

Proyecto Fin de Grado  
Ingeniería de Tecnologías Industriales  
Especialidad de Organización y Producción

Optimización de la topología de una red wifi  
mediante un proceso Heurístico

Autor: Borja Pérez de Guzmán Bermúdez-Coronel

Tutor: Pedro L. González Rodríguez

**Dep. Organización Industrial I**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2016





Trabajo Fin de Grado  
Organización Industrial y Gestión de empresas I.

# **Optimización de la topología de una red wifi mediante un proceso Heurístico**

Autor:

Borja Pérez de Guzmán Bermúdez-Coronel

Tutor:

Pedro L. González Rodríguez

Profesor titular

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016



Proyecto Fin de Grado: Optimización de la topología de una red wifi mediante un proceso Heurístico

Autor: Borja Pérez de Guzmán Bermúdez-Coronel

Tutor: Pedro L. González Rodríguez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal



*A mis padres.*





# Agradecimientos

---

Tras una larga carrera, finalmente se ve la luz al fondo de este primer pasillo, recuerdo todos esos momentos de angustias y alegrías. Todo esto no habría sido posible sin la ayuda de todos aquellos que me han apoyado, por ello quiero agradecer:

A mi padre Javier por su afable carácter y su incondicional apoyo por la causa, por su cariño su afecto y su incontable conocimiento.

A mi madre Silvia, por todas esos tapers para aprovechar el día de biblioteca, por sus consejos en otras cosas menos técnicas pero igual de importantes y porque siempre está ahí para ayudar.

A mis hermanos, porque siempre han sabido animarme y ayudarme a desconectar, porque siempre se han preocupado y me han apoyado.

A el Colegio Tabladilla que me ha formado para llegar hasta aquí.

A mi tutor Pedro Luis que me ha ayudado cuando no veía la forma de hacer las cosas, por su apoyo y sus ánimos para finalizar este proyecto.

*Borja Pérez de Guzmán Bermúdez-Coronel*

# Resumen

---

Los servicios de la luz, el agua y el gas en España están sometidos a distintas leyes de carácter europeo en las que se intenta inmiscuir a los usuarios en el mercado. A razón de esto, se han fomentado políticas que someten a los distribuidores de dichos servicios a un conocimiento de los distintos consumos y obtención general de los datos de forma automatizada y telemática.

La Ley que incumbe a las redes de agua aún no ha entrado en vigor, (entrará en vigor en 2017) sin embargo con el conocimiento previo que se tiene hasta ahora de esta ley, las compañías procurarán adaptarse a las políticas que en ella se exigen (como implantación de un sistema telemático para obtener las facturas) buscando diversas alternativas y evaluando cuál de ellas es la mejor. Entre dichas alternativas, se encuentra el desarrollo de un sistema telemático que comunica los datos almacenados en los distintos contadores de agua hacia la oficina central o local de la empresa suministradora mediante el uso de routers, terminales y concentradores como se explicará más adelante.

El sistema telemático recogerá los consumos de los contadores que estarán instalados en cada vivienda. Los consumos serán emitidos desde los terminales (instalados en las viviendas) hacia a los routers mediante señales. A través de una red de routers conectados entre sí, las señales se irán repitiendo hasta llegar a los concentradores, destino final de dichas señales.

Las condiciones de este problema vienen dadas por otro proyecto (ver Caro 2014) en el que se estudió de forma exacta las distintas soluciones a partir de tres modelos distintos.

El problema a estudiar es la situación óptima de los routers, los cuales recogen la lectura de unos terminales cuya localización no es variable ya que los terminales están situados en los edificios de las localidades. Dichos routers tendrán que hallar un camino hacia los distintos concentradores que de antemano tampoco están localizados. Los objetivos de este Trabajo Fin de Grado son ver cuáles son las mejores rutas a seguir por las señales de los routers así como cuál es el mínimo número necesario de routers y concentradores.

En este Trabajo de Fin de Grado, se va a elaborar un modo constructivo para obtener una solución inicial porque los tiempos de ejecución de los modelos lineales enteros son excesivamente lentos y en cambio, con el método constructivo aquí propuesto, los tiempos de ejecución son mínimos.

<b>Agradecimientos</b>	<b>9</b>
<b>Resumen</b>	<b>10</b>
<b>Índice</b>	<b>11</b>
<b>Índice de Tablas</b>	<b>14</b>
<b>Índice de Figuras</b>	<b>15</b>
<b>1 Justificación y objetivos</b>	<b>17</b>
1.1 Justificación	17
1.2 Objetivos	18
1.2.1 Objetivo General	18
1.2.2 Objetivos Específicos (O.E) y estructura del documento	18
1.2.3 O.E.1: Comprensión detallada del problema	18
1.2.4 O.E.2: Realizar una revisión o estado del Arte	18
1.2.5 O.E.3: Modelo matemático y Grafo asociado	19
1.2.6 O.E. 4: Desarrollo de un proceso heurístico	19
1.2.7 O.E. 5: Discutir las implicaciones prácticas de los resultados obtenidos	19
1.3 Estructura del documento	19
<b>2 Descripción del problema</b>	<b>21</b>
2.1 Redes de comunicación Wireless	21
2.1.1 Clasificación de las redes Wireless	21
2.1.2 Arquitectura de las redes Wireless	22
2.1.3 Redes centralizadas	22
2.1.4 Redes inalámbricas distribuidas	22
2.2 Arquitectura del problema objeto de estudio	23
2.3 Elementos y sus características	24
<b>3 Estado del Arte</b>	<b>11</b>
3.1 Revisión de la literatura del problema de la localización de routers	11
3.2 Caracterización del problema según literatura	11
3.3 Revisión de la literatura de los Árboles de Expansión Mínima	12
3.3.1 Algoritmo de Kruskal	13
3.3.2 Algoritmo de Prim	15
3.3.3 Diferencias	18
<b>4 Grafo asociado y Modelo Matemático</b>	<b>20</b>
4.1 Tipos de Nodos	20
4.2 Tipos de conexiones o arcos	20
4.3 Parámetros y variables	22
4.3.1 Parámetros	22
4.3.2 Variables	22
4.4 Función Objetivo y Restricciones	23
4.5 Modelo Completo	24
4.6 Análisis y topología de las soluciones	24

---

4.7	Ejemplo en un barrio de la localidad de Ginés	27
<b>5</b>	<b>Proceso Constructivo</b>	<b>31</b>
5.1	Pseudo Código	31
5.2	Fase 0: Preprocesado y Ordenación de datos	32
5.3	Fase 1: Asignación de Terminales	32
5.3.1	Fase 1.1 y 1.2: Terminales asignados a un único Router y Determinación de los Routers obligatoriamente activos	32
5.3.2	Fase 1.3: Asignar Terminales a Routers Activos	33
5.3.3	Fase 1.4: Asignar Terminales al Resto de Routers	34
5.4	Fase 2: Conexión de Routers	35
5.4.1	Fase 2.1 Construir Grafo no dirigido	35
5.4.2	Fase 2.2 Construir Bosque	36
5.4.3	Fase 2.3: Construir Árbol de Expansión Mínima	37
5.5	Fase 3: Conexión con concentradores	38
5.6	Fase 4: Obtención del Grafo dirigido y mejoras	39
5.6.1	Mejora 1: Reasignación por niveles	40
5.6.2	Mejora 2: Ruta a la salida más cercana	41
5.7	Fase 5: Cálculo de la Función Objetivo	41
<b>6</b>	<b>Experimentación y análisis de los Resultados</b>	<b>42</b>
6.1	Experimentación	42
6.2	Análisis de los resultados	44
6.3	Ejecución en Matlab	45
6.3.1	Tutorial	45
<b>7</b>	<b>Conclusiones</b>	<b>56</b>
	<b>Referencias</b>	<b>58</b>
	<b>Anexo I. Batería 20-200</b>	<b>59</b>
	<b>Anexo II. Batería 20-400</b>	<b>66</b>



# ÍNDICE DE TABLAS

---

Tabla 2-1. Resumen de características.	26
Tabla 2-2. Resumen de conexiones.	26
Tabla 4-1. Conexiones entre Nodos.	21
Tabla 5-1. Codificación de los datos.	32
Tabla 5-2. Asignaciones obligadas	33
Tabla 5-3. Terminales con conexión a alguno de los routers obligatorios	33
Tabla 5-4. Routers Activos e Inactivos.	35
Tabla 5-5. Selección de Concentradores.	38
Tabla 6-1. Resultados Con Densidad de 10 Terminales por Calle.	43
Tabla 6-2. Resultados Con Densidad de 20 Terminales por Calle.	43

# ÍNDICE DE FIGURAS

---

Figura 2-1. Ejemplo de Red Centralizada.	22
Figura 2-2. Ejemplo de Red Distribuida.	23
Figura 2-3. Ejemplo de TPL.	24
Figura 2-4. Ejemplo de router.	25
Figura 2-5. Ejemplo de concentrador de red.	25
Figura 3-1. Ejemplo HLP.	11
Figura 3-2. Ejemplo del Problema de los puentes de Königsberg y su Grafo Asociado.	13
Figura 3-3. Ejemplo algoritmo de Kruskal.	14
Figura 3-4. Solución Algoritmo Kruskal.	15
Figura 3-5. Ejemplo utilizado para la solución de Prim.	17
Figura 3-6. Solución utilizando el algoritmo de Prim.	18
Figura 4-1. Arcos y nodos.	21
Figura 4-2. Grafo de la solución óptima.	25
Figura 4-3. Grafo de la solución en forma de árbol.	25
Figura 4-4. Nodo ejemplo.	26
Figura 4-5. Algunos arcos no son necesarios.	26
Figura 4-6. Foto aerea de un barrio de la localidad de Gines.	27
Figura 4-7. Arcos permitidos en uno de los barrios de Ginés.	28
Figura 4-8. Representación de la solucionen la imagen aérea.	29
Figura 4-9. Representación del grafo solución.	30
Figura 5-1. Asignación terminales al resto de routers.	34
Figura 5-2. Grafo de conexiones Inicial.	35
Figura 5-3. Arbol de Expansión mínima inicial.	36
Figura 5-4. Grafo de Conexiones Final.	37
Figura 5-5. Árbol de Expansión mínima final.	38
Figura 5-6. Ruta dada por el Árbol de Expansión Mínima.	39
Figura 5-7. Posibles Rutas Alternativas.	40
Figura 5-8. Niveles de los nodos.	40
Figura 5-9. Camino a la salida más cercana.	41
Figura 6-1. Pantalla principal de Matlab.	45
Figura 6-2. Seleccionar archivo.	46
Figura 6-3. Importar datos.	46
Figura 6-4. Fichero asignación obligados.	47

---

Figura 6-5. Fichero asignación terminales sin conectar.	47
Figura 6-6. Fichero cuenta cuantos lleva.	48
Figura 6-7. Fichero buscador sin colocar.	48
Figura 6-8. Fichero terminales sin conexión.	49
Figura 6-9. Fichero routers más conectables.	49
Figura 6-10. Fichero routers definitivos.	50
Figura 6-11. Fichero nuevo grafo routers.	50
Figura 6-12. Fichero ejecución buena.	51
Figura 6-13. Fichero busca sin árbol.	51
Figura 6-14. Fichero busca camino más corto.	52
Figura 6-15. Árbol de expansión mínima y grafo asociado de conexiones.	52
Figura 6-16. Fichero selecciona concentradores.	53
Figura 6-17. Fichero concentrador elegido.	53
Figura 6-18. Fichero calcula con arcos.	54
Figura 6-19. Fichero reasignación de terminales.	54
Figura 6-20. Fichero reasignación por niveles.	55
Figura 6-21. Fichero camino salida cercana.	55



# 1 JUSTIFICACIÓN Y OBJETIVOS

---

*"Si quieres cambiar al mundo, cámbiate a ti mismo."*

*- Mohandas Gandhi -*

## 1.1 Justificación

Actualmente vivimos en un mundo cada vez más conectado, donde las cosas más sencillas se van añadiendo a una red global. El desarrollo de la tecnología conlleva nuevas formas de trabajo y al mismo tiempo desaparecen otras. Estas nuevas formas de funcionar, intentan hacer la vida más cómoda a las personas.

La globalización y la comunicación son dos adjetivos del mundo en el que vivimos actualmente. Esto conlleva la aparición de nuevas normativas cada vez más restrictivas que hay que cumplir. A partir de 2017, una ley de carácter europeo entrará en vigor exigiendo la información de los consumos de distintos servicios como son la luz, el gas o el agua corriente queden telematizados con una disposición de información sobre los consumos instantáneos.

El objetivo de la telematización de los servicios (luz, gas, agua) es la cohesión de todos ellos en una sola red. En esta única red se obtendrá la información de los consumos de todos los servicios, quedando incluidos el alumbrado, semáforos, parkings, etc. Esta unificación conllevaría la aparición de un único centro de control donde estarían disponibles los informes de cada uno de los servicios, facilitando así su análisis.

La distribución y transporte de energía eléctrica y agua, recursos cada vez más escasos y costosos, necesita de redes de gestión donde se pueda conocer a tiempo real todos los parámetros que influyen en el funcionamiento de estas redes, con el objeto de hacerlas cada día más eficaces y fiables. Además, la tendencia de las redes de distribución de transporte de energía y agua es que éstas se integren con otras redes (alumbrado, semáforos, parkings...etc.) con el objeto de que todos los servicios en una ciudad estén telegestionados desde un único centro de control.

Originalmente, la obtención de los diversos consumos era leída de forma analógica y con el paso del tiempo se ha ido renovando por nuevos terminales digitales. La lectura tanto de los manuales como de los digitales se viene haciendo de forma manual por un operario al que se le asigna una determinada ruta, donde miden en los distintos puntos de lectura instalados en cada edificio con un sistema de Walk-By hacia los distintos TPL (Terminal de Punto de Lectura) de cada edificio.

Se han planteado por tanto, una serie de mejoras para las distintas compañías de agua en los sistemas de telelectura que son:

1. Ahorro en la obtención de datos por la sustitución del sistema Walk-By por un sistema completamente automatizado y remoto, donde toda la información esté disponible en las oficinas centrales, pudiendo así ser integrado en sus propios sistemas de gestión.
2. Conocimiento continuo y real de las lecturas.
3. Posibilidad de una mejor gestión de los recursos mediante una predicción de la demanda basada en los datos obtenidos de los consumos lo que a su vez permitirá la monitorización de la red.
4. Un sistema más eficaz en la detección de fugas gracias a la disposición de los tiempos reales de consumo.
5. La detección de posibles fraudes.

6. La aplicación de diferentes políticas de ahorro de agua.
7. Las grandes ciudades evolucionan hacia lo que se conoce como Smart City, donde cada vez más los servicios como el alumbrado público, las redes de semáforos o el tráfico es dirigido como un conjunto para una mejora global de los servicios.

En el presente Trabajo Fin de Grado (TFG) se estudiará cómo conseguir una primera solución al problema de la localización de routers de una red de comunicación inalámbrica de contadores de agua. Esta solución se servirá como punto de partida de posteriores metodologías de resolución.

## 1.2 Objetivos

### 1.2.1 Objetivo General

El objetivo principal de este Proyecto es aportar una primera solución admisible cercana al óptimo mediante un proceso constructivo de la solución. Esta solución serviría posteriormente como punto de arranque de otras heurísticas para acabar de optimizar el problema de la localización de los routers.

### 1.2.2 Objetivos Específicos (O.E) y estructura del documento

A continuación describimos brevemente los objetivos específicos que se van a alcanzar en el estudio y resolución del problema.

#### 1.2.3 O.E.1: Comprensión detallada del problema

En el problema que se estudia en el presente Proyecto se detectan tres elementos: terminales, concentradores y routers. En cada localidad, hay una serie de terminales con unas localizaciones conocidas a priori (en los edificios), estos terminales enviarán los consumos mediante señales hacia los routers. La localización de estos routers no está definida pero sí se conocen las posibilidades de su localización (esquinas de las calles). A estos routers se les asigna los terminales anteriormente nombrados. En último lugar quedan los concentradores, que tienen unas posibles localizaciones y habrá que decidir dónde ubicarlos y crear las rutas desde los routers hasta ellos.

En el siguiente capítulo (Descripción del problema, 2) se hará una descripción más detallada de los elementos anteriormente descritos, así como de sus especificaciones y limitaciones.

#### 1.2.4 O.E.2: Realizar una revisión o estado del Arte

En el tercer capítulo realizaremos un análisis bibliográfico de los trabajos que se han desarrollado hasta el momento y trataremos de buscar similitudes con nuestro problema para tener material en el que apoyarnos.

Clasificaremos nuestro problema en función de las características que este tenga, es decir, en función de cómo sea la solución, del tipo de función objetivo, según el número de Hubs, según la capacidad, etc.

Finalmente en este apartado haremos una revisión de los algoritmos para la creación del árbol de expansión mínima y veremos algunos de ellos.

### **1.2.5 O.E.3: Modelo matemático y Grafo asociado**

Este Proyecto propone un nuevo enfoque a un Proyecto que ya se expuso (Sevilla, 2014. Localización Óptima de Routers de una Red de Comunicación Inalámbrica de Contadores de Agua.), cuyo autor Salvador Caro Jiménez desarrolló el modelo matemático del problema Localización Óptima de Routers de una Red de Comunicación Inalámbrica de Contadores de Agua. Con el estudio de dicho modelo tendremos una mayor perspectiva del problema con el objetivo de facilitar la comprensión del método constructivo propuesto.

En este apartado se verá a partir de un ejemplo como las soluciones óptimas a estos problemas tienen estructura de árbol, en concreto al Árbol de Expansión Mínima.

La heurística desarrollada se aplicará en grafos con los rangos usuales de terminales por calles, lo que significa que el número de terminales por calle oscilará entre 10 y 20. Se verá el RPD (relative percent difference) entre la solución constructiva y la óptima que obtuvo Salvador Caro.

### **1.2.6 O.E. 4: Desarrollo de un proceso heurístico**

En este apartado se desarrollará paso a paso el proceso de construcción y se mostrarán los pasos a partir de un ejemplo. Para la explicación de la heurística se mostrará un ejemplo y se ejecutará sobre el la heurística desarrollada.

Por otro lado, se explicará detalladamente cada parte del proceso constructivo y se justificará como se ha decidido construir de esta manera la solución.

Se mostrarán los resultados del ejemplo y se verán las distintas representaciones gráficas del ejemplo así como los cambios.

### **1.2.7 O.E. 5: Discutir las implicaciones prácticas de los resultados obtenidos**

Durante este capítulo se analizarán los resultados obtenidos y los posibles nuevos algoritmos para aplicar a estas soluciones de partida.

Así mismo, se verán las ventajas y las limitaciones que se han tenido durante el desarrollo del trabajo.

En último lugar se explicará cómo hacer una ejecución en Matlab de lo que se ha desarrollado en el TFG con uno de los ejemplos de la batería aportada por el tutor.

## **1.3 Estructura del documento**

La estructura que se va a seguir a lo largo del documento aquí presente tendrá la siguiente forma:

El presente documento estará formado por siete capítulos donde se desarrollará cada uno de los apartados. Además se incluyen dos anexos en los que se muestran un ejemplo en cada uno de los dos tamaños de problema a los que se ha aplicado el algoritmo.

En primer lugar haremos una descripción del problema en el apartado 2. En este apartado veremos los distintos tipos de redes Wireless, sus tipos y sus distintos tipos de arquitecturas. Una vez hecho esto pasaríamos a ver la estructura de nuestra red, sus elementos y las características de ellos.

En segundo lugar en el apartado 3 haremos una revisión del estado del arte de los problemas de localización de routers. En este apartado veremos algunos de los problemas similares al nuestro para ver si podemos enmarcarlos entre ellos y finalmente hablaremos de las características y restricciones específicas de nuestro problema. Finalmente haremos una revisión de los distintos tipos de árboles de expansión mínima y los algoritmos más utilizados para obtenerlos.

En el cuarto apartado veremos el modelo matemático que se utiliza para la obtención de la solución óptima. En este apartado hablaremos de los tipos de nodos existentes y los arcos permitidos entre ellos. Así mismo estudiaremos las soluciones óptimas dadas por el modelo y la similitud que este tiene con el árbol de expansión

mínima.

En el quinto apartado estudiaremos el algoritmo que se ha desarrollado y se completará la explicación del mismo con un ejemplo. Del mismo modo en este apartado se desarrollara el pseudocódigo asociado a la heurística y se profundizará en el análisis de la heurística.

En el sexto apartado veremos las soluciones que la heurística propuesta nos da. Veremos si las soluciones son cercanas al óptimo o se alejan mucho de él. Del mismo modo, en este apartado se mostrará como ejecutar el algoritmo desarrollado en el programa informático Matlab.

Además, el documento contará con un apartado para las referencias y dos anexos. En ambos anexos se muestran los datos de dos problemas que se han utilizado en la heurística.

Finalmente en el sexto capítulo se estudiarán las conclusiones del proyecto. Se verá si se han cumplido los objetivos específicos y se comentarán. Así mismo se verán las posibles ampliaciones de este proyecto.

# 2 DESCRIPCIÓN DEL PROBLEMA

---

*“Mientras me quede algo por hacer, no habré hecho nada”*

*- Julio Cesar (100-44 A.C.) -*

**E**n este apartado se estudiarán las restricciones tecnológicas. A su vez se explicará el tipo de redes y sus arquitecturas, conexiones disponibles, las limitaciones y los modelos de los dispositivos con sus características.

## 2.1 Redes de comunicación Wireless

Las redes inalámbricas son sistemas de comunicación, las cuales emplean el aire como medio transmisor donde las ondas de radiofrecuencia son las que llevan la información entre los diferentes dispositivos. Estas redes, cada vez más desarrolladas y de un mayor alcance, no necesitan de cableado, diferenciándose así de las redes convencionales.

Las redes inalámbricas también tienen algunas limitaciones y en general algunos problemas, como por ejemplo:

- Las interferencias que se producen entre diferentes dispositivos. Estas interferencias pueden ser provocadas mediante inhibidores o simplemente por otros dispositivos, ya que envían señales a una misma frecuencia.
- Otro problema es que la potencia de alcance de las diversas señales se ve afectada por los elementos físicos que han de atravesar, (árboles, farolas o muros).

### 2.1.1 Clasificación de las redes Wireless

Las redes inalámbricas (Wireless) se agrupan según el alcance máximo de las ondas de transmisión y se clasifican principalmente en tres tipos de redes:

1. WPAN (Wireless Personal Área Network): son las redes de menor alcance. Dicho alcance ronda los diez metros. Entre ellas se encuentran los RFID (Radio Frequency IDentification) y el Bluetooth, los cuales tienen bajos consumos energéticos y sirven para comunicar distintos dispositivos.
2. WLAN (Wireless Local Área Network): estas redes se caracterizan por poseer un alcance aproximado de cien metros. El uso habitual de las redes WLAN es conectar ordenadores en un mismo espacio de trabajo o en hogares, así como para establecer conexiones entre las redes tipo WPAN.
3. WMAN (Wireless Metropolitan Area Network): Este tipo de redes son empleadas a nivel metropolitano, con el objetivo de mantener conectada una ciudad o localidad.
4. Globales: Estas redes serían el siguiente nivel de conexiones de tipo Wireless, las cuales conectan por completo Comunidades Autónomas o países.

## 2.1.2 Arquitectura de las redes Wireless

La arquitectura de las redes Wireless se divide en dos grupos: las redes centralizadas y distribuidas. La principal diferencia entre ambas reside en cómo se regentan las conexiones, como están distribuidas y donde está el corazón de las mismas.

### 2.1.3 Redes centralizadas

En este tipo de redes hay un único dispositivo que dirige la red y sus accesos. En grandes espacios, con diversos accesos y envíos de información, para una mayor cohesión de los datos entrantes o salientes y una mejor relación, este tipo de red permite un ahorro de costes así como de una mayor fiabilidad de la red.

Del mismo modo, estas redes permiten crear unos sistemas de seguridad más fuertes y un mayor control de los distintos puntos de acceso. Gracias a esto, el administrador de la red puede conocer la localización de los accesos mediante un sistema de triangulación.

Otra característica de este tipo de red es que permitirá al administrador la creación de túneles de conexiones entre los dispositivos existentes en caso de que uno de los elementos de la red esté estropeado, evitando así el parón completo de las transmisiones en la red.



Figura 2-1. Ejemplo de Red Centralizada.

Como vemos en la figura 2-1, el portátil y la PDA se conectan con el router el cual es el que administra la red a la que se conectan.

### 2.1.4 Redes inalámbricas distribuidas

Este tipo de conexiones son las que utilizan las redes WPAN y WLAN y se definen por no tener un administrador. En la red habrá distintos emisores-receptores, estos son usuarios y administradores de la red. Los distintos usuarios serán a su vez creadores de la propia red lo que significa que los dispositivos más alejados de la red pueden acceder a ella utilizando los canales creados por la red de dispositivos.

Actualmente esta configuración de redes de dispositivos está progresando a gran velocidad y permite un gran abanico de posibilidades gracias a su adaptabilidad a distintos entornos (bibliotecas, aeropuertos, colegios,

etc.).

Al no existir un punto donde se canalice toda la información, cuando uno de los dispositivos queda desconectado, la red no se ve perjudicada al no haber dependencia entre unos usuarios y otros. Del mismo modo, todos los elementos de la red emiten a una misma frecuencia y no será necesario de ningún tipo de traductor de frecuencia.

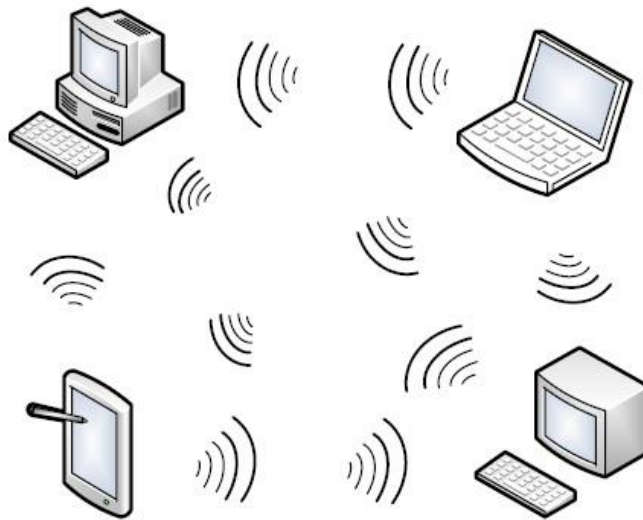


Figura 2-2. Ejemplo de Red Distribuida.

Como vemos en la figura 2-2, todos los dispositivos de la red se conectan entre ellos a partir de las señales que envía cada uno.

## 2.2 Arquitectura del problema objeto de estudio

Con el conocimiento de los tipos de red y con el entendimiento del problema llegamos a la conclusión de que nuestra red será WPAN con distancias máximas de 500 metros en función de los obstáculos según el estándar IEEE 802.15.4 de.

Mediante esta red creada entre los distintos dispositivos el flujo de información entre los router se hará de tal forma que la información se reenvíe desde los routers más alejados de los puntos de la red y se vaya repitiendo utilizando los routers existentes hacia la salida en el concentrador.

Un correcto conocimiento de la estructura de la red nos permitirá buscar una forma correcta de crear una forma constructiva para crear la solución, como se mueven los flujos, cuales son los puntos frecuentes, cuales son necesarios y los más probables.

Hace unos años, los contadores de agua eran de tipo analógico pero se han ido actualizando a hacia modelos digitales más modernos, sin embargo, la recolección de los datos se seguía haciendo de forma manual mediante la programación de rutas de operarios con un sistema llamado Walk-By. Los operarios utilizaban los TPL (Terminal de Punto de Lectura) para la obtención de los datos de los terminales.

En el nuevo sistema los distintos terminales de lectura enviarán los datos obtenidos del contador o contadores hacia el router al que esté asignado, estos routers tienen un límite máximo de conexiones. Los routers se conectarán directamente con la Sede Central donde se encuentra el Concentrador, o bien se conectarán con otro router de la ruta que tiene asignada hasta llegar finalmente al Concentrador. La ubicación de los Routers será una variable con posibilidades discretas, en concreto se podrán poner los routers en las esquinas de cada una de las calles.

Al igual que los routers, los concentradores tienen sus limitaciones en el número de conexiones posibles, debido a esto, por la ciudad habrá distintas sedes locales, en las que la información obtenida en ellos será enviada por fibra óptica hacia la sede central de la compañía suministradora. La localización de los concentradores será en aquellos lugares predefinidos donde haya conexión con la red de fibra óptica de la ciudad.

**En este proyecto pues se va a buscar de forma constructiva una primera solución de partida que sirva para futuras heurísticas a aplicar en este tipo de problemas. Se va a tratar de poner un mínimo número de Routers y de Concentradores cumpliendo con las limitaciones técnicas de los mismos. Con todo esto, siempre tratando que el coste sea el mínimo.**

## 2.3 Elementos y sus características

A continuación se expondrán los elementos que conforman el problema y se explicarán detalladamente sus características:

### Terminal de Lectura

El terminal de lectura es aquel que cada edificio tiene asignado y que recoge los datos de cada uno de los contadores que tiene. Dicho terminal estará asignado a un router determinado al que le enviará la información.

Los contadores estarán en distintos puntos en los edificios, típicamente se hayan en las fachadas de ellos aunque también podrán estar en los sótanos o en el interior. Dependiendo de donde se sitúen tendrán un mayor o menor alcance debido a los impedimentos físicos (paredes, árboles, etc.).



Figura 2-3. Ejemplo de TPL.

### Router

El router es aquel al que le llegan los datos de los distintos terminales de cada edificio así como los datos de los otros routers de la red. El router podrá ir conectado a otro router creando el camino de una ruta o bien conectado directamente al concentrador que es el destino final de la red.

