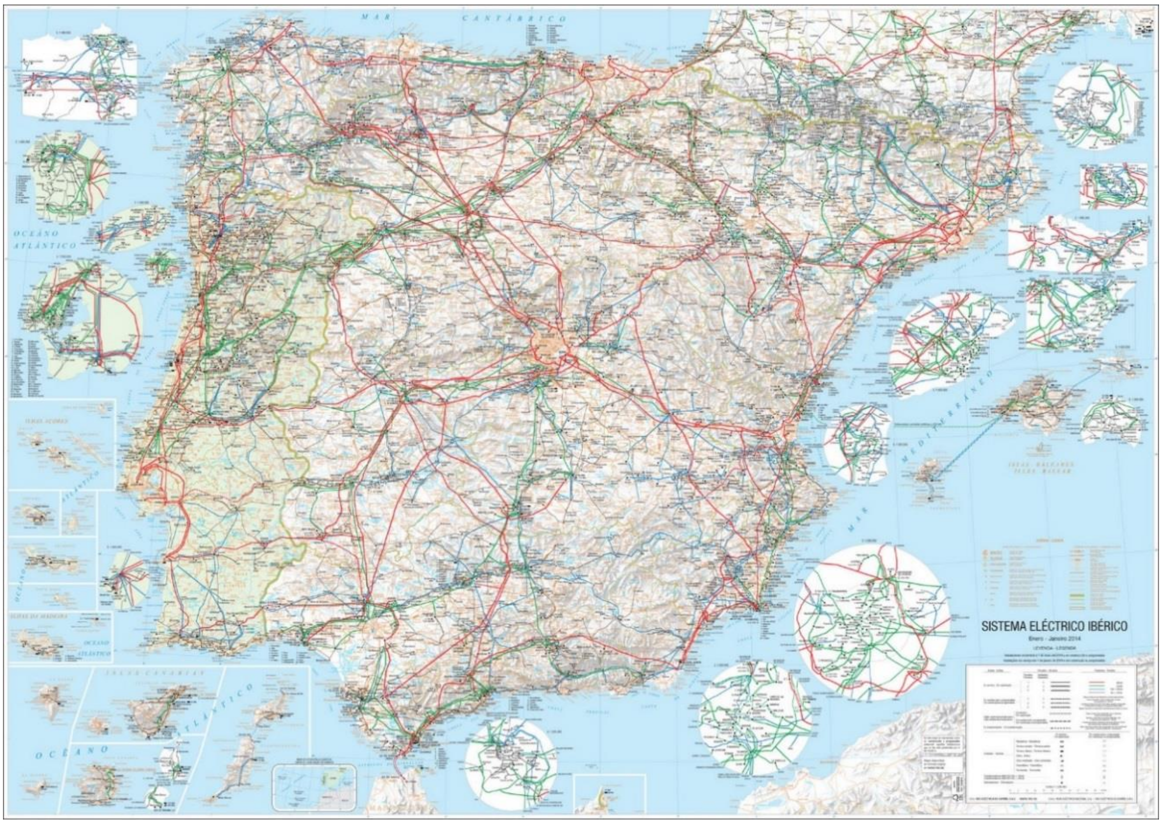

Figure 3-1. Spanish Power Network. Source: Red Eléctrica de España.

3.1. Power Grids

In a traditional power system, generation plants are connected to demand through the transmission and distribution grids. This project will only consider the high-voltage transmission network.

As mentioned above, the transmission grid is a special type of undirected graph, in which nodes are geographically distributed. Therefore, it is a spatially embedded network. The graph is undirected because capacity does not impose any direction in the physical flows that will traverse the network.

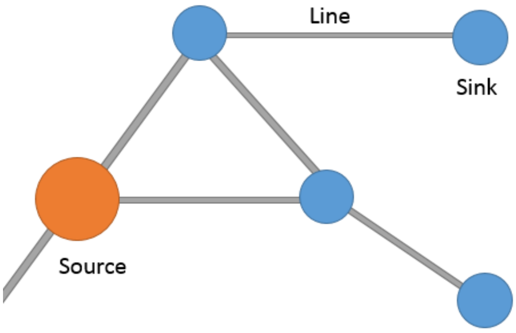

Figure 3-2. Power Grid Graph

There can be two main types of nodes in the power system: sinks and sources. In this text, as it can be seen in Figure 3-2, an orange circle represents a generator (source), and a blue one represents demand (sink). The network can include interconnection substations; they will be modeled as a sink with no load. Nodes have different attributes depend on their nature, such as generation technology, capacity, or operating cost.

We assume that the length of lines is the Euclidean distance and their capacity is limited. The power flow through the lines is governed by Kirchhoff's circuit laws (we will use a linearized DCOPF to model them)

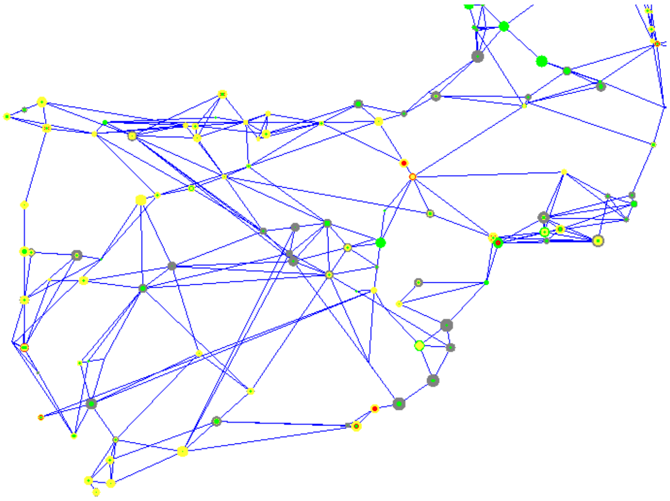

Figure 3-3. Spanish Power Grid.

3.1.1. *Node & Line Attributes*

In this chapter, the power transmission network (generation, demand and lines) will be characterized following the attributes in reference (Pereda et al., 2015).

Source nodes will be characterized following Table 3-1. The number of nodes in each technology will depend on their relative abundance.

Sinks are exponentially distributed with an average value of 66 MW for each node. Table 3-2 shows these line characteristics. We suppose that the reactance is the same for all line types and the electrical resistance is assumed zero, as power losses are not considered.

Table 3-1. Generation Characteristics

TECHNOLOGY	RELATIVE ABUNDANCE (%)	CAPACITY LOWER BOUND [MW]	CAPACITY UPPER BOUND [MW]	MARGINAL COST [EUR/MWh]
WIND	14.10%	10	500	0
SUN	9.10%	10	500	0
HYDRO	15.50%	10	500	0
NUCLEAR	13.40%	1000	1500	15
COAL	18.40%	500	1000	20
CCGT	22,00%	100	500	25
OCGT	4.80%	100	500	50

Table 3-2. Line Characteristics

VOLTAGE [kV]	NTC[3]	COST [MEUR/km]	REACTANCE [Ω pu/km]
400	100	0.08	0.008
400	250	0.15	0.008
400	500	0.3	0.008
400	1000	0.5	0.008
400	2000	1	0.008
400	4000	20	0.008

[3] NTC, Net transmission capacity.

3.1.2. *Epsilon-Disc Model Description*

The first mechanism to generate random networks that was tested in the context of this project was the ***Epsilon-Disc Model***, used in (Patania et al., 2015). Nodes are connected only based on their placement. The algorithm can be described in three steps:

- **Node location** is randomly fixed according to a probabilistic distribution function. This work uses a uniform distribution. The inverse transform method assigns the **attributes** (Table 3-1)**,** using a uniform distribution. It is necessary to define the area where the network is located and the number of nodes (sinks and sources). The number of sinks and sources is related through equation 3-1. Demand must be lower than the upper generation limit, and higher than the lower generation limit. The lower generation limit depends on the availability of the generation commitment.

$$\sum_i \underline{G_i} \cdot x_i \leq \sum_i D_i \leq \sum_i \overline{G_i} \qquad (2\text{-}1)$$

- **Node connections** will depend on the Euclidean distance between two nodes. The idea that provides the basis of the algorithm is to consider discs of epsilon radius centered around each node. The radius will grow ε units at each step, from a minimum value ε_{min} up to a maximum value ε_{max}. Nodes that are inside the disc will be connected to the central node.

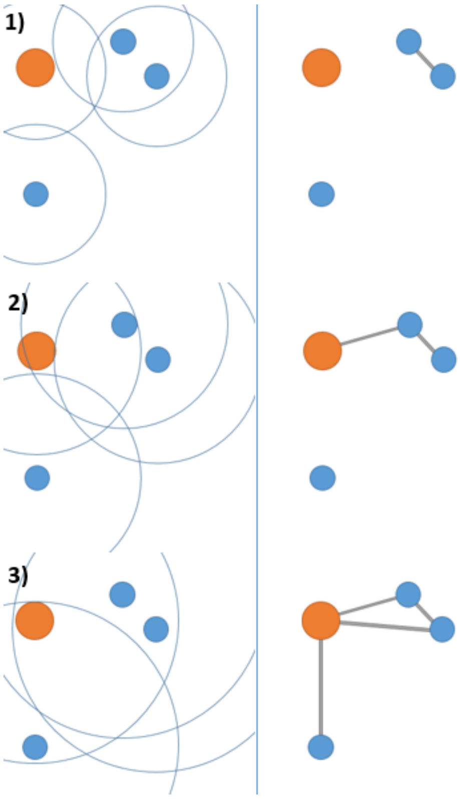

Figure 3-4. Epsilon-disc model process.

- **Line capacity** will be determined solving a power flow, in this case, a DC power flow model. An approximation of unit commitment is required and it will be applied as a simple economic dispatch. Although the power flow results are continuous values, line capacity will be set following a discrete distribution (see table 3-2) in order to reflect the discrete capacities of transmission lines (a line can either be installed or not). The result will be quasi-optimal line capacities for a given network. If the flow through a line becomes less than a threshold, this line is removed and the DC power flow is recalculated.

3.1.3. *Epsilon-Disc Model Simulation*

Figure 3-5 shows four graphs of 166 nodes (146 sinks and 20 sources) in a square of 5000 km with different values for the parameter ε_{max}.

A first conclusion from the figure below is that the number of lines will depend on ε_{max}. This parameter will condition the supply, since with a low ε_{max} there may not be enough line capacity to meet demand. The election of the maximum radius has therefore deep implications, as we will obtain graphs with varying properties, from less to more meshed.

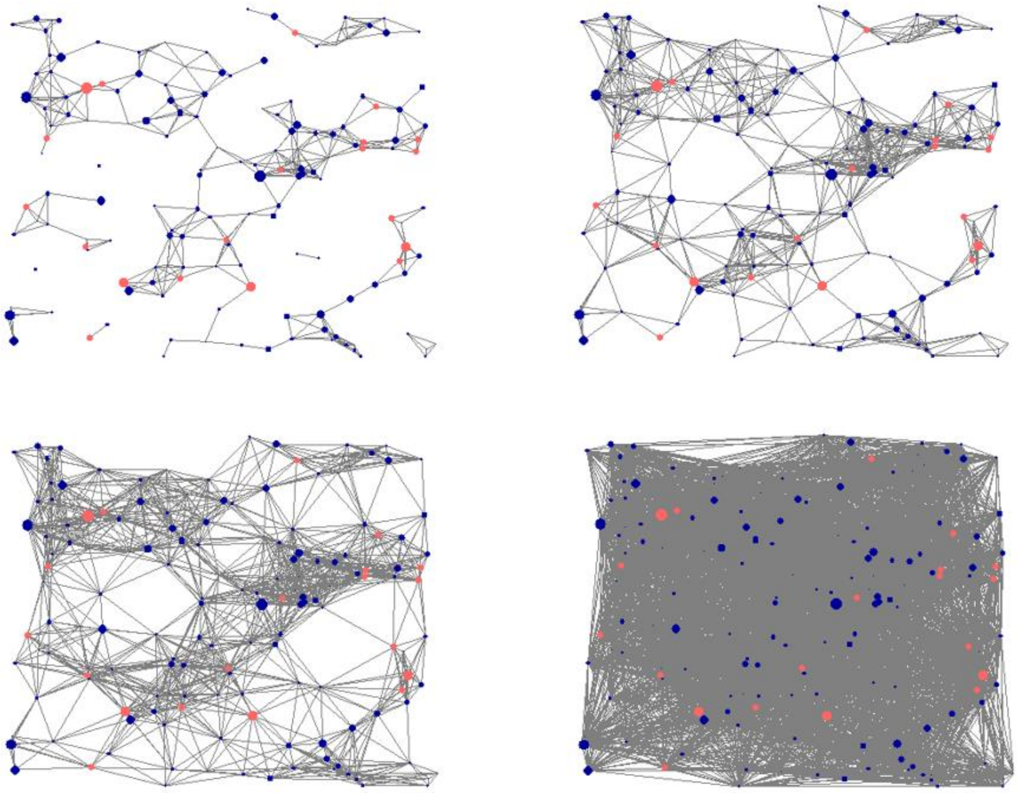

Figure 3-5. Network of 166 nodes with different ε_{max}, 100 km (top-left), 150 km (top-right), 200 km (bottom-left) and 500 km (bottom-right).

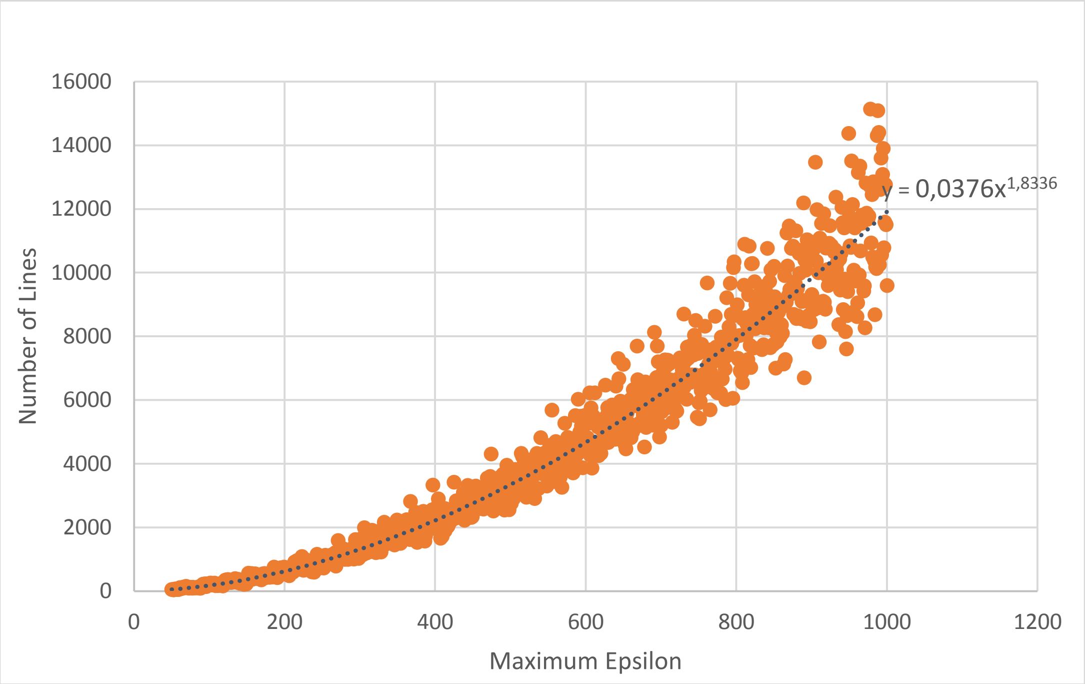

Figure 3-6. Relationship ε_{max}, – Number of lines. Case 166 nodes (146 sources and 20 sinks).

To see the relationship between number of lines set by the epsilon-disc model and the maximum radius specified, we run the algorithm fixing only the number of nodes. We can see in figure 3-6, that this relationship follows a power law.

This model does not consider the electrical properties of the network. This will make that one node with a low demand can be linked to a high number of nodes if they are placed in the latest disc. To avoid nodes connected to a high number of lines, a new step is introduced in the algorithm. Before fixing capacities, some lines are removed.

