



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Organización Industrial**

**Obtención del mínimo árbol expandido  
en el problema de Steiner Euclídeo**

**Autor:**

**Díez Parra, Gonzalo**

**Tutora:**

**Mata Crespo, Raquel**

**Departamento:**

**Estadística e Investigación**

**Operativa**

**Valladolid, julio 2023.**



## **Agradecimientos**

Quisiera agradecer en primer lugar a Raquel Mata, mi tutora, ya que, sin su ayuda y asesoría continua, este trabajo no habría sido posible. Raquel me ha servido de guía durante estos meses, siempre disponible y amable.

También quiero agradecer a mi familia y amigos, los cuales siempre han estado presentes y me han prestado apoyo y consejos durante todo el proceso de elaboración del trabajo.



## Resumen

Se analizará el problema de Steiner Euclídeo con el objetivo de buscar el mínimo árbol expandido que conecte un conjunto finito de puntos del plano euclídeo. Se representarán gráficamente los árboles de Steiner en el contexto de grafos no dirigidos y con pesos ponderados en sus aristas.

Se estudiará la modelización del problema de Steiner como un diseño de redes con costes fijos y, para ejemplificar la resolución manual del problema de Steiner, se expondrán algunos ejemplos relativamente sencillos para su implementación con intención de poder aplicarse en problemas de mayor complejidad que requieran cierta dificultad y rapidez computacional.

Se comentará también la versión con distancias rectilíneas donde las conexiones estarán restringidas, sólo podrán ser horizontales o verticales.

**Palabras clave:** Optimización Combinatorial, Programación Entera, Teoría de Grafos.



## **Abstract**

The Euclidean Steiner problem will be analyzed in order to find the minimum expanded tree that connects a finite set of points in the Euclidean plane. Steiner trees will be represented graphically in the context of undirected graphs and with weighted weights on their edges.

The modeling of the Steiner problem will be studied