El router es el elemento del que sabemos sus posibles localizaciones y que tenemos que minimizar su cantidad. Cada una de las esquinas de una ciudad es un punto susceptible de poseer uno de los routers lo cual da un número muy grande para calcular los puntos de forma manual.

La máxima distancia que puede haber entre dos routers para que la conexión entre ellos sea buena es sobre unos 500 metros cuando no hay entre ellos obstáculos. Los routers tienen que tener visión directa debido a restricciones de la tecnología y frecuencia de transmisión elegida, por lo que las ubicaciones donde interfiera cualquier arboleda u otro elemento de tipo arquitectónico, entre Terminal- Router, Router-Router o Router-Concentrador, no será una ubicación candidata.



Otra limitación de los Routers es la cantidad de terminales de los que puede transmitir los datos. El límite está situado en 20 terminales por router. En estos 20 terminales se incluyen tanto los que vienen directamente transmitidos por los TPL hacia el router, como los que han sido recibidos por otro router. Es decir, si un terminal tiene asociados 15 TPL directamente y un router conecta con él y le envía una señal con los datos de más de 5 terminales, en ese nodo habría que poner otro router.

El coste cada uno de los Routers será de 98 €, el cual incluye los costes del propio router, el estudio de las posibles ubicaciones y la instalación del mismo.

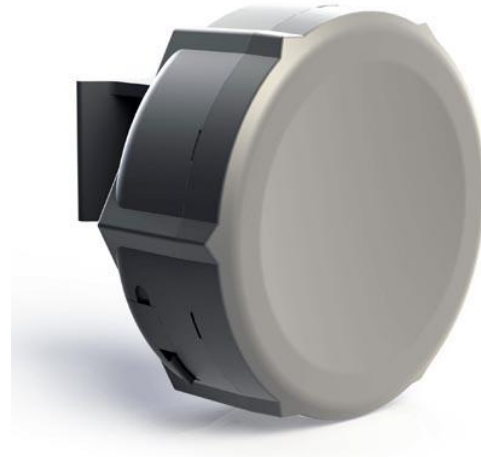


Figura 2-4. Ejemplo de router.

### **Concentrador**

Los concentradores son los puntos de salida de la red. El concentrador recibe las señales emitidas por los distintos routers y la envía por la red de fibra óptica hacia la sede central.

Los concentradores tienen limitados el número de conexiones con routers en 20 y el máximo flujo de información sobre terminales es de 500.

El concentrador no está localizado a priori pero sí que se conocen las posibles ubicaciones, las cuales son lugares donde hay acceso a la red de fibra óptica.

La distancia máxima que puede haber entre concentrador que recibe la información de un router es de 500 metros sin obstáculos.

El coste por cada uno de los concentradores es de 350 euros, donde se incluyen el coste del propio concentrador y los costes de montaje y diseño de la red.



Figura 2-5. Ejemplo de concentrador de red.

En la siguiente tabla se muestra un resumen de lo anteriormente comentado:  
Donde “Sí” significa que hay un arco posible entre ellos y “No” lo contrario.

Tabla 2-1. Resumen de características.

	<b>Límite de conexiones</b>	<b>Alcance (m)</b>	<b>Coste (€)</b>
<b>TPL</b>	1	100	Indeterminado
<b>Router</b>	20	500	98
<b>Concentrador</b>	20	500	350

Hay que señalar que en la tabla 2-1 el coste del TPL no es relevante ya que será el mismo en todas las alternativas admisibles de las soluciones.

Tabla 2-2. Resumen de conexiones.

	<b>TPL</b>	<b>Router</b>	<b>Concentrador</b>
<b>TPL</b>	No	Sí	No
<b>Router</b>	No	Si	Si
<b>Concentrador</b>	No	No	No

En la tabla 2-2 tenemos los arcos permitidos. Para ver los arcos permitidos para un TPL miramos la fila (origen) y en función de sí en las distintas columnas aparece un “Sí” es que el arco está permitido en caso contrario no lo estaría.

# 3 ESTADO DEL ARTE

## 3.1 Revisión de la literatura del problema de la localización de routers

EL problema que se ha comentado en el apartado anterior, estudia donde localizar una serie de routers y por ello, parece necesario hacer una revisión del Estado del Arte sobre lo que se ha hecho hasta el momento sobre el tema que aquí nos concierne.

La localización de los Routers (Hub Location Problem, HLP), es un problema de estrategia donde se aborda como enviar un determinado producto o información entre una serie de localizaciones o nodos de una red particular. Este problema fue investigado clásicamente para decidir las localizaciones de unos centros de transporte aéreo. Subsiguientemente se ha utilizado en gran variedad de problemas dentro del área de la logística y las telecomunicaciones.

La situación en este tipo de problemas, es que la información o los bienes han de pasar por una serie de puntos, a partir de ellos se define la ruta o grupo de nodos por los que ha de pasar la información. La ruta ha de ser seleccionada entre las posibilidades de los distintos nodos para trabajar como puntos de conmutación del tráfico.

El ejemplo más sencillo es aquel donde entre una serie de nodos origen, deben enviar un determinado flujo hacia una serie de nodos destino conociendo el coste unitario de enviar una unidad de flujo entre dichos arcos.

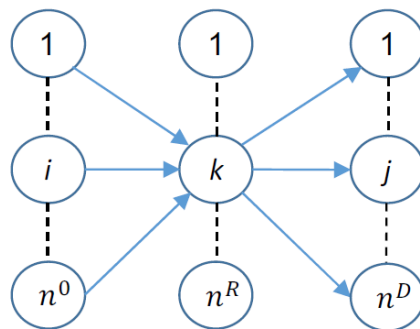


Figura 3-1. Ejemplo HLP.

Como vemos en la figura, el nodo  $i$  se conecta con el  $j$  a partir de un nodo-hub  $k$ .

Este tipo de problemas son de tipo NP-Completo y se encuadran en el campo de la optimización combinatoria. Para su resolución óptima es necesario un gran trabajo computacional, por ello los tiempos necesarios para hallar la solución óptima necesitan de tiempos muy elevados. Muchos de los modelos son de tipo NP-Duro, inclusive aquellos en los que la localización de los distintos Hubs es fija, la parte correspondiente a la asignación sigue siendo NP-Duro (Kara, 1999).

Existen muchos trabajos sobre la localización de routers, y se pueden encontrar algunas revisiones como ReVelle y Eiselt (2005), ReVelle *et al.* (2008), Alumur y Kara (2008), Hekmatfar y Pishvaei (2009).

Tras esto, vamos a mostrar una clasificación de los distintos problemas existentes para tratar de encuadrar nuestro problema entre alguno de ellos.

## 3.2 Caracterización del problema según literatura

Para clasificar los distintos tipos de HLP, vamos a utilizar el trabajo de Hekmatfar and Pishvaei (2009) para ser capaces de enmarcar de forma estructurada los diversos modelos. Los parámetros que enmarcan los problemas en distintos tipos son:

1. Según la solución sea Discreta o Continua.
2. Según la Función Objetivo sea de MiniMax o de MiniSum.
3. Según el número de Hubs sea único o mayor de uno.
4. Según la capacidad sea restringida o ilimitada.
5. Según tenga coste o no la ubicación del Hub.
6. Según los números de conexiones de Nodo a Hub.
7. En función de la existencia de algún coste fijo o variable entre nodo y Hub.

Caracterizando lo anterior a nuestro problema:

Nuestra **solución** será **discreta** ya que el número de terminales, routers y concentradores son siempre números enteros. En aquellos lugares donde finalmente no haya sido activado ninguno de los routers el valor de la variable será cero.

La función objetivo será de **MiniSum** ya que al tener un número de dispositivos hará posible que se obtenga un menor coste.

En nuestro caso el número de **Hubs** será **mayor de uno**, de hecho, por la situación del problema, habrá lugares donde habrá que poner en un mismo punto más de un Hub.

La **capacidad** de nuestros **Hubs** será **limitada**. Tanto los concentradores como los routers tendrán limitados el número de conexiones a 20.

La **ubicación del Hub** en cualquiera de los puntos tiene un coste el cual **siempre es constante**.

El **número de conexiones** de Nodo a Hub será **conocido pero variará** en función del lugar.

**No** existirá ningún **coste fijo ni variable entre nodo y Hub**.

### 3.3 Revisión de la literatura de los Árboles de Expansión Mínima

El ser humano desde su aparición se ha visto obligado a llegar a distintos puntos, por ello el hombre siempre ha tratado de encontrar la ruta óptima, con la cual necesitaría un menor esfuerzo, tiempo, energía, ... Por el estudio de este tipo de problemas, surgen análisis para estudiar cómo resolverlos.

En la actualidad podemos apreciar muchas cosas que nos pueden parecer de lo más habitual, caminos, líneas de comunicación telefónica, televisión por cable, el transporte ferroviario, líneas aéreas, circuitos eléctricos de nuestras casas, automóviles, etc. Los cuales, muchas veces se han calculado a partir de la ejecución del Árbol de Expansión Mínima.

Por ejemplo, cuando en un barrio nuevo se va a hacer la instalación eléctrica, una manera de calcular la cantidad mínima de cableado necesario es minimizando la distancia. Este problema puede ser resuelto aplicando este tipo de árboles.

Para la resolución del problema propuesto va a utilizarse el Árbol de Expansión Mínima y por tanto vamos a hacer un análisis de las distintas metodologías para formarlo.

Un árbol de nodos es aquel que enlaza todos los nodos de un grafo de manera directa (todos conectados entre sí) o indirecta (hay una ruta posible entre nodos para llegar desde un nodo  $i$  a un nodo  $j$ ).

En un árbol no existen ciclos, es decir, para ir de un nodo  $i$  a un nodo  $j$  hay un único camino, el cual es el mismo pero inverso cuando es desde  $j$  a  $i$  y no hay rutas alternativas.

Los ramales de un nodo no están orientados, es decir, la ruta de un nodo  $i$  a un nodo  $j$  es posible siempre en ambos sentidos.

Por último, cada uno de los arcos posibles tiene asignado un peso determinado y el objetivo es el de minimizar la suma de los pesos de los arcos activos.

Este tipo de problemas fue originado por Leonhard Euler en 1736 tratando de resolver el “Problema de los puentes de Königsberg”.

En la ciudad de Königsberg hay siete puentes que unen ambos márgenes del río Pregel con dos de sus islas. Dos de los puentes unen la isla mayor con la margen oriental y otros dos con la margen occidental. La isla menor está conectada a cada margen por un puente y el séptimo puente une ambas islas. El problema planteaba que si sería posible dar un paseo comenzando desde cualquiera de estas regiones, pasando por todos los puentes, recorriendo solo una vez cada uno y regresando al mismo punto de partida.

De este estudio se obtiene que el problema de Königsberg no tenía solución y que no era posible regresar al nodo de salida sin volver a pasar por alguno de los nodos ya pisados. <sup>1</sup>

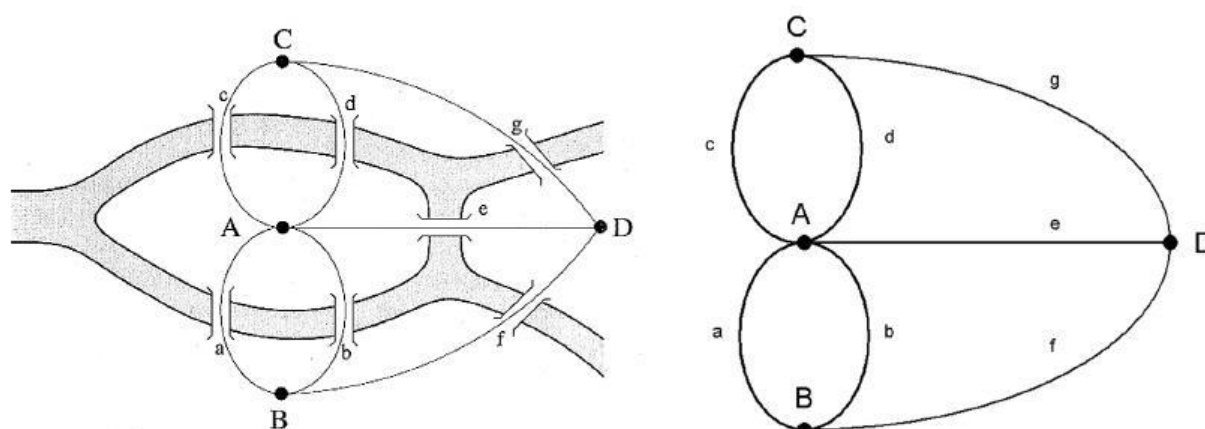


Figura 3-2. Ejemplo del Problema de los puentes de Königsberg y su Grafo Asociado.

Para la obtención del Árbol de Expansión Mínima hay dos enfoques, el algoritmo de Prim (ver Prim, 1957) y el de Kruskal (ver Kruskal, 1956) que utilizan algoritmos voraces (también conocidos como ávido, devorador o goloso) que son aquellos que siguen una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. Vamos a explicar cada uno de ellos con sus detalles:

### 3.3.1 Algoritmo de Kruskal

En esta heurística todos los nodos del problema se consideran un árbol. En primer lugar el algoritmo elegiría la arista con un menor peso asignado. En cada una de las iteraciones posteriores localizamos el arco de menor peso que aún no haya sido utilizado y comprobamos a cuál de los árboles pertenece cada uno de los nodos.

Puede ocurrir que los dos nodos pertenecieran a un único árbol, lo cual significaría que estaríamos formando un ciclo y por ello no tomaríamos ese arco.

En caso de que los nodos del arco pertenecieran a dos árboles diferentes, tomaríamos el arco como válido y juntaríamos los dos árboles en un único árbol. Este arco elegido sería una de las ramas de árbol.

El pseudocódigo para la ejecución del Algoritmo de Kruskal sería:

<sup>1</sup> Para más información visitar: <https://eltrasterodepalacio.wordpress.com/2011/12/01/euler-y-los-siete-puentes-de-konigsberg/>

### 3.3.1.1 Pseudocódigo Algoritmo de Kruskal

**Fase 0: Ordenar las aristas del grafo menor a mayor peso**

**Fase 1: Inicializamos todos los nodos como arboles iniciales**

**Fase 2: Buscamos la arista de menor peso**

**Fase 3: Evaluamos la conexión**

**Fase 3.1: La arista pertenece a dos árboles distintos**

**Fase 3.1.1: Se unen los árboles**

**Fase 3.1.2: Actualizamos el número de árboles y de arcos disponibles**

**Fase 3.1.2: Volvemos a la fase 2**

**Fase 3.2: La arista pertenece a un mismo árbol**

**Fase 3.2.1: Se cancela la arista**

**Fase 3.2.2: Volvemos a la fase 2**

### 3.3.1.2 Ejemplo Algoritmo de Kruskal

En el siguiente ejemplo veremos cómo funciona el Algoritmo de Kruskal.

En el ejemplo tenemos 9 nodos que hay que conectar, cada uno de ellos tiene al menos dos aristas disponibles. El número dentro de cada uno de los círculos (nodos) es la etiqueta de cada uno de los nodos. Las barras (arcos) que unen los distintos nodos son las aristas disponibles. Por ejemplo, el nodo 8 tiene dos aristas disponibles, una que une el nodo 8 con el 0 y otra que une el 8 con el 6.

Los arcos tienen a su vez un peso asociado y el objetivo es el de unir todos los nodos obteniendo la menor suma posible de los pesos que hay de cada uno de esos arcos. En el caso del nodo 8, la arista que une 8 con 0 tiene un coste de 169.71 y la arista que une 8 con 6 tiene un coste de 200.

El ejemplo se ha montado con el programa Grafos cuya distribución es gratuita. Dicho programa pertenece a Alejandro Rodríguez Villalobos.

En la figura que hay a continuación los círculos son los nodos que hay en el problema y las flechas los arcos posibles entre ellos. El número que hay en la parte de arriba del círculo es el número de nodo. El valor que hay sobre las flechas es la distancia que hay entre los nodos que une.

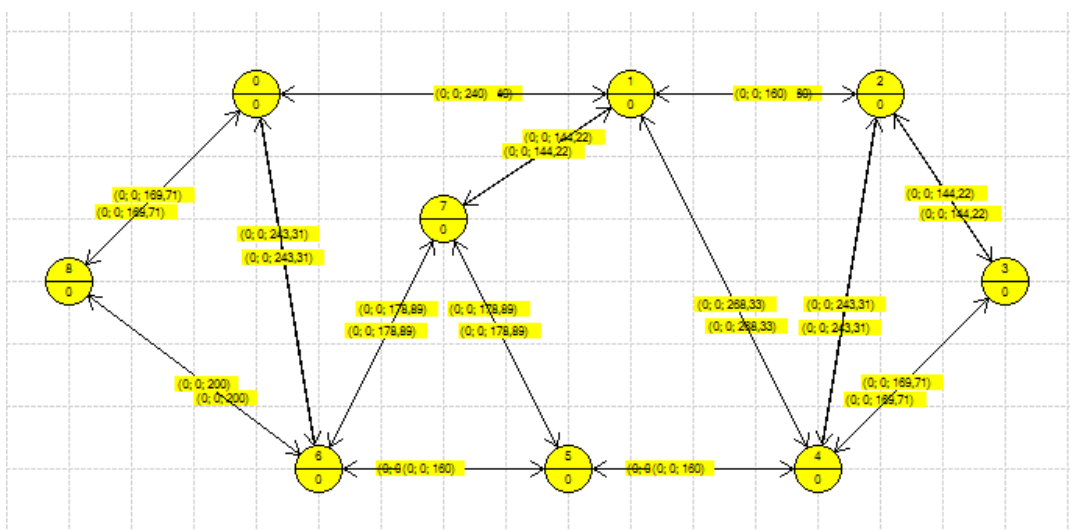


Figura 3-3. Ejemplo algoritmo de Kruskal.

Como vemos, cada uno de los arcos tiene asociado un coste asociado y además el vemos que hay siempre dos posibles direcciones, en sentido de ida y de vuelta.

La solución del problema aplicando el método de Kruskal ha sido:

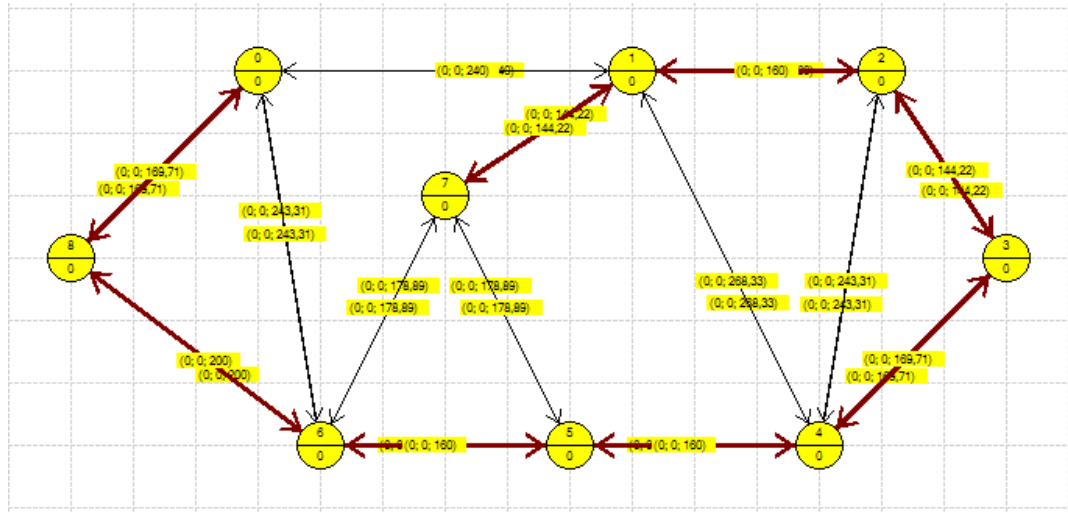


Figura 3-4. Solución Algoritmo Kruskal.

Las flechas rojas que vemos en la figura son los arcos que ha elegido para la formación del Árbol de Expansión Mínima.

Donde en cada una de las iteraciones las operaciones han sido:

- \* 2 ---- (160) ----> 1
- \* 3 ---- (144,22) ----> 2
- \* 4 ---- (169,71) ----> 3
- \* 5 ---- (160) ----> 4
- \* 6 ---- (160) ----> 5
- \* 7 ---- (144,22) ----> 1
- \* 8 ---- (169,71) ----> 0
- \* 8 ---- (200) ----> 6

Coste total = 1307,86.

Como particularidades vemos que el arco 5-4 y el 6-5 tienen un mismo peso pero el peso total de la solución no se ve afectada.

Como observamos, el algoritmo ha ido buscando los arcos de menor peso y ha evaluado si está en un mismo árbol o si va a pasar a formar parte de la solución.

Finalmente, cuando todos los nodos han quedado conectados el algoritmo ha quedado finalizado.

### 3.3.2 Algoritmo de Prim

El Algoritmo de Prim es un algoritmo utilizado para hallar el árbol de expansión mínima de un grafo conexo y

no dirigido cuyas aristas tienen unos pesos asignados.

El algoritmo selecciona las aristas con un menor peso, ya que este es el objetivo de la función. En caso de que el grafo fuera no conexo, el algoritmo buscaría un árbol para cada uno de los componentes del grafo.

El Algoritmo de Prim, también llamado Algoritmo DJP o Algoritmo de Jarnik, fue diseñado por Prim (1957). También fue diseñado por Jarnik (1930) y Dijkstra (1959) de forma independiente.

El algoritmo comienza eligiendo la arista de menor peso, lo cual añadiría dos nodos que ya estarían conectados. Una vez hecho esto, el algoritmo va buscando en cada iteración añadir las aristas de menor peso que vayan conectando con los nodos restantes.

El algoritmo antes de seleccionar una arista, comprueba que esta no está ya conectada al árbol que ha ido formando ya que en caso de que esto ocurriera estaríamos formando un ciclo.

### 3.3.2.1 Pseudocódigo Algoritmo de Prim

**Fase 1: Elegimos el arco con un menor peso;**

**Fase 2: Evaluación del arco con menor peso**

**Fase 2.1: Sí el arco elegido no conecta con nodos que ya están activos**

**Fase 2.1.1: Añadimos el arco al árbol;**

**Marcamos los nodos como conectados;**

**Fase 2.2: Sí el arco elegido conecta dos nodos que ya están activos**

**Fase 2.2.1: Se cancela el arco;**

**Fase 2.3: Comprobar si todos los nodos están conectados**

**Fase 2.3.1: Sí todos los nodos están conectados**

**Fase 2.3.1: Fin**

**Fase 2.3.2: Sí quedan nodos por conectar volvemos a la fase 2.**

### 3.3.2.2 Ejemplo Algoritmo de Prim

En el siguiente ejemplo veremos cómo funciona el Algoritmo de Prim.

Para estudiar el problema vamos a utilizar el mismo ejemplo que para el caso de Kruskal.

El ejemplo se ha montado con el programa Grafos cuya distribución es gratuita. Dicho programa pertenece a Alejandro Rodríguez Villalobos.



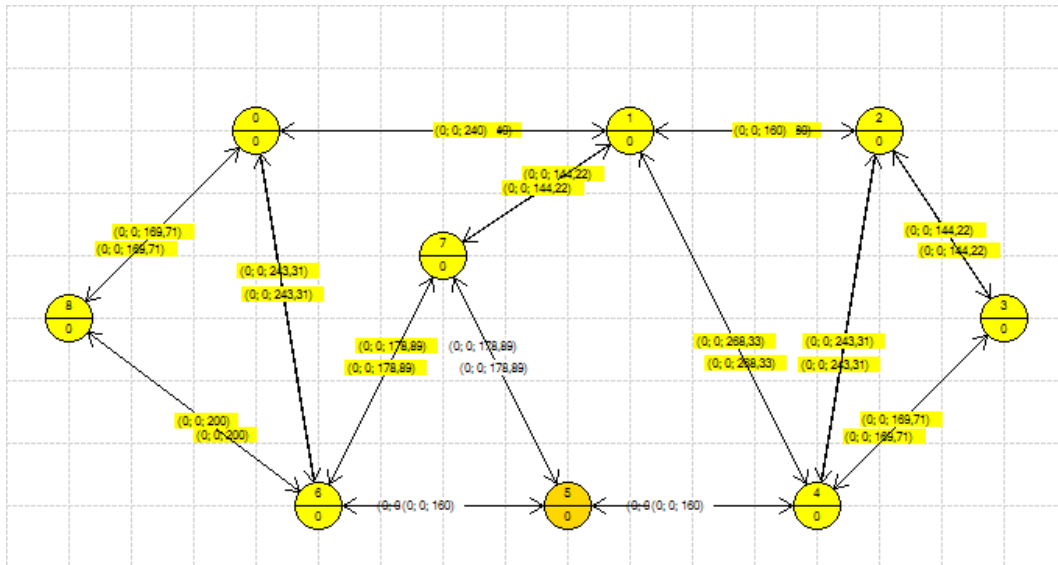


Figura 3-5. Ejemplo utilizado para la solución de Prim.

- \* 0 ---- (169,71) ----> 8
- \* 1 ---- (160) ----> 2
- \* 1 ---- (144,22) ----> 7
- \* 2 ---- (144,22) ----> 3
- \* 3 ---- (169,71) ----> 4
- \* 4 ---- (160) ----> 5
- \* 5 ---- (160) ----> 6
- \* 6 ---- (200) ----> 8

Coste total = 1307,86.

La representación de la solución dada por el Algoritmo de Prim sería:

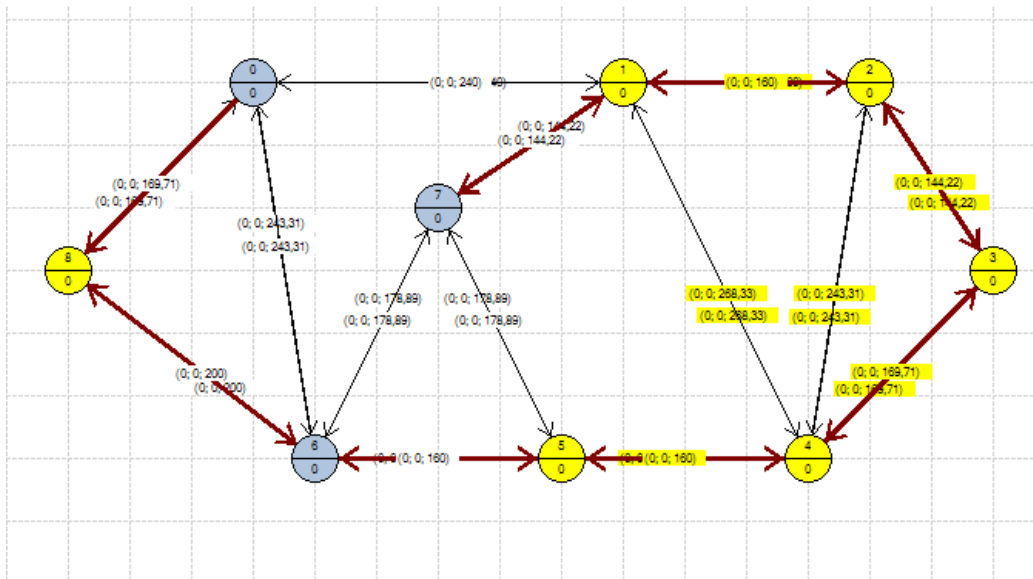


Figura 3-6. Solución utilizando el algoritmo de Prim.

Como observamos, el coste total obtenido ha sido el mismo pero el orden en el que se obtienen los arcos ha sido diferente en ambas ocasiones.

### 3.3.3 Diferencias

Como vemos, la principal diferencia es el propio algoritmo.

En el Algoritmo de Prim partimos de un primer nodo y vamos añadiendo nodos cuyo coste sea el menor. Es decir, si partimos del nodo 0, tenemos tres posibilidades, hacia 1, 6 y 8 cuyos arcos que los conectan pesan 240, 243.31 y 169.71 respectivamente. Por ello el Algoritmo de Prim decide elegir el arco que lo une con 8 que es el de menor peso. Ahora tendríamos los nodos 1 y 8 para seguir conectando el árbol y evaluaríamos los arcos de igual manera que antes.

Sin embargo, en el Algoritmo de Kruskal lo primero que se hace es ordenar todas las aristas de menor a mayor peso. Este algoritmo va eligiendo la de menor coste y evaluando con dos condiciones, nunca se puede repetir un nodo y no se pueden generar bucles.

Por otro lado, cada uno de los algoritmos no siempre da una misma solución, es decir, ambas llegan a la solución óptima pero puede ser que las soluciones sean distintas, esto puede ser por la existencia de arcos con un mismo peso en localizaciones distintas u otras razones. Lo cual significa que puede haber más de una solución óptima<sup>2</sup>.

El Algoritmo de Kruskal no necesita que todos los nodos hayan quedado conectados ya que sí esto pasa se dedica a crear un bosque en cambio el Algoritmo de Prim sí que necesita para poder ejecutarse que exista alguna ruta entre todos los nodos.

Respecto a la complejidad, la diferencia principal es que Prim añade nodos sabiendo que llegamos a ellos con menor coste y Kruskal ordena las aristas para obtener el menor coste.

Teniendo ello presente obtenemos que:

Complejidad de Prim:  $O(n^2)$

Complejidad de Kruskal:  $O(a \log a)$  (siendo  $a$  el número de arcos).

La función que utilizaremos para obtener el arco contempla ambas opciones y elige la mejor de ellas en función

<sup>2</sup> Ver también <http://www.ing.ula.ve/~aguilar/actividad-docente/AYDA/Clase4MiniSem.pdf>

del problema.