As Figure 3-7 shows, in a case of three nodes, power supply is ensured with only two lines, so one line is redundant. The reference (Patania et al., 2015) establishes a flow criterion to remove redundant lines. However, if there are two nodes linked by two similar lines, the flow will be similar in both lines and it will be difficult to remove one line strictly following a flow criterion.

In this text, the algorithm has been modified to obtain basic networks where all the nodes are linked. After that, new lines will be added following a reliability criterion. The new step removes one line when three nodes are linked by three edges. The algorithm chooses the line which has the longest distance.

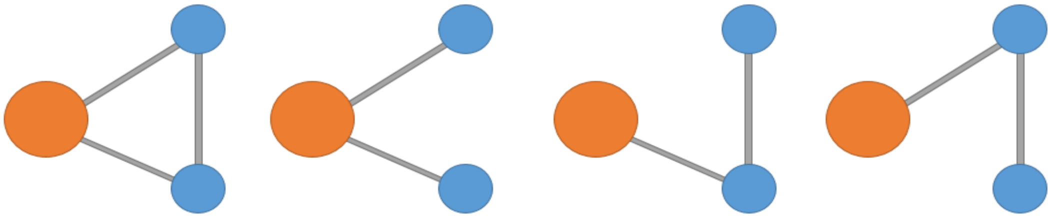

Figure 3-7. Different ways of linking three nodes ensuring the power supply.

In Figure 3-8 we can see the effect of the new step introduced in the algorithm. In an instance of 166 nodes, with a maximum radius of 150 km, the number of lines is reduced from 902 to 278. Despite these removals, there are some areas where the number of lines per node is still intuitively too high compared to real networks.

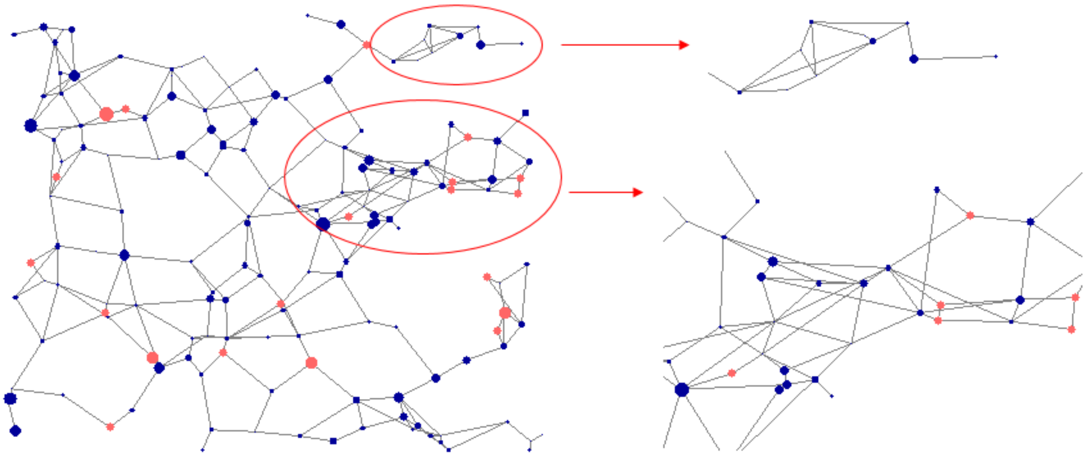

Figure 3-8. Power network where triangles are removed.

As previously explained, the degree of a node in a network is the number of lines linked to it. We will use it to compare a random grid generated with a real one (in this section, the Spanish network).

In Figures 3-9 and 3-10, we can see how the pattern followed in the three cases is not similar. In the second histogram (epsilon-disc model), the degree reaches higher values than the Spanish network (first histogram). After removing triangles (third histogram), the highest degrees have been reduced.

The main difference among the three histograms is the most frequent degree. While in the Spanish instances is around 1, in random networks is over 3. The epsilon-disc model requires new steps to remove lines from extremely linked nodes.

The algorithm links nodes that are close without taking into account their current degree. We can think that the higher the degree of a node, the lower the number of new lines should be added to that node.

One option is to link nodes based not only on distance but also on degree. This idea is developed in the **preferential attachment** model (See section 2). This algorithm, introduced by reference (Barabasi & Albert, 1999), proposes a growth model where the probability of linking two nodes depends on their current degree. This idea is also developed in (Manna & Sen, 2002), where the probability of linking two nodes decreases with the Euclidean distance between them.

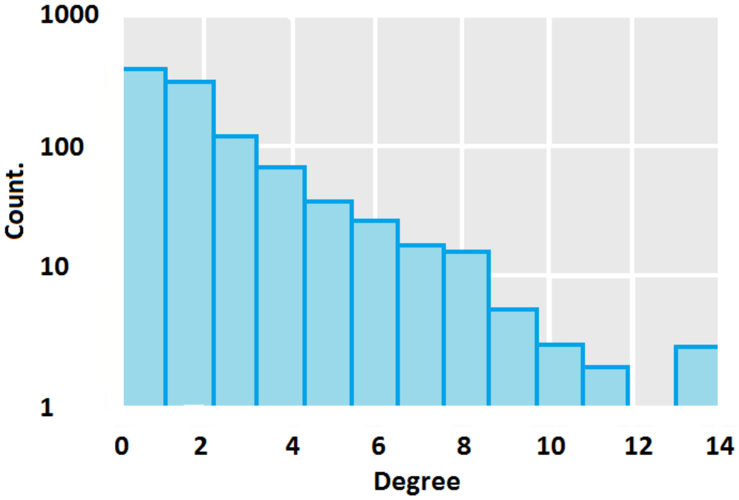

Figure 3-9. Degree frequency histogram of Spanish network. Source: (Patania et al., 2015).

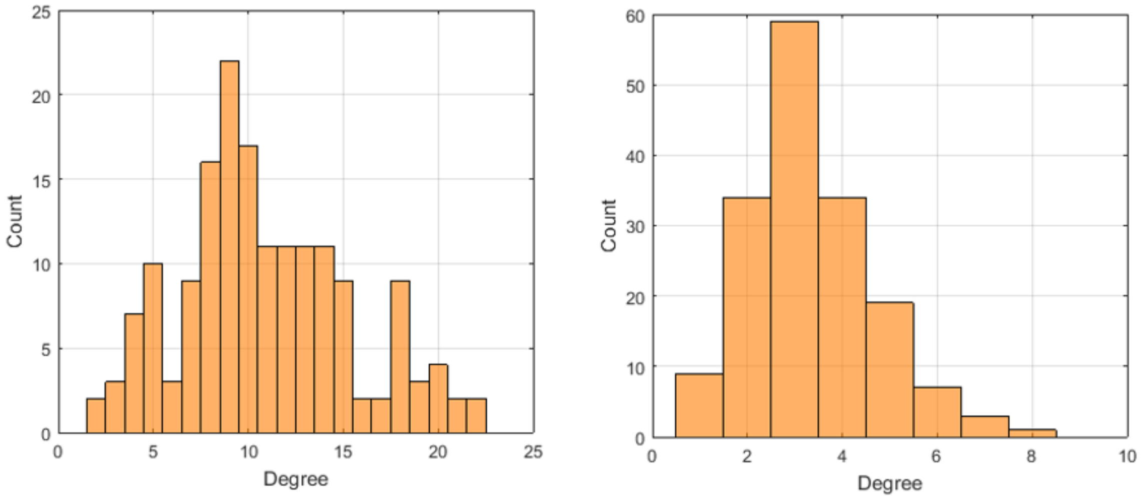

Figure 3-10. Degree frequency histogram of random networks.

We adapt the epsilon-disc model to take into account not only distance but also degree. Now the distance between two nodes will be increased with the number of lines attached to each one. If one node is inside the disc of radius ε (considering only euclidean distance), it will be attached depending on how connected to other lines they are. The distance will be increased based on the degree and also on a weighting factor.

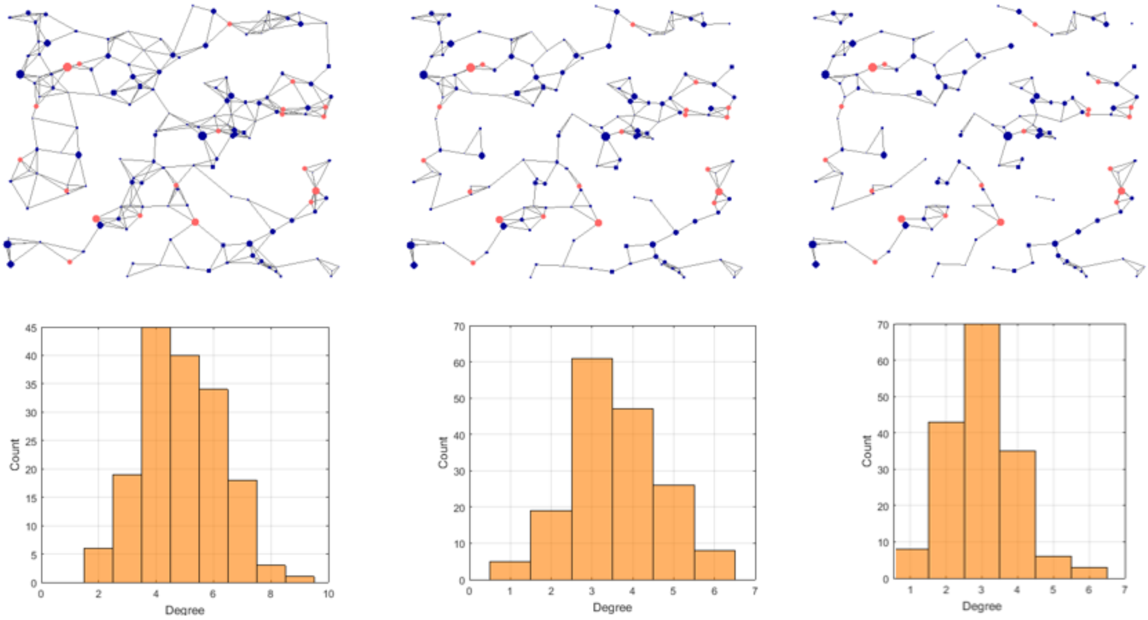

Figure 3-11. Random networks generated and their degree histogram.

Figure 3-11 shows different networks for the same node definition (same node capacities and locations). The higher the weighting factor, the lower the node's degree. These results are quite similar to the previous ones. There are only a few nodes with low degree and the most frequent degree is 3 or 4 depending on the case. More lines should be removed again. As we can see in the third network, some areas are isolated and there are even two sinks that are not linked to any source. There is also the problem that the algorithm does not take into account the features of the generators. Some areas can be supplied only with renewable energy, which means that demand will not be met in the cases where renewable power availability is not sufficient. In addition, there is not an objective criterion to follow as lines are removed. However, it is easier to think in a growth model.

Based on the state-of-the-art of Section 2, where some algorithms used to create networks are described, we can think on a new model that combines the best ideas of them. From preferential attachment we can take that the higher the degree the higher the possibilities to be linked to another node. It is also clear that the connection should be conditioned to the features of the nodes, so the process should take into account a fitting function. At the same time, the process should try to look for a network that minimizes the cost, although it should not be considered an optimization problem. We are not looking for the minimum cost network but to a process that can generate a wide range of reasonable power networks.

After analyzing the epsilon-disc model, we have seen its strength in taking into account that the transmission network is spatially embedded. This geographical aspect also should be considered in a new algorithm to generate power networks.

The ideas described above can be combined into a single algorithm. We propose to ensure the power supply in a first step and support the reliability of the system in a second step. In this way, we cover one of the main weakness of the epsilon-disc- model where we could not be sure when demand is met. For meeting the demand we can think that it is not necessary to have a high number of lines per node, so we will avoid the problem of high degree for most of the nodes. This new algorithm is explained and implemented in the next chapter.

4. THE TRADE-OFF NETWORK MODEL

The Epsilon-disc model only considers the spatial component but no other features that are relevant to power networks. For instance, we should consider generation technology in order to avoid having sinks which demand is only met with renewable sources with no other backup technology.

In addition, the Epsilon-disc algorithm takes into account the cost of installing a line (cost is directly related with distance €/km) but it does not take into account the generation cost. In general, we can say that customers try to meet demand at the lowest cost. Generators that are cheaper should be preferable.

In a real network, nodes are not only linked to satisfy their demand, they are also connected to reach a suitable level of reliability of the system. Transport System Operators (TSO) apply an N-1 redundancy criterion to ensure the reliability of the power network when all elements but one (hence the name N-1) are available. This criterion tries to answer the question: What happens if a line or a generator fails?

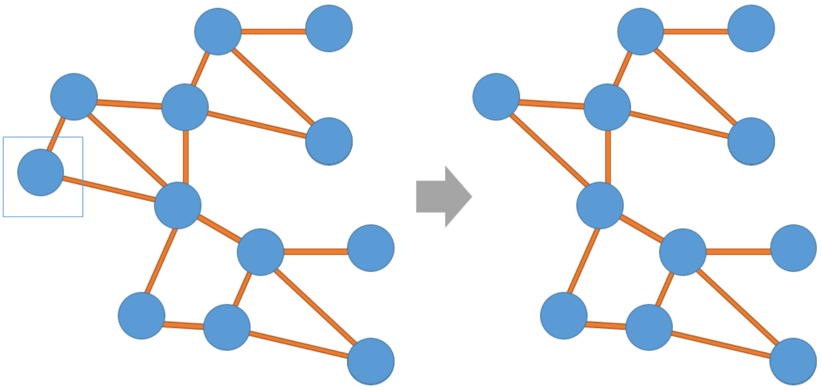

Figure 4-1. N-1 criterion where a node is removed.

To answer the question, we can simulate the behaviour (power flows) of the system after removing one edge or one node as we can see in Figure 4-1. In case of removing a generator, a new dispatch is required. Once the generation is fixed, the power flows through the network are estimated. If power flows respect the line capacity limits, it means that neither lines nor nodes should be added to the system. If any line is overloaded or demand cannot be met, it is necessary to study which line or power plant should be added to the system to satisfy it.

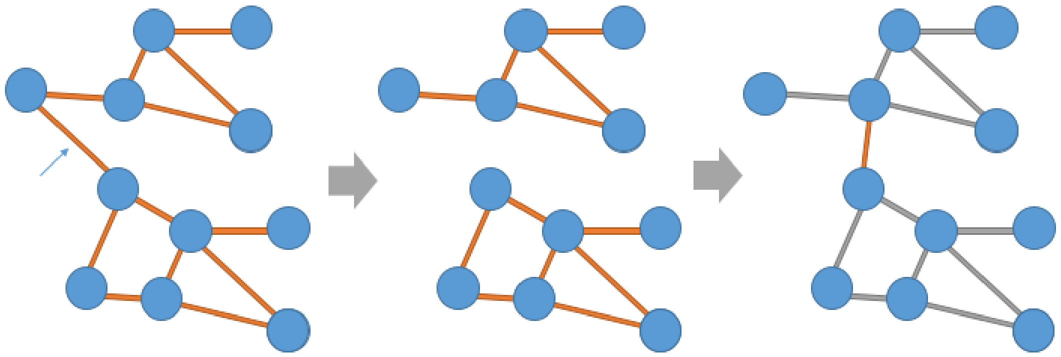

Figure 4-2.- Process followed in the N-1 criterion

We propose the Trade-off model as the generation algorithm for spatially embedded power networks. This model exhibits properties that are more consistent with the ones of actual power networks. The algorithm is not only able to ensure energy supply but also to create networks that follow specific degree distributions. This is the main contribution of this research, as it can be used for other types of networks. To reach a degree distribution, we propose a preferential attachment model that effectively shifts low-degree nodes towards the high-degree region by adding edges in a guided fashion.

The model can be described in two steps: meeting demand cost-effectively (1) and fitting a degree distribution (2).

The algorithm has been implemented in Matlab[4], using existing Matlab functions to work with graphs.

4.1. Meeting demand efficiently

The first step of the algorithm simulates a clearing process between demand (customers) and generators (suppliers). Each sink will try to find the generator that is able to supply its energy needs at the lowest cost.