# 4 GRAFO ASOCIADO Y MODELO MATEMÁTICO

*“Cada hora de tiempo perdido en la juventud es una posibilidad más de desgracia en la adultez.”*

Claude Shannon, 1948

En el modelo matemático habrá que definir las distintas localizaciones de los terminales, los routers y los concentradores así como las restricciones. Las diversas localizaciones formarán un grafo no dirigido que será el objeto de estudio. Cada uno de los terminales tendrá asignada una unidad de flujo que viene dada por el nodo origen  $O$ . Los nodos intermedios son los nodos routers, los cuales se conectan entre sí o con los nodos concentradores. Finalmente, los nodos concentradores se conectan con el nodo final  $E$ , el cual tendrá que tener el mismo número de unidades de flujo que salieron en  $O$ .

## 4.1 Tipos de Nodos

1. Nodo  $O$ : comienzo del grafo, de  $O$  salen tantas unidades de flujo como terminales hay en el grafo. Hay un nodo  $O$  por grafo.
2. Nodos  $T_i$ : Son los terminales de cada edificio de la localidad y habrá que asignarlo a alguno de los posibles routers. El número total de terminales será  $N^T$ .
3. Nodos  $P_i$ : Son las Posibles localizaciones de los routers. Tendrán asignados terminales y/o routers e irán a otro router o concentrador. Las posibles localizaciones de los routers serán  $N^P$ .
4. Nodos  $C_i$ : Son las posibles localizaciones de los concentradores, reciben los datos de los routers y reenvían al nodo  $E$  la información. El total de las localizaciones de los Concentradores serán  $N^C$ .
5. Nodo  $E$ : Final del Grafo. Todas las unidades de flujo que entraron por el nodo  $O$  han de ser las mismas que le lleguen ahora a  $E$ . Solo hay un nodo  $E$  por grafo.

## 4.2 Tipos de conexiones o arcos

Una vez definidos los tipos de Nodos, se desarrollan los tipos de conexiones entre los distintos nodos, los cuales son:

1. Nodo Origen ( $O$ ): Origen del nodo que tiene una dirección única del nodo  $O$  hacia cada uno de los terminales  $T_i$ , el nodo  $O$  le asigna una unidad de flujo a cada uno de los  $T_i$  del grafo.
2. Nodos Terminales ( $T_i$ ): cada uno de los terminales del grafo llegará a uno o más posibles nodos routers. Cada  $T_i$  lleva asignada una unidad de flujo que enviará a uno de los arcos posibles que tiene hacia los routers. Los terminales solo pueden enviar la información hacia alguno de los routers que le llegan.
3. Nodos Routers ( $P_i$ ): tienen dos conexiones permitidas. Como entrada pueden tener distintos terminales y routers. Como salidas pueden conectarse con otros routers o bien conectar con alguno de los concentradores.
4. Nodos Concentradores: a los concentradores les llegan uno o varios routers de los posibles como entradas y como salidas siempre conectan con el nodo  $E$ .

5. **Nodo E:** El nodo E es el nodo de salida y solo tiene como entradas los distintos concentradores posibles del grafo.

Tabla 4-1. Conexiones entre Nodos.

Origen	Destino
<b>Nodo O</b>	Terminal
<b>Terminal</b>	Router
<b>Router</b>	Router
	Concentrador
<b>Concentrador</b>	Nodo E

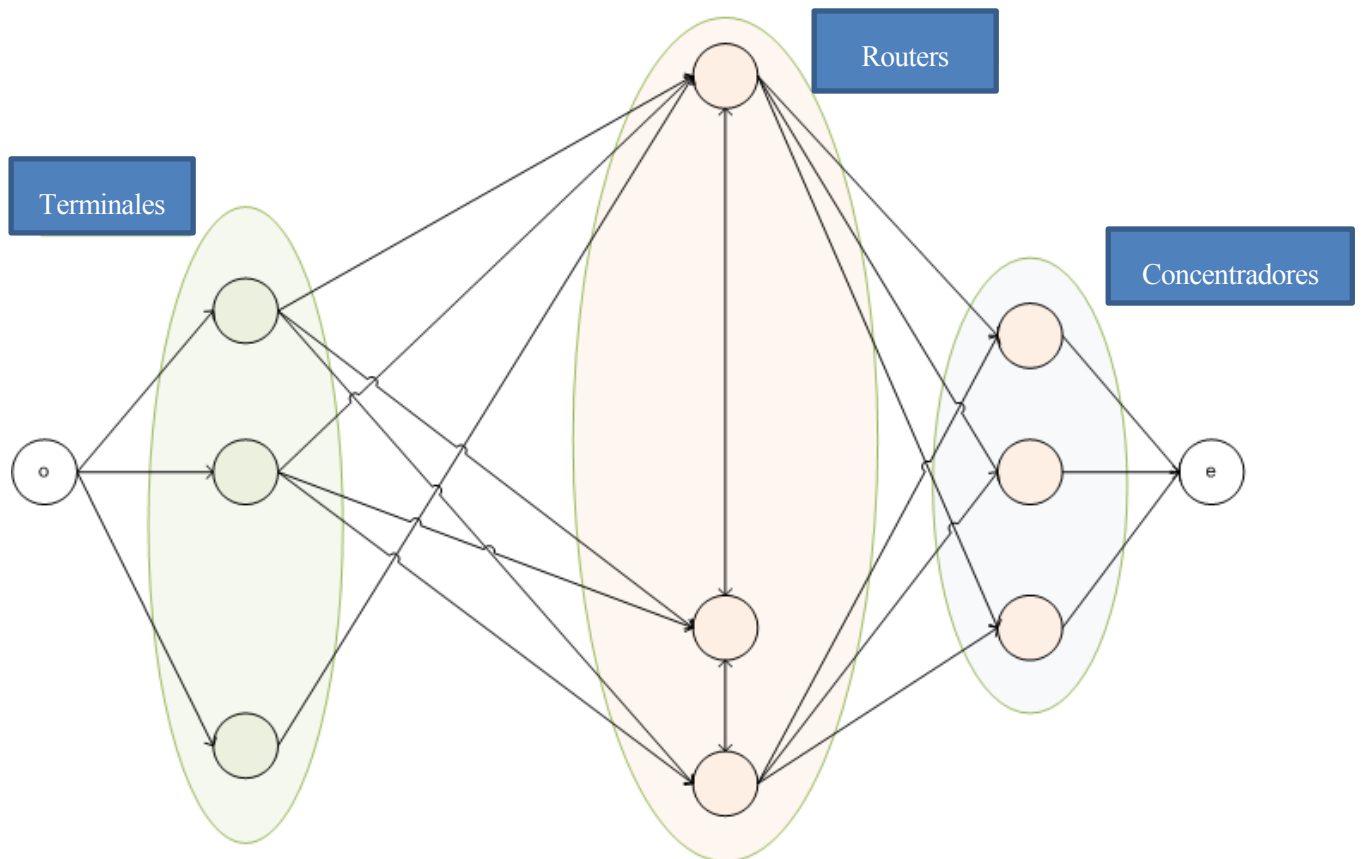
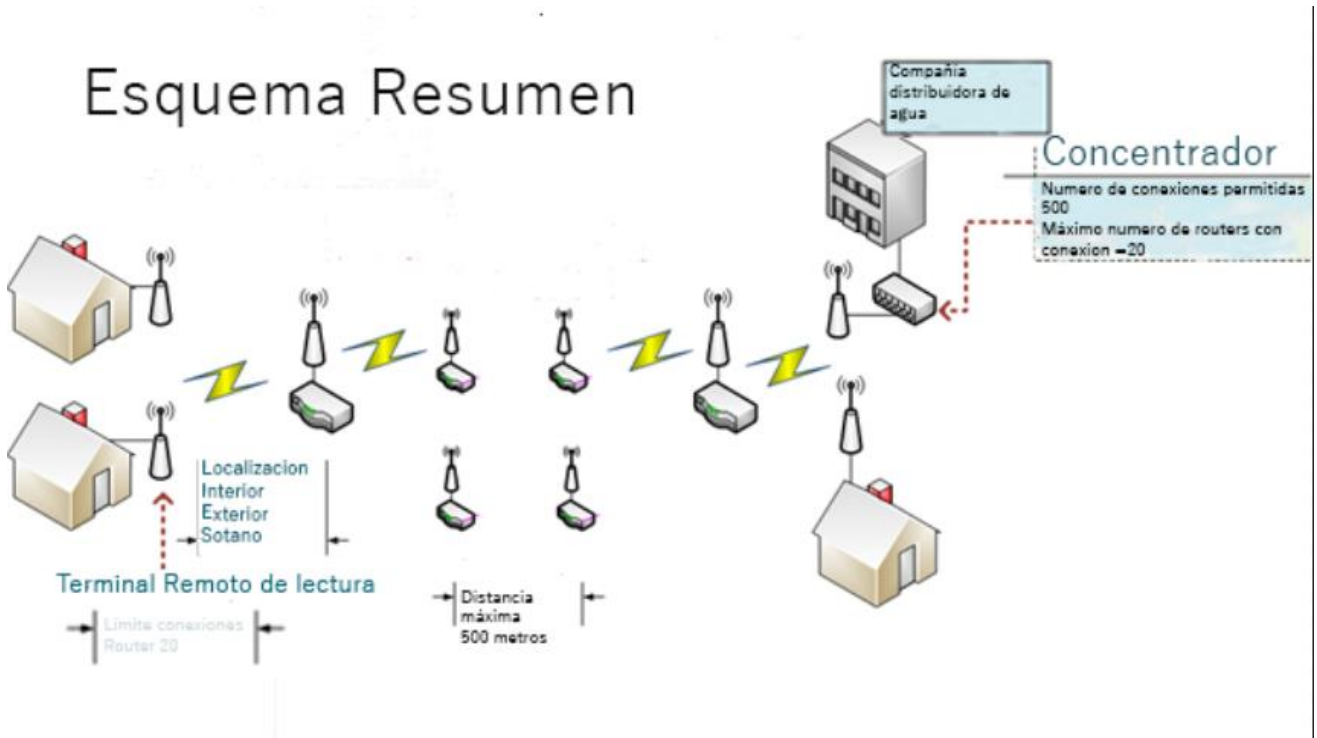


Figura 4-1. Arcos y nodos.

En concreto, la distribución de la red quedaría:



Donde vemos como el terminal de un edificio envía la señal al router que tiene asignado y este router va reenviando la señal hacia otro router y así sucesivamente hasta el punto de salida en el concentrador.

### 4.3 Parámetros y variables

#### 4.3.1 Parámetros

A continuación se describen los parámetros de los distintos dispositivos así como los costes unitarios de cada uno de ellos:

Conexiones máximas de un Router  $n^{TR}$ : 20.

Límite de terminales que recoge un concentrador  $n^{TC}$ : 500.

Límite de routers que admite un concentrador,  $n^{RC}$ : 24.

Precio por cada router, CR: 98.

Coste de cada concentrador, CC: 350.

#### 4.3.2 Variables

La variable  $n_j^r$  es una variable entera la cual es la suma de todos los routers que se han colocado en el grafo de entre todas las posibilidades.

La variable  $n_k^c$  es una variable entera la cual es la suma de todos los concentradores que se han colocado de entre todas las posibilidades.

La variable  $y_{ij}$  es la variable que me dice que flujo asociado tiene cada uno de los nodos del grafo en el arco  $(i,j)$ .

#### 4.4 Función Objetivo y Restricciones

El propósito de este Proyecto es el de minimizar el coste por lo que la función objetivo del modelo lineal será una función de minimar los costes:

$$\min f(x) = CR \times \sum_{\forall j \in N^P} n_j^r + CC \times \sum_{\forall k \in N^C} n_k^c$$

El coste vendrá dado por el precio de cada uno de los Routers (CR), multiplicado por el número de routers que se han puesto en el grafo. A esto habrá que sumarle el coste asociado al número de concentradores que será la suma de concentradores que finalmente se han colocado por el precio de cada uno de los concentradores (CC).

El modelo estará sometido a las siguientes restricciones:

Restricción 1(R1): Inyección del flujo. Desde el nodo origen O se inyecta una unidad de flujo a cada uno de los terminales que conforman el nodo.

$$\sum_{\forall j \in N^T} y_{oj} = 1, \forall j \in N^T$$

R2: Balance de Flujo. El número de unidades de flujo que entra en un nodo ha de ser igual al que sale.

$$\sum_{\forall (j,i) \in A} y_{ji} = \sum_{\forall (i,j) \in A} y_{ij}, \forall i \in N$$

R3: Sumidero de Flujo. Todo el flujo que sale de los nodos asociados a posiciones candidatas de los concentradores llega al nodo final (e), siendo su suma el número de elementos que está contenido en el conjunto.

$$\sum_{\forall i \in N^C} y_{ie} = |N^T|$$

R4: Contador de Routers. El número de routers en cada uno de los nodos se puede saber a partir del flujo de entrada o de salida de los mismos, redondeándolo al entero superior obteniendo los routers necesarios.

$$n_i^r \geq \sum_{\forall (j,i) \in A} \frac{y_{ji}}{n^{TR}}, \forall i \in N^P$$

R5: Contador de Concentradores. El número de concentradores se puede saber a partir del flujo de entrada o de salida de los mismos, redondeando al entero superior, obteniendo así el número de concentradores necesarios.

$$n_i^c \geq \sum_{\forall (j,i) \in A} \frac{y_{ji}}{n^{TC}}, \forall i \in N^C$$

#### 4.5 Modelo Completo

Seguidamente se expondrá el modelo completo, con la Función Objetivo (F.O.) y sus Restricciones.

$$\min f(x) = CR \times \sum_{\forall j \in N^P} n_j^r + CC \times \sum_{\forall k \in N^C} n_k^c \quad \text{F.O.}$$

Sujeto a (s.a.):

$$\sum_{\forall j \in N^T} y_{oj} = 1, \forall j \in N^T \quad \text{R1}$$

$$\sum_{\forall (j,i) \in A} y_{ji} = \sum_{\forall (i,j) \in A} y_{ij}, \forall i \in N \quad \text{R2}$$

$$\sum_{\forall i \in N^C} y_{ie} = |N^T| \quad \text{R3}$$

$$n_i^r \geq \sum_{\forall (j,i) \in A} \frac{y_{ji}}{n^{TR}}, \forall i \in N^P \quad \text{R4}$$

$$n_i^c \geq \sum_{\forall (j,i) \in A} \frac{y_{ji}}{n^{TC}}, \forall i \in N^C \quad \text{R5}$$

En resumen, la función objetivo calcula el coste de cada una de las soluciones posibles. La R1 le asigna una unidad de flujo a cada uno de los terminales. La R2 obliga a que el flujo que entra por en un router sea la misma que sale. La restricción 3 obliga a que el todo el flujo que llega a los concentradores sea el mismo que el flujo que se ha inyectado a los terminales. La R4 calcula el número de routers de cada una de las iteraciones. La R5 calcula el número de concentradores en cada una de las iteraciones.

#### 4.6 Análisis y topología de las soluciones

Analizando las soluciones de la batería de problemas llegamos a la conclusión de que poseen una gran similitud con el Árbol de Expansión Mínima (Minimum Spanning Tree (MST)). Las soluciones no son directamente un MST sin embargo quitando algunos de los arcos activos en la solución óptima la solución se vuelve un MST.

Para la explicación de la heurística desarrollada se ha utilizado el problema 20-200-1 de una batería de problemas aportada por el tutor (ver González, 2015).

Para la representación de la solución se ha utilizado el programa Grafos – 1.3.5 de Alejandro Rodríguez Villalobos.



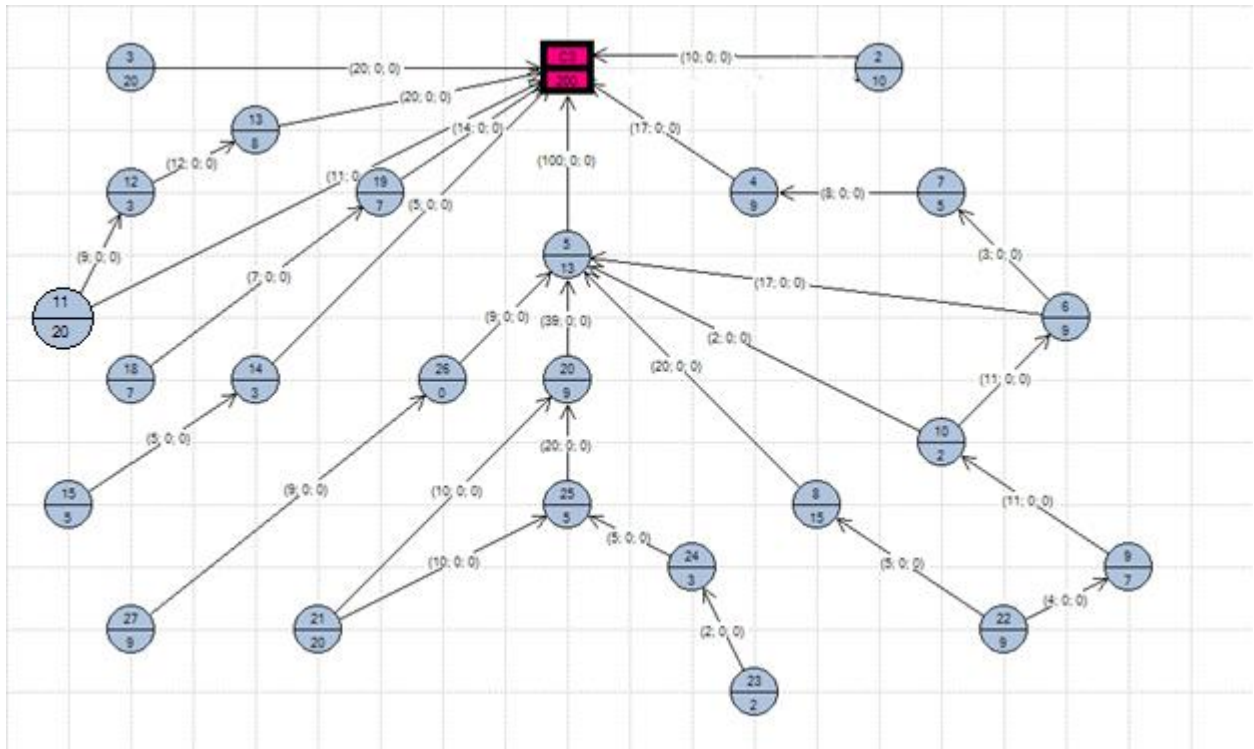


Figura 4-2. Grafo de la solución óptima.

La anterior figura con los arcos podados quedaría:

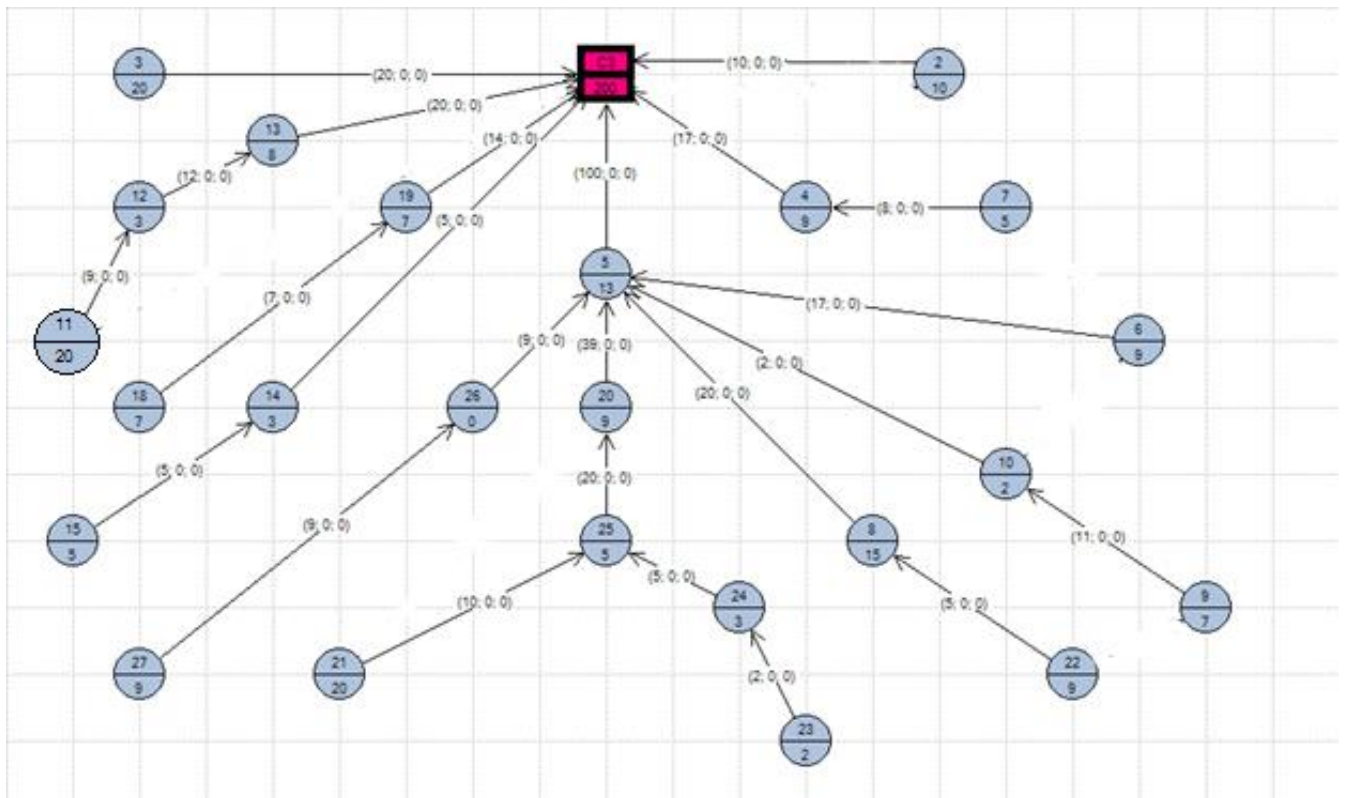


Figura 4-3. Grafo de la solución en forma de árbol.



Figura 4-4. Nodo ejemplo.

Donde el 11 superior del círculo significa el número de nodo que tiene asignado. Por ejemplo, en este recorte sería el nodo 11 (esquina 11).

El número inferior 20 nos dice el número de TPL que tiene asignado dicho nodo. Es decir, el número de los terminales que le envían la información.

Como vemos, al quitar el arco  $22 \rightarrow 9$  hay 4 unidades de flujo que van a tener que ir por el arco  $22 \rightarrow 8$ , esto va a repercutir en el resultado ya que el nodo 8 tiene 15 TPL asignados, lo cual significa que en el nodo 15 vamos a tener un total de 29 unidades de flujo lo cual va a aumentar en un router extra en el nodo 8.

La diferencia entre las dos imágenes es que se han borrado 5 arcos, lo cual ha permitido convertir la solución óptima en un árbol. En esta nueva solución no se permite el envío de la señal hacia dos routers distintos lo cual empeora la solución pero no siempre.

En concreto, el arco  $11 \rightarrow C3$  lo hemos borrado pero si hubiéramos borrado el arco  $11 \rightarrow 12$  y hubiésemos enviado todo el flujo directamente hacia el concentrador la solución no hubiera empeorado ni mejorado, la solución hubiera quedado igual.

Lo mismo ocurre con el arco  $21 \rightarrow 25$ , si hubiéramos quitado este arco en vez del arco  $21 \rightarrow 20$  la solución se hubiera quedado igual, en concreto hubiéramos disminuido el flujo que entra en 25 pero hubiera seguido siendo necesario un mismo número de routers.

Vamos a ver cómo quedaría la solución tras elegir los arcos adecuados.

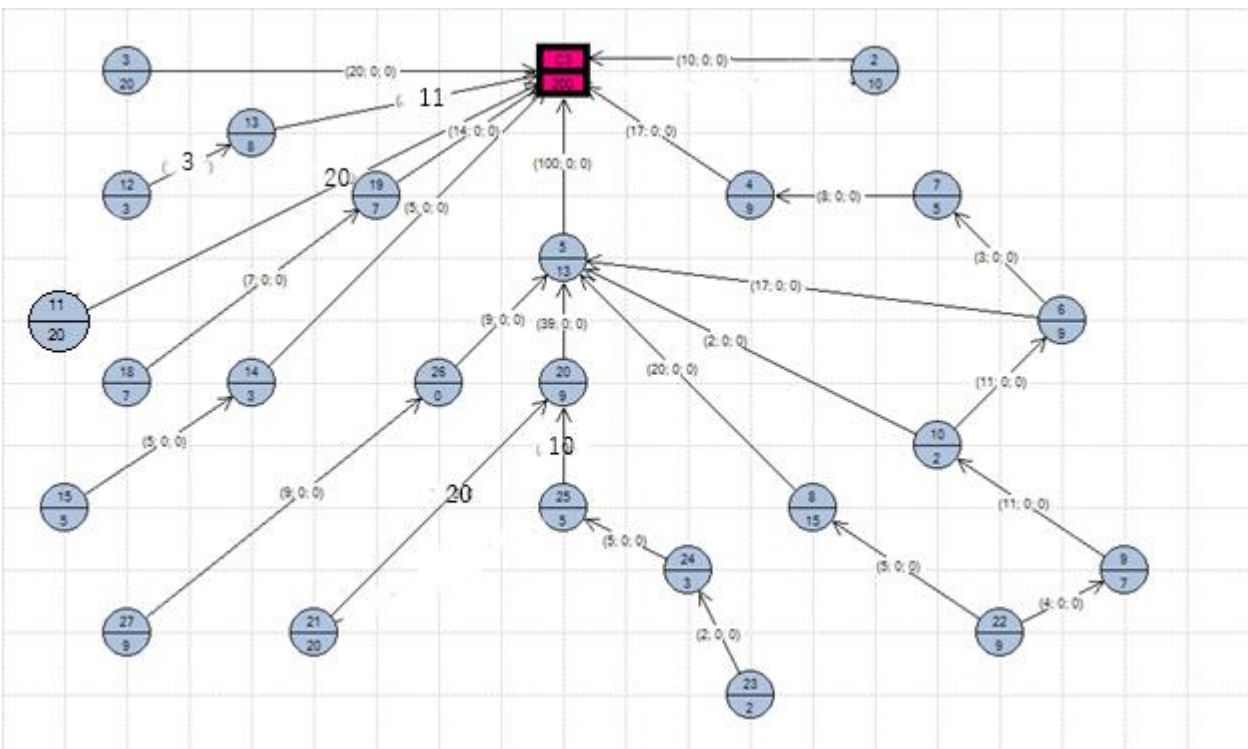


Figura 4-5. Algunos arcos no son necesarios.

La diferencia está en que los routers pueden dividir la señal que envían entre varios routers a los que llega y por eso el árbol quedaría incompleto.

Por ello, partiendo de una solución de un árbol, podemos ir añadiendo con alguna otra heurística arcos y ver cómo va evolucionando la F.O. y ver si esta mejoraría.

Por tanto, del análisis de las soluciones se ha obtenido la conclusión de que cualquiera de las soluciones tiene una similitud con el MST y es por ello que se ha desarrollado esta heurística.

#### 4.7 Ejemplo en un barrio de la localidad de Ginés

Para una mayor comprensión del problema, vamos a ver cómo sería la solución con un árbol para un barrio situado en la localidad de Ginés. Consideremos la siguiente imagen aérea del barrio:



Figura 4-6. Foto aérea de un barrio de la localidad de Gines.

Los rectángulos azules son puntos con terminales de cada una de las casa o de varias y los cuadrados rojos son posibles localizaciones posibles para los routers.

Los cuadrados rojos marcados con una cruz son aquellos lugares donde además de poder tener un router, hay acceso a la fibra óptica y se podría por tanto poner un concentrador.

Los terminales (cuadrados azules) se conectarían a los routers (cuadrados rojos). Cada terminal podrá conectarse al menos uno de los routers.

Para este conjunto de nodos, los arcos permitidos serían:



Figura 4-7. Arcos permitidos en uno de los barrios de Ginés.

En el mapa podemos observar que:

- Tenemos un total de 42 localizaciones con terminales que hay que enviar al concentrador a partir de los routers.
- Tenemos un total de 14 nodos con ubicaciones posibles para los routers. Todos los arcos entre nodos se ven directamente o bien hay obstáculos pero están lo suficientemente cerca.
- Tenemos 4 ubicaciones posibles para los concentradores.

Comentario: los nodos rojos y azules no tienen ningún tipo de diferencia.

Para este barrio, utilizando el Algoritmo de Prim obtenemos la solución óptima cuyo peso es 2707,11.

La representación de la solución óptima por este algoritmo sería:



Figura 4-8. Representación de la solución en la imagen aérea.

Donde las flechas rojas es la solución dada por el Algoritmo de Prim que coincide con la solución dada por el Algoritmo de Kruskal.

Quedaría por tanto seleccionar que concentrador sería el mejor de ellos y ver si hay algunas posibilidades de mejoras en la solución.

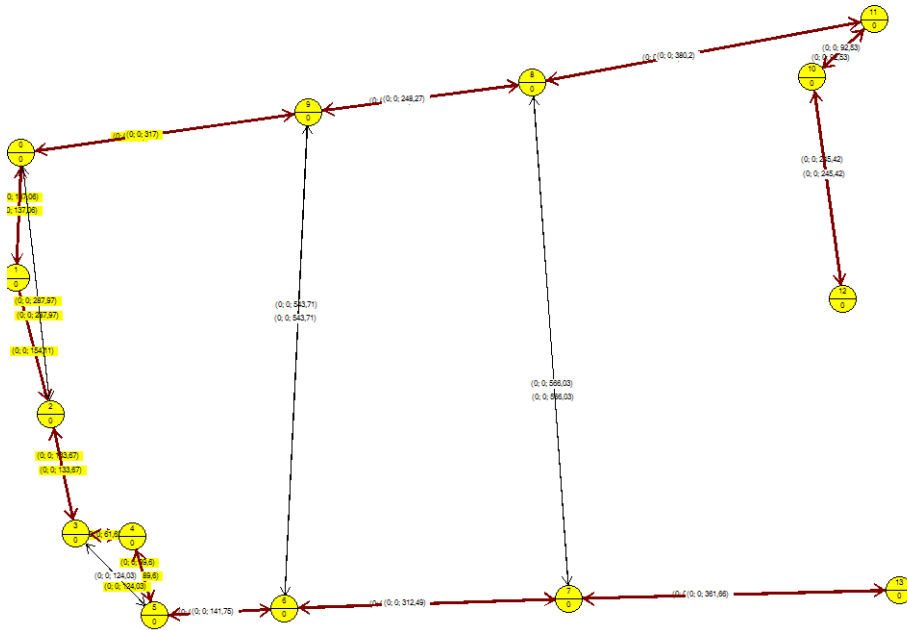


Figura 4-9. Representación del grafo solución.

Finalmente en la figura 4-9 vemos como han quedado todos los nodos conectados en el grafo.

# 5 PROCESO CONSTRUCTIVO

---

*“Cada hora de tiempo perdido en la juventud es una posibilidad más de desgracia en la adultez.”*

Claude Shannon, 1948

Por la complejidad del problema que se está estudiando y su carga computacional, se estudian nuevas formas de resolver el mismo problema de una forma aproximada y como se comentó en el capítulo 4, la heurística aquí propuesta se basa en el MST (Minimum Spanning Tree).

Estudiando los resultados obtenidos en el modelo matemático, nos damos cuenta de que la estructura de las soluciones tiene algo en común, esto es que **las soluciones de este tipo de problemas tienen una gran similitud a un árbol**, en concreto, nosotros desarrollaremos un Árbol de Expansión Mínima. Es por ello que se desarrolla este TFG.

Una vez comprendido el modelo matemático y entendiendo que las soluciones tienen una similitud con un MST, podemos empezar a desarrollar el proceso constructivo. A continuación se explican detalladamente cada uno de los pasos que se ha seguido en la construcción del método constructivo en la localización de los Routers.

Para la explicación del método se ha utilizado el problema 20-200-1 de una batería de problemas aportada por el tutor. (Ver González, 2015).

## 5.1 Pseudo Código

**Fase 0: Preprocesado y Ordenación de datos;**

**Fase 1: Asignación de Terminales**

```
{  
  Fase 1.1 Terminales asignados a un único Router;  
  Fase 1.2 Determinación de los Routers Obligatoriamente Activos;  
  Fase 1.3: Asignar Terminales a Routers Activos;  
  Fase 1.4: Asignar Terminales al Resto de Routers;  
}
```

**Fase 2: Conexionado de Routers**

```
{  
  Fase 2.1: Construir Grafo No Dirigido;  
  Fase 2.2 Construir Bosque;  
  Fase 2.3 Construir Árbol de Expansión Mínima (Minimum Spanning Tree (MST));  
}
```

**Fase 3: Conexión con Concentradores;**

**Fase 4: Obtención Grafo Dirigido y mejoras;**

## Fase 5: Cálculo de la Función Objetivo;

A continuación se desarrollan las anteriores ideas de manera extendida.

### 5.2 Fase 0: Preprocesado y Ordenación de datos

En primer lugar se procede a una ordenación de los datos de las conexiones:

En primer lugar las conexiones Terminal-Router, las cuales siempre tienen una unidad de flujo asignadas. Estarán guardadas en un vector origen con un índice negativo y su correspondiente destino solo podrán ser routers con un destino positivo. Así mismo tendrán asignado una distancia entre el terminal y los routers a los que llega.

En segundo lugar tenemos las conexiones con origen en un Router y destino un Router. En el vector origen tendremos un número determinado, que será el número de router que tiene asignado y este será positivo y mayor que cero. En el destino de este router estará otro número de otro router al que llega que también será positivo y mayor que cero. Entre dichos routers habrá una distancia determinada.

En tercer lugar tenemos las conexiones desde los Routers hacia los concentradores. En el origen tendremos un valor positivo entero, uno de los routers, y en el destino tendremos un número negativo, que es el que cada concentrador tendrá asociado.

Tabla 5-1. Codificación de los datos.

Origen	Destino	Dirección		
<b>Terminal (-)</b>	Router(+)	(-)	→	(+)
<b>Router (+)</b>	Router(+)	(+)	→	(+)
<b>Router (+)</b>	Concentrador(-)	(+)	→	(-)

El resto de las combinaciones no estarían permitidas.

Los datos se ordenan en Excel, abriendo los .txt y ordenándolos según el origen según este en un Router primero y después los Terminales, teniendo así una constante llamada cambio que guarda donde en los vectores origen, destino y distancia cambian de origen en terminal y origen en router.

Teniendo los vectores así ordenados nos ahorramos tener que ir buscando las conexiones y sabiendo donde están de antemano.

### 5.3 Fase 1: Asignación de Terminales

#### 5.3.1 Fase 1.1 y 1.2: Terminales asignados a un único Router y Determinación de los Routers obligatoriamente activos

Debido a las restricciones del problema, algunas de las esquinas obligarán siempre ha que determinados routers estén siempre activos. Esto se produce porque algunos terminales solo llegan a un único router lo cual nos permite de antemano que esquinas van a estar activas con un 100% de seguridad.



Gracias a la ordenación de los datos, empezamos en el vector origen desde el elemento cambio, donde estaría el primero de los terminales (-1) y comprobamos si el siguiente es igual que él o no. En caso de que el siguiente inmediato no fuera y el anterior tampoco, significa que dicho terminal solo tiene alcance a un único router y quedaría automáticamente asignado para el resto del problema.

Tabla 5-2. Asignaciones obligadas

Origen	Destino	Distancia
-45	10	110.8
-45	20	98.8
-46	9	1
-47	10	79.32
-47	9	58.68

Como vemos en el ejemplo, el terminal 46 tiene como único arco permitido el 9 por lo que queda asignado a ese router para el resto del problema.

### 5.3.2 Fase 1.3: Asignar Terminales a Routers Activos

Tras este primer paso ya tendremos una serie de esquinas activas a las cuales les asignaremos todos los terminales que tengan de un arco disponible hacia ellas. Es decir, tenemos unas esquinas activas con algún terminal que lo ha activado de forma obligatoria, y ahora en este paso los terminales que no están asignados puede que lleguen a alguno de los obligatorios y por ello se los asignamos.

Recorremos el vector origen por los terminales y vemos cuál de ellos tiene como destino alguno de los routers que ya están asignados.

Tabla 5-3. Terminales con conexión a alguno de los routers obligatorios

Origen	Destino	Distancia
-48	23	200.4
-48	9	100.4
-48	20	25.6

Como vemos, el terminal 48 tiene un arco posible con el router número 9, por ello se asigna ese terminal a dicho router.

Procediendo de esta manera vamos asignando todos los terminales posibles a los distintos routers obligatorios y vamos reduciendo el número de terminales sin asignación.

### 5.3.3 Fase 1.4: Asignar Terminales al Resto de Routers

Una vez tenemos asignados todos los terminales posibles a los routers que eran obligatorios, vamos a asignar aquellos que han quedado sin conectarse.

Para ello, vamos a evaluar cuál de las posibles esquinas tiene más posibilidades de ser elegida, es decir, cuál de las posibles ubicaciones de los distintos routers tiene más arcos de entrada entre los terminales que no han quedado asignados.

Para esto definimos un vector en el cual cada componente me dirá cuántos de los terminales que no están asignados podría asignar a un determinado router. Eligiendo el máximo de los elementos de este vector estaremos activando dicho router y asignándole todos los terminales que le llegan.

Este proceso es iterativo, es decir, cada vez que asignamos uno de los routers tenemos que actualizar el vector cuantos le llegan a cada router e ir seleccionando en cada iteración al router que más terminales le pueda asignar.

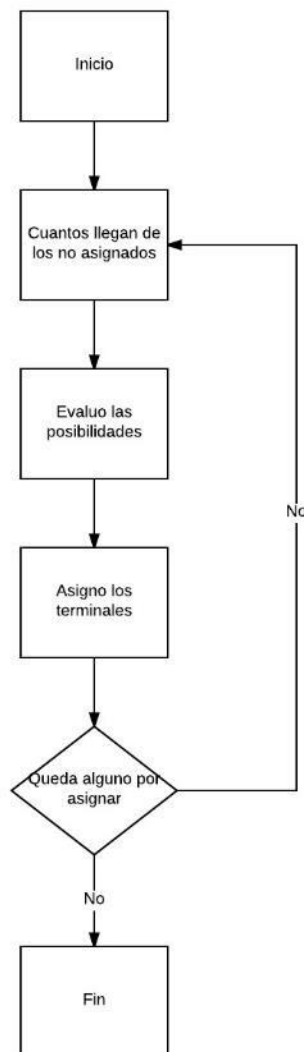


Figura 5-1. Asignación terminales al resto de routers.

Esta forma de proceder nos deja los siguientes routers activos. En la siguiente tabla un 1 indicará que el nodo está activo y un cero al contrario:

Tabla 5-4. Routers Activos e Inactivos.

Nodo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1/0	1	0	0	1	0	1	0	1	1	0	1	0	1	1	1	1	1	0	0	1	1	1	0	1	0	1	1	1

Como vemos, esto nos dice que con 18 esquinas activas somos capaces de asignar todos los flujos de cada uno de los terminales a algún router, sin embargo, no nos dice que todos hayan quedado conectados entre sí.

## 5.4 Fase 2: Conexionado de Routers

### 5.4.1 Fase 2.1 Construir Grafo no dirigido

Con estos pasos, ya tenemos todos los terminales asignados a alguno de los routers. A partir de los routers activos montamos un grafo de conexiones entre los routers. En la matriz habrá filas y columnas de ceros, donde una fila o columna de ceros significará que el router de esa fila no ha sido activado.

Con esta matriz y la función de Matlab `biograph` dibujamos el grafo no dirigido de nodos que nos ha quedado con los nodos activos.

```
View (biograph (UG, [], 'ShowArrows', 'off', 'ShowWeights', 'on'))
```

La ejecución de esta función nos dará un grafo donde veremos todas las conexiones posibles.

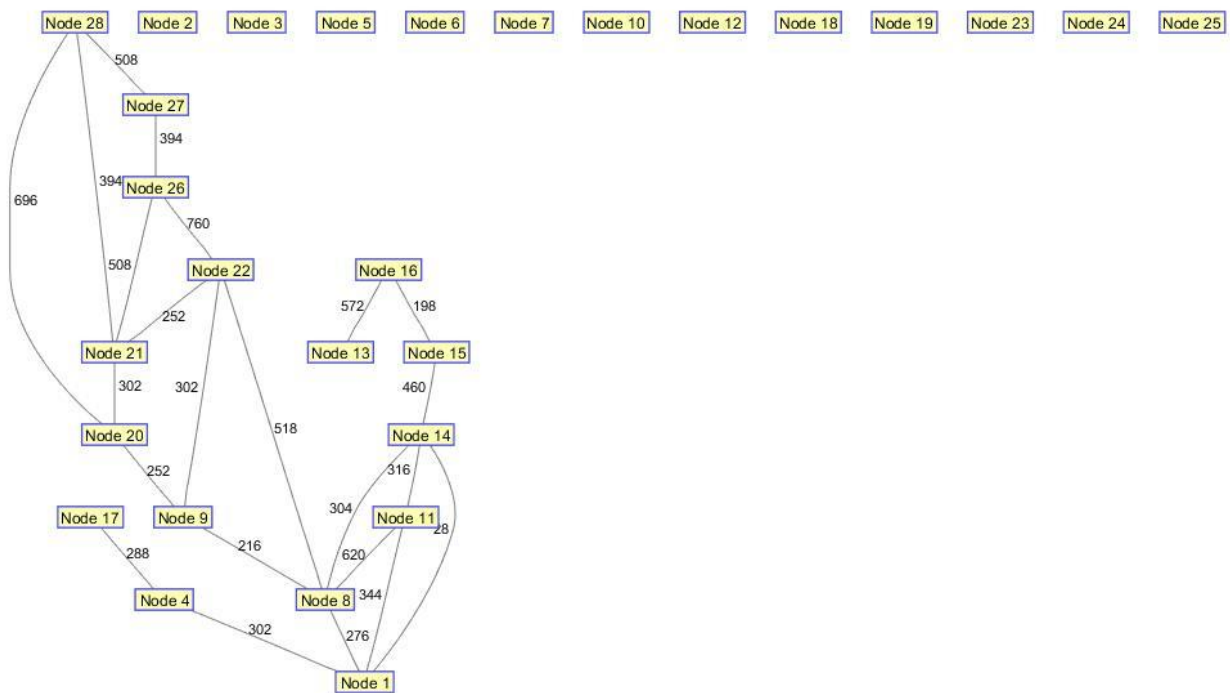


Figura 5-2. Grafo de conexiones Inicial.

Donde los nodos que están sin conexiones son aquellos que no tienen conexiones, o bien aquellos que si tienen asignados algunos de los terminales pero no tienen conexiones entre los routers que hay disponibles.

## 5.4.2 Fase 2.2 Construir Bosque

A este primer grafo obtenido le calculamos el árbol de expansión mínima para hacer que todo quede conectado y obtener una primera ruta del problema. Esto no tiene porqué ocurrir y entonces se forman bosques. Los cuales son pequeños arboles desconectados entre sí.

Esto lo hacemos con una función de Matlab `graphissspantree (ST)`.

Esta función necesita como entrada una matriz dispersa de conexiones y nos devolverá el árbol de expansión mínima en el cual no estarán conectados aquellos que no están activos, o bien, aquellos activos que no tienen conexión posible entre los activos.

La función `graphissspantree (ST)` nos dará un vector con las direcciones de las conexiones y en los no conectados nos dará un error (Not a Number (NaN)). Sabiendo cuales son las esquinas activas y comparando con los que han salido NaN, cuando un nodo activo tenga asignado un NaN significará que el nodo activo ha quedado desconectado del árbol y que hay que conectarlo.

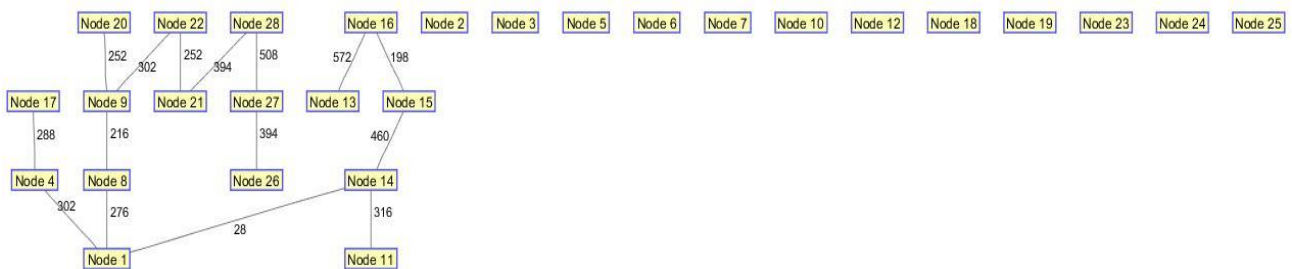


Figura 5-3. Arbol de Expansión mínima inicial.

Como podemos apreciar, los nodos 6 y 24 se incluían entre los nodos activos y vemos que no han quedado conectados en el árbol. Por ello y para que todo quede conectado buscamos un camino para conectarlos.

### 5.4.2.1 Fase 2.2.1 Conexionado del Bosque

Los nodos que han quedado sin conexión hay que conectarlos con el árbol que tenemos, para ello calculamos el camino más corto desde el nodo a conectar hacia todos los nodos activos pasando por los inactivos. Es decir, construimos la matriz de conexiones de los routers entera y buscamos el camino más corto, que será el que necesite un menor número de saltos (routers extra) hacia uno de los nodos activos.

Siguiendo este modo de proceder, obtenemos que el nodo 6 puede conectarse al árbol si añadimos como router activo el nodo 7 necesitando únicamente un router extra.

Del mismo modo, el nodo 24 queda unido a la red añadiendo como router extra el 25. Quedando finalmente un total de 20 esquinas activas y un grafo donde todos los nodos activos están conectados con al menos un arco.

La función que calcula el camino más corto es:

```
[Dist, path, pred] = graphshortestpath (B, k, i)
```

Donde B será una matriz dispersa que incluya todas las conexiones Router-Router.

Dist: la distancia que hay entre los nodos k e i.

Path: es el más importante, pues nos dirá el número de saltos necesarios (routers extra) para hacer la conexión entre el árbol y los nodos desconectados.

Pred: nos dirá la dirección de los saltos.

### 5.4.2.2 Fase 2.2.2 Construir nuevo grafo de conexiones

Sabiendo ya por tanto que todos los nodos van a estar conectados, montamos el grafo de conexiones entre los routers que están activos. Ahora el número de conexiones ha cambiado así como el número de arcos posibles, por lo que el grafo será distinto y tendrá más posibilidades.

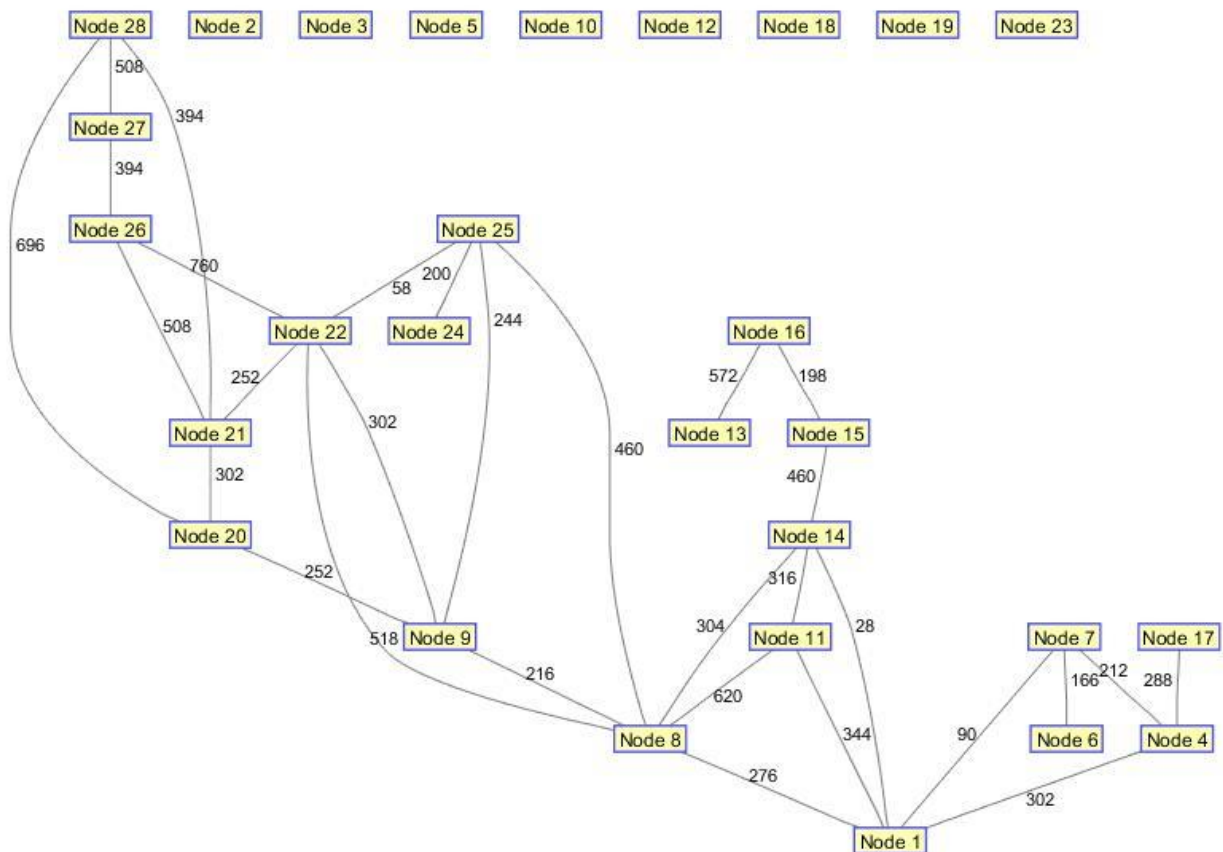


Figura 5-4. Grafo de Conexiones Final.

Como podemos observar, el nodo 6 solo tiene como conexión disponible el 7, al igual que ocurre con el 25 pero ha quedado todo conectado.

### 5.4.3 Fase 2.3: Construir Árbol de Expansión Mínima

Con el grafo calculado en el anterior apartado, creamos su árbol de expansión mínima que utilizaremos como una primera solución. Como han cambiado las conexiones en el grafo, el árbol de expansión mínima habrá cambiado a su vez.

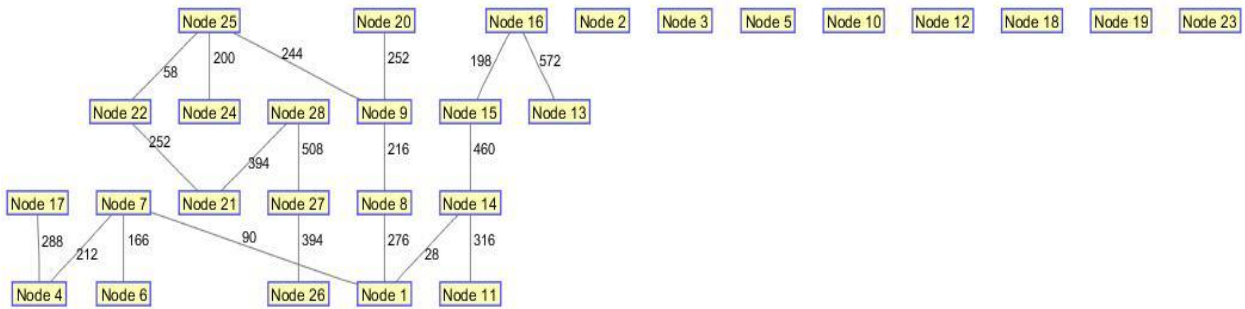


Figura 5-5. Árbol de Expansión mínima final.

Como se observa, el grafo ha cambiado y además el orden de sus conexiones. Una vez tenemos todo conectado, pasamos a decidir los concentradores que van a quedar activos.

**5.5 Fase 3: Conexión con concentradores**

Una vez tenemos el árbol conectado, hay que estudiar cuál de los concentradores sería mejor. Para esto, vemos cuál de los concentradores posibles tiene un mayor número de conexiones. Con ello presente, vemos cual tiene más arcos posibles con los nodos activos.

El concentrador con más conexiones a los arcos activos producirá un mayor número de salidas en los arcos. Así mismo, todos aquellos nodos que tengan un arco hacia el concentrador elegido no necesitarán de una ruta por la que ir repitiendo la información, por lo que en principio estaríamos ahorrando routers en el reenvío de señales.

Siguiendo con este procedimiento, evaluamos todos lo concentradores y nos queda que la cantidad de arcos activos en cada concentrador es de:

Tabla 5-5. Selección de Concentradores.

Concentrador	1	2	3	4	5	6	7	8
Cantidad que llegan	3	3	6	3	4	2	3	3

Como se observa, en el concentrador 3 tenemos 6 arcos activos, esto quiere decir que en nuestro grafo tenemos 6 puntos de salida del mismo y que en ellos no habrá que crear ningún tipo de ruta sino que directamente se enlazarán con el concentrador.

De los 20 nodos activos, solo tendremos que crear 14 rutas para cada uno de ellos tratando de llenar siempre los que están más cercanos a la salida, ya que estos tendrán que ir repitiendo la señal un número de veces menor que cuando están más alejados de esta, dándonos siempre una solución mejor o igual a la que ya tenemos.

Tras la elección del primer concentrador buscamos la adición de más concentradores. Los siguientes

concentradores más valorados serán aquellos que creen nuevos arcos con nodos que no conectan directamente con el concentrador anterior. Es decir, si el concentrador elegido me ha producido tener en nuestro grafo seis puntos de salida, el siguiente mejor concentrador a elegir será aquel que ponga produzca un mayor incremento en el número de salidas.

Una vez tenemos el concentrador que más salidas nos da, para meter otro concentrador usaríamos el anterior elegido y buscaríamos cuál de los otros concentradores nos incrementa más el número de nodos activos como salidas y así sucesivamente.

### 5.6 Fase 4: Obtención del Grafo dirigido y mejoras

El uso de la función en la creación del árbol de expansión mínima nos ha dado una primera ruta a seguir. Como ya tenemos los terminales asignados a los distintos nodos, lo que hacemos es partir de los extremos del árbol, nodos hoja, y vamos utilizando los arcos dados por la función desde estos hacia las distintas salidas.

Lo cual nos da ya una primera solución, sin embargo, esta es fácilmente mejorable, ya que esta solución solo utiliza un número de arcos ya dados cuando en realidad hay un mayor número de posibilidades que pueden mejorar nuestra solución.

Como vemos en el ejemplo que vamos estudiando, en el árbol de expansión mínima el nodo el camino a seguir por el nodo 17 sería de tres saltos hasta la salida uno.

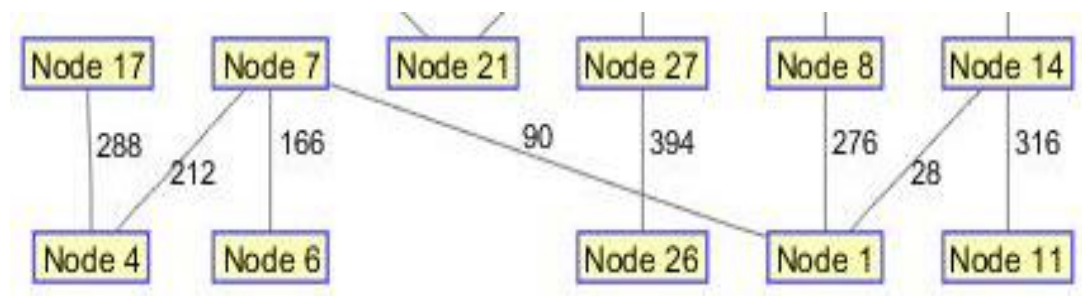
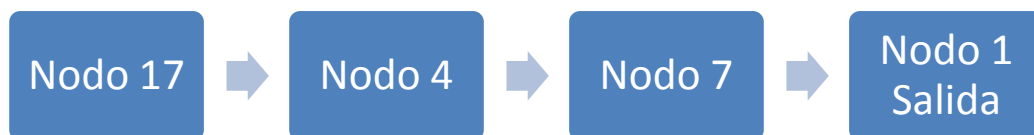


Figura 5-6. Ruta dada por el Árbol de Expansión Mínima.

Es decir, la ruta puesta como una serie de saltos sería:



Sin embargo, en el grafo completo hay una alternativa que dará **una solución que siempre será mejor o igual:**

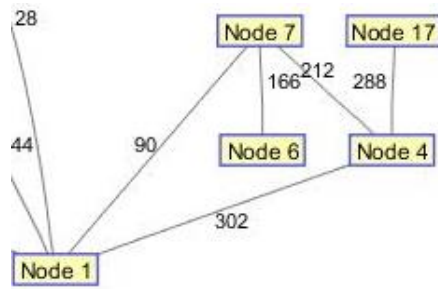


Figura 5-7. Posibles Rutas Alternativas.

Es decir, del nodo 17 vamos al cuatro y del cuatro al uno. Con esta manera de proceder no tenemos que repetir la señal desde siete hasta uno, lo cual significa que si los terminales asignados a 6, 7, 4 y 17 suman más de 20, en 7 tendríamos que poner dos routers y sin embargo al hacer directamente el salto desde 4 hacia uno puede que nos estemos ahorrando un Router.

### 5.6.1 Mejora 1: Reasignación por niveles

Llegados a este punto, vemos que cuanto más cerca está un nodo de la salida menos saltos va a necesitar, por ello se hace una reasignación de los terminales, donde van a estar priorizados por niveles. El nivel cero lo tendrán los nodos salida, ya que estos no tienen que seguir ningún tipo de ruta y por tanto vamos a tratar de asignar todos los terminales posibles a las distintas salidas. El siguiente nivel sería para aquellos que necesitan de un salto para llegar a la salida y así sucesivamente.

De esta manera, no vamos a ser capaces de reducir en ningún número el número de nodos activos, sin embargo sí que podemos disminuir el número de routers necesarios, ya que la repetición de señales puede hacer que en uno de los nodos sea necesario más de un router y al ejecutar de esta manera disminuimos la cantidad de routers.

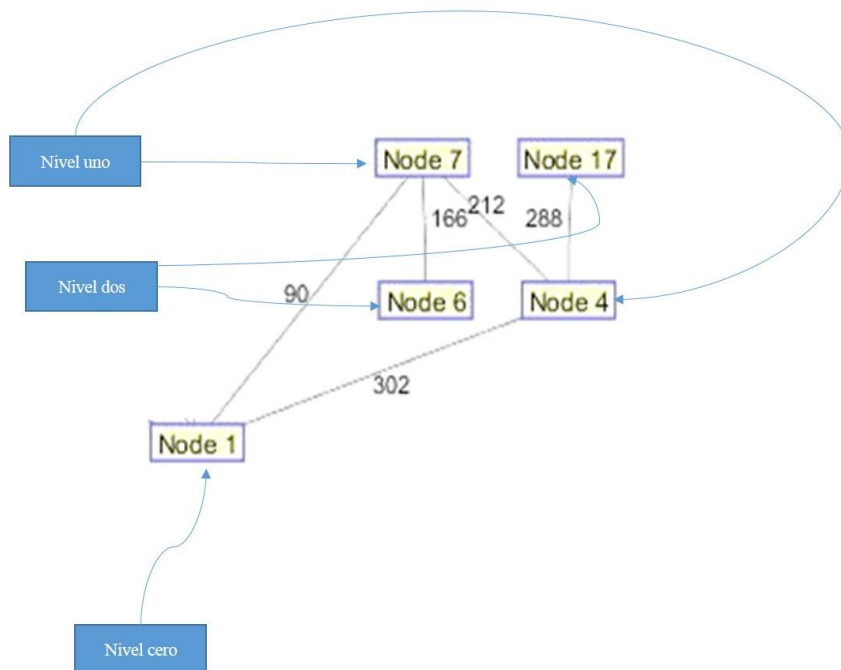


Figura 5-8. Niveles de los nodos.