The priority of finding an available generator is based on size, in terms of power demand. The bigger the node the higher the priority to find a source node.

Before starting the trade-off model, nodes are generated as described in the Epsilon-disc model (Chapter 3). Sinks are defined by their location and power demand. Power plants are characterized by location, maximum and minimum generation limits, generation variable cost and type of technology.

[4] Matlab, http://www.mathworks.com/products/matlab/index.html

The cost of linking two nodes considers the cost of line installation and the cost of generation (equation 4-1). The cost is the net present value of future cash flows. Therefore, it is necessary to define a period of time and a discount rate to evaluate the cost.

$$C = \sum_{t=1}^{n} \frac{C_{installation+} \, C_{generation}}{(1+r)^{t}} \tag{4-1}$$

When a single generator is not able to supply all the demand of a node, it will supply as much as it can. New iterations linking other generators to that sink will be required until demand is satisfied.

The cost associated to installing a line is linear with the length of the line, modelled as the Euclidean distance between two nodes. The model chooses the lowest-cost option between a direct connection between two nodes or connecting them through already existing lines. When nodes are not connected directly, more than one line could be required.

The algorithm will look for the shortest path between nodes. As nodes are not connected yet, it is necessary to define an imaginary graph. In this network, the weight of edges will be the geographical distance, if there is no line or 1 if there is a line connecting two nodes.

To avoid high computational requirements, it is necessary to define a graph where is more probable to find a connection between closed nodes than between those which are located far away.

This is performed with the epsilon-disc model. As we saw in the previous section, all the nodes that are inside the disc of a node will be linked to it. The disc will grow until a maximum radius. The problem of defining the maximum radius does not affect to this case. If the maximum epsilon is not big enough, some areas will be isolated, nevertheless the algorithm will ensure the supply connecting two nodes directly. If the maximum value is high, there are more possible paths to check and the algorithm will be slower. It has been tested that there is not linear correlation between computational time and number of edges of a graph. The algorithm is much slower when the graph increases in size.

Figure 4-3 shows the base network for a given example. The maximum radius has been defined as the minimum of the maximum distance between nodes. It is supposed that all nodes should be connected. As we can see in the figure, some areas are more meshed than others.

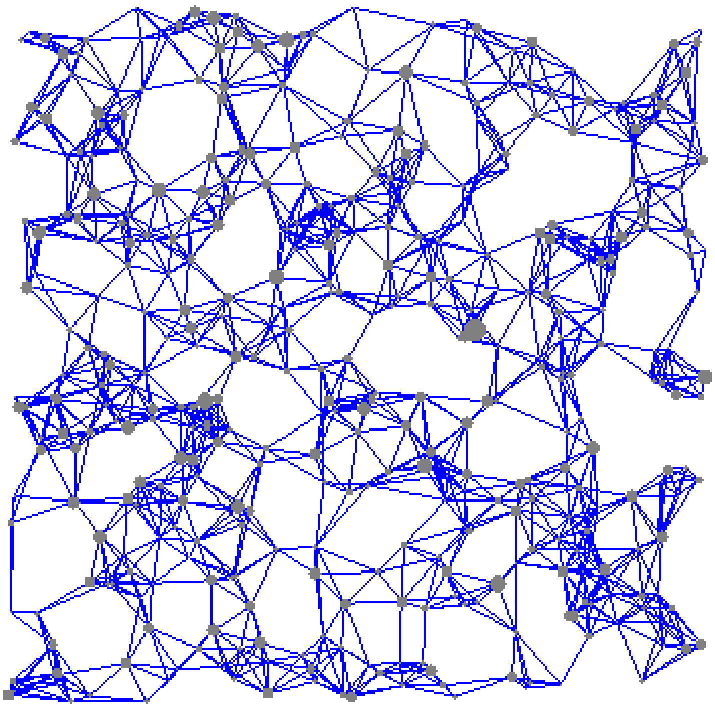

Figure 4-3. Auxiliary graph based on epsilon-disc model.

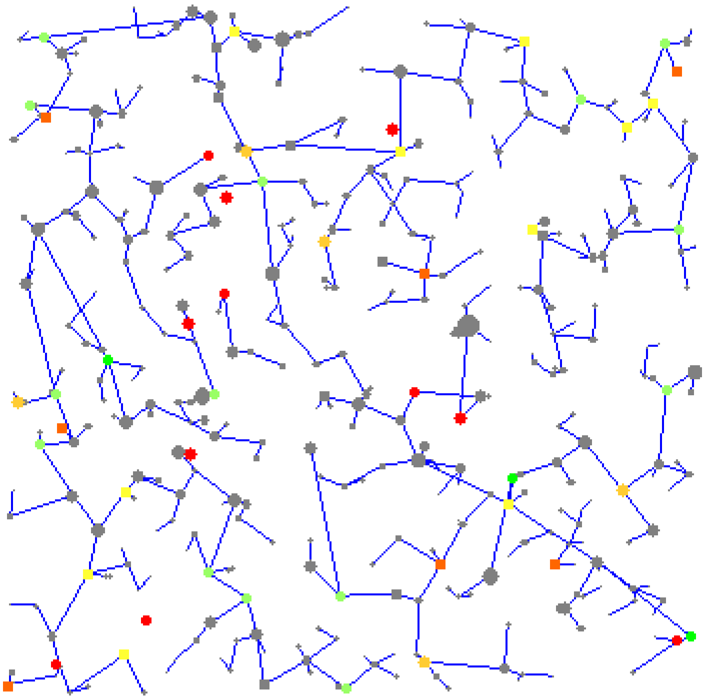

Figure 4-4. An example of the first step in the trade-off model.

Figure 4-4 displays the result of this first step for a random network. Nodes have been generated following the parameters defined in Section 3. Grey bubbles represent demand. The colour of source nodes is related to the generation cost of that node, with green representing relatively cheap and red representing relatively expensive generation.

As we can see, all the sink nodes have at least one connection. We suppose that all the demand can be supplied. We can check this assumption with the proportion generation/demand. If there is more power available than demand this ratio will be higher than 1. In this case, the algorithm can ensure all the demand is met. In cases in which the ratio is lower than 1 the system will not able to satisfy its demand. Networks with energy not supplied are interesting from the point of view of the optimization algorithms but in fact they are not realistic.

In addition, it is important to consider the resource availability in the case of renewable generation: it is necessary to introduce an improvement to ensure nodes will meet demand in all scenarios.

If we want to create a network similar to a real one, we need to consider the time evolution of the network. To do this, the trade-off model can be subdivided in different stages. Each stage represents an increase in power demand and generation. Stages can be based on generation technologies, as the historical development of the power system seemed to be articulated around periods where there was one single dominant technology that was more extensively deployed

The simplest division would be to split generation expansion in two stages. In the first one, thermal and hydroelectric plants are used while in the second stage renewable generation is introduced. This implies that sink nodes seek to satisfy their demand using conventional generation first, and only then are linked to renewable generators as well. These stages can be generalized to consider other technology waves, as it has happened in Spain with coal, nuclear or CCGT units. The stages can also be used to consider national versus international connections.

Another possibility is to consider that renewable plants only supply a percentage of the total demand. The sink node will have more generation available than what it needs. These kind of model would be useful for a greenfield expansion, where new plants (most of the renewable power plants) are being developed at the same time that the lines.

From the point of view of demand, if we consider different stages, each stage can represent a growth in demand. The demand increase trajectory is pre-specified at the beginning.

Line capacity is fixed based on the theoretical flow through the line after the trade-off model is executed. It also considers the limits of the biggest line that can be installed (which is an input). If the flow is higher than the maximum line capacity, the algorithm will look for a new way to supply demand.

A typical degree distribution after the first step in the trade-off model is shown in Figure 4-5. The distribution is similar to the shown in Chapter 3 for the Spanish network. The most frequent degree is one (that is, most of the nodes are only linked to another one).

Nevertheless, it does not fit well to the simplified European Network where degree does not follow an exponential distribution.

Depending on the specific network, different distributions will be required. It is necessary to define a new step to make it possible to follow any type of degree distribution. We propose a method to do this by adding lines in a specific way. We can shift the distribution to the right (increasing degree) stablishing new lines. If we need to shift it to the left (decreasing degree) is necessary to remove existing lines. If lines are removed, we should check if demand is still met. For the sake of simplicity, we only consider the possibility of adding lines, and therefore we implicitly assume that we will not need to shift the distribution to the left. The number of lines added in step 1 is considered relatively small with respect to the total number of lines.

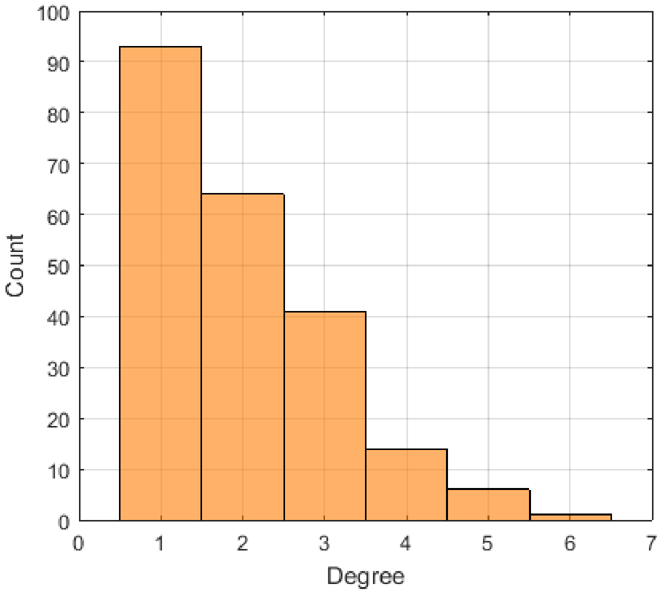

Figure 4-5. Typical degree distribution for the network obtained in the first step of the algorithm.

4.2. Fitting a degree distribution

This step can be summarized as adding new lines smartly to obtain a defined distribution.

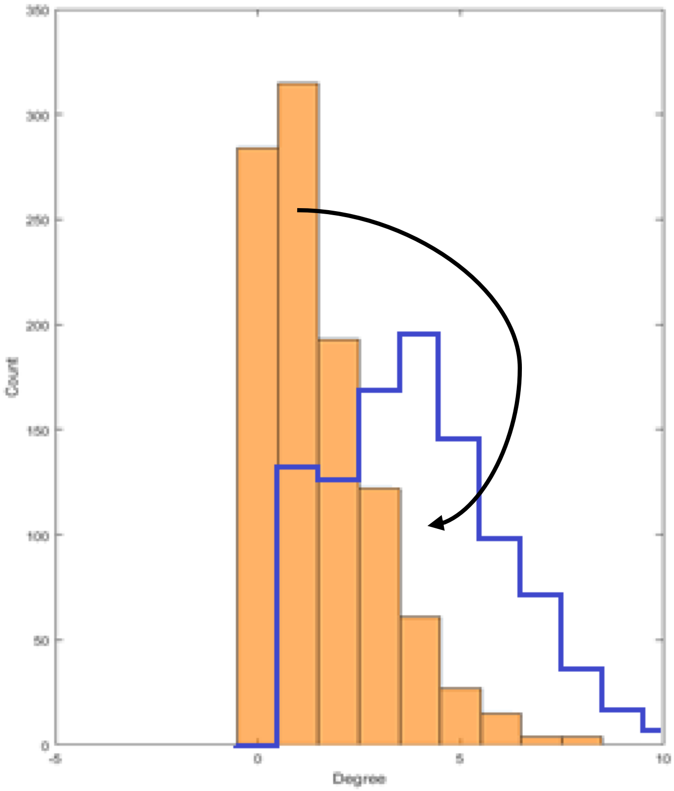

Figure 4-6. Current degree distribution (in orange) and target function (blue line).

An input of the algorithm is the degree distribution that is wanted to achieve. This can be given from the distribution for an existing system, for instance. It can be also defined randomly. The distribution can be introduced as an equation or as a histogram. We use a bin for each value of degree in order to get the most accurate results.

The algorithm also requires setting a tolerance level for the fitting. The process ends if the difference between the target and the current distribution is smaller than the tolerance or if after a specific number of iterations the model does not add any line. The lower the tolerance the most accurate the histogram and the greater computational time spent.

The algorithm will analyse all the possible combinations of nodes to decide if a line is added to the system. Constraints related to distance can be introduced to avoid studying lines that are too long.

As in the first step of the algorithm, there is a priority to analyse if a line is installed. It is based on size.

The process will analyse the current degree of the two candidate nodes to be linked. If the line contributes to reach the target distribution, it is assigned a positive probability of being installed. If that connection will not contribute to the fitting process, the probability of linking those nodes will be zero.

Equation 4-2 defines when a line contributes to reach a desired distribution based on the degree of two nodes.

$$\begin{gathered}\left(number\ of\ nodes\ \big(\ degree\ (node)\big) > target\right)\ \cap \\ (number\ of\ nodes\ (\ degree\ (node\ i) + 1 = < target)\end{gathered} \quad (4\text{-}2)$$

The objective is to shift the degree distribution to the right. We assume that the number of nodes with a low degree is much higher than the number of nodes with a high degree.

When the connection between two nodes is studied, their degree is analyzed. If the number of nodes with a given degree is higher than the target value, then the distribution must be shifted to the right.

As we have explained, power networks are spatially embedded. We assume that the probability of linking two nodes is related to the distance between them. We will condition the connection between two nodes not only to its contribution to the fitting process but also to a fitness function based on distance. Considering distance we are taking into account the cost of installing a line.

The fitness function can be defined as the probability of installing a line. For instance, we can use a Gaussian distribution. Another possibility is to introduce a desired distribution for length of lines. Knowing the maximum number of lines in a gap, the algorithm will decide if the probability of connecting a line that contributes to the reach the target distribution is high or low. If the difference between the target number of lines in an interval and the current interval is zero, that lines will not be installed.

If a line contributes to fitting the degree distribution and has a positive probability to be installed, the algorithm will decide whether to install it or not based on generating a random number.

After a new line is installed, it is necessary to update the difference between the target distribution and the current one.

When the fitness distribution is defined it is necessary that it is in accordance to the properties of the case. Otherwise, lines will not be added and the algorithm will not be able to reach the desired distribution.

If after the Step 2 there are still isolated nodes, they will be linked randomly to one of their closest neighbours.

The result of this second step will be shown in the next chapter. This will be used also to check the degree of accuracy of the algorithm, testing it with a real case of the European network. Node features will be introduced as inputs. The objective will be to know the goodness of the algorithm to link nodes. The artificial network will be compared to the real one from a statistical point of view.

If the same nodes are used in different simulations, the result of the first step will be identical. There will not be any difference between the lines installed in different cases. Nevertheless, the second step will give different results for each simulation.

5. STATISTICAL ANALYSIS

A real case of the European network will be used to test the algorithm explained in the previous section. The objective is to check if the result, that is, the artificial networks generated with the trade-off model, are consistent with the real ones.

5.1. European network

The simplified network used is a combination of 14 countries in Europe. Each country has a different number of nodes, Table 5-1 shows the number of nodes per country. Nodes have their geographical position defined. At each node we can find demand, one or more generator or a combination of demand and generators. Figure 5.1 shows the network that will be the benchmark. Grey bubbles represent demand. Generation is represented with different colors based on generation cost (red represents more expensive and green shows cheaper plants). Lines can be classified as international connections and national connections. All nodes are connected in the same graph.

Table 5-1. Number of nodes considered in each country.