## 5.6.2 Mejora 2: Ruta a la salida más cercana

Una vez reasignados los terminales, y viendo las mejoras posibles en saltarnos nodos hasta llegar a la salida, buscamos la ruta más corta hacia la salida, esta vez con todos los arcos de los nodos activos. Esto lo hacemos buscando el camino más corto entre los nodos y la salida con la función `graphshortestpath (B, k, i)` que nos dará el camino más corto.

De esta manera estaremos evaluando si el nodo ha quedado asignado a una salida a la que tiene que llegar dando un número de saltos que se podría reducir.

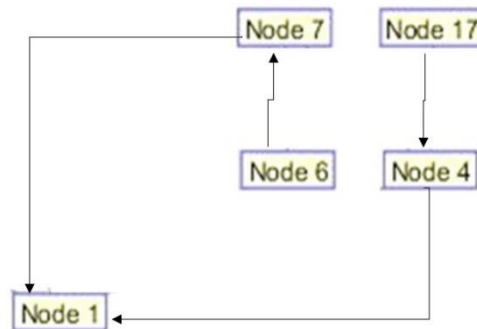


Figura 5-9. Camino a la salida más cercana.

## 5.7 Fase 5: Cálculo de la Función Objetivo

Una vez tenemos la solución con las mejoras pasamos a calcular el valor de la función objetivo. Para calcular el valor de la función objetivo multiplicamos el número de routers que hemos calculado como necesario y lo multiplicamos por el coste de cada uno de los routers. A ese valor le sumamos el número de concentradores por el número de concentradores.

$$\text{Coste} = N^{\circ}\text{Routers} * 98 \left( \frac{\text{€}}{\text{Router}} \right) + N^{\circ}\text{Concentradores} * 350 \left( \frac{\text{€}}{\text{Concentrador}} \right)$$

Lo que nos daría el valor de la función objetivo en euros.

Y finalmente en el ejemplo que hemos venido desarrollando, el coste total sería de 3290€.

# 6 EXPERIMENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

*“Cada hora de tiempo perdido en la juventud es una posibilidad más de desgracia en la adultez.”*

Claude Shannon, 1948

El método propuesto se ha aplicado a una serie de grafos que incluyen la densidad de terminales por calle habituales, que son entre 10 y 20 terminales por calle. A continuación se ponen los resultados de la ejecución del problema.

## 6.1 Experimentación

El HL-MST (Hub Location-Minimum Spanning Tree) se ha programado en Matlab usando sus funciones para el cálculo del árbol de expansión mínima y el resto de las operaciones.

Las instancias vienen clasificadas como AA\_BBB\_C donde cada uno de los elementos significa qué:

AA: es la densidad de terminales por calle, la cual suele variar entre 10 y 20 terminales por calle.

BBB: es el número de terminales que hay en el grafo.

C: es el número de instancia de cada ejemplo.

Como se ha comentado previamente, los datos vienen en un archivo .txt y se ordenarán en Excel. Posteriormente serán cargados en el campo de trabajo de Matlab donde se les aplicara el HL-MST.

Así mismo, se compararán los resultados obtenidos aplicando la heurística con los del MILP (Mixed interger linear programming) para ver a la distancia que quedan las soluciones propuestas y la distancia de su cota superior inicial (GAP).

En la Tabla 6-1 y 6-2 cada elemento significa:

$C_{MILP}^*$ : sería el coste obtenido al calcular con el modelo lineal que viene dado en cada uno de los .txt que se han utilizado. En todos ellos se ha alcanzado la solución óptima.

$C_{HL-MST}$ : es el coste obtenido a partir de la heurística desarrollada que siempre será mayor o igual al coste obtenido por el MILP.

El RPD (relative percent difference) es el tanto por ciento de la diferencia relativa entre el  $C_{MILP}^*$  y el  $C_{HL-MST}$  obtenido con la siguiente fórmula:

$$RPD = \frac{(C_{HL-MST} - C_{MILP}^*)}{C_{HL-MST}} * 100$$

MIP-GAP: Es la diferencia relativa entre las cotas inferior y superior en la ejecución del modelo MILP.

La fórmula utilizada para calcular el MIP-GAP en un instante  $t$  es:

$$MIP - GAP_t = \frac{(CS - CI)}{CS} * 100$$

Donde CS es la cota superior y CI es la cota inferior del problema MILP.

En primer lugar para los grafos con una densidad de 10 terminales por calle los resultados son:

Tabla 6-1. Resultados Con Densidad de 10 Terminales por Calle.

Función Objetivo (Coste €)				
Instancia	$C_{MILP}^*$	$C_{HL-MST}$	RPD (%)	MIP-GAP (solución inicial)
20_200_1	3192	3290	3.97	85.3%
20_200_2	2898	3290	11.91	80.0%
20_200_3	2996	3388	11.57	75.3%
20_200_4	3052	3192	4.38	77.5%
20_200_5	2898	3290	11.91	78.3%
20_200_6	3346	3486	4.02	78.6%
20_200_7	2856	2996	4.67	90.4%
20_200_8	3094	3290	5.96	85.1%
<b>RPD medio</b>			7.33	

Para la segunda batería de grafos aportada por el tutor, se aplica el mismo método pero con una densidad de 20 terminales por calle. Los resultados serían los siguientes:

Tabla 6-2. Resultados Con Densidad de 20 Terminales por Calle.

Instancia	$C_{MILP}^*$	$C_{HL-MST}$	RPD (%)	MIP-GAP (solución inicial)
20_400_1	4594	5642	18.57	82.9%
20_400_2	3542	4662	24.02	81.3%
20_400_3	5792	8386	30.93	83.3%
20_400_4	4270	5544	22.97	87.7%
20_400_5	4382	5894	25.65	82.2%
20_400_6	3738	4326	13.59	83.8%
20_400_7	4130	4816	14.24	71.3%
20_400_8	3934	4620	14.84	74.8%

---

<b>RPD medio</b>	20.60
------------------	-------

---

## 6.2 Análisis de los resultados

A la vista de los resultados obtenidos hay que destacar que:

1. RDP muy bajo para las instancias de menor tamaño (inferior al 7.33% de media). En este tipo de problemas tenemos 20 calles y 200 terminales, para este tamaño la heurística da muy buenos resultados con las asignaciones y las rutas.
2. En instancias de mayor tamaño se obtiene un RDP mayor (inferior al 20.6% de media). En este tipo de problemas tenemos 20 calles y 400 terminales, para este tamaño de problemas la heurística da buenos resultados a pesar de tener una alta densidad de terminales por calle.
3. Hay que tener en cuenta que aunque el RPD para los de mayor densidad parezca en principio elevado, hay que contrastar su valor con la solución obtenida mediante el problema relajado. El problema MILP relajado da una solución admisible del problema real que es la que utiliza el solver como solución inicial.
4. La utilidad de esta heurística es para ser utilizada como método de arranque para posteriores tratamientos. Por ejemplo, la utilización de esta heurística como arranque para el ALNS (Algorithm Large Neighborhood Search, (Pisinger y Ropke, 2009)). Con la aplicación de esta heurística se utilizaría el grafo completo y se reubicarían los distintos terminales tratando de buscar nuevas alternativas. Es decir, con el ALNS estaríamos contemplando la posibilidad de enviar la información de un router a varios y dividir así la señal en vez de únicamente a uno.
5. Aunque el tamaño de los problemas sea limitado y no muy grande, esto se ha hecho para poder visualizar los resultados obtenidos. Es seguro que el método propuesto dará una primera solución de partida para nuevas formas de investigación para problemas de mayor tamaño o más reales.
6. Tiempo de resolución: todo el proceso constructivo de la heurística se hace en un tiempo polinómico por lo que es una ventaja respecto al modelado y resolución del problema MILP que para instancias mayores no logra la solución óptima en un tiempo razonable (según González-R, 2015 quien ha empleado 7200s sin conseguir solución óptima).

Inconvenientes del proyecto:

1. El principal inconveniente es una posible pérdida de calidad en instancias de mayor tamaño, aunque la solución sigue siendo admisible (puede emplearse como solución de partida para MILP y así tener desde el comienzo una cota superior más ajustada).
2. La codificación en Matlab puede ser no eficiente por lo que quizá sea más rápido en lenguajes compilados como C++.
3. Otro inconveniente es que la codificación se ha ido haciendo a partir de las ideas que se iban estudiando para desarrollar el algoritmo lo cual ha hecho no que sea un código desarrollado para aplicar a muchas baterías.
4. Otro inconveniente es que la heurística desarrollada, aunque se ha probado en instancias basadas en parámetros reales, no se ha contrastado con datos de aplicación reales.

## 6.3 Ejecución en Matlab

El algoritmo que se ha explicado se ha implementado en Matlab (MATrix LABoratory).

MATLAB una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

Para explicar cómo arrancar una solución vamos a utilizar la instancia 20\_200\_1.

### 6.3.1 Tutorial

En primer lugar, vamos a necesitar de una computadora que tenga instalada Matlab.

Arrancamos el programa y lo primero que veremos será una imagen similar a la de la figura 6-1:

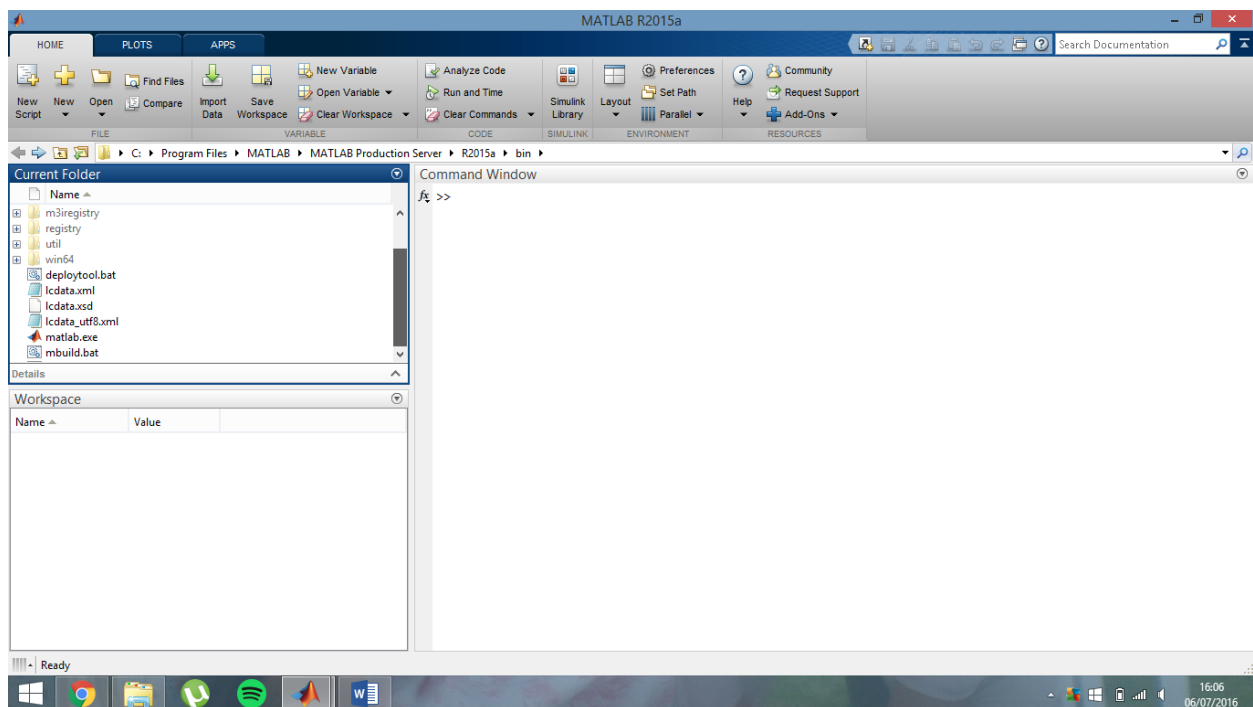


Figura 6-1. Pantalla principal de Matlab.

Una vez abierto hay que esperar a que se cargue el programa y cuando esté listo nos vamos a la carpeta en la que está el archivo 20\_200\_1\_Tutorial.

Una vez en esa carpeta tendremos que darle clic con el botón derecho de ratón como vemos en la figura 6-2 y pulsar la opción Import Data...

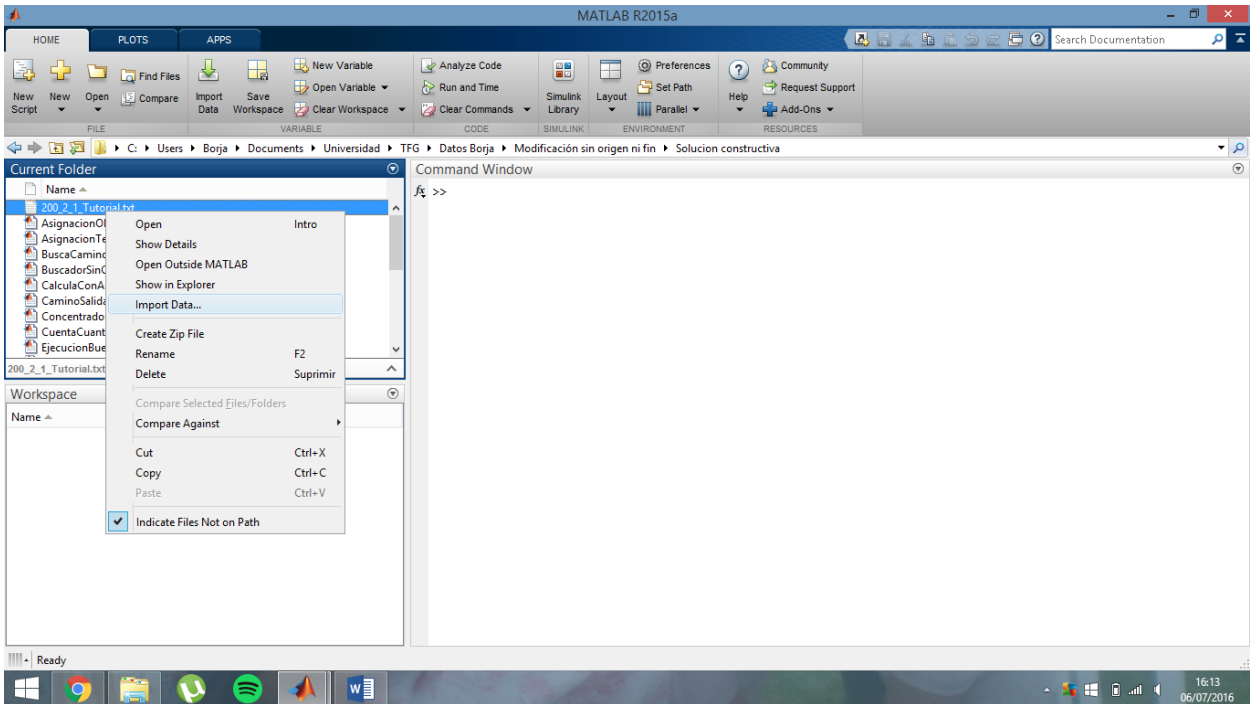


Figura 6-2. Seleccionar archivo.

Una vez hecho esto nos aparecerán tres columnas como vemos en la figura 6-3 con los datos ya ordenados y en el menú de arriba a la derecha le damos a Import selection.

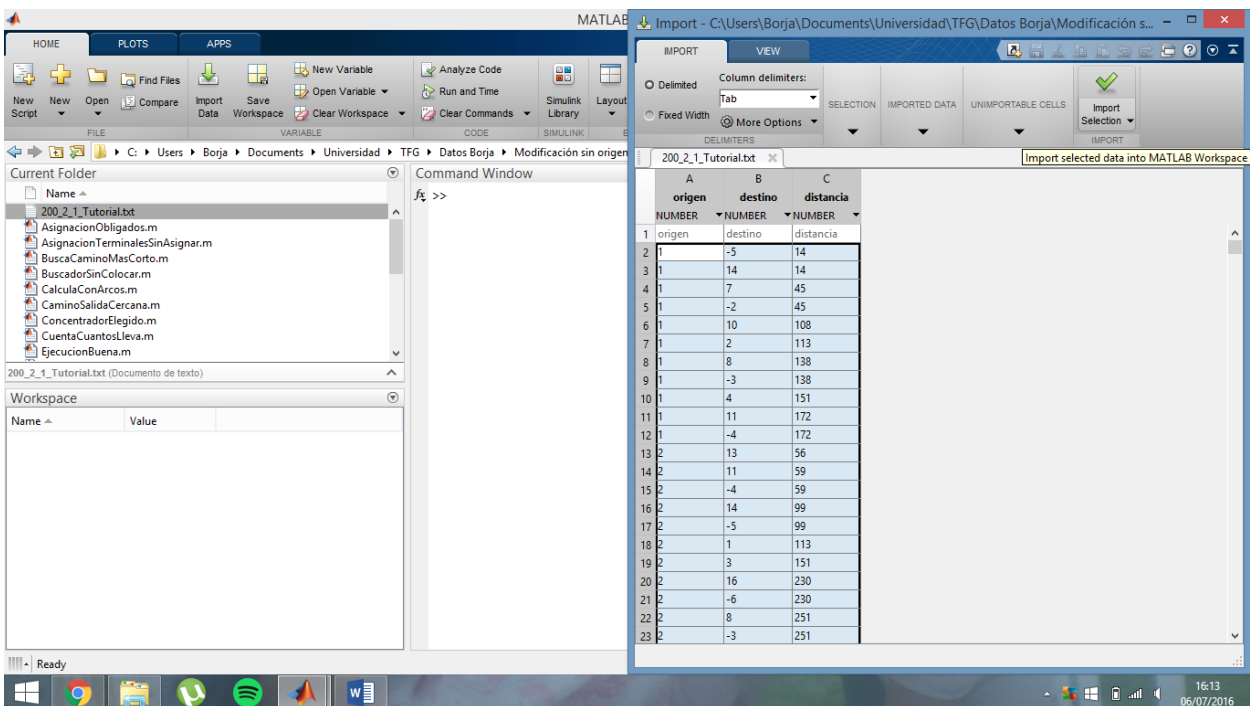
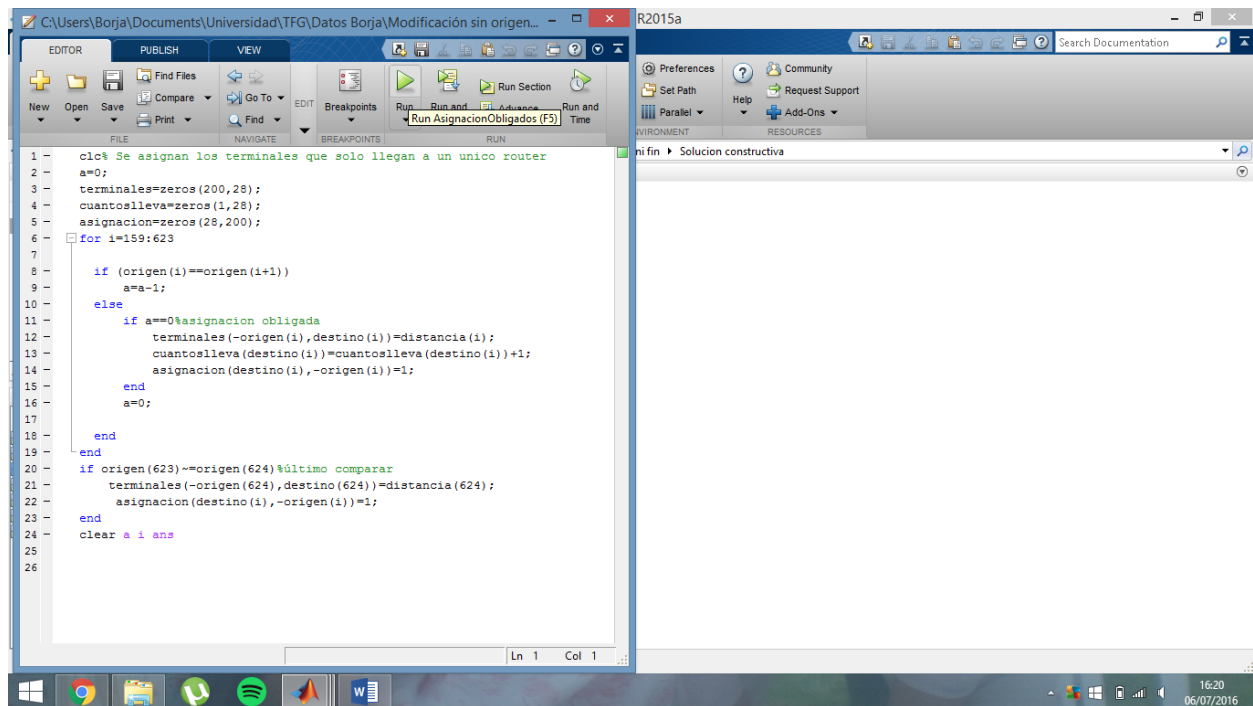


Figura 6-3. Importar datos.

Con este paso se crearán tres vectores, origen, destino y distancia que son los datos de entrada.

Tras esto, empezamos la ejecución.

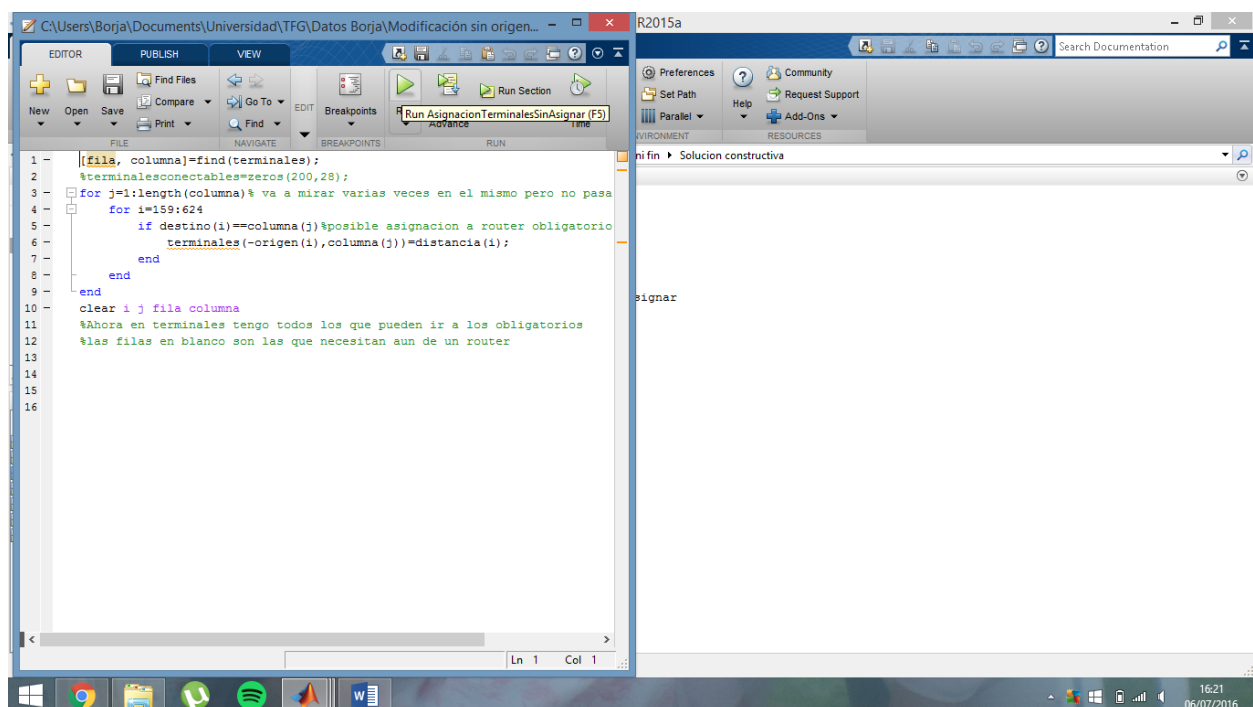
Abrimos el fichero AsignacionObligados y nos aparecerá una imagen como la figura 6-4. Hacemos clic en el botón de Run(play).



```
1 - clc % Se asignan los terminales que solo llegan a un unico router
2 - a=0;
3 - terminales=zeros(200,28);
4 - cuantoslleva=zeros(1,28);
5 - asignacion=zeros(28,200);
6 - for i=159:623
7 -
8 -     if (origen(i)==origen(i+1))
9 -         a=a-1;
10 -    else
11 -        if a==0%asignacion obligada
12 -            terminales(-origen(i),destino(i))=distancia(i);
13 -            cuantoslleva(destino(i))=cuantoslleva(destino(i))+1;
14 -            asignacion(destino(i),-origen(i))=1;
15 -        end
16 -        a=0;
17 -    end
18 - end
19 - end
20 - if origen(623)~=origen(624)%ultimo comparar
21 -     terminales(-origen(624),destino(624))=distancia(624);
22 -     asignacion(destino(i),-origen(i))=1;
23 - end
24 - clear a i ans
25 -
26 -
```

Figura 6-4. Fichero asignación obligados.

Tras este paso, abrimos AsignacionTerminalesSinAsignar.m y nos aparece una ventana como la figura 6-5 y le damos a Run. Con este paso ya tenemos asignados los terminales que llegaban a alguno de los routers obligatorios.



```
1 - [fila, columna]=find(terminales);
2 - %terminalesconectables=zeros(200,28);
3 - for j=1:length(columna)% va a mirar varias veces en el mismo pero no pasa
4 -     for i=159:624
5 -         if destino(i)==columna)%posible asignacion a router obligatorio
6 -             terminales(-origen(i),columna(j))=distancia(i);
7 -         end
8 -     end
9 - end
10 - clear i j fila columna
11 - %Ahora en terminales tengo todos los que pueden ir a los obligatorios
12 - %las filas en blanco son las que necesitan aun de un router
13 -
14 -
15 -
16 -
```

Figura 6-5. Fichero asignación terminales sin conectar.

Tras esto, abrimos CuentaCuantosLleva.m y nos aparece una ventana como la figura 6-6 y le damos a Run.

Obtenemos la cantidad de terminales que han quedado asignados.

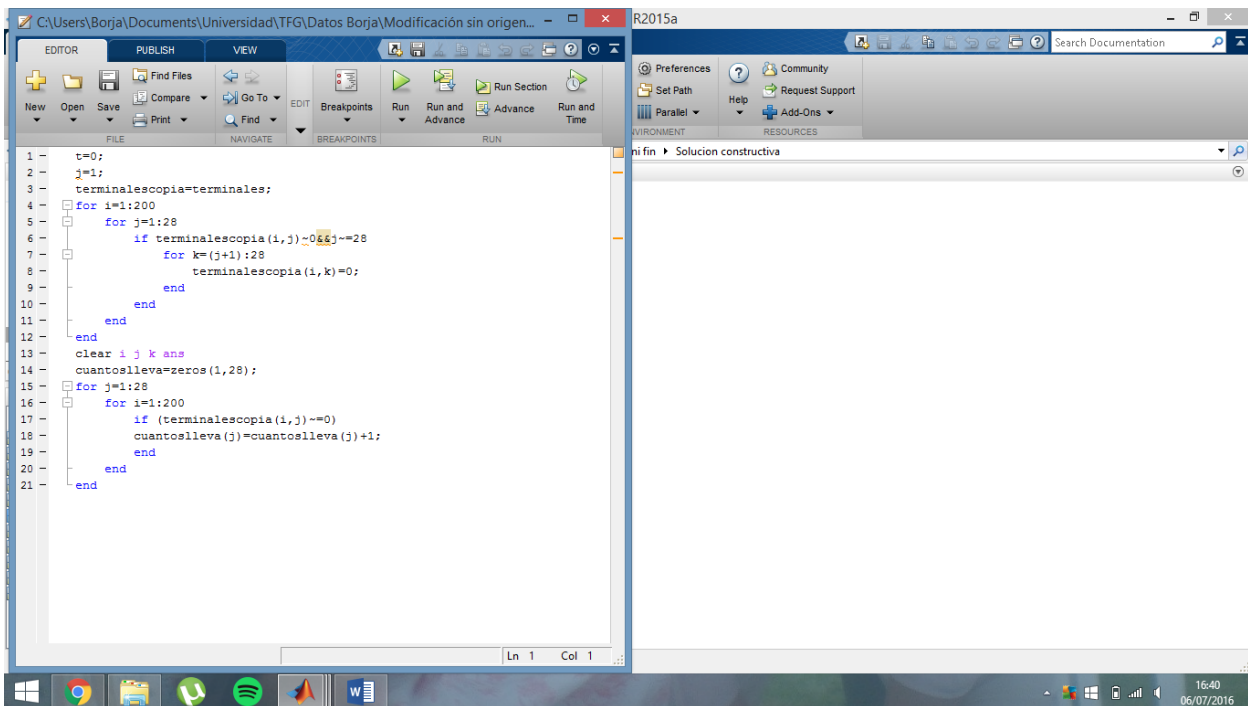


Figura 6-6. Fichero cuenta cuantos lleva.

Ahora abrimos el archivo BuscadorSinColocar.m y nos aparece una ventana como la figura 6-7 que se dedicará a buscar cual de los terminales no ha quedado asignado. Una vez abierto le damos a Run.

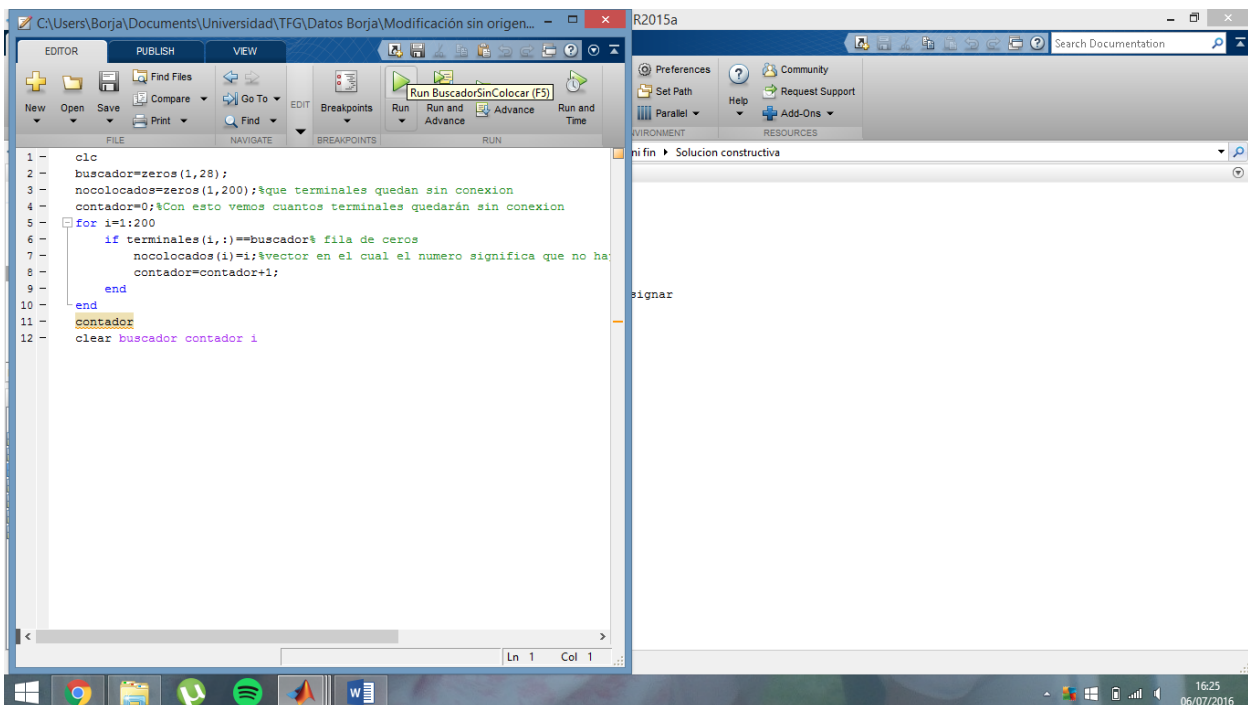


Figura 6-7. Fichero buscador sin colocar.

Ahora abrimos TerminalesSinConexion.m y nos aparece una ventana como la figura 6-8 y le damos a Run. Esto creará una matriz con los terminales no colocados.



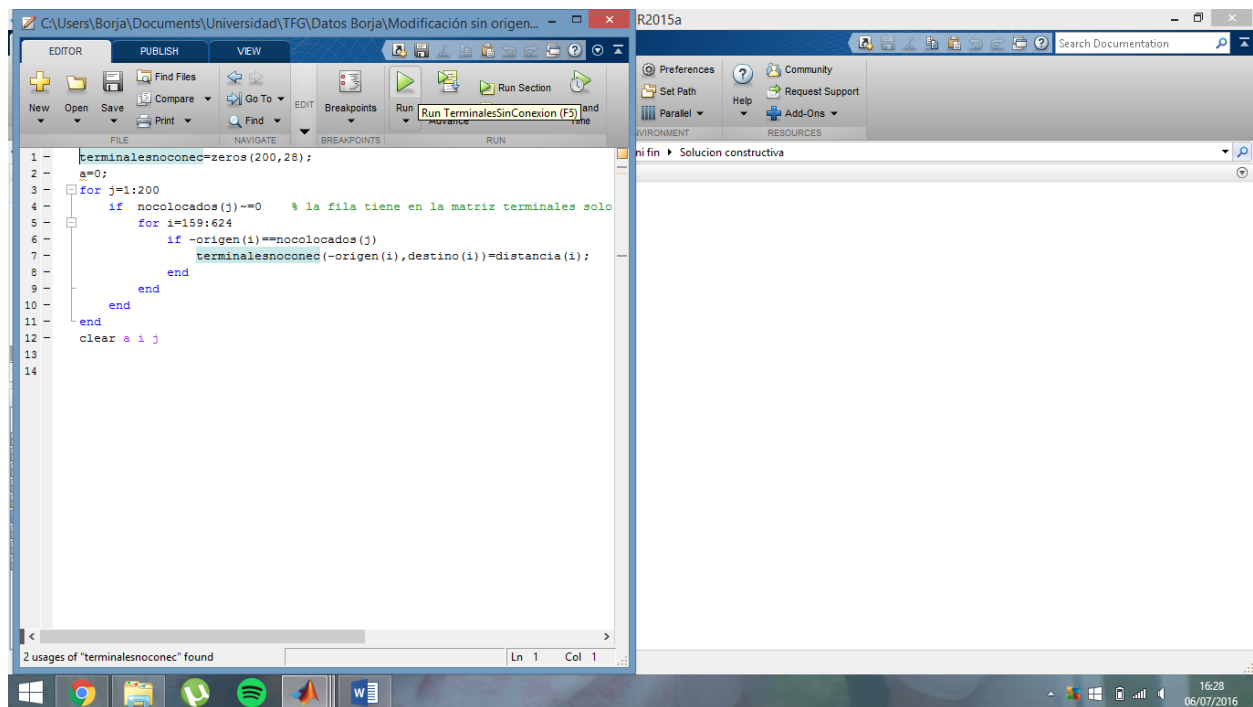


Figura 6-8. Fichero terminales sin conexión.

Ahora abrimos el archivo RoutersMasConectables.m y nos aparece una ventana como la figura 6-9 que evaluará cada uno de los routers no colocados y nos dará cuantos terminales llegan a cada uno de los routers. Una vez abierto el archivo le damos a Run.

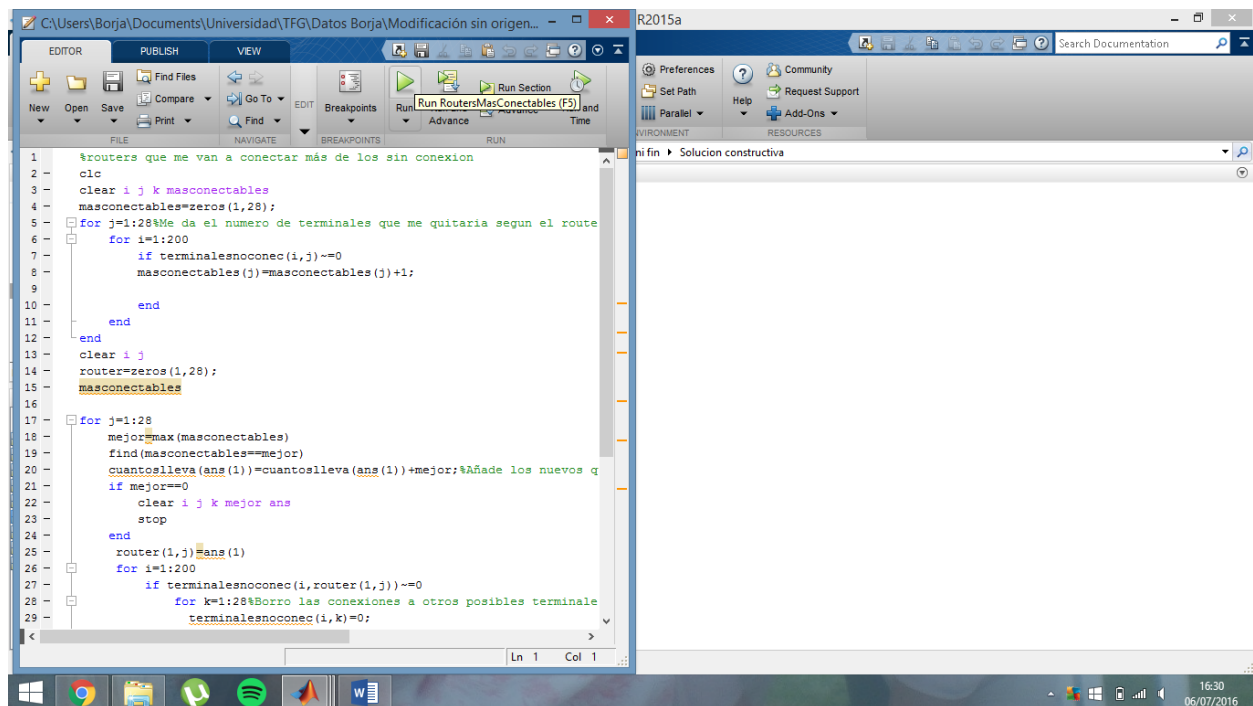
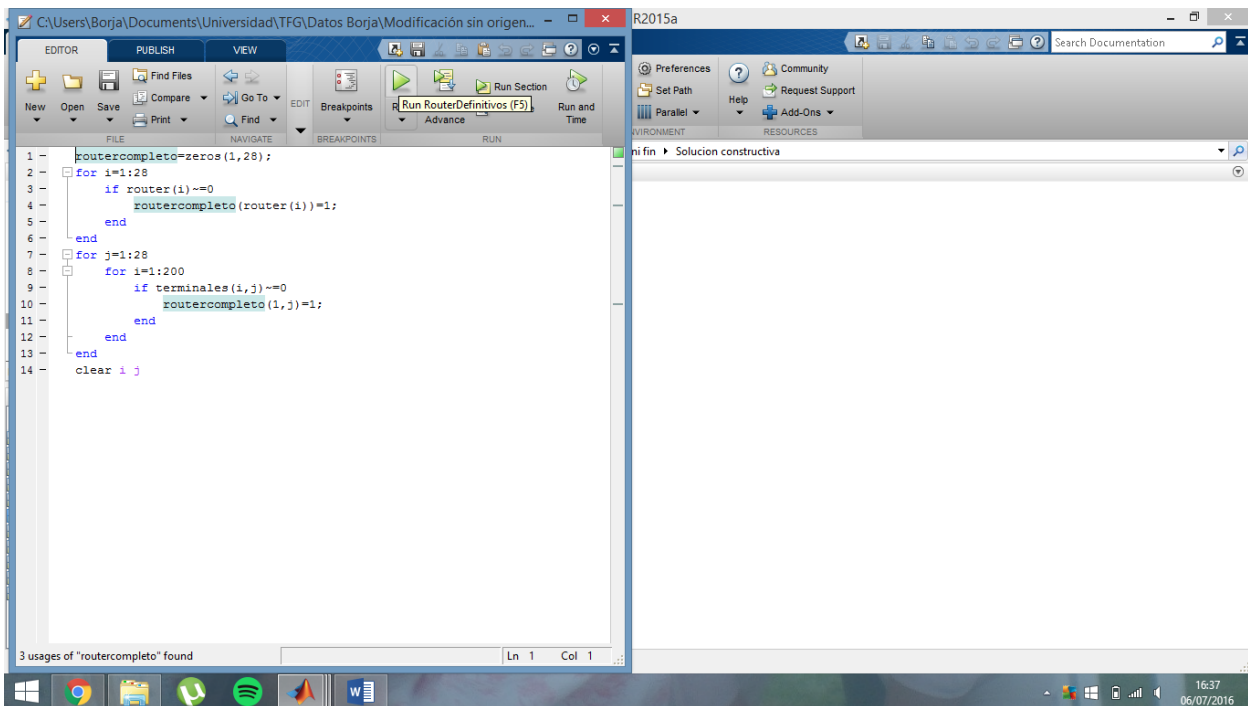


Figura 6-9. Fichero routers más conectables.

Tras esto, abrimos el archivo RouterDefinitivos.m y nos aparece una ventana como la figura 6-10 y cuando esté

abierto le damos a Run. Aquí se eligen el resto de routers necesarios para tener todos los terminales asignados.

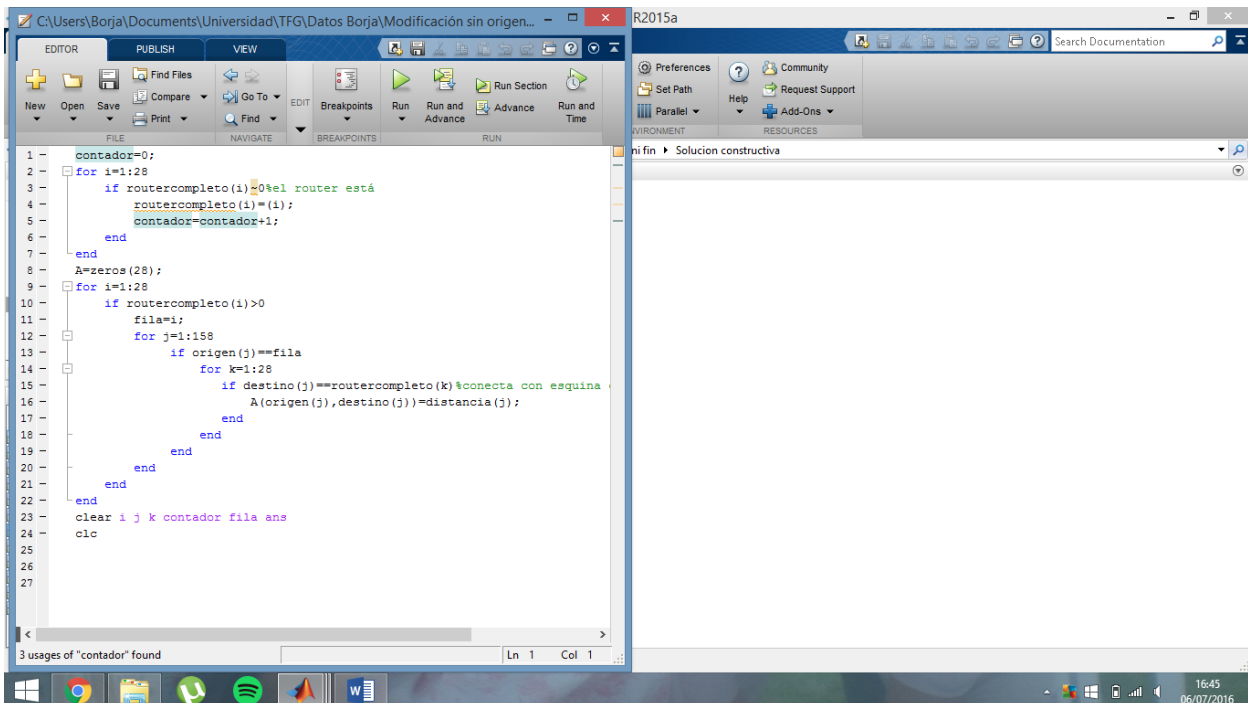


```

1 routercompleto=zeros(1,28);
2 for i=1:28
3     if router(i)~=0
4         routercompleto(router(i))=1;
5     end
6 end
7 for j=1:28
8     for i=1:200
9         if terminales(i,j)~=0
10            routercompleto(i,j)=1;
11        end
12    end
13 end
14 clear i j
  
```

Figura 6-10. Fichero routers definitivos.

Tras esto pasamos a abrir NuevoGrafoRouters.m y nos aparece una ventana como la figura 6-11 donde se creará la matriz con las conexiones entre los routers que han quedado activos. Una vez abierto le damos a Run.



```

1 contador=0;
2 for i=1:28
3     if routercompleto(i)~=0%el router está
4         routercompleto(i)=(i);
5         contador=contador+1;
6     end
7 end
8 A=zeros(28);
9 for i=1:28
10    if routercompleto(i)>0
11        fila=i;
12        for j=1:158
13            if origen(j)==fila
14                for k=1:28
15                    if destino(j)==routercompleto(k) %conecta con esquina
16                        A(origen(j),destino(j))=distancia(j);
17                    end
18                end
19            end
20        end
21    end
22 end
23 clear i j k contador fila ans
24 clc
  
```

Figura 6-11. Fichero nuevo grafo routers.

Ahora tenemos la matriz de conexiones y abrimos el archivo EjecucionBuena.m y le damos a Run. Esto generará un dibujo del grafo y del árbol de expansión mínima.

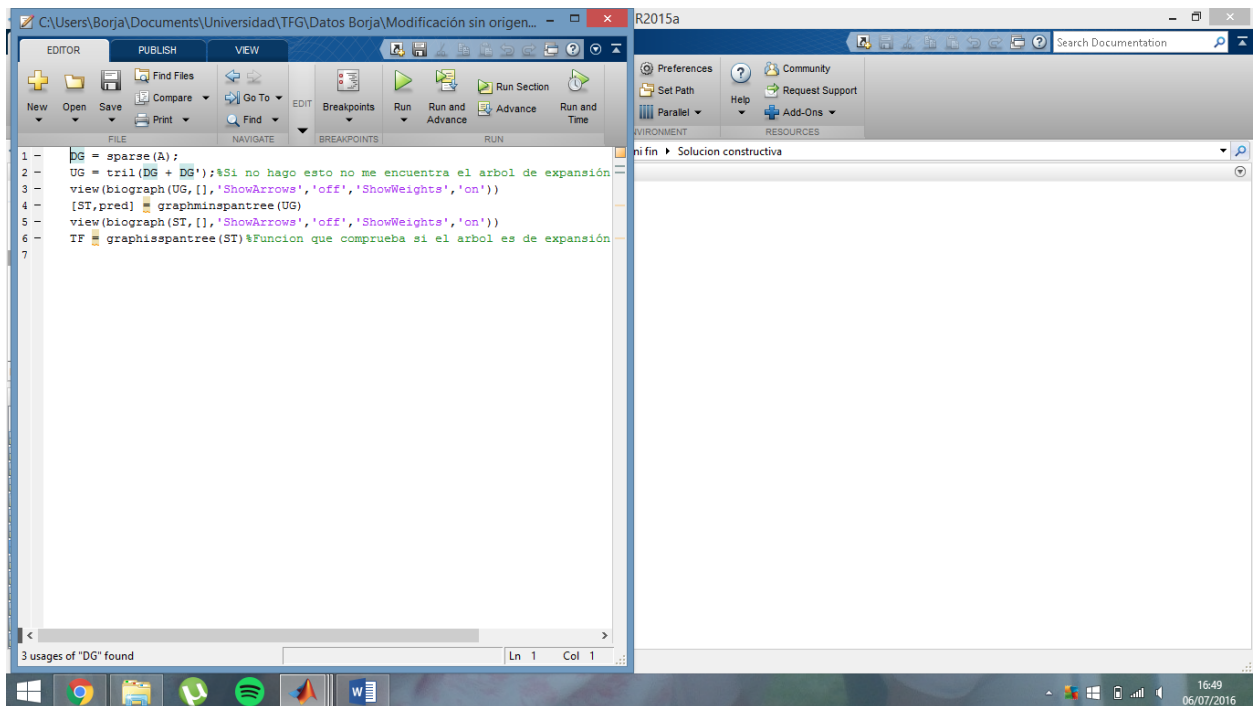


Figura 6-12. Fichero ejecución buena.

Ahora buscamos cuales de los nodos no han quedado conectados con el arbol. Para esto abrimos BuscaSinArbol.m y nos aparece una ventana como la figura 6-13 y le damos a Run.

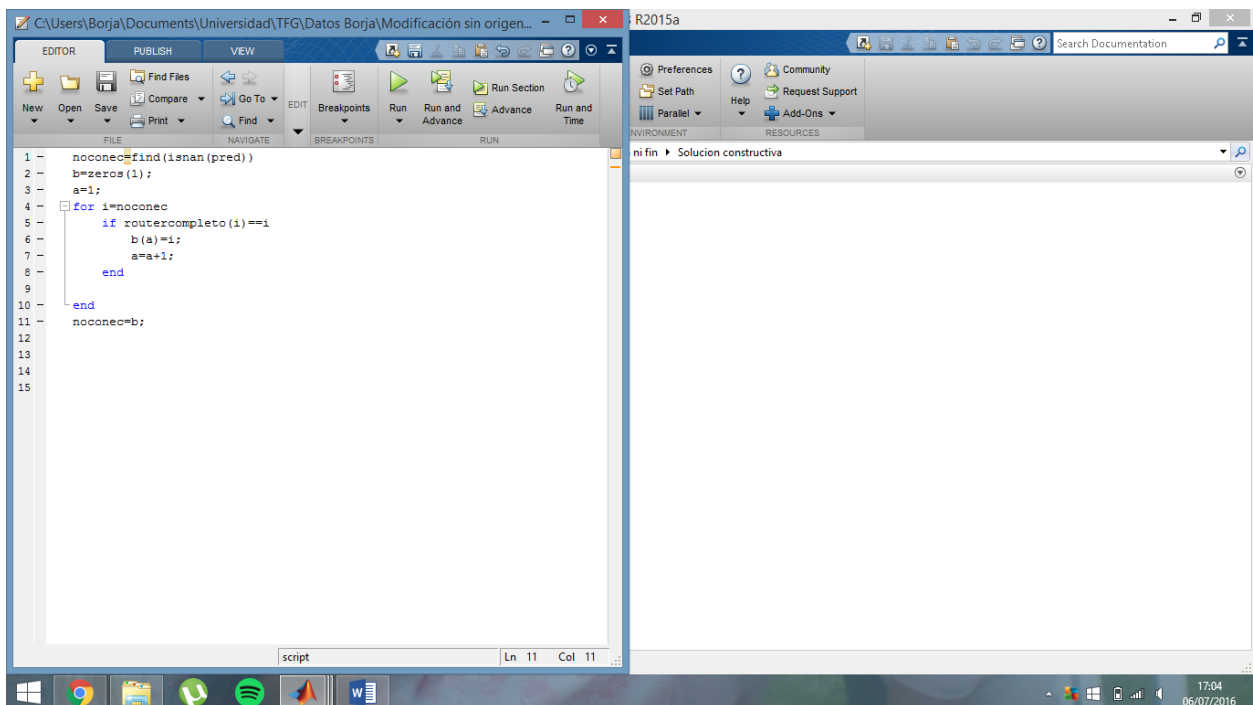


Figura 6-13. Fichero busca sin árbol.

Ahora ya tenemos los nodos que han quedado desconectados y buscamos la forma de unirlos con el resto del arbol. Para esto abrimos BuscaCaminoMasCorto y nos aparece una ventana como la figura 6-14 y le damos a

Run.

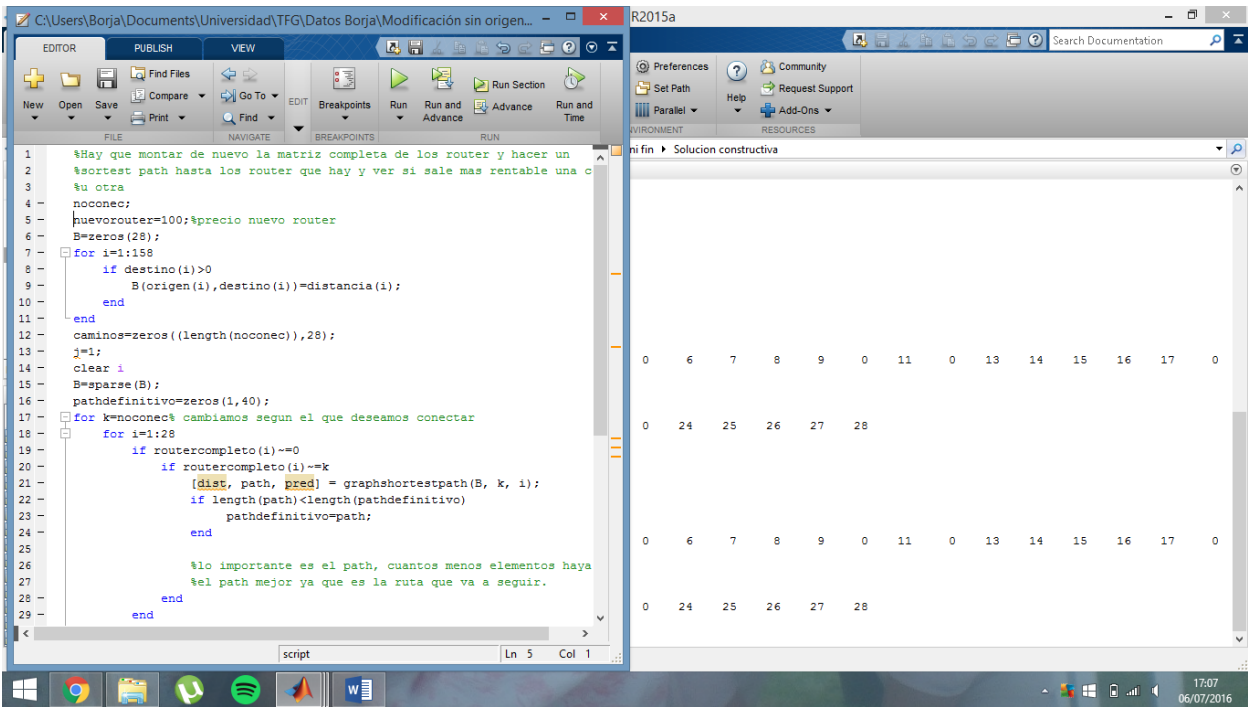


Figura 6-14. Fichero busca camino más corto.

Ahora tenemos que crear una matriz donde todos los nodos están conectados entre sí. Ahora volvemos a ejecutar la función NuevoGrafoRouters y nos dará la nueva matriz de conexiones. Este paso es igual que la otra ejecución de NuevoGrafoRouters.m.

Ahora ejecutamos otra vez EjecucionBuena.m y nos dará el grafo definitivo de conexiones y el árbol de expansión mínima definitivo.

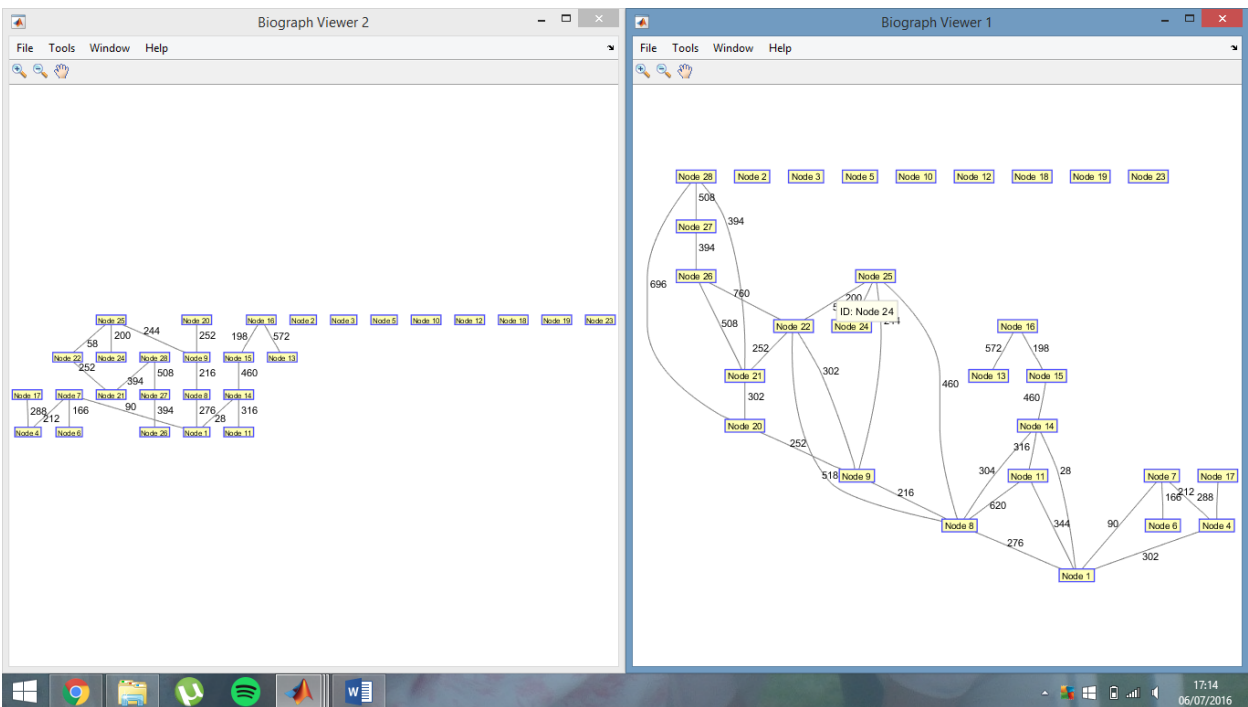


Figura 6-15. Árbol de expansión mínima y grafo asociado de conexiones.

Ahora abrimos la funcion SeleccionaConcentradores.m y nos aparece una ventana como la figura 6-16 que me dirá cuales de los concentradores son mejores. Una vez abierto, le damos a Run.