COUNTRY	NUMBER OF NODES
1	32
2	29
3	15
4	218
5	92
6	245
7	162
8	7
9	53
10	20
11	30
12	22
13	32
14	68

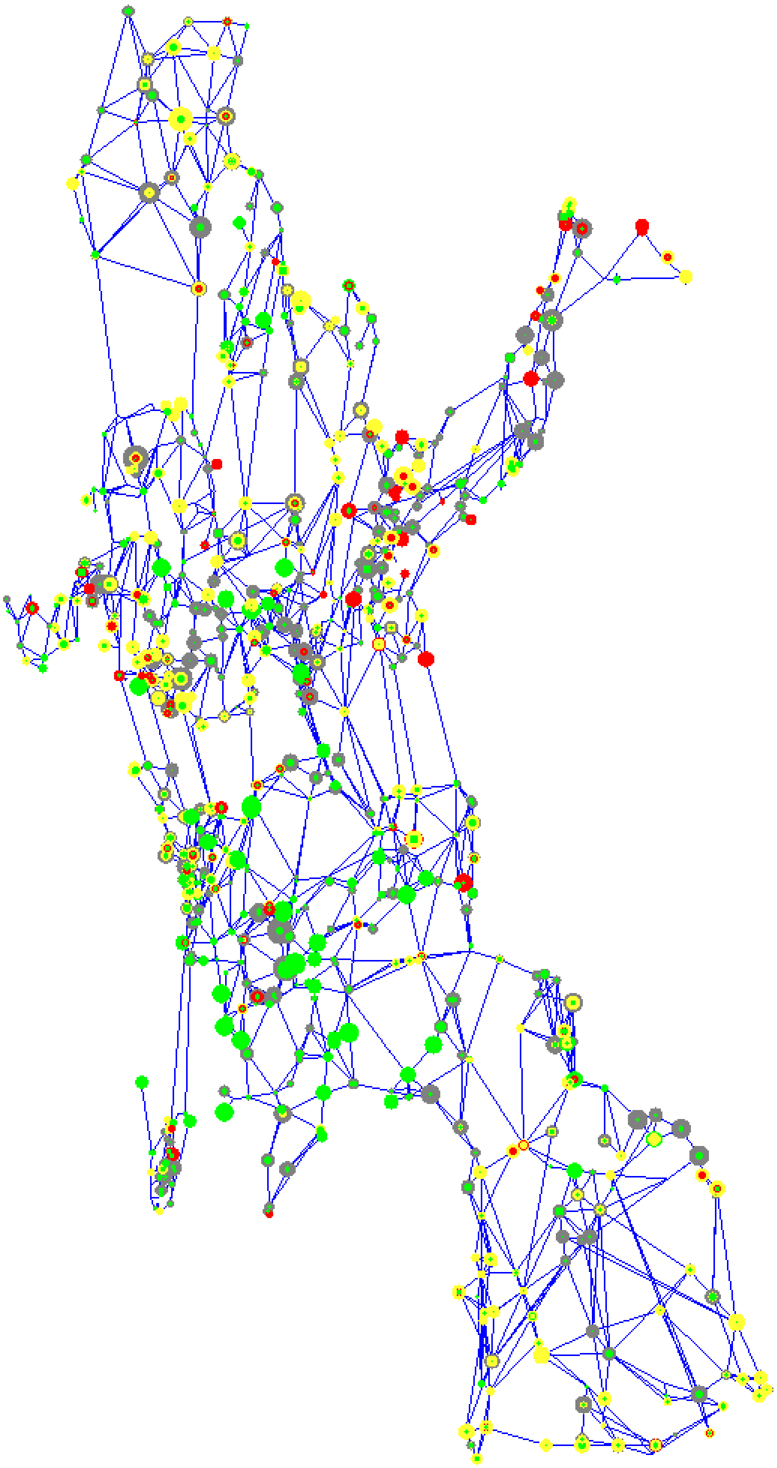

Figure 5-1. Simplified European network.

5.2. Random European network generated with Trade-off model

Node location and features (power demand, power supplied and type of generation) have been introduced based on the real case. Only lines will be generated randomly by the algorithm.

In attempt to mimic the real historical development of the network, two stages have been considered. As in the real network renewables were only installed later, we have distinguished between conventional and renewable power plants. Conventional plants are: nuclear, coal, gas, hydro and oil. Renewables include wind and solar. We have considered the growth in demand when renewables were installed was small, so all the demand was considered in the first stage. As there is not an increase in demand during the second stage, renewable plants will be linked randomly in the second step of the algorithm, where all the demand is met.

In the first step the algorithm will try to ensure that all the demand is supplied, in this case only with conventional power plants. Sink nodes will look for the cheapest power plant that is able to supply all their demand. Generation cost for each type of power plant is shown in Table 5-3. The present value is calculated considering a discount rate of 10 % for the next 40 years. Line power losses have not been considered. Table 5-2 shows the cost of installing a line.

Figure 5-2 displays the network generated after the first step. In this instance, ratio between total conventional generation power and total power demand is 1.4353, which is reasonable compared to existing system. We can observe that all sink nodes are connected to at least one line.

Table 5-2. Lines cost in European case.

CAPACITY (MW)	COST (MEUR/KM)
100	0.08
250	0.15
500	0.3
100	0.5
2000	1
4000	2

Table 5-3. Generation cost in European case.

TECHNOLOGY	COST (EUR/MWH)
Gas	50
Oil	110
Biomass	16
Peak	120
Nuclear	14
Coal	50
Hydro	0
Wind on-shore	0
Wind off-shore	0
Solar	0

With the second step, the algorithm will try to fit the desired degree distribution. In this step, nodes with renewable plants have been introduced.

During step 1, sinks are connected to sources only considering generation cost. No geographical constraints were taken into account. Nevertheless, in the second step the connection of two nodes will be conditioned by the country of location of the nodes.

As the algorithm will try to reach a degree distribution connecting nodes of the same country, we can say that the algorithm is fitting not only one-degree distribution but also fourteen distributions, one for each country. This is more representative of the European network given that the national networks were created independently only to be interconnected relatively later.

Although we lose the possibility of mimicking international connections, we think is the best way to have a good approach.

The fitness function considered in this step is based on the length of the line. For each candidate line the probability of installing it is correlated to the distances between the two nodes. There will be a different fitness function for each country, as length of lines cannot be considered similar in the case of Spain or Sweden for instance.

Figure 5-3 shows the artificial network for Europe obtained with the Trade-off model.

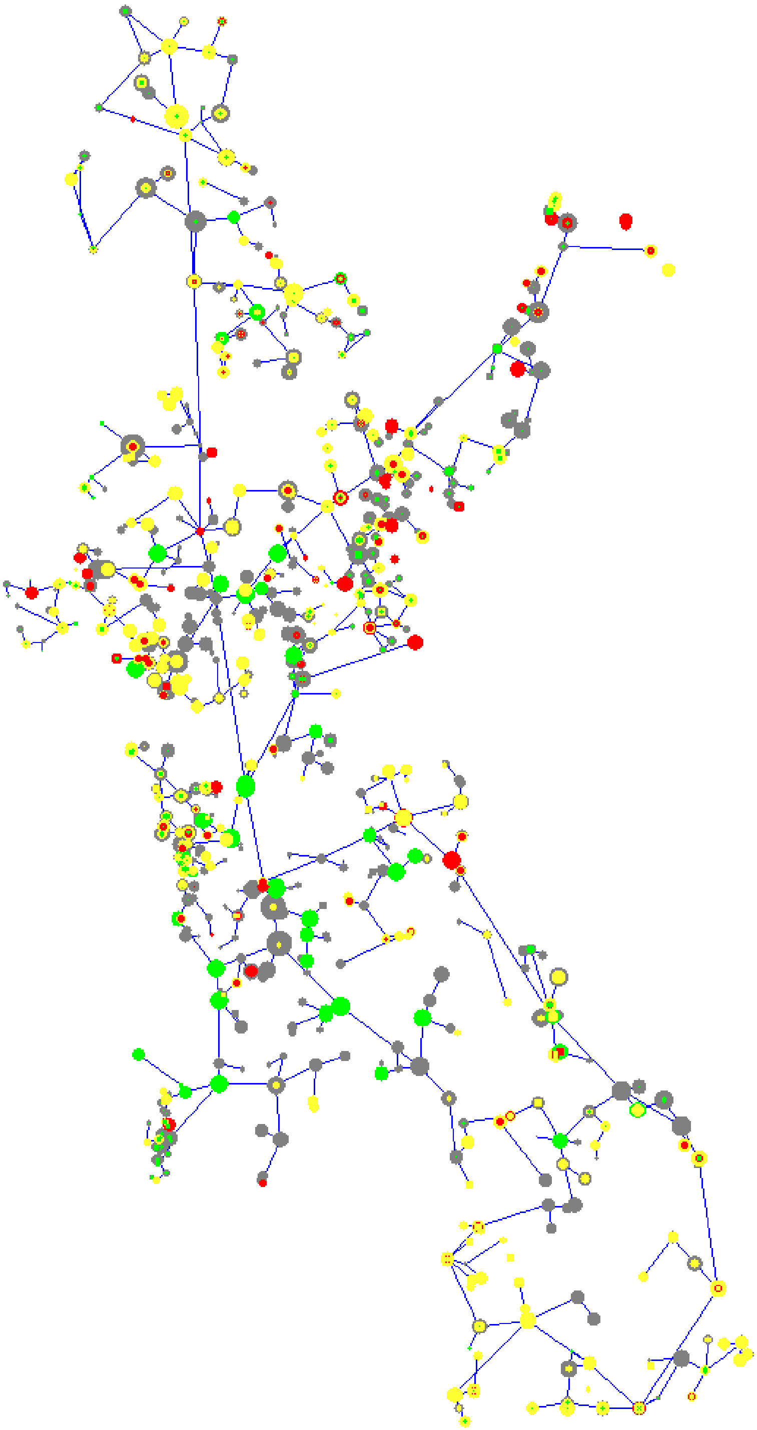

Figure 5-2. Artificial European network after the first step of Trade-off mode.

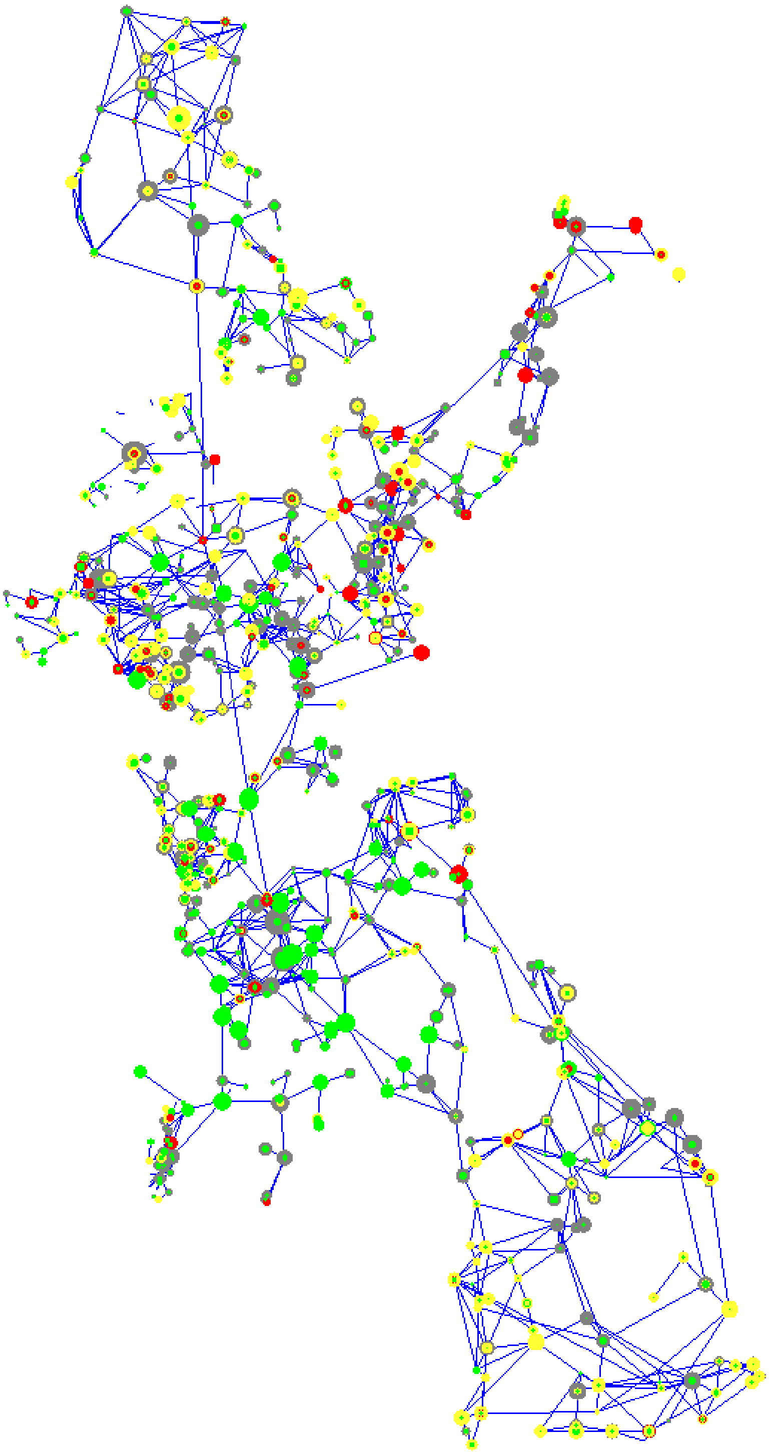

Figure 5-3. Artificial European network obtained with the Trade-off model.

Figure 5-4 shows the degree distribution after the first step and the target distribution. We can observe that there is a high number of nodes whose degree is zero. Nodes with no lines are those which have renewable plants or have not been linked in the first step (remember that the ratio between power generation and power demand was higher than 1).

The most frequent degree is one, as most of the nodes can met its demand only with one power plant. There are no nodes with a very high degree; the most connected node has 8 lines connected to it. This is far from the 16 lines that some nodes have in the real network.

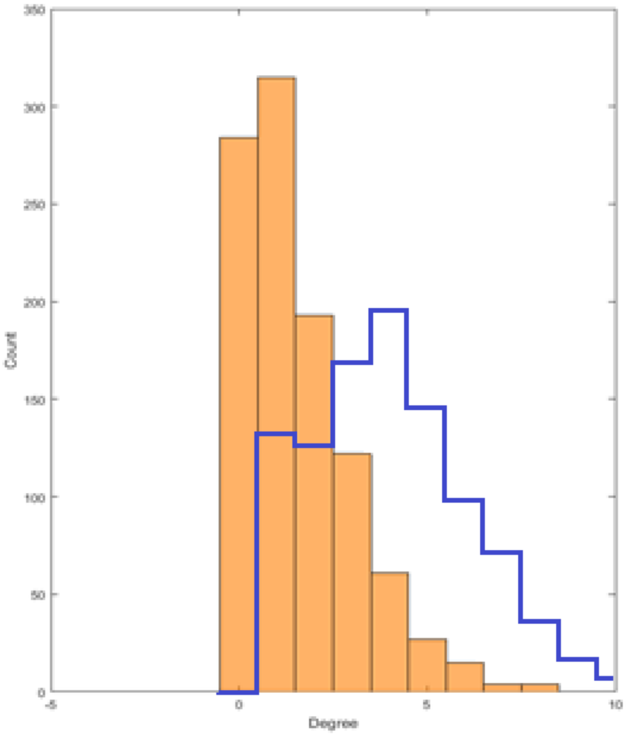

Figure 5-4. Degree distribution after the first step and target distribution for the European case.

Although the tolerance for this case is only 2, the error in the overall degree distribution can be high as international connections have not been considered. The degree will be adjusted based on ten bins: nodes with 1, 2, 3, 4, 5, 6, 7, 8, 9, >9 lines per node.

The fitness distribution for the length of lines has been defined as the difference between the number of lines in the real network and in the artificial network for a length interval. If that difference is positive, the probability of installing that line is also positive. This difference is updated after a new line is added. This function is divided in ten classes equally distributed from the minimum and up to the maximum length of a line in each country.

5.3. Statistical analysis

We can observe that the real and the artificial network are very similar. Different measures are used to compare them from a statistical point of view.

5.3.1. *Degree distribution*

The degree distribution, as explained, is the main descriptor of a network used in network science. As we saw in Section 3, in the case of a power network degree is the number of lines linked to a node (a demand or a power plant).