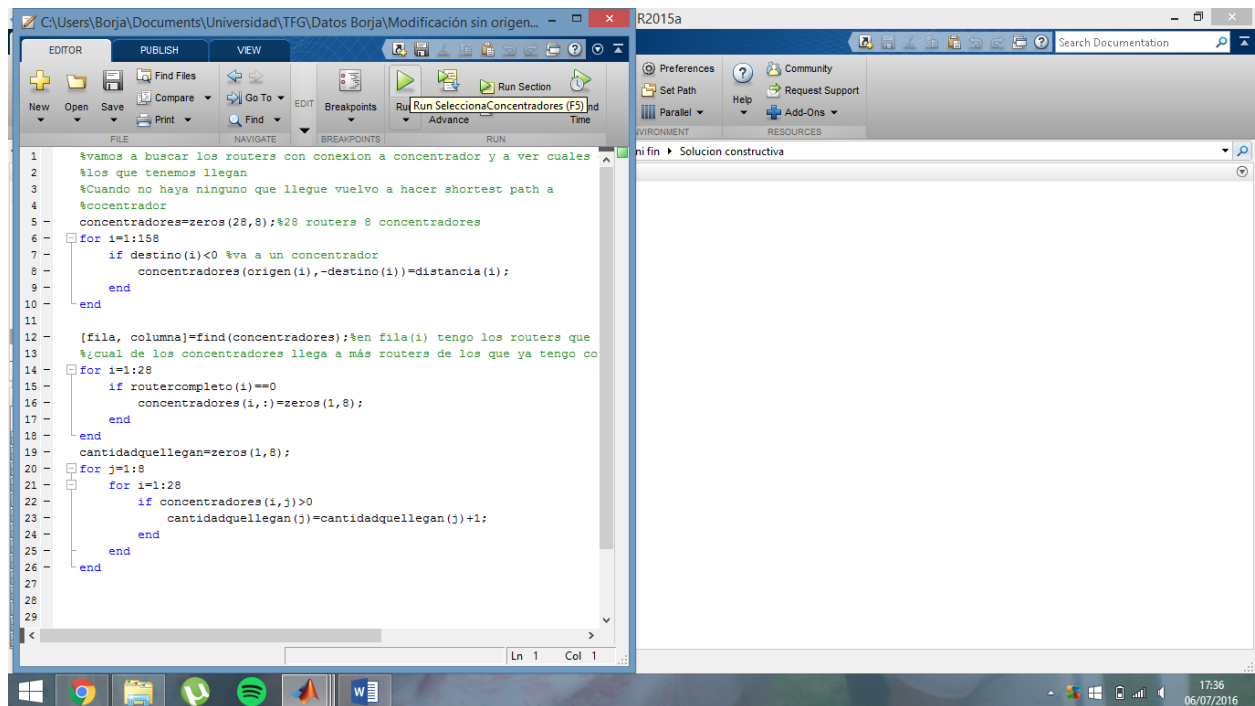


Figura 6-16. Fichero selecciona concentradores.

Ahora abrimos ConcentradorElegido.m y nos aparece una ventana como la figura 6-17 y le damos a run, donde utilizando la evaluación de la anterior funcion selecciona concentrador.

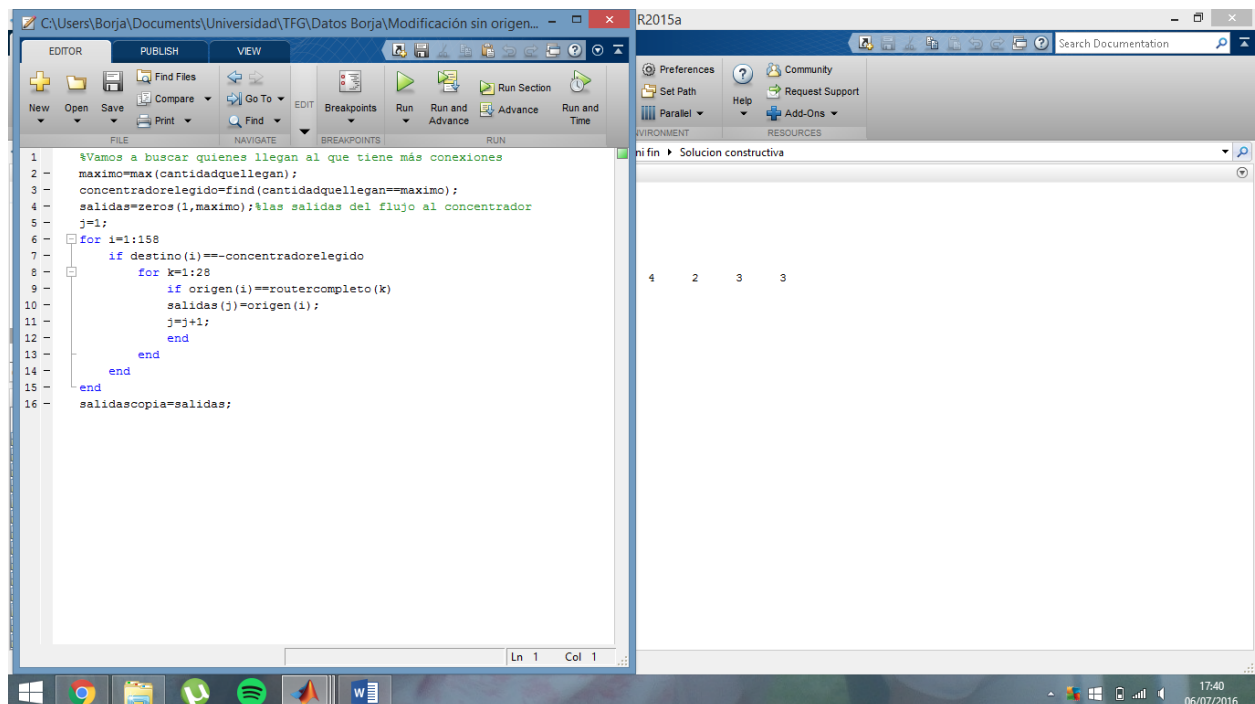


Figura 6-17. Fichero concentrador elegido.

Ahora abrimos CalculaConArcos.m y nos aparece una ventana como la figura 6-18. Una vez abierto el archivo

le damos a Run.

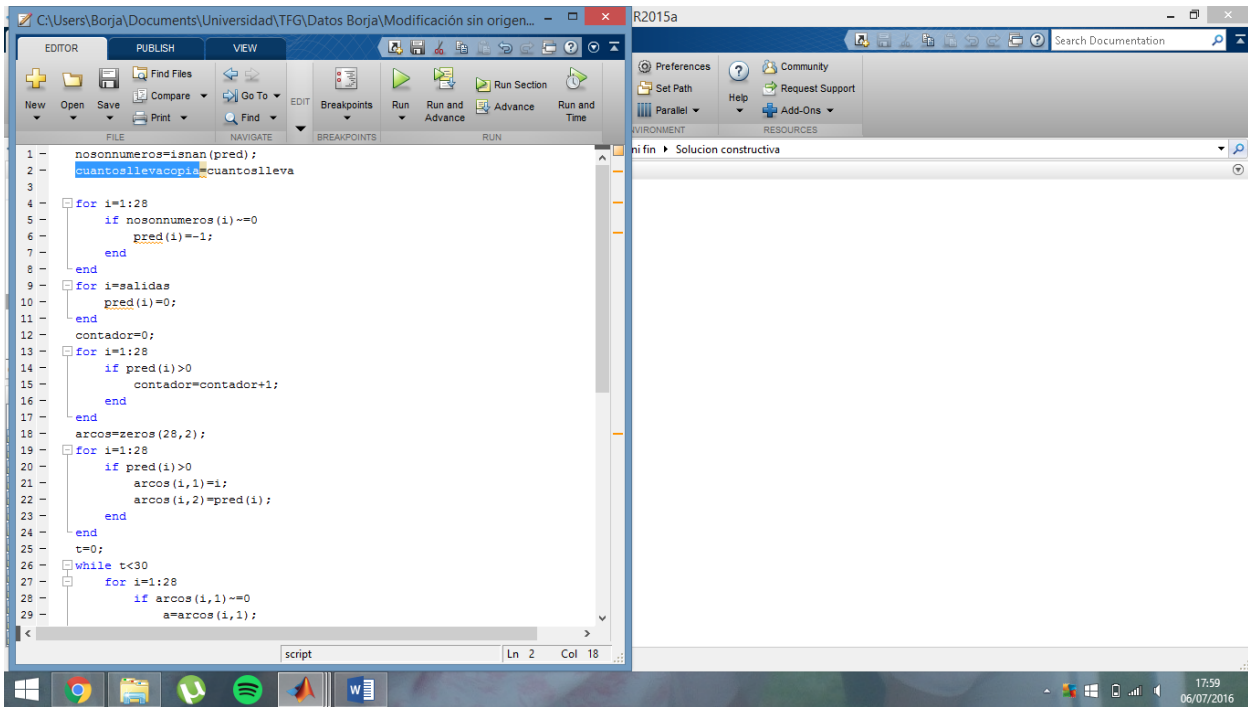


Figura 6-18. Fichero calcula con arcos.

Ahora abrimos ReasignacionDeTerminales.m y nos aparece una ventana como la figura 6-19 donde comenzaremos a reasignar los terminales en otro orden. Una vez abierto el archivo le damos a Run.

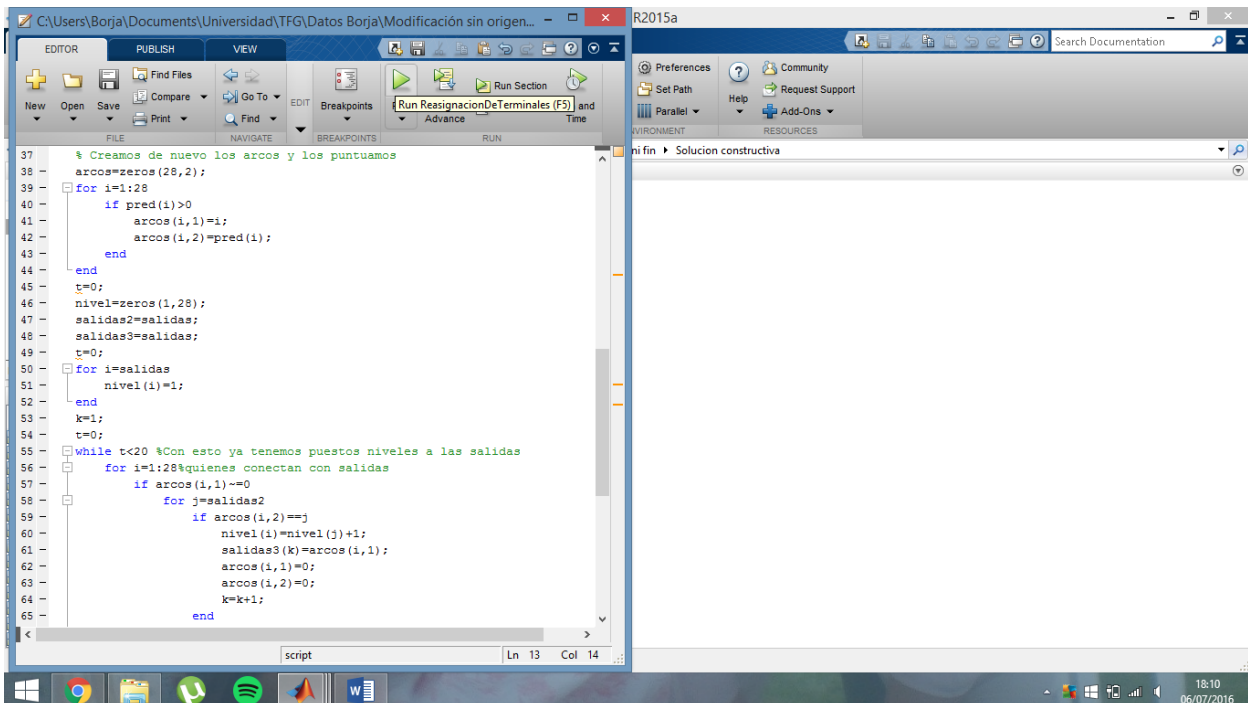


Figura 6-19. Fichero reasignación de terminales.

Ahora abrimos ReasignacionPorNiveles.m y nos aparece una ventana como la figura 6-20 y le damos a Run. Con esta función hemos reasignado los terminales en función de la distancia al concentrador. Una vez abierto el script le damos a Run.

```

1 for k=2:max(nivel) % Vamos a asignar primero a los que están más cerca
2 candidatosactuales=find(k==nivel);
3 for i=candidatosactuales
4     for j=159:624
5         if destino(j)==1%Se asigna unicamente a una salida
6             nosirve=0;
7             for k=1:28
8                 if asignacion(k,-origen(j))==1%Ya está asignado
9                     nosirve=1;
10                end
11            end
12            if nosirve==0
13                asignacion(destino(j),-origen(j))=1;
14            end
15        end
16    end
17 end
18 end
19 resultadodeasignaciones=zeros(1,28);
20 for i=1:28
21     resultadodeasignaciones(i)=sum(asignacion(i,:));
22 end
23
24
25 arcos=zeros(28,2);
26 for i=1:28
27     if pred(i)>0
28         arcos(1,1)=1;
29         arcos(i,2)=pred(i);

```

Figura 6-20. Fichero reasignación por niveles.

Y finalmente abrimos CaminoSalidaCercana.m y nos aparece una ventana como la figura 6-21 y le damos a Run. Esto nos dará la solución definitiva del algoritmo.

```

12 [dist, path, pred3] = graphshortestpath(DC, i, j);
13 a=length(path);
14 if length(path)<length(pathdefinitivo)
15     if a>0
16         pathdefinitivo=path;
17     end
18 end
19 end
20
21 (pathdefinitivo)
22 pathdefinitivo(:,1)=[ ]
23 for j=(pathdefinitivo)
24     resultadodeasignaciones3(j)=resultadodeasignaciones
25 end
26 end
27
28 end
29 pathdefinitivo=zeros(1,40);
30 end
31
32 cuantosencada2=zeros(1,28);
33 numerototal=0;
34 j=1;
35 for i=1:28
36     if resultadodeasignaciones(i)>0
37         while (resultadodeasignaciones3(1)/(20*j))>1
38             j=j+1;
39         end
40         numerototal=numerototal+j;

```

Figura 6-21. Fichero camino salida cercana.

## 7 CONCLUSIONES

Para concluir este TFG se van a comentar las conclusiones que hemos obtenido después de realizar el análisis de los resultados del anterior capítulo.

El problema objeto de estudio aborda la localización de routers en un sistema de recogida de datos para una red de abastecimiento de aguas. Como se ha argumentado en la revisión de la literatura dicho problema es considerado NP duro, por lo que los modelos matemáticos tienen una aplicación limitada. Debido a esto, surgen maneras alternativas para resolver este tipo de problemas. En ocasiones puede ser interesante renunciar a la optimalidad del problema a favor de obtener soluciones admisibles rápidamente con una calidad adecuada.

En el segundo capítulo hemos estudiado el problema y sus condiciones. Aquí hemos analizado los distintos tipos de redes y cómo se comportan para poder caracterizar nuestro problema a las redes existentes y entender cómo funcionan los distintos tipos, particularmente hemos detallado nuestro problema, sus condiciones y como son cada uno de los elementos que lo forman. De esta manera, hemos cumplido el primer objetivo específico, comprensión detallada del problema.

En el tercer capítulo hemos revisado varios trabajos ya desarrollados por otros autores en el problema de la localización de routers. Así mismo, hemos caracterizado nuestro problema y lo hemos enmarcado entre ellos. En este capítulo también se ha hecho una revisión de los algoritmos más utilizados en la creación de Árboles de Expansión Mínima, ya que la técnica constructiva propuesta se basa en la generación de un árbol de expansión mínima y hemos analizado sus diferencias. Con esta forma de proceder hemos cumplido el objetivo específico número dos, Realizar una revisión o estado del Arte.

En el cuarto capítulo hemos estudiado el grafo asociado al problema, los tipos de nodos del problema y los diferentes tipos de arcos que hay entre ellos. Además en este capítulo hemos estudiado el modelo matemático del problema y hemos visto la topología de las soluciones óptimas de donde hemos obtenido la conclusión de que las soluciones tienen una gran similitud con el Árbol de Expansión Mínima. Por ello concluimos que el objetivo específico número tres, Modelo matemático y Grafo asociado, ha sido cumplido.

En el quinto capítulo hemos desarrollado el proceso Heurístico constructivo y hemos ido detallando en cada uno de los pasos del proceso. Así mismo en este capítulo se ha explicado la heurística utilizando un ejemplo y viendo los distintos resultados. Debido a esto obtenemos la conclusión de que el objetivo específico número cuatro, Proceso Constructivo, ha sido correctamente cumplido.

En el sexto capítulo hemos aplicado la heurística a una pequeña batería de problemas y hemos obtenido los resultados correspondientes. Así mismo, hemos analizado los resultados y hemos visto que las soluciones son siempre admisibles y muy buenas para problemas pequeños y para problemas más grandes la solución sigue siendo admisible y aunque siguen siendo buenas nunca llegan al valor de la solución óptima. Por ello, obtenemos la conclusión de que el objetivo específico número cinco, Discutir las implicaciones prácticas de los resultados obtenidos, ha sido cumplido.

Podemos entonces afirmar que hemos logrado alcanzar el objetivo general de este TFG ya que hemos podido cumplir todos los objetivos específicos que nos habíamos puesto.

A pesar de que los resultados obtenidos con la heurística nunca llegan a ser igual de buenos que las soluciones obtenidas por el modelo MILP, la heurística aquí propuesta obtendrá siempre una solución inicial de partida admisible en un tiempo de ejecución es mínimo.

Analizando los resultados, vemos que para problemas con un número menor de terminales las soluciones obtenidas por la heurística son mejores que aquellos casos en los que hay una mayor cantidad de terminales.

Hemos de comentar que en este TFG se ha utilizado la heurística propuesta para un modelo donde no hay routers GPRS (donde se permitiría la conexión directa de cada nodo router al nodo final) y una ampliación de esta heurística sería contemplar el caso de que sí existiera y ver cómo se comporta.

Una futura ampliación de este proyecto sería el utilizar esta heurística como punto de arranque para nuevas



heurísticas para seguir mejorando la solución. Una de las opciones sería aplicar el algoritmo ALNS (Algorithm Large Neighborhood Search , ver por ejemplo Pisinger y Ropke, 2009).

Otra posible ampliación de este TFG sería el aplicar el método desarrollado a problemas reales y a otros problemas similares de abastecimiento de otros servicios y ver cómo se comporta.

Otra opción sería ampliar el número de problemas, con mayor diversidad de tamaños, a los que se le ha desarrollado la heurística y desarrollar la heurística en alguno de los lenguajes compilados.

## REFERENCIAS

---

- Alumur, S. y Kara, B.Y. (2008) “Network Hub Location Problems: state of the art”, *European Journal of Operational Research*, 190(1), 1-21.
- González-R, P.L. (2015), “A set of instances for the capacitated hub location model for WIFI data reading in a water supply company”, Technical Report.
- Hekmatfar, M. y Pishvae, M. (2009). “Facility Location: Concepts, Models, Algorithms and Case Studies”. *Springer-Verlag*.
- Kruskal, J. B. (1956) “On the shortest spanning subtree and the traveling salesman problem”. *Proceedings of the American Mathematical Society*. 7, pp. 48–50.
- Pisinger, David y Ropke, Stefan (2007) “A general heuristic for vehicle routing problems”, *Computers & Operations Research* 34, 2403–2435.
- Prim, R. C. (1957). “Shortest connection networks and some generalisations”. *Bell System Technical Journal*, 36, pp. 1389–1401.
- ReVelle, C.S. y Eiselt, H.A. (2005). “Location analysis: A synthesis and survey”. *European Journal of Operational Research*, 165(1), 1-19.
- ReVelle, C.S., Eiselt, H.A. y Daskin M.S. (2008). “A bibliography for some fundamental problem categories in discrete location science”. *European Journal of Operational Research*, 184, 817–848.

# ANEXO I. BATERÍA 20-200

A continuación se muestran los datos de la batería de problemas en forma de tabla proporcionada por el tutor (González-R, 2015).

En esta tabla se contemplan los problemas con una densidad de 10 terminales por calle.

Donde cada columna de la tabla significa que:

Origen: es el comienzo del arco, cuando este valor sea positivo es un arco que comienza en un router. En caso contrario el origen será en un terminal.

Destino: Es el final del arco, cuando este valor es positivo, es un arco entre routers y cuando sea negativo es un arco hacia concentrador.

Distancia: es la longitud del arco.

Los datos de las tablas se mostrarán en dos columnas por cada una de las páginas

Los datos para el 20-200-1 son:

Origen	Destino	Distancia	Origen	Destino	Distancia
1	4	122.0	4	8	472.0
1	-1	186.0	4	3	183.0
1	7	186.0	4	-4	64.0
1	17	285.0	4	-4	183.0
1	2	183.0	4	-4	122.0
1	-1	0.0	5	20	458.0
2	-2	122.0	5	6	64.0
2	16	32.0	5	8	181.0
2	3	122.0	5	4	291.0
2	1	183.0	5	11	350.0
2	19	196.0	5	10	63.0
2	13	334.0	5	-5	474.0
2	17	102.0	5	26	281.0
2	-2	183.0	5	3	474.0
2	-2	32.0	6	7	291.0
3	2	122.0	6	-6	291.0
3	11	124.0	6	10	1.0
3	4	183.0	6	5	64.0
3	13	212.0	7	4	64.0
3	19	318.0	7	-7	186.0
3	-3	90.0	7	-7	0.0
3	16	90.0	7	6	291.0
3	5	474.0	7	1	186.0
3	-3	0.0	8	26	100.0
3	14	150.0	8	22	247.0
4	11	59.0	8	20	277.0
4	5	291.0	8	9	63.0
4	7	64.0	8	4	472.0
4	1	122.0	8	5	181.0
4	14	333.0	9	10	181.0

Origen	Destino	Distancia	Origen	Destino	Distancia
9	8	63.0	19	16	228.0
9	22	310.0	20	25	158.0
10	5	63.0	20	21	247.0
10	6	1.0	20	28	76.0
10	9	181.0	20	8	277.0
11	4	59.0	20	26	177.0
11	12	212.0	20	5	458.0
11	5	350.0	21	28	171.0
11	-11	124.0	21	22	277.0
11	14	274.0	21	20	247.0
11	3	124.0	21	23	207.0
12	13	124.0	21	25	89.0
12	11	212.0	22	23	484.0
13	-13	302.0	22	9	310.0
13	3	212.0	22	21	277.0
13	-13	212.0	22	8	247.0
13	16	302.0	23	22	484.0
13	2	334.0	23	21	207.0
13	12	124.0	23	24	89.0
14	-14	150.0	24	23	89.0
14	15	90.0	24	25	207.0
14	4	333.0	25	20	158.0
14	3	150.0	25	24	207.0
14	11	274.0	25	21	89.0
15	16	150.0	25	28	82.0
15	14	90.0	26	8	100.0
15	-15	150.0	26	20	177.0
16	2	32.0	26	27	76.0
16	15	150.0	26	5	281.0
16	19	228.0	27	26	76.0
16	-16	90.0	27	28	177.0
16	3	90.0	28	27	177.0
16	-16	0.0	28	20	76.0
16	13	302.0	28	25	82.0
17	18	196.0	28	21	171.0
17	-17	285.0	-1	1	92.1600036621
17	1	285.0	-1	2	90.8399963379
17	2	102.0	-2	2	40.3800048828
18	19	102.0	-2	17	142.380004883
18	17	196.0	-2	1	142.619995117
19	18	102.0	-3	17	177.930000305
19	-19	228.0	-3	1	107.069999695
19	3	318.0	-3	2	75.9300003052
19	-19	318.0	-4	2	60.1600036621
19	2	196.0	-4	1	122.839996338

Origen	Destino	Distancia	Origen	Destino	Distancia
-5	2	33.1100006104	-27	11	21.4100036621
-5	17	68.8899993896	-28	5	99.8399963379
-6	2	6.80000305176	-29	26	40.700012207
-6	17	95.1999969482	-29	8	59.299987793
-7	2	45.2299957275	-30	5	75.3200073242
-7	17	56.7700042725	-31	5	131.130004883
-8	4	0.689999997616	-32	4	214.369995117
-8	11	59.6899999976	-32	5	76.6300048828
-9	3	10.6100006104	-33	14	38.5599975586
-10	26	124.040008545	-33	3	111.440002441
-10	8	24.0400085449	-34	5	48.1300048828
-10	5	156.959991455	-35	5	72.1799926758
-11	14	50.2699890137	-36	8	216.160003662
-11	3	99.7300109863	-36	5	35.1600036621
-12	5	86.0100097656	-37	4	138.910003662
-12	8	94.9899902344	-37	11	79.9100036621
-13	14	108.529998779	-37	3	44.0899963379
-13	3	41.4700012207	-38	11	144.430000305
-14	8	24.8999938965	-38	4	85.4300003052
-14	26	75.1000061035	-39	11	108.419998169
-15	8	84.0	-39	14	165.580001831
-15	5	97.0	-39	3	15.5800018311
-16	8	10.3900146484	-39	4	167.419998169
-16	26	89.6099853516	-40	20	20.8099975586
-17	3	18.9499969482	-40	26	156.190002441
-18	3	131.510009766	-41	4	86.7799987793
-18	14	18.4899902344	-41	11	145.779998779
-19	20	5.28002929688	-42	4	108.839996338
-20	8	22.2399902344	-43	14	98.6999969482
-20	26	77.7600097656	-43	4	234.300003052
-21	5	66.4400024414	-43	11	175.300003052
-22	11	216.11000061	-43	3	51.3000030518
-22	4	157.11000061	-44	14	1.66000366211
-22	5	133.88999939	-44	3	148.339996338
-23	11	49.2299995422	-45	8	26.1300048828
-23	3	173.229999542	-46	11	96.4099998474
-23	4	9.77000045776	-46	4	37.4099998474
-24	8	49.1400146484	-47	20	20.7299804688
-24	26	50.8599853516	-48	11	10.3400001526
-25	11	82.0200042725	-48	4	48.6599998474
-25	3	41.9799957275	-49	8	94.4199829102
-26	8	14.1799926758	-49	26	5.58001708984
-27	4	80.4100036621	-50	20	167.789978027
-27	3	102.589996338	-50	8	109.210021973

Origen	Destino	Distancia	Origen	Destino	Distancia
-50	26	9.21002197266	-71	7	14.0399999619
-51	11	243.320007324	-72	6	107.559997559
-51	4	184.320007324	-73	6	110.550003052
-51	5	106.679992676	-74	6	130.619995117
-52	4	122.279998779	-75	6	7.92001342773
-53	4	76.4899978638	-76	6	14.549987793
-53	11	135.489997864	-77	6	127.080001831
-54	5	4.70001220703	-78	6	172.0
-55	11	15.0699996948	-78	7	119.0
-55	4	74.0699996948	-79	7	28.7199993134
-56	3	7.46000671387	-80	6	69.8200073242
-57	3	130.850006104	-81	6	135.009994507
-57	14	19.1499938965	-82	7	24.8400001526
-58	3	45.7400054932	-83	10	14.8999938965
-58	11	78.2599945068	-84	9	51.2399902344
-59	8	191.16998291	-85	10	35.2200012207
-59	26	91.1699829102	-85	9	145.779998779
-59	20	85.8300170898	-86	10	42.8299865723
-60	14	56.1799926758	-87	9	76.049987793
-60	3	93.8200073242	-88	10	86.2799987793
-61	11	164.470001221	-88	9	94.7200012207
-61	4	105.470001221	-89	15	29.2699890137
-61	5	185.529998779	-90	16	72.1499938965
-62	11	79.610006104	-90	15	77.8500061035
-62	4	20.610006104	-91	15	37.1300048828
-63	11	80.9199981689	-91	16	112.869995117
-63	3	43.0800018311	-92	16	45.6499938965
-63	4	139.919998169	-92	15	104.350006104
-63	14	193.080001831	-93	16	120.899993896
-64	8	101.280029297	-93	15	29.1000061035
-64	20	175.719970703	-94	21	7.95001220703
-64	26	1.28002929688	-94	23	199.049987793
-65	4	119.5	-95	21	81.950012207
-65	11	60.5	-95	23	125.049987793
-65	3	63.5	-96	22	14.6000061035
-65	14	213.5	-97	23	73.8400268555
-66	5	96.950012207	-97	21	133.159973145
-66	8	84.049987793	-98	21	227.609985352
-67	8	81.2600097656	-98	22	49.3900146484
-67	5	99.7399902344	-99	21	70.7100219727
-68	11	117.119995117	-100	22	66.0200195312
-68	3	6.88000488281	-101	21	115.83001709
-69	7	14.6000003815	-101	23	91.1699829102
-70	6	42.0	-102	21	69.1799926758

Origen	Destino	Distancia	Origen	Destino	Distancia
-102	23	137.820007324	-131	27	116.559997559
-103	23	80.299987793	-131	28	60.4400024414
-104	23	159.41998291	-132	27	91.7600097656
-104	21	47.5800170898	-132	28	85.2399902344
-105	21	82.549987793	-133	27	82.0700073242
-105	23	124.450012207	-133	28	94.9299926758
-106	22	40.9099731445	-134	4	115.579999924
-107	23	123.440002441	-134	7	179.579999924
-107	21	83.5599975586	-134	1	6.42000007629
-108	22	139.890014648	-135	7	94.9599990845
-108	21	137.109985352	-135	1	91.0400009155
-109	21	133.070007324	-135	4	30.9599990845
-109	22	143.929992676	-136	4	45.1300048828
-110	23	64.7700195312	-136	1	167.130004883
-111	22	51.8900146484	-136	7	18.8699951172
-112	21	36.6099853516	-137	7	140.659999847
-113	22	63.6199951172	-137	1	45.3400001526
-114	23	137.25	-137	4	76.6599998474
-114	21	69.75	-138	4	62.4100036621
-115	21	2.34002685547	-138	1	184.410003662
-115	23	204.659973145	-138	7	1.58999633789
-116	21	81.6599731445	-139	16	85.9000015259
-116	22	195.340026855	-139	3	4.09999847412
-117	23	47.7600097656	-140	3	202.079986572
-117	21	159.239990234	-140	13	9.92001342773
-118	21	135.909973145	-141	19	41.2799987793
-118	22	141.090026855	-142	2	44.4399986267
-119	21	115.530029297	-142	16	76.4399986267
-119	23	91.4699707031	-143	13	3.35000610352
-120	22	102.710021973	-144	19	38.9700012207
-120	21	174.289978027	-144	16	189.029998779
-121	25	54.4799804688	-144	2	157.029998779
-122	25	51.2700195312	-145	16	25.2699999809
-123	25	96.2600097656	-145	2	6.73000001907
-124	24	55.8800048828	-146	2	57.2000007629
-125	24	26.200012207	-146	19	138.799999237
-126	25	49.9400024414	-146	16	89.2000007629
-126	24	157.059997559	-147	19	19.8200073242
-127	27	34.5900268555	-148	3	42.9000015259
-128	27	108.570007324	-148	16	47.0999984741
-128	28	68.4299926758	-148	2	79.0999984741
-129	27	68.700012207	-149	2	69.6999969482
-130	28	12.6599731445	-149	3	52.3000030518
-130	27	164.340026855	-149	16	37.6999969482