Figure 5-5 shows the degree distribution for the real and artificial networks. Both histograms are quite similar. As a consequence of the inputs used we can observe that there are no nodes with more than ten incident lines. If we wanted to avoid this, we could divide the degree distribution into a larger number of bins (in this case we used one single bin for all degrees higher than 9). Although the tolerance for the algorithm was 2, we can observe that the error is bigger than this. This was to be expected, as the fit will not always be possible. For nodes with one connection, the difference between the target and the current degree distribution is almost 20. We should consider that the algorithm has been adjusted by country, so for the whole network the error is the sum of the errors of each country.

Another cause that justify that the algorithm does not adjust perfectly the distribution is that international connections are considered in the target distribution but we did not take them into account for fitting the curve. Adding international connections the curve will shift to the right so the difference with the real distribution will be lower.

A solution to improve this error can be to divide the network in different areas that do not correspond to countries. For instance, in Central Europe, a set of countries can be merged into a single area.

Figure 5-6 and 5-7 show the degree distribution of the real and artificial network for each country. As explained above, the difference between distributions is lower. It is in accordance with the tolerance. In most of the cases the error is one node per bin.

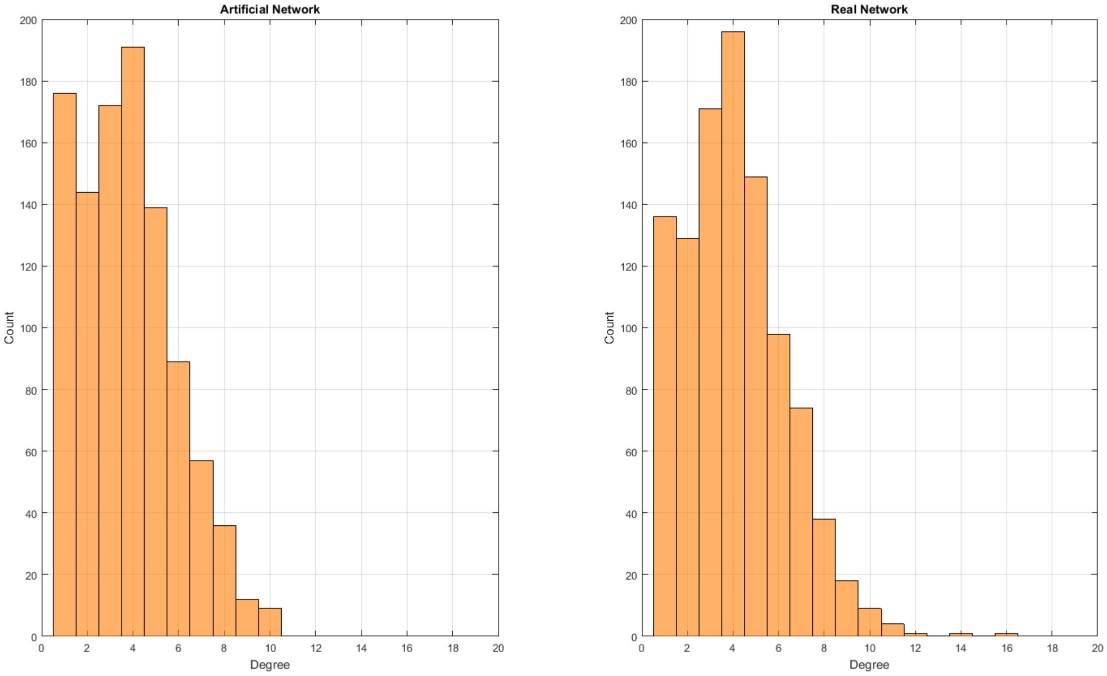

Figure 5-5. Degree distribution for artificial (left) and real (right) European networks.

Table 5-4 shows statistical measures for the degree distribution of both networks. Mean (the average value), Standard deviation (the amount of dispersion of a set of values) and mode (most frequent value) have similar values. The difference is a little bit higher for the moment of order 3 (indicator of symmetry) and 4 (kurtosis, indicator of 'tailedness').

In both cases moment 3 is positive, there is an asymmetry respect to the mean value. The moment 4 is higher in the real network as the tail of the distribution is fatter. In the artificial network values are more concentrated close to the mean value.

Table 5-4. Statistical measures for degree distribution.

	REAL NETWORK	ARTIFICIAL NETWORK
Mean	4.05	3.80
Standard deviation	2.20	2.12
Mode	4	4
Skewness	7.86	5.49
Kurtosis	92.87	56.49

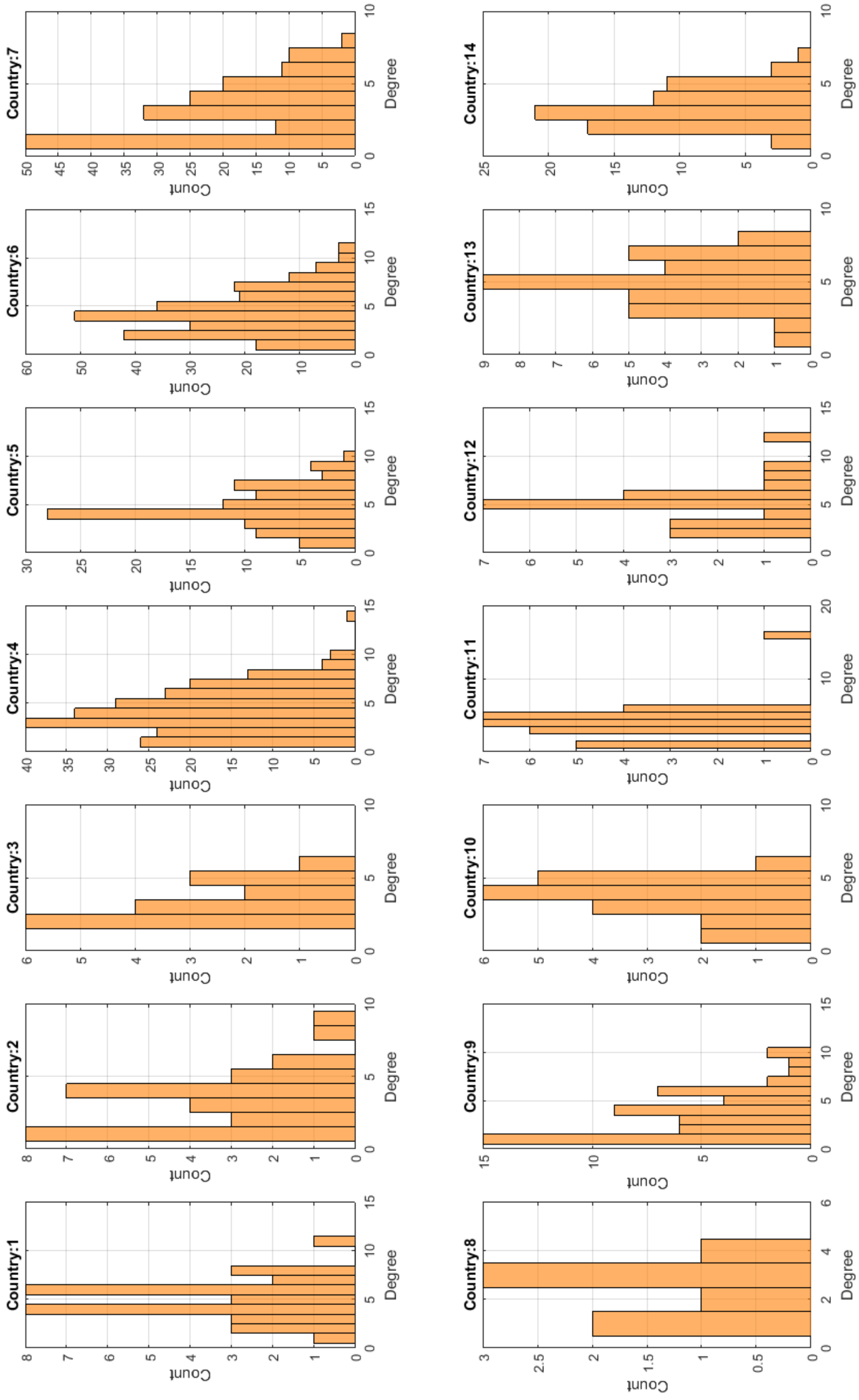

Figure 5-6. Degree distribution for countries in the real European Network.

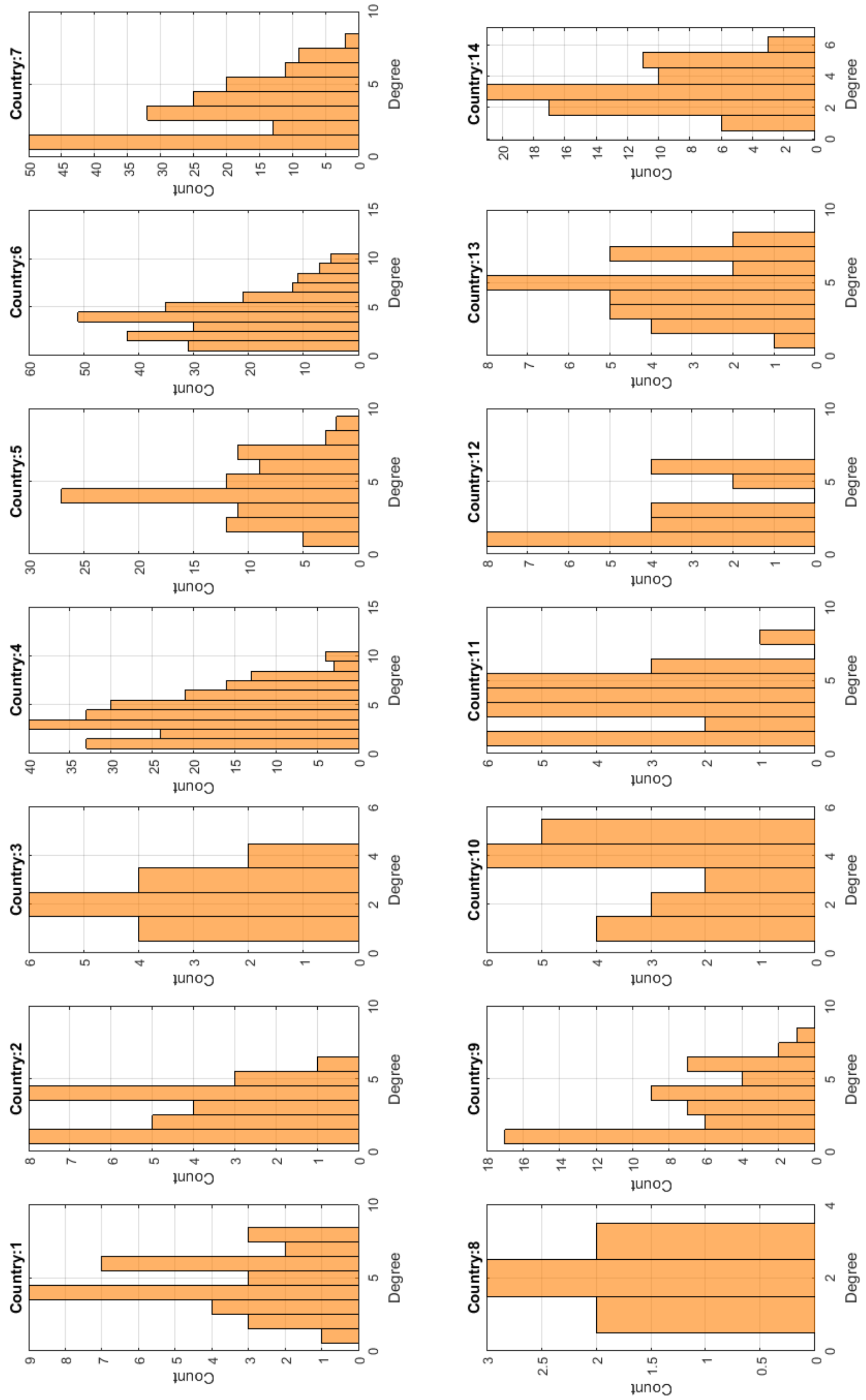

Figure 5-7. Degree distribution for countries in the artificial European network.

5.3.2. *Betweenness centrality*

It can be defined as the number of times a node is located in the shortest path between two nodes, considering all the possible combinations in the graph. It represents a measure of the centrality of a node in the network.

A node with high level of betweenness centrality has a central position in the graph. It is in the shortest path among a high share of combination of pairs. For instance, in Figure 5-8, we can see that the light blue node has a central position, this node is in the path among nodes located at the top and at the bottom.

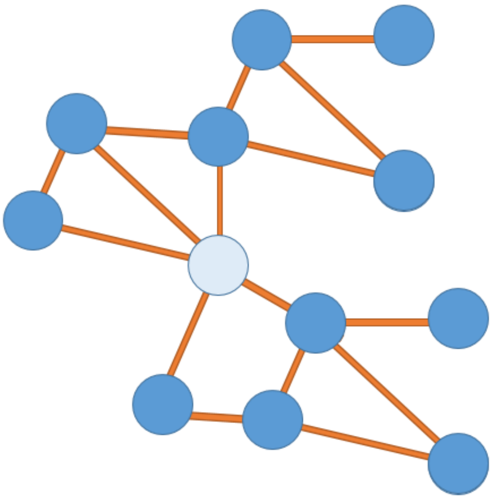

Figure 5-8. An example of graph where light blue vertex has a high value of centrality.

A comparison between the betweenness centrality of the real network and the artificial network is shown in the Figure 5-9.

As in the case of degree, both distributions are similar. In both cases the most frequent value is zero (Table 5-5). In the artificial network there are more nodes that are not in the shortest path between other two. Increasing the number of lines, for instance with international connections, this value can be lowered.

Table 5-5. Statistical measures for betweenness centrality distribution.

	REAL NETWORK	ARTIFICIAL NETWORK
Mean	1.318 e4	1.05 e4
Standard deviation	2.637 e4	3.71 e4
Mode	0	0
Skewness	6.51 e13	3.92 e14
Kurtosis	9.92 e18	1.52 e20

In both cases the mean of the distribution is similar. We find the highest difference in the kurtosis, as we said for the degree, it is due to the tail of the distribution. In the artificial network values are more dispersed, and the tail is bigger.

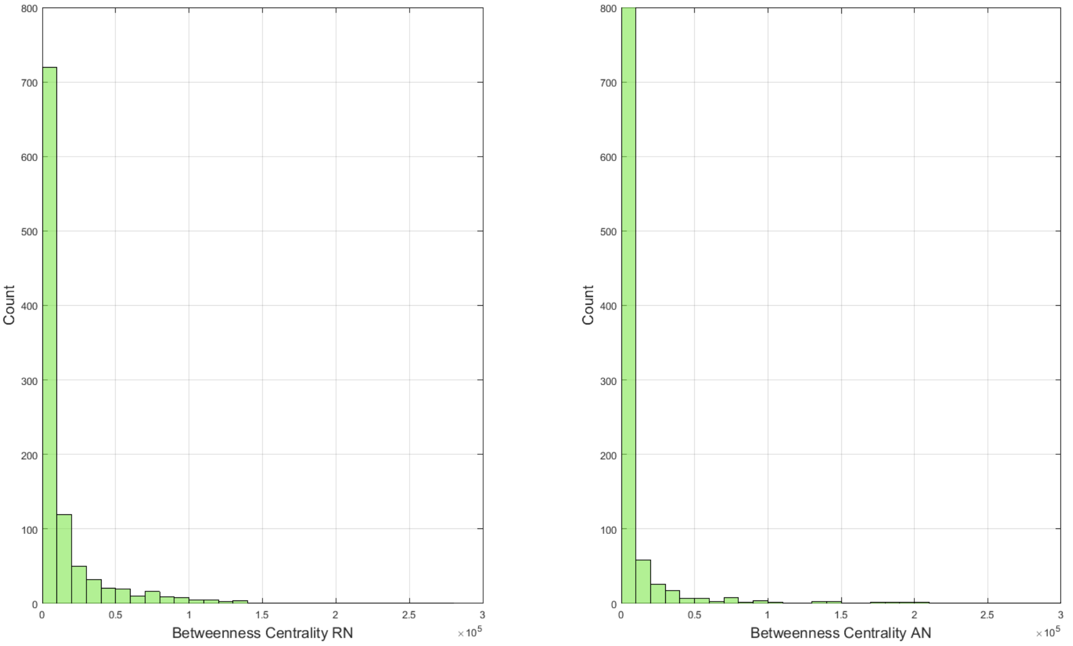

Figure 5-9. Betweenness centrality for the real (left) and artificial (right) European networks.