Origen	Destino	Distancia	Origen	Destino	Distancia
-150	19	139.369998932	-164	2	179.720001221
-150	2	56.6300010681	-165	8	61.0399932861
-150	16	88.6300010681	-165	9	1.96000671387
-151	16	192.380004883	-166	22	122.059999943
-151	2	160.380004883	-166	8	124.940000057
-151	19	35.6199951172	-167	9	24.0700073242
-152	3	32.6600036621	-167	8	38.9299926758
-153	3	94.3899993896	-168	22	23.8799972534
-153	13	117.61000061	-169	22	209.809997559
-154	16	107.460006714	-169	9	100.190002441
-154	2	139.460006714	-169	8	37.1900024414
-154	13	194.539993286	-170	22	12.0400009155
-154	3	17.4600067139	-171	9	43.1000061035
-155	3	91.7600002289	-171	8	19.8999938965
-155	16	1.76000022888	-172	8	72.5999984741
-155	2	30.2399997711	-173	8	232.279998779
-156	3	129.820007324	-173	22	14.7200012207
-156	13	82.1799926758	-174	22	63.2099990845
-157	16	66.9800033569	-175	22	125.360000014
-157	3	23.0199966431	-175	8	121.639999986
-157	2	98.9800033569	-176	22	1.80999755859
-158	3	61.25	-177	12	31.8800048828
-158	16	151.25	-177	11	180.119995117
-158	2	183.25	-178	12	50.6000061035
-158	13	150.75	-179	11	10.8300018311
-159	16	15.8600006104	-179	12	201.169998169
-159	2	47.8600006104	-180	11	2.81999969482
-159	3	74.1399993896	-181	11	90.4900054932
-159	19	243.86000061	-182	15	4.93000030518
-160	16	9.22999954224	-182	14	85.0699996948
-160	2	22.7700004578	-183	15	13.0299987793
-160	3	99.2299995422	-183	14	76.9700012207
-161	16	132.339996338	-184	15	83.5899963379
-161	2	100.339996338	-184	14	6.41000366211
-161	19	95.6600036621	-185	17	171.899993896
-162	13	77.0700073242	-185	18	24.1000061035
-162	16	224.929992676	-186	17	112.080001831
-162	3	134.929992676	-186	18	83.9199981689
-163	3	47.3300018311	-187	18	5.74000549316
-163	16	42.6699981689	-187	17	190.259994507
-163	2	74.6699981689	-188	18	18.8000030518
-164	13	154.279998779	-188	17	177.199996948
-164	3	57.7200012207	-189	17	140.779998779
-164	16	147.720001221	-189	18	55.2200012207



Origen	Destino	Distancia
-190	18	44.1100006104
-191	18	9.4700012207
-192	28	41.0599975586
-192	20	34.9400024414
-193	21	61.0
-193	25	28.0
-194	21	194.959999084
-194	25	105.959999084
-194	28	23.9599990845
-194	20	52.0400009155
-195	28	36.2299995422
-195	21	134.770000458
-195	25	45.7700004578
-195	20	112.229999542
-196	20	108.590000153
-196	21	138.409999847
-196	25	49.4099998474
-196	28	32.5900001526
-197	21	89.4799995422
-197	28	81.5200004578
-197	25	0.479999542236
-198	20	132.239999771
-198	25	25.7600002289
-198	28	56.2399997711
-198	21	114.760000229
-199	26	10.2600021362
-199	27	65.7399978638
-200	27	41.4300003052
-200	26	34.5699996948

El resto de elementos de la batería tienen una similar estructura pero entre ellos varían la disponibilidad de arcos entre nodos.

## ANEXO II. BATERÍA 20-400

A continuación se muestran los datos de la batería de problemas en forma de tabla proporcionada por el tutor (González-R, 2015).

En esta tabla se contemplan los problemas con una densidad de 20 terminales por calle.

Donde cada columna de la tabla significa que:

Origen: es el comienzo del arco, cuando este valor sea positivo es un arco que comienza en un router. En caso contrario el origen será en un terminal.

Destino: Es el final del arco, cuando este valor es positivo, es un arco entre routers y cuando sea negativo es un arco hacia concentrador.

Distancia: es la longitud del arco.

Los datos para el 20-400-1 son:

Origen	Destino	Distancia	Origen	Destino	Distancia
1	-3	122	4	-2	183
1	-1	0	4	-1	122
1	2	183	4	1	122
1	4	122	4	3	183
1	7	186	4	5	291
1	17	285	4	7	64
2	-5	32	4	8	472
2	-2	122	4	11	59
2	-1	183	4	14	333
2	1	183	5	-4	350
2	3	122	5	-3	291
2	13	334	5	-2	474
2	16	32	5	3	474
2	17	102	5	4	291
2	19	196	5	6	64
3	-5	90	5	8	181
3	-4	124	5	10	63
3	-3	183	5	11	350
3	-2	0	5	20	458
3	2	122	5	26	281
3	4	183	6	5	64
3	5	474	6	7	291
3	11	124	6	10	1
3	13	212	7	-3	64
3	14	150	7	-1	186
3	16	90	7	1	186
3	19	318	7	4	64
4	-4	59	7	6	291

Origen	Destino	Distancia	Origen	Destino	Distancia
8	4	472	16	15	150
8	5	181	16	19	228
8	9	63	17	-1	285
8	20	277	17	1	285
8	22	247	17	2	102
8	26	100	17	18	196
9	8	63	18	17	196
9	10	181	18	19	102
9	22	310	19	-5	228
10	5	63	19	-2	318
10	6	1	19	2	196
10	9	181	19	3	318
11	-4	0	19	16	228
11	-3	59	19	18	102
11	-2	124	20	5	458
11	3	124	20	8	277
11	4	59	20	21	247
11	5	350	20	25	158
11	12	212	20	26	177
11	14	274	20	28	76
12	-4	212	21	20	247
12	11	212	21	22	277
12	13	124	21	23	207
13	-5	302	21	25	89
13	-2	212	21	28	171
13	2	334	22	8	247
13	3	212	22	9	310
13	12	124	22	21	277
13	16	302	22	23	484
14	-4	274	23	21	207
14	-3	333	23	22	484
14	-2	150	23	24	89
14	3	150	24	23	89
14	4	333	24	25	207
14	11	274	25	20	158
14	15	90	25	21	89
15	-5	150	25	24	207
15	14	90	25	28	82
15	16	150	26	5	281
16	-5	0	26	8	100
16	-2	90	26	20	177
16	2	32	26	27	76
16	3	90	27	26	76
16	13	302	27	28	177

Origen	Destino	Distancia	Origen	Destino	Distancia
28	20	76	-18	17	130.6799927
28	21	171	-19	1	67.26000214
28	25	82	-19	2	115.7399979
28	27	177	-19	17	217.7399979
-1	2	39.08999634	-20	1	76.54000092
-1	17	62.91000366	-20	2	106.4599991
-2	2	92.92001343	-20	17	208.4599991
-2	17	9.079986572	-21	1	109.6500015
-3	1	137.8699951	-21	2	73.34999847
-3	2	45.13000488	-21	17	175.3499985
-3	17	147.1300049	-22	2	15.88999939
-4	1	83.36000061	-22	17	86.11000061
-4	2	99.63999939	-23	2	47.27000427
-4	17	201.6399994	-23	17	54.72999573
-5	1	98.30999756	-24	1	82.86000061
-5	2	84.69000244	-24	2	100.1399994
-6	2	75.16000366	-24	17	202.1399994
-6	17	26.83999634	-25	2	12.32000732
-7	2	27.16000366	-26	3	147.3099976
-7	17	74.83999634	-26	14	2.690002441
-8	2	52.11000061	-27	3	58.33999634
-9	1	126.5100021	-27	4	241.3399963
-9	2	56.48999786	-27	11	182.3399963
-10	2	50.22000122	-27	14	91.66000366
-11	1	118.2200012	-28	5	79.55999756
-11	2	64.77999878	-28	8	101.4400024
-11	17	166.7799988	-29	5	151.9200134
-12	1	72.66999817	-29	8	29.07998657
-12	2	110.3300018	-29	26	129.0799866
-13	2	35.08000183	-30	8	81.21002197
-13	17	66.91999817	-30	26	18.78997803
-14	1	201.9499969	-31	4	213.7700043
-14	2	18.94999695	-31	5	77.22999573
-14	17	83.05000305	-32	8	104.9000244
-15	1	94.83999634	-32	20	172.0999756
-15	2	88.16000366	-32	26	4.900024414
-15	17	190.1600037	-33	8	175.7399902
-16	1	247.8500061	-33	20	101.2600098
-16	2	64.8500061	-33	26	75.73999023
-16	17	37.1499939	-34	4	10.61999989
-17	1	151.3800049	-34	11	48.38000011
-17	2	31.61999512	-35	5	54.82000732
-17	17	133.6199951	-35	8	126.1799927
-18	2	28.67999268	-36	3	126.0799866

Origen	Destino	Distancia	Origen	Destino	Distancia
-36	14	23.92001343	-53	20	85.4699707
-37	5	52.51998901	-53	26	91.5300293
-38	3	46.72000122	-54	5	21.45001221
-38	4	229.7200012	-55	5	163.1900024
-38	11	170.7200012	-55	8	17.80999756
-38	14	103.2799988	-55	26	117.8099976
-39	3	34.1499939	-56	8	10.75
-39	4	217.1499939	-56	26	89.25
-39	11	158.1499939	-57	3	7.949996948
-39	14	115.8500061	-58	8	93.94000244
-40	8	114.7199707	-58	26	6.059997559
-40	20	162.2800293	-59	8	103.1400146
-40	26	14.7199707	-59	26	3.140014648
-41	4	240.3200073	-60	5	89.27999878
-41	5	50.67999268	-60	8	91.72000122
-41	8	231.6799927	-61	8	69.92999268
-42	8	133.9899902	-61	26	30.07000732
-42	20	143.0100098	-62	4	14.52999973
-42	26	33.98999023	-62	11	73.52999973
-43	5	25.94000244	-63	4	66.55000305
-44	20	20.65997314	-63	11	7.550003052
-45	3	149.2200012	-64	3	10.94999695
-45	4	33.77999878	-64	4	172.0500031
-45	11	25.22000122	-64	11	113.0500031
-46	3	143.5299988	-64	14	160.9499969
-46	4	39.47000122	-65	5	35.20999146
-46	11	19.52999878	-65	8	145.7900085
-47	5	203.1600037	-66	5	159.2900085
-47	8	22.16000366	-66	8	21.70999146
-47	26	77.83999634	-66	26	121.7099915
-48	3	73.16000366	-67	5	30.20001221
-48	14	76.83999634	-68	8	145.8200073
-49	4	231.2599945	-68	20	131.1799927
-49	5	59.74000549	-68	26	45.82000732
-49	8	240.7400055	-69	3	54.94000244
-50	8	236.0100098	-69	11	69.05999756
-50	20	40.98999023	-70	5	128.2799988
-50	26	136.0100098	-70	8	52.72000122
-51	3	125.5499992	-71	5	6.559997559
-51	4	57.45000076	-71	8	187.5599976
-51	11	1.549999237	-72	5	77.05999756
-52	8	35.29000854	-73	3	72.19999695
-52	26	64.70999146	-73	4	110.8000031
-53	8	191.5300293	-73	11	51.80000305

Origen	Destino	Distancia	Origen	Destino	Distancia
-74	26	59.55999756	-94	26	44.78997803
-75	5	242.2399902	-95	8	111.6699829
-75	8	61.23999023	-95	20	165.3300171
-75	20	215.7600098	-95	26	11.66998291
-75	26	38.76000977	-96	3	69.33000183
-76	8	63.60998535	-96	11	54.66999817
-77	4	6.050000191	-97	5	39.13999939
-77	11	65.05000019	-98	4	178.2400055
-78	20	33.79998779	-98	5	112.7599945
-78	26	143.2000122	-98	11	237.2400055
-79	20	53.34002686	-99	3	13.63999939
-79	26	123.6599731	-99	4	196.6399994
-80	3	69.77999878	-99	11	137.6399994
-80	11	193.7799988	-99	14	136.3600006
-80	14	80.22000122	-100	5	73.01998901
-81	3	33.55999756	-101	5	117.9299927
-81	11	90.44000244	-101	8	63.07000732
-82	5	7.239990234	-102	8	195.6900024
-83	8	72.23999023	-102	20	81.30999756
-84	4	22.5	-102	26	95.69000244
-84	11	81.5	-103	5	27.17999268
-85	4	90.76999664	-104	8	29.75
-85	11	149.7699966	-104	26	70.25
-86	5	64.60998535	-105	4	2.210000038
-86	8	116.3900146	-105	11	56.78999996
-86	26	216.3900146	-106	8	159.2000122
-87	3	36.71000671	-106	20	117.7999878
-87	4	219.7100067	-106	26	59.20001221
-87	11	160.7100067	-107	5	220.1000061
-87	14	113.2899933	-107	8	39.1000061
-88	8	8.769989014	-107	20	237.8999939
-88	26	91.23001099	-107	26	60.8999939
-89	3	96.16000366	-108	3	110.5500031
-89	11	220.1600037	-108	4	72.44999695
-89	14	53.83999634	-108	11	13.44999695
-90	5	99.74000549	-109	3	88.5
-91	3	86.33999634	-109	14	61.5
-91	14	63.66000366	-110	14	29.82000732
-92	8	50.59997559	-111	4	173.2700043
-92	26	49.40002441	-111	5	117.7299957
-93	3	81.51000214	-111	11	232.2700043
-93	11	42.48999786	-112	5	142.5100098
-94	8	144.789978	-112	8	38.48999023
-94	20	132.210022	-112	26	138.4899902

Origen	Destino	Distancia	Origen	Destino	Distancia
-113	5	20.14001465	-137	8	138.8099976
-114	20	10.36999512	-137	20	138.1900024
-115	4	39.61999893	-137	26	38.80999756
-115	11	19.38000107	-138	7	77.30999756
-116	8	93.69000244	-139	7	17.12999916
-116	26	6.309997559	-140	6	135.1499939
-117	5	0.75	-141	6	1.440002441
-118	8	110.5100098	-142	6	217.5899963
-118	26	10.51000977	-142	7	73.41000366
-119	3	135.1300011	-143	7	40.09999847
-119	4	47.86999893	-144	6	0.619995117
-119	11	11.13000107	-145	7	56.52000046
-120	5	89.3999939	-146	6	115.1300049
-120	8	91.6000061	-146	7	175.8699951
-121	3	127.9599915	-147	7	35.29999924
-121	14	22.04000854	-148	7	44.74000168
-122	8	113.7000122	-149	7	36.68000031
-122	26	13.70001221	-150	6	6.350006104
-123	5	93.92001343	-151	6	97.32000732
-123	8	87.07998657	-151	7	193.6799927
-124	4	205.1600037	-152	6	99.07000732
-124	5	85.83999634	-152	7	191.9299927
-125	4	99.76999664	-153	7	8.710000038
-126	20	68.96002197	-154	6	166.7699966
-127	5	119.1099854	-154	7	124.2300034
-127	8	61.89001465	-155	6	241.1300011
-128	8	4.079986572	-155	7	49.86999893
-128	26	95.92001343	-156	6	54.47999573
-129	3	99.93000031	-157	7	121.0400009
-129	4	83.06999969	-158	7	17.88999939
-129	11	24.06999969	-159	6	34.88000488
-129	14	249.9300003	-160	6	60.3999939
-130	8	21.55999756	-161	7	65.66000366
-130	26	78.44000244	-162	6	47.00999451
-131	20	11.23999023	-163	6	202.2099991
-132	3	77.76998901	-163	7	88.79000092
-132	14	72.23001099	-164	9	33.35998535
-133	3	33.58999634	-165	9	76.82000732
-133	11	90.41000366	-166	10	55.41000366
-134	20	12.64001465	-167	9	46.32000732
-135	3	96.93000031	-167	10	134.6799927
-135	4	86.06999969	-168	10	25.67001343
-135	11	27.06999969	-169	9	42.07000732
-136	3	30.33000183	-170	9	107.7099915

Origen	Destino	Distancia	Origen	Destino	Distancia
-170	10	73.29000854	-200	21	44.78997803
-171	9	26.57998657	-201	21	14.30999756
-172	10	77.33999634	-202	23	3.429992676
-173	15	68.67999268	-203	21	46.54998779
-173	16	81.32000732	-204	23	12.17999268
-174	16	31.33999634	-205	21	144.5200195
-175	15	43.29998779	-205	22	132.4799805
-175	16	106.7000122	-206	21	62.76000977
-176	15	49.23001099	-207	21	95.67999268
-177	15	45.3999939	-208	21	45.46002197
-178	15	18.10998535	-209	22	86.26000977
-178	16	131.8900146	-210	21	76.58001709
-179	16	43.97000122	-211	22	68.51000977
-180	15	75.11999512	-212	21	125.5999756
-180	16	74.88000488	-212	23	81.40002441
-181	18	97.75	-213	21	193.2800293
-181	19	4.25	-213	23	13.7199707
-182	18	5.130004883	-214	21	4.25
-182	19	96.86999512	-214	23	211.25
-183	18	77.91999817	-215	21	25.76000977
-183	19	24.08000183	-216	21	57.59997559
-184	18	61.3500061	-216	23	149.4000244
-184	19	40.6499939	-217	21	12.15002441
-185	18	8.25	-217	23	219.1500244
-185	19	93.75	-218	21	43.33001709
-186	21	162.75	-218	22	233.6699829
-186	22	114.25	-219	21	68.19000244
-187	21	220.7700195	-219	23	138.8099976
-187	22	56.22998047	-220	21	134.5200195
-188	21	138.4500122	-220	23	72.47998047
-188	22	138.5499878	-221	22	10.32998657
-189	21	115.2999878	-222	21	98.10998535
-190	21	73.78997803	-222	22	178.8900146
-191	21	174.5999756	-223	21	9.020019531
-191	22	102.4000244	-224	22	119.9899902
-192	22	0.359985352	-225	21	33.84002686
-193	21	116.7000122	-226	21	179.0300293
-194	22	16.26998901	-226	23	27.9699707
-195	21	116.7800293	-227	21	143.6099854
-195	23	90.2199707	-227	23	63.39001465
-196	23	52.15002441	-228	21	199.8099976
-197	21	31.86999512	-228	22	77.19000244
-198	21	79.21002197	-229	22	31.98001099
-199	21	99.42999268	-230	21	137.710022



Origen	Destino	Distancia	Origen	Destino	Distancia
-230	23	69.28997803	-259	1	55.20999908
-231	21	124.3200073	-259	4	66.79000092
-232	23	24.59997559	-259	7	130.79000009
-233	21	95.57000732	-260	1	164.7799988
-233	23	111.4299927	-260	4	42.77999878
-234	21	11.16998291	-260	7	21.22000122
-235	22	72.30999756	-261	4	19.8500061
-236	21	181.7000122	-261	7	44.1499939
-236	23	25.29998779	-262	1	119.6600037
-237	25	69.79998779	-262	4	2.339996338
-238	24	160.0900269	-262	7	66.33999634
-238	25	46.90997314	-263	1	42.54999924
-239	24	95.16998291	-263	4	79.45000076
-240	24	58.36999512	-263	7	143.4500008
-240	25	148.6300049	-264	4	8.940002441
-241	25	44.59997559	-264	7	55.05999756
-242	24	206.2000122	-265	1	44.33000183
-242	25	0.799987793	-265	4	77.66999817
-243	25	14.47998047	-266	1	90.33999634
-244	24	17.33001709	-266	4	31.66000366
-244	25	189.6699829	-266	7	95.66000366
-245	24	194.4199829	-267	1	15.67000008
-245	25	12.58001709	-267	4	106.3299999
-246	24	95.78997803	-268	1	29.12000084
-246	25	111.210022	-268	4	92.87999916
-247	24	125.25	-269	1	150.5800018
-247	25	81.75	-269	4	28.58000183
-248	25	41.14001465	-269	7	35.41999817
-249	24	97.41998291	-270	1	118.0199966
-249	25	109.5800171	-270	4	3.980003357
-250	24	107.0300293	-270	7	67.98000336
-250	25	99.9699707	-271	1	38.52000046
-251	27	30.30999756	-271	4	83.47999954
-251	28	146.6900024	-271	7	147.4799995
-252	27	39.25	-272	3	6.650001526
-253	27	49.11999512	-272	16	83.34999847
-254	27	88.01000977	-273	2	238.5800018
-254	28	88.98999023	-273	3	116.5800018
-255	27	9.770019531	-273	13	95.41999817
-255	28	167.2299805	-273	16	206.5800018
-256	28	63.46002197	-274	2	33.70999908
-257	28	6.820007324	-274	3	88.29000092
-258	27	41.84002686	-274	16	1.709999084
-258	28	135.1599731	-275	13	31.25

Origen	Destino	Distancia	Origen	Destino	Distancia
-276	2	15.60999966	-291	16	229.1700134
-276	3	137.6099997	-292	2	41.00999832
-276	16	47.60999966	-292	3	80.99000168
-277	3	7.070007324	-292	16	9.009998322
-277	16	97.07000732	-292	19	237.0099983
-278	2	76.08999634	-293	2	64.44000244
-278	3	198.0899963	-293	3	57.55999756
-278	16	108.0899963	-293	16	32.44000244
-278	19	119.9100037	-294	2	83.91000366
-279	2	137.7899933	-294	16	115.9100037
-279	3	15.78999329	-294	19	112.0899963
-279	13	196.2100067	-295	2	232.3300018
-279	16	105.7899933	-295	3	110.3300018
-280	3	84.28999329	-295	13	101.6699982
-280	13	127.7100067	-295	16	200.3300018
-281	2	86.70999908	-296	13	17.76998901
-281	3	35.29000092	-297	2	7.420000076
-281	13	247.2900009	-297	3	129.4200001
-281	16	54.70999908	-297	16	39.42000008
-282	2	6.940000057	-297	19	188.5799999
-282	16	38.94000006	-298	2	127.0800018
-283	2	131.0800018	-298	3	249.0800018
-283	3	9.080001831	-298	16	159.0800018
-283	16	99.08000183	-298	19	68.91999817
-284	2	96.76999664	-299	2	30.48999977
-284	3	218.7699966	-299	3	91.51000023
-284	16	128.7699966	-299	16	1.510000229
-284	19	99.23000336	-300	13	68.76998901
-285	13	40.29998779	-301	2	161.1900024
-286	2	106.6600037	-301	16	193.1900024
-286	3	228.6600037	-301	19	34.80999756
-286	16	138.6600037	-302	2	140.3500061
-286	19	89.33999634	-302	16	172.3500061
-287	2	65.19000244	-302	19	55.6499939
-287	16	97.19000244	-303	2	162.1600037
-288	3	195.4200134	-303	16	194.1600037
-288	13	16.57998657	-303	19	33.83999634
-289	2	127.8000031	-304	2	70.22000122
-289	19	68.19999695	-304	3	51.77999878
-290	2	193.0399933	-304	16	38.22000122
-290	16	225.0399933	-305	13	18.66000366
-290	19	2.960006714	-306	2	58.45000076
-291	3	139.1700134	-306	3	63.54999924
-291	13	72.82998657	-306	16	26.45000076

Origen	Destino	Distancia	Origen	Destino	Distancia
-307	2	163.4400024	-326	8	92.95000076
-307	16	195.4400024	-326	9	155.9500008
-307	19	32.55999756	-326	22	154.0499992
-308	2	86.62000275	-327	8	12.88999939
-308	3	208.6200027	-327	9	50.11000061
-308	16	118.6200027	-328	8	66.36999893
-308	19	109.3799973	-328	9	129.3699989
-309	2	14.31000042	-328	22	180.6300011
-309	3	107.6899996	-329	8	128.04
-309	16	17.68999958	-329	9	191.04
-310	2	121.5199966	-329	22	118.96
-310	3	0.480003357	-330	8	25.74000549
-310	16	89.51999664	-330	9	37.25999451
-311	2	208.8600006	-331	22	59.91999817
-311	3	86.86000061	-332	8	189.6900024
-311	13	125.1399994	-332	22	57.30999756
-311	16	176.8600006	-333	22	9.279998779
-312	2	23.10000038	-334	22	8.209999084
-312	16	55.10000038	-335	22	67.88000107
-313	3	55.96000671	-336	8	204.8000031
-314	3	109.0399933	-336	22	42.19999695
-314	13	102.9600067	-337	22	90.27999878
-315	2	76.88999939	-338	8	35.65000153
-315	3	45.11000061	-338	9	98.65000153
-315	16	44.88999939	-339	8	157.2000008
-316	2	87.04000092	-339	9	220.2000008
-317	2	71.56999969	-339	22	89.79999924
-318	2	39.20000076	-340	8	66.79000092
-318	3	82.79999924	-340	9	129.7900009
-318	16	7.200000763	-340	22	180.2099991
-318	19	235.2000008	-341	8	244.7300034
-319	8	42.94999695	-341	22	2.269996643
-319	9	105.9499969	-342	8	91.13999939
-320	22	58.34999847	-343	8	59.00999451
-321	8	145.2600002	-343	9	3.990005493
-321	9	208.2600002	-344	8	20.88000488
-321	22	101.7399998	-344	9	42.11999512
-322	8	18.66000366	-345	12	35.32998657
-322	9	44.33999634	-346	11	138.8599854
-323	8	80.31999969	-346	12	73.14001465
-324	8	55.44999695	-347	11	69.61000061
-325	8	144.8299999	-347	12	142.3899994
-325	9	207.8299999	-348	11	37.94999695
-325	22	102.1700001	-349	11	3.629997253

Origen	Destino	Distancia	Origen	Destino	Distancia
-350	12	76.94000244	-380	25	110.8799973
-351	11	67.3500061	-380	28	28.87999725
-352	11	68.28999329	-381	25	23.48999977
-352	12	143.7100067	-381	28	58.51000023
-353	11	84.36999512	-382	21	13.66999817
-353	12	127.6300049	-382	25	75.33000183
-354	11	58.75	-383	20	151.2099991
-354	12	153.25	-383	21	95.79000092
-355	11	37.30000305	-383	25	6.790000916
-356	11	62.55000305	-383	28	75.20999908
-357	11	67.75	-384	21	53.33999634
-357	12	144.25	-384	25	35.66000366
-358	12	17.19000244	-385	20	52.5
-359	11	52.33000183	-385	21	194.5
-359	12	159.6699982	-385	25	105.5
-360	18	86.55000305	-385	28	23.5
-361	18	24.63000488	-386	21	82.50999832
-362	17	18.39999962	-386	25	6.490001678
-363	17	24.67000008	-386	28	88.49000168
-364	18	20.66000366	-387	20	105.6599998
-365	18	31.38000488	-387	21	141.3400002
-366	17	81.81999969	-387	25	52.34000015
-367	17	117.2900009	-387	28	29.65999985
-367	18	78.70999908	-388	20	196.9599991
-368	17	72.44000244	-388	21	50.04000092
-368	18	123.5599976	-388	25	38.95999908
-369	17	45.75999832	-388	28	120.9599991
-370	17	138.7799988	-389	20	145.6800003
-370	18	57.22000122	-389	21	101.3199997
-371	18	33.88999939	-389	25	12.31999969
-372	17	125.4100037	-389	28	69.68000031
-372	18	70.58999634	-390	21	90.02999878
-373	17	139.4100037	-390	25	1.029998779
-373	18	56.58999634	-390	28	80.97000122
-374	17	121.9100037	-391	20	69.33000183
-374	18	74.08999634	-391	25	88.66999817
-375	17	91.75	-391	28	6.669998169
-376	17	70.84999847	-392	20	219.1500015
-376	18	125.1500015	-392	21	27.84999847
-377	18	13.80000305	-392	25	61.15000153
-378	17	159.8200073	-392	28	143.1500015
-378	18	36.17999268	-393	20	82.13000107
-379	17	34.83000183	-393	21	164.8699989
-380	20	47.12000275	-393	25	75.86999893

Origen	Destino	Distancia
-393	28	6.130001068
-394	25	32.13000011
-394	28	49.86999989
-395	20	204.3300018
-395	21	42.66999817
-395	25	46.33000183
-395	28	128.3300018
-396	21	71.70000076
-396	25	17.29999924
-396	28	99.29999924
-397	20	23.94000244
-397	21	223.0599976
-397	25	134.0599976
-397	28	52.05999756
-398	21	13.69999695
-398	25	75.30000305
-399	21	68.25999832
-399	25	20.74000168
-400	21	19.48999786
-400	25	69.51000214

El resto de elementos de la batería tienen una similar estructura pero entre ellos varían la disponibilidad de arcos entre nodos.