5.3.3. *Length of lines*

To define the length of a line we used the Euclidean distance between two nodes. This was used as the fitness function that conditions the decision of adding a line to the system.

We should remember that in the first step of the algorithm there was no length restriction, so some outliers can appear.

The rank of distances will vary among countries. It is supposed that the higher the area where the grid is located the longer the lines that are uses to connect nodes.

Figure 5-10 shows the length of lines distribution for the whole European network.

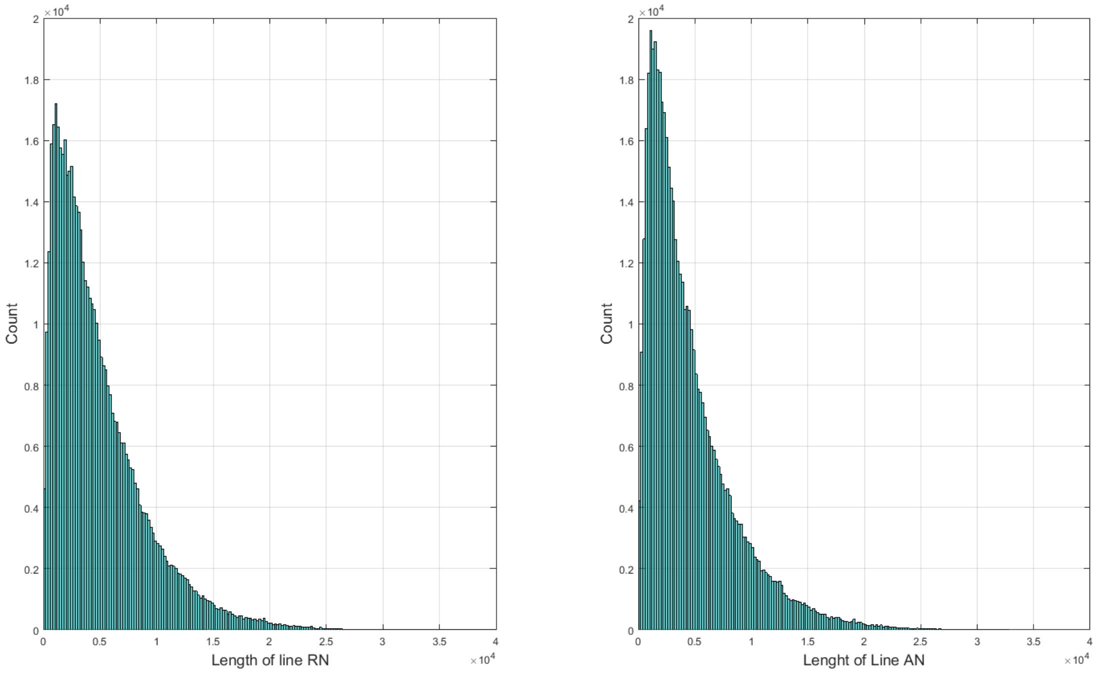

Figure 5-10. Length of lines distribution for the real (left) and artificial (right) European networks.

We observe that the shape of both functions is really similar. In this case, mean and standard deviation have identical values. The value of mode is not significant in this case as it is really difficult that two lines have the same distance (discrete values are considered).

The measure of tailedness is analogous in both cases and both distributions have the same asymmetry.

If needed, we could get a better fit it more than ten bins were used. We should also consider the possibility of add a new step in the algorithm to remove lines. As in the first step length of line is not checked, the model can install lines with a length that does not correspond to the real distribution.

Table 5-6. Statisticial measures for length of lines distribution.

	REAL NETWORK	ARTIFICIAL NETWORK
Mean	4.8982 e3	4.5674 e3
Standard deviation	4.1042 e3	3.8941 e3
Mode	754	1089
Skewness	1.18 e11	9.6928 e10
Kurtosis	2.33 e15	1.463 e15

5.3.4. *Shortest-path length*

The shortest-path length of a graph is the minimum number of edges there are between two nodes.

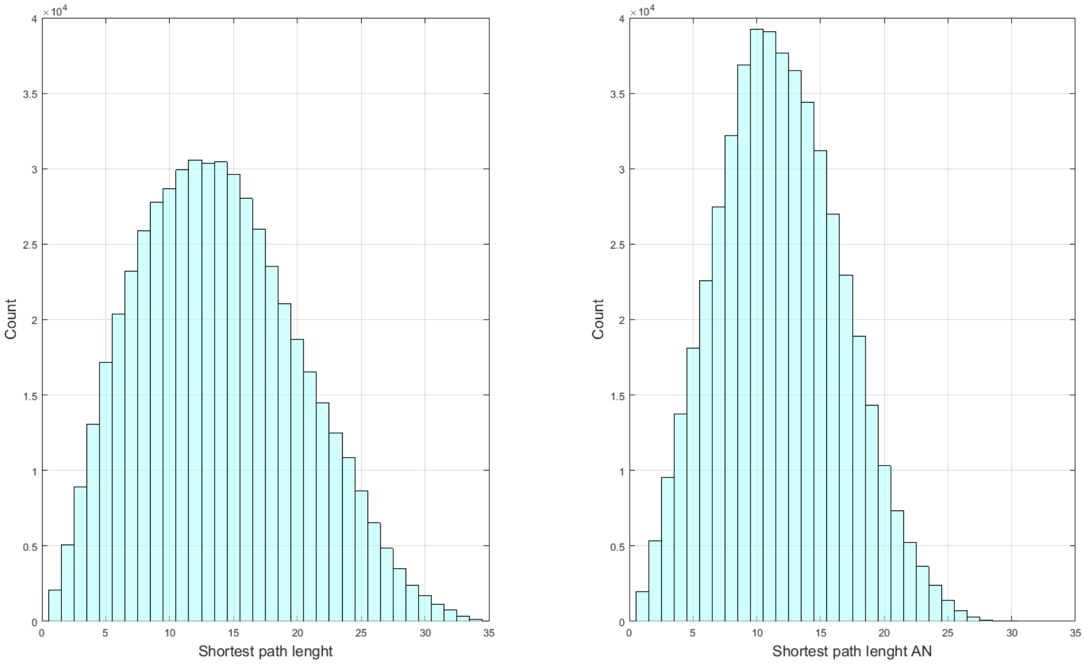

Figure 5-11. Shortest-path distribution for the real (left) and artificial European networks.

If we compare the real and artificial networks, we can observe that in the artificial one distances are shorter than in the real one. In the real network usually there are two edges more than in the artificial one in the shortest path length. Standard deviation is also lower for the artificial network. In the same way, the kurtosis is much bigger in the real distribution. Values are more closed in the synthetic network.

The lower data dispersion can be understood from the point of view of how the network is meshed. If lines are well spread distributed in the area, the path between two nodes will be shorter.

Table 5-7. Statistical measures for shortest path length distribution.

	REAL NETWORK	ARTIFICIAL NETWORK
Mean	13.87	11.82
Standard deviation	6.29	4.83
Mode	12	10
Skewness	77.77	24.71
Kurtosis	4.016 e3	1.453 e3

5.3.5. *Clustering coefficient*

The clustering coefficient is used as a measure to know if nodes tend to form communities inside which there are more links than among them. This metric is really useful in social networks, as it represents the average probability that a pair of one’s friends are friends of each other’.

In many networks clustering is found to have a rough dependence on degree, with vertices with a higher degree having a lower local clustering coefficient on average. Local clustering can be used as a proof for the existence of so-called "structural holes" in a network. It is common in some types of networks, that for the neighbors of a vertex to be connected among themselves. It happens sometimes that these expected connections between neighbors are missing. The missing links are called structural holes. When missing links appear it reduces the number of alternative routes that power can take through the network (Newman, 2010), which has important consequences for reliability.

The clustering coefficient for a network can be calculated following the equation 5-1:

$$C = \frac{3 \cdot number\ of\ triangles}{number\ of\ connected\ triplets\ of\ vertices} \tag{5-1}$$

A triplet is a set of three nodes that are connected.

Figure 5-12 shows an example of calculating clustering coefficient for a simple graph. The value of clustering coefficient is: a) 1, b) 1/3, c) 0.

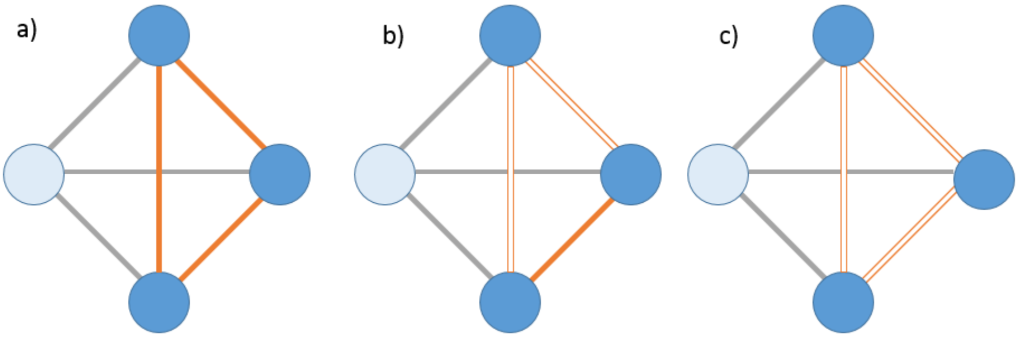

Figure 5-12. Different clustering coefficient for a given graph with different edges.

Figure 5-13 displays the clustering coefficient distribution for the networks we are studying.

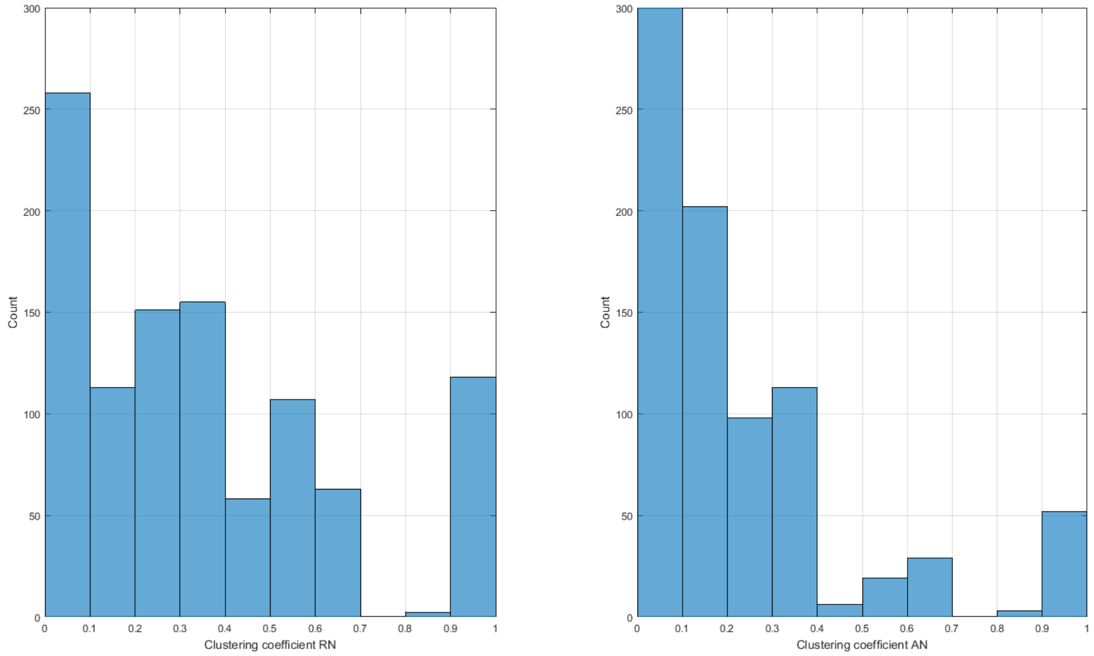

Figure 5-13. Clustering coefficient distribution for the real (left) and artificial (right) European networks.

This metric is the most different among the ones we have analyzed. The mean value is the double in the real network. Neighbors of a node are more linked in that case. The kurtosis is the most identical measure, nevertheless this not have a relevant significance here as the distribution is enclosed between 0 and 1.

We can think that as in previous metrics, adding more lines the distributions will be more similar.

Table 5-8. Statistical measures for clustering coefficient distribution.

	REAL NETWORK	ARTIFICIAL NETWORK
Mean	0.34	0.17
Standard deviation	0.30	0.25
Mode	0	0
Moment (order 3)	0.02	0.03
Moment (order 4)	0.02	0.02

As we have mentioned in the shortest path length, if lines are well spread through all the geographical space, there will not be highly meshed areas and the connections among neighbors of a node (clustering coefficient) will be lower.

Further analyses should be performed in order to identify a pattern in how the network is meshed.

Figure 5-14 displays the correlation between the clustering coefficient and the degree of a node. Taking into account that in the artificial network there are no nodes with a degree higher than 10, both graphs are quite analogous. In both cases we can identify a trend, the higher the degree of a node the lower connect that its neighbors are.

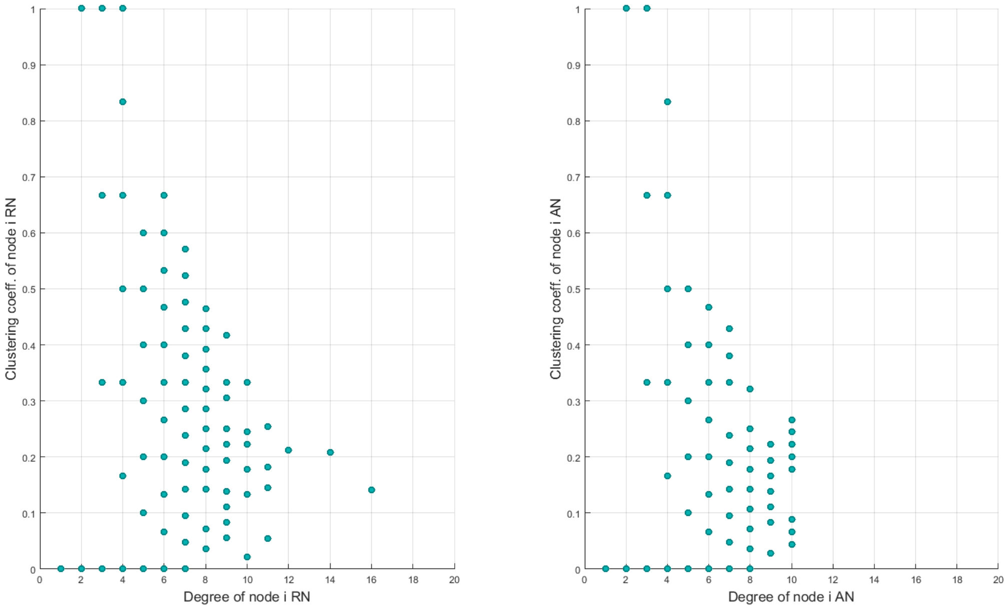

Figure 5-14. Correlation between clustering coefficient and degree of a node for the real (right) and the artificial (left) European network.

6. Conclusions

6.1. Epsilon Disc-Model

This work has implemented a state-of-the-art method for generating spatially embedded networks applied to the power grid (developed by this project's supervisors in reference (Patania et al., 2015)).

Epsilon-disc models have been used in the network science literature to generate spatially-embedded networks. These networks take into account the spatial component but no other additional features that are relevant for power networks.

This project has identified the flaws of the Epsilon-disc model in terms of not fitting the descriptive distributions of the network and not taking into account economic information such as generation cost in the development of the network.

Fixing the value of the maximum epsilon radius was the main problem we found. We did not find a way to establish it. A high value will give networks with a high degree of mesh. Low radius will create grids with isolated areas where demand may be not supplied.

We consider that the algorithm can be useful for other types of networks as social ones, nevertheless it is not completely useful for power networks as it does not consider some features like generation costs or reliability of the system that are essential in the real design of them. Some improvements could have been introduced, nevertheless we considered that it was better to develop a new model.

6.2. The Trade-off Model

This project has developed a new generative network model that overcomes the flaws observed in the epsilon-disc model. The new model displays properties that are more consistent with the ones of actual networks.

With this model we not only ensure that all the demand is supplied but also we can fit any degree distribution adding lines in a flexible way.

This is the main contribution of this research. The developed method can be used not only to generate power networks but also in other types of networks.

The Trade-off Model is based on a preferential attachment model that effectively shifts low-degree nodes towards the high degree region by adding edges smartly.

The results proved that the algorithm is able to fit a general degree distribution that is taken from input data.

The second step of the algorithm does not consider the cost of the system. To avoid that far nodes have the same probability to be linked than nearby ones, the action of adding a line is conditioned to the length distribution of lines.

It is difficult to emulate the process followed by TSOs to add new lines to increase reliability. Several factors can influence this decision, including economic or political ones. The algorithm will generate lines randomly based only on distance with is correlated to the cost of installing a line.

We have generated random networks taking the European one as a template. Our first conclusion is that the algorithm can be used to create networks that are similar to existing ones. Degree distributions are almost identical. They are also similar from the point of view of betweenness centrality, length of lines and shortest path length.

There was a larger difference in the case of the clustering coefficient.

We think that this difference is due to the way the grid is meshed. If we see the real network, there is not a pattern in the mesh of the network. Depending on whether the mesh is spread across the space or is concentrated in specific areas, we will have more or less similar networks. This can justify the fact that in the artificial network the shortest path length between nodes is lower.

From the point of view of the length of the lines, both networks are similar. Only some outliers appear in the artificial network as no restrictions about distance were made in the first step.

The line length distribution can be approached more accurately considering a high number of bins in the histogram for the fitness distribution.

It will be interesting to replicate the European system considering random nodes as well (in this case study nodes were fixed and only connections were generated), the area should be well defined, as it is not a simple square. The case of Spain and Portugal can be approached better with a square. The European network can be replicated dividing it into different subnetworks.

We also concluded that we can create networks that have different properties depending on the area. We applied this to Europe, where the network is a set of 14 small different networks. We can create completely independent random networks that can be linked, resulting a network with clear differenced areas. This could be a way to avoid the problem of some networks in which the degree of mesh depends on the zone.

It is clear that we are not going to create networks that are exactly equal to the one that we are using as a template. That is not the objective of the project. We try to create artificial networks that are consistent with the real ones. There is not only one pattern that have been follow in the generation of the existing network. There are a lot of properties that should be considered. We have only considered the ones we think have more relevant in the process of network design. They are good to recreate existing ones, and to be used in studies about the expansion of existing networks.

If we want to use the algorithm to create new networks for emerging countries this second step it is not the most accurate, as the process should be a pseudo-optimization process based on cost, emissions, etc.

6.3. Future Research

The goal of this work was to create a model that can generate artificial networks that are similar to the real ones. Although there are some papers that have faced this question for other types of networks, this had not been properly studied for the power network. We have proposed an algorithm that can generate artificial networks with a high degree of similarity as we have tested in section five. Nevertheless, the algorithm can increase its degree of accuracy. With the improvements, the result will be more accurate for greenfield expansion, where optimization algorithms require a high computational time.

Some of the improvements that can be developed in next steps are:

To improve the fitting to a degree distribution, a function should be implemented to remove lines. For instance, in the case of Europe, in the first step, the trade-off was implemented not taking into account constraints of nationality. In this case we can have really long lines that are undesirable as they do not appear in the real case. Even the degree distribution obtained in the first step will require not only shift it to the right but also to the left, so following the same pattern to add line a new process can be defined to remove existing lines. Here, the main consideration should be that the process should ensure that all demand is met. If an essential line for the system is removed, another one should be added.

As explained, an N-1 criterion is used by the TSO to ensure the reliability of the system. Our algorithm added random transmission lines to fit degree distribution. However, the N-1 criterion is not checked explicitly. A new step can be introduced before fitting the degree distribution to add new lines following this criterion. As a first consequence of the N-1 criterion, there will not be any node with only one line, all of them will have at least 2 lines. As we saw in the example of Europe, the number of nodes with only one lines was high. Even in the example shown in reference (Patania et al., 2015) in the Spanish system nodes with only one line were the most frequent. Therefore, the correct way to implement this criterion should be studied carefully.

One of the flaws of the proposed algorithm is that it does not take into account interconnection substations. It could install different lines very close geographically, so that it should make more sense to join them by means of a substation. It should be possible to establish the connection to an intermediate point of an existing line if there are not active capacity constraints (the existing line can carry more power). This should be implemented in a computational-efficient way, as we are increasing the number of ways to connect at the first step.

An improvement for the algorithm can be introduced in the node location. In the algorithm described you can introduce the location of nodes or the algorithm can generate them randomly. With the random option, they are placed following a uniform distribution. With topological information about the place where the network will be located, the algorithm should be able to decide where to place these nodes. For instance, on a river the probability of have a hydroelectric plant will be higher than zero, while outside it should be zero. This system should read not only topological information but also information related to wind or solar radiation. This can be a really useful improvement if this tool is used to generate completely new power networks.

Furthermore, the improvements for the algorithm described and following the motivation for this project, the next step will be to use random networks to test the optimization algorithms. With a large number of instances and following the framework of K. Smith-Miles we will be able to approach a solution to answer our motivation question of when it is better to use a Genetic Algorithm or Benders' decomposition. .

7. REFERENCES

Barabasi, A. L., & Albert, R. (1999). Emergence of scaling in random networks. Science (New York, N.Y.), 286(5439), 509-512.

Barmpoutis, D., & Murray, R. M. (2010). Networks with the smallest average distance and the largest average clustering. ArXiv Preprint arXiv:1007.4031,

Bianconi, G., & Barabási, A. (2001a). Competition and multiscaling in evolving networks. EPL (Europhysics Letters), 54(4), 436.

Bianconi, G., & Barabási, A. (2001b). Bose-einstein condensation in complex networks. Physical Review Letters, 86(24), 5632.

Chaniotis, T. (2012). Security of supply , sustainability, competitiveness. A european approach.Entso-e.

i Cancho, R. F., & Solé, R. V. (2003). Optimization in complex networks. Statistical mechanics of complex networks (pp. 114-126) Springer.

Duro, C. (2014). Transmission expansion planning using a Genetic Algorithm. Proyecto Fin de Carrera, ICAI. Madrid 2014.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, 7(1), 48-50.

Lumbreras, S. (2014). Decision support methods for large-scale flexible transmission expansion planning. Unpublished

Lumbreras, S., & Ramos A. (2015). Improvements to Benders' Decomposition. A Practical Evaluation Using a Transmission Expansion Planning Problem. Madrid 2012.

Manna, S. S., & Sen, P. (2002). Modulated scale-free network in euclidean space. Physical Review E, 66(6), 066114.

Newman, M. (2010). Networks: An introduction Oxford University Press.

Patania, A., Young, J., Lumbreras, S., Pereda, M., Bertazzi, I., Citron, D., et al. (2015) Part I: Generating random networks that are consistent with power transmission.

Pereda, M., Bertazzi, I., Lumbreras, S., Patania, A., Young, J., Citron, D., et al. (2015). Part II: An agent-based model for transmission expansion planning.

PRICE, D. J. (1965). Networks of scientific papers. Science (New York, N.Y.), 149(3683), 510-515.

Prim, R. C. (1957). Shortest connection networks and some generalizations. Bell System Technical Journal, 36(6), 1389-1401.

Smith-Miles, K., Baatar, D., Wreford, B., & Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. Computers & Operations Research, 45, 12-24.

Watts, D. J., Dodds, P. S., & Newman, M. E. (2002). Identity and search in social networks. Science (New York, N.Y.), 296(5571), 1302-1305.

8. EXHIBIT 1: TRADE-OFF MODEL FIRST STEP

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                             INPUTS                              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Area localización:

length_a=1000;

length_b=1000;

%Generación:

Mix = [0.15 , 0.3, 0.45 , 0.6, 0.75, 0.9, 1];

%Demanda:

expomean=66;

n_sink=50;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%     FASE 1: Localizar generación y demanda                      %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Source_0=source_generation(n_source, length_a, length_b, Mix);

Sink_0=sink_generation(n_source, length_a, length_b, expomean,
                                             sum(Source_0(:,4)),
sum(Source_0(:,5)));

n_source=max(size(Source_0));

n_node=n_sink+n_source;

Source=ordenasource(Source_0);

Sink=ordenarsink(Sink_0);


Node=[Source(:,1), Source(:,2) ; Sink(:,1), Sink(:,2)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%   FASE 2: Desarrollo básico de la red                            %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Dist=node_distance(Node);

fac=1.2;

e_max=min(max(Dist));

Lines_eb0=(node_linkage(delta_e, e_max, Dist, fac));

plot_lines(Node, Lines_eb0, Source,Sink, n_sink, n_source)

grafo=graph(Dist.*Lines_eb0);

Lines_f2=f_tradeoff_perd(grafo, Dist, Source, Sink);

plot1_lines(Node, Lines_f2(:,:,1), Source,Sink, n_sink,
               n_source,Lines_f2(:,:,2))
```

9. EXHIBIT 2: FUNCTIONS I

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                  Caracterización Demanda                         %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Sink=sink_generation(n_source, length_a, length_b, expomean,
lb, ub, coef )

n_sink=round(ub/(expomean*coef));

Sink=ones(n_sink,3);

Sink(:,1)=randi(length_a,n_sink,1);

Sink(:,2)=randi(length_b,n_sink,1);

Sink(:,3)=random('Exponential', expomean, n_sink, 1);

return Sink;
```

Exhibit 2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                   Caracterización Generación                    %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function sources=source_generation(n_source, length_a, length_b, Mix_c)

Source=ones(n_source,7);

Source(:,1)=randi(length_a,n_source,1);

Source(:,2)=randi(length_b,n_source,1);

for i=1:n_source

            aux2=rand(1);

            aux3=rand(1);

            if aux2<Mix_c(1)

            Source(i,3)=1;

            Source(i,4)=10*(1+randi([0,10])/10);

            Source(i,5)=500*(1-randi([0,9])/100);

            Source(i,6)=0;

            elseif (aux2<Mix_c(2) && aux2>Mix_c(1))
            Source(i,3)=2;

            Source(i,4)=10*(1+randi([0,10])/10);

            Source(i,5)=500*(1-randi([0,9])/100);

            Source(i,6)=0;

            elseif (aux2<Mix_c(3) && aux2>Mix_c(2))

            Source(i,3)=3;

            Source(i,4)=10*(1+randi([0,10])/10);

            Source(i,5)=500*(1-randi([0,9])/100);

            Source(i,6)=0;
```

```
elseif (aux2<Mix_c(4) && aux2>Mix_c(3))

Source(i,3)=4;

Source(i,4)=1000*(1+randi([0,2])/20);

Source(i,5)=1500*(1-randi([0,4])/30);

Source(i,6)=15;

elseif (aux2<Mix_c(5) && aux2>Mix_c(4))

Source(i,3)=5;

Source(i,4)=500*(1+randi([0,4])/20);

Source(i,5)=1000*(1-randi([0,4])/20);

Source(i,6)=20;

elseif (aux2<Mix_c(6) && aux2>Mix_c(5))

Source(i,3)=6;

Source(i,4)=100*(1+randi([0,10])/10);

Source(i,5)=500*(1-randi([0,5])/100);

Source(i,6)=25;

elseif (aux2<1 && aux2>Mix_c(6))

Source(i,3)=7;

Source(i,4)=100*(1+randi([0,10])/10);

Source(i,5)=500*(1-randi([0,5])/100);

Source(i,6)=50;

end

if (aux3<0.95)

    Source(i,7)=1;

Else
```

Exhibit 2

```
                Source(i,7)=0;
            end
end
sources=Source;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                       Ordenar Demanda                           %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function ordenado=ordenarsink(Sink)

[B,k]=sort(Sink(:,3), 'descend');

orden=[Sink(k,1) Sink(k,2) B Sink(k,4) Sink(k,5)];

ordenado=orden;
```

Exhibit 2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                        Ordenar Generación                       %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function ordenado=ordenasource(Source)

[B,k]=sort(Source(:,5), 'descend');

orden=[Source(k,1) Source(k,2) Source(k,3)  Source(k,4) B Source(k,6)
Source(k,7) Source(k,8) ];

ordenado=orden;
```

Exhibit 2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                  Distancia entre nodos                          %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function matrix=node_distance( node)


Dist=(zeros(size(node)));

n_node=max(size(node));

for i=1:(n_node)

    for j=(i+1):(n_node)

        Dist(i,j)=round(sqrt((node(i,1)-node(j,1))^2+(node(i,2)-
node(j,2))^2));

        Dist(j,i)=Dist(i,j);

    End

end


matrix=Dist;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                    Unión nodos EDM                              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function lines_0=node_linkage(delta_e, e_max,  Dist, fac)

e=0;

Lines=(zeros(size(Dist)));

n_node=max(size(Dist));

lines_per_node=int8(zeros(1, n_node));

aux=1;

while(e<e_max)

    e=e+delta_e;

    for i=1:n_node

        aux=aux+lines_per_node(i)*fac;

        for j=i+1:n_node

            if (Lines(i,j)==0 && Dist(i,j)*aux<e )

                Lines(i,j)=1;

                Lines(j,i)=1;

            end


        end

        aux=1;

    end

    lines_per_node=sum(Lines);
end


lines_0=Lines;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                    PLOT                                          %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function plot4(Lines_red,nodes, Source_0_con,Sink_0, Source_0_ren,
Source_0)

nodes=nodes*8000;

axis equal

figure

hold on

for i=1:1025

    for j=i+1:1025

        if(Lines_red(i,j)==1)

           plot([nodes(i,2),nodes(j,2)],[nodes(i,3),nodes(j,3)], 'b');

        end

    end

end

aux1=max(Source_0_con(:,5));

aux2=min(Source_0_con(:,5));

aux3=aux1-aux2;

pr=(Source_0(:,6));

c1=(1*(pr>(aux1-0.6*aux3))+0.8*(pr<(aux1-0.6*aux3) & pr>(aux1-
0.7*aux3))+0.6*(pr<(aux1-0.7*aux3) & pr>(aux1-0.8*aux3))+0*(pr<(aux1-
0.8*aux3) ));

c2=(0*(pr>(aux1-0.2*aux3))+0.4*(pr<(aux1-0.2*aux3) & pr>(aux1-
0.3*aux3))+0.6*(pr<(aux1-0.3*aux3) & pr>(aux1-0.4*aux3))+0.8*(pr<(aux1-
0.4*aux3) & pr>(aux1-0.5*aux3))+1*(pr<(aux1-0.5*aux3) ));

c3=(0*(pr>(aux1-0.3*aux3))+0.2*(pr<(aux1-0.3*aux3) & pr>(aux1-
0.6*aux3))+0.4*(pr<(aux1-0.6*aux3) & pr>(aux1-0.8*aux3))+0*(pr<(aux1-
0.8*aux3) ));

colour=[c1 c2 c3];

grey=[0.5 0.5 0.5];
```

Exhibit 2

```
max1=max(Sink_0(:,3));

s1=(Sink_0(:,3)/max1*75);

scatter(Sink_0(:,1),Sink_0(:,2),s1*2.5,grey, 'filled')

s2=(Source_0(:,4)/max1*75);

scatter(Source_0(:,1),Source_0(:,2),s2*2.5,colour, 'filled')

axis equal

set(gcf, 'color', [1 1 1])

axis off

end
```

10. EXHIBIT 3: TRADE-OFF MODEL SECOND STEP

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                             TOM 2°                              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Lines=Lines_f2(:,:,1);

gnode=nodes(:,1);

ngnode=max(size(gnode));

gLines=zeros(ngnode,ngnode);

[a,b]=size(Lines);

n_lines=a;

for i=1:n_lines

   for j=i+1:n_lines

        if(Lines(i,j)==1)

            aux1=Node(i,3);

            aux2=Node(j,3);

            aux3=find(gnode==aux1);

            aux4=find(gnode==aux2);

            if(aux3~=aux4)

            gLines(aux3,aux4)=1;

            gLines(aux4,aux3)=1;

            end

        end

    end
```

Exhibit 6

```
end

geoDist=node_distance([nodes(:,2)*8000, nodes(:,3)*8000]);

gnLines=gLines;

currentgrafo=graph(gLines);

grad=degree(currentgrafo);

info=[grad country];

clear currentgrafo


[Lines_red]=f_pat_v2(gLines, info,geoDist );

plot4(Lines_red,nodes, Source_0_con, Sink_0, Source_0_ren, Source_0);
```

11. EXHIBIT 4: FUNCTIONS II

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                      P. Attachment                               %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [n_Lines]=f_pat_v2(Lines,info1,Dist)

[auxa rdist]=f_d_c(info1(:,2),Dist,LinEurope);
auxb=f_d_c(info1(:,2),Dist,Lines);
ma_prob=auxa-auxb;
ma_prob(ma_prob<0)=0;

matriz_auxiliar=f_objective_values();
n_node=max(size(Lines));
n_Lines=Lines;
cond=zeros(14,1);
q=0;
while (q<20)
q=q+1;
for c=1:14
    if (cond(c)==0)
        aux1=find(info(:,2)==c);
        grado=info(:,1);
        a=grado(aux1);
        da=ones(10,1);
```

Exhibit 4

```
da0=max(size(find(a==0)));

da(1)=max(size(find(a==1)));

da(2)=max(size(find(a==2)));

da(3)=max(size(find(a==3)));

da(4)=max(size(find(a==4)));

da(5)=max(size(find(a==5)));

da(6)=max(size(find(a==6)));

da(7)=max(size(find(a==7)));

da(8)=max(size(find(a==8)));

da(9)=max(size(find(a==9)));

da(10)=max(size(find(a>9)));


r1=rdist(c,1);

r2=rdist(c,2);

r3=rdist(c,3);

r4=rdist(c,4);

r5=rdist(c,5);

r6=rdist(c,6);

r7=rdist(c,7);

r8=rdist(c,8);

r9=rdist(c,9);

r10=rdist(c,10);

r11=rdist(c,11);
```

```
dt=matriz_auxiliar(c,:)';

pd=da-dt;


    for i=1:n_node
       for j=1:n_node


           D=Dist(i,j);
           r=(1*( D<=r2)+2*(r2<D && D<=r3)+3*(r3<D &&
          D<=r4)+4*(r4<D && D<=r5)+5*(r5<D && D<=r6)+6*(r6<D &&
          D<=r7)+7*(r7<D && D<=r8)+8*(r8<D && D<=r9)+9*(r9<D &&
          D<=r10)+10*(r10<D && D<=r11));

           if(r>0)

               if(info(i,2)==c && info(j,2)==c && i~=j)

                  c1=1;

                  if(info(i,1)==0 || info(j,1)==0 &&
                  n_Lines(i,j)==0 )

                      c1=0;

                      if(info(i,1)==0 && info(j,1)==0)

                          if(ma_prob(c,r)>0 && rand(1)<0.3)

                              n_Lines(i,j)=1;

                              n_Lines(j,i)=1;

                              info(i,1)=info(i,1)+1;

                              info(j,1)=info(j,1)+1;

                              da0=da0-2;

                              da(1)=da(1)+1;

                              da(1)=da(1)+1;

                              pd(1)=pd(1)+2;

                              ma_prob(c,r)=ma_prob(c,r)-1;
                  end

              elseif (info(i,1)==0 && info(j,1)>0)

                  aux1=info(j,1);

                  if (aux1>9)

                  aux1=10;

                  end

                  if(aux1<10)

                      if(max(pd(1:aux1))>0 &&

                                    (pd(aux1+1))<0)

                          if(ma_prob(c,r)>0 &&

rand(1)<0.3)

                              n_Lines(i,j)=1;

                              n_Lines(j,i)=1;

                              info(i,1)=info(i,1)+1;

                              info(j,1)=info(j,1)+1;

                              da(aux1)=da(aux1)-1;

                              da0=da0-1;

                              da(aux1+1)=da(aux1+1)+1;

                              da(1)=da(1)+1;

                              pd(1)=pd(1)+1;

                              pd(aux1+1)=pd(aux1+1)+1;

                              pd(aux1)=pd(aux1)-1;
```

Exhibit 4

```
                                ma_prob(c,r)=ma_prob(c,r)-
                                   1;
                            end
                        end
                    end

                elseif (info(i,1)~=0 && info (j,1)==0)
                    aux1=info(i,1);
                    if (aux1>9)
                              aux1=10;
                    end
                    if (aux1<10
                        if(pd(aux1)>0 &&(pd(aux1+1))<0)
                            if((ma_prob(c,r)>0 &&

rand(1)<0.3) || (D<r5 &&
               rand(1)<0.6) )
                                n_Lines(i,j)=1;
                                n_Lines(j,i)=1;
                                info(i,1)=info(i,1)+1;
                                info(j,1)=info(j,1)+1;
                                da(aux1)=da(aux1)-1;
                                da0=da0-1;
                                da(aux1+1)=da(aux1+1)+1;
                                da(1)=da(1)+1;
                                pd(1)=pd(1)+1;
```

Exhibit 4

```
                     pd(aux1+1)=pd(aux1+1)+1;
                     pd(aux1)=pd(aux1)-1;
                 end
             end
         end
     end

end

 if (c1==1)
     aux1=info(i,1);
     if (aux1>9)
      aux1=10;
     end
     aux2=info(j,1);
     if (aux2>9)
                   aux2=10;
     end
     if(pd(aux1)>0 && aux1<10)
         if( aux2<10)
             if((pd(aux2+1))<0)
                 if(ma_prob(c,r)>0)
                     p=0.3;
                 else
                     p=0;
```

Exhibit 4

```
                           end

                           if(rand(1)<p &&

n_Lines(i,j)==0 &&

                     n_Lines(i,j)==0 )

                                 n_Lines(i,j)=1;

                                 n_Lines(j,i)=1;

                                 da(aux1)=da(aux1)-1;

                                 da(aux2)=da(aux2)-1;

                                 da(aux1+1)=da(aux1+1)+1;

                                 da(aux2+1)=da(aux2+1)+1;

                                 pd(aux1)=pd(aux1)-1;

                                 pd(aux2)=pd(aux2)-1;

                                 pd(aux1+1)=pd(aux1+1)+1;

                                 pd(aux2+1)=pd(aux2+1)+1;

                                 info(j,1)=info(j,1)+1;

                                 info(i,1)=info(i,1)+1;

                                ma_prob(c,r)=ma_prob(c,r)
                                -1;


                               end

                           end

                       end

                   end

               end
```

Exhibit 4

```
                        end

                   end

               end

            end


           if(max(abs(da-dt))<2)

               cond(c)=1;
           end

           if(sum(cond)==13)

               x=x+1;

               if (x==20)
                   cond(c)=1;

               end

           end

    end

end

end

aa=find(info(:,1)==0);

for i=1:max(size(aa))

   v=1;

   for j=1:n_node

       if (Dist(aa(i),j)~=0 && Dist(aa(i),j)<100 && v==1)

           n_Lines(aa(i),j)=1;

           n_Lines(j, aa(i))=1;

           info(j,1)=info(j,1)+1;
```

Exhibit 4

```
           info(aa(i),1)=info(aa(i),1)+1;

           v=0;
       end

   end

end
```

12. Exhibit 5: Functions III

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                             Trade-off                           %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function result=f_tradeoff_perd(grafo, Dist, Source, Sink)
r=0.1;
t=40;
n_source=max(size(Source));
n_sink=max(size(Sink));
Lines=zeros(n_source+n_sink);
capac_lines=Lines;
flow=Lines;
P_disponible=Source(:,5);
for i=1:n_sink
    Pi=Sink(i,3);
    aux= P_disponible<Pi;
    if (length(find(aux==1))==length(P_disponible))
        v=1;
    else
        v=0;
    end

    if(v==0)
        Pi=Sink(i,3);
        aux= P_disponible<Pi;
        generation_cost=Source(:,6);
```

Exhibit 5

```
    c1=100;

    c2=250;

    c3=500;

    c4=1000;

    c5=2000;

    c6=4000;

    p1=8;

    p2=15;

    p3=30;

    p4=50;

    p5=100;

    p6=200;

    cost_instal=p1*(Pi<100 && Pi>0)+p2*(Pi<250 && Pi>100)+p3*(Pi<500
             && Pi>250)+p4*(Pi<1000 && Pi>500)+p5*(Pi<2000 &&
                           Pi>1000)+p6*(Pi<4000 && Pi>2000);

    total_generation_cost=Pi.*generation_cost.*24*365*((1-
                                          1/(1+r)^t)/(r));

    lines_cost=ones(n_source,2);

    losses=0.001;

  lines_cost(:,1)=Dist(1:n_source,i+n_source)*cost_instal+total_gene
  ration_cost.*losses.*Dist(1:n_source,i+n_source);

    dist_2=distances(grafo, 1:n_source, i+n_source);


  lines_cost(:,2)=dist_2*cost_instal+total_generation_cost.*losses.*
  dist_2;

  [min_cost_line,aux1]=min(lines_cost,[],2);

  cost=min_cost_line+total_generation_cost;

  cost(aux)=-1;

   if(sum(cost)==length(cost)*-1)

       break

   end
```

Exhibit 5

```
        j=find(cost==(min(cost(cost>10))));

        j=j(1);

         if aux1(j)==1

             Lines(i+n_source,j)=1;

             Lines(j,i+n_source)=1;

             flow(i+n_source,j)=Pi;

             flow(j,i+n_source)=-Pi;

             capac_lines(i+n_source,j)=c1*(Pi<100 && Pi>0)+c2*(Pi<250 &&
                               Pi>100)+c3*(Pi<500 && Pi>250)+c4*(Pi<1000
&&
Pi>500)+c5*(Pi<2000 && Pi>1000)+c6*(Pi<4000 && Pi>2000);

            capac_lines(j,i+n_source)=capac_lines(i+n_source,j);

             grafo=rmedge(grafo,i+n_source,j);

             grafo=addedge(grafo,i+n_source,j,1);

         else

               path = shortestpath(grafo,i+n_source,j);

               ii=path(1:end-1);

               jj=path(2:end);

               ij=sub2ind(size(Lines), ii, jj);

               ji=sub2ind(size(Lines), jj, ii);

               Lines(ij)=1;

               Lines(ji)=1;

               flow(ij)=(Pi+flow(ij));

               flow(ji)=-Pi+flow(ji);

               Paux=flow(ij);

              capac_lines(ij)=capac_lines(ij)+c1*(Paux<100 &
                                                       Paux>0)+c2*(Paux<250
& Paux>100)+c3*(Paux<500 &
                 Paux>250)+c4*(Paux<1000 & Paux>500)+c5*(Paux<2000 &

Paux>1000)+c6*(Paux<4000 &

Paux>2000).*(capac_lines(ij)==0);
```

Exhibit 5

```
         capac_lines(ji)=capac_lines(ij);

         grafo=rmedge(grafo,ii,jj);

         grafo=addedge(grafo,ii,jj,1);


    end


P_disponible(j)=P_disponible(j)-Pi;


else

    b=1;

    Piaux3=Sink(i,3);

    while(b==1)

        Piaux2=max(P_disponible);

        if(Piaux3>Piaux2)

            b=1;

            Pi=Piaux2;

        Else

            b=0;

            Pi=Piaux3;

        end

        Piaux3=Piaux3-Piaux2;

        aux= P_disponible<Pi;

        generation_cost=Source(:,6);

        c1=100;

        c2=250;

        c3=500;

        c4=1000;

        c5=2000;
```

Exhibit 5

```
            c6=4000;

            p1=8;

            p2=15;

            p3=30;

            p4=50;

            p5=100;

            p6=200;

            cost_instal=p1*(Pi<100 && Pi>0)+p2*(Pi<250 &&
                                                   Pi>100)+p3*(Pi<500 &&
Pi>250)+p4*(Pi<1000 &&
                     Pi>500)+p5*(Pi<2000 && Pi>1000)+p6*(Pi<4000 &&
Pi>2000);

            total_generation_cost=Pi.*generation_cost.*24*365*((1-
                                              1/(1+r)^t)/(r));

            lines_cost=ones(n_source,2);

            losses=0.001;

            lines_cost(:,1)=Dist(1:n_source,i+n_source)*cost_instal+tota
               l_generation_cost.*losses.*Dist(1:n_source,i+n_source);

            dist_2=distances(grafo, 1:n_source, i+n_source);

            lines_cost(:,2)=dist_2*cost_instal+total_generation_cost.*lo
            sses.*dist_2;

            [min_cost_line,aux1]=min(lines_cost,[],2);

            cost=min_cost_line+total_generation_cost;

            cost(aux)=-1;

           if(sum(cost)==length(cost)*-1)

               break

           end

           j=find(cost==(min(cost(cost>10))));

           j=j(1);

            if aux1(j)==1

                Lines(i+n_source,j)=1;
                Lines(j,i+n_source)=1;
```

Exhibit 5

```
                 flow(i+n_source,j)=Pi;
                 flow(j,i+n_source)=-Pi;

                 capac_lines(i+n_source,j)=c1*(Pi<100 &&

Pi>0)+c2*(Pi<250 && Pi>100)+c3*(Pi<500 &&
                                                  Pi>250)+c4*(Pi<1000 &&
Pi>500)+c5*(Pi<2000 &&
Pi>1000)+c6*(Pi<4000 && Pi>2000);

                 capac_lines(j,i+n_source)=capac_lines(i+n_source,j);

                 grafo=rmedge(grafo,i+n_source,j);

                 grafo=addedge(grafo,i+n_source,j,1);

            else


                  path = shortestpath(grafo,i+n_source,j);

                  ii=path(1:end-1);

                  jj=path(2:end);

                  ij=sub2ind(size(Lines), ii, jj);

                  ji=sub2ind(size(Lines), jj, ii);

                 Lines(ij)=1;

                 Lines(ji)=1;

                 flow(ij)=(Pi+flow(ij));

                 flow(ji)=-Pi+flow(ji);

                 Paux=flow(ij);

                 capac_lines(ij)=capac_lines(ij)+c1*(Paux<100 &

Paux>0)+c2*(Paux<250 & Paux>100)+c3*(Paux<500 &
                                         Paux>250)+c4*(Paux<1000 &
Paux>500)+c5*(Paux<2000 &
Paux>1000)+c6*(Paux<4000 &

Paux>2000).*(capac_lines(ij)==0);

                 capac_lines(ji)=capac_lines(ij);

                 grafo=rmedge(grafo,ii,jj);

                 grafo=addedge(grafo,ii,jj,1);
```

Exhibit 5

```
            end

         P_disponible(j)=P_disponible(j)-Pi;

        End


    end



end



nuevo(:,:,1)=Lines;

nuevo(:,:,2)=flow;

nuevo(:,:,3)=capac_lines;

result=nuevo;
```

13. EXHIBIT 6: EPSILON-DISC MODEL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                             INPUTS                             %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

delta_e=75 ;

n_source=25;

length_a=1000;

length_b=1000;

Mix=[0.141 , 0.232, 0.387 , 0.521, 0.705, 0.925, 1];

expomean=66;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%                             MODELO                             %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Source_0=source_generation(n_source, length_a, length_b, Mix);

Sink_0=sink_generation(n_source, length_a, length_b, expomean,
sum(Source_0(:,4)), sum(Source_0(:,5)));

n_sink=max(size(Sink_0));

n_node=n_sink+n_source;

Source=ordenasource(Source_0);

Sink=ordenarsink(Sink_0);

Node=[Source(:,1), Source(:,2); Sink(:,1), Sink(:,2)];

Dist=node_distance(Node);

fac=1.2;

e_max=100;
```

Exhibit 6

```
Lines_eb0=(node_linkage(delta_e, e_max, Dist, fac));

plot_lines(Node, Lines_eb0, Source,Sink, n_sink, n_source)

                      grafo=graph(Dist.*Lines_eb0);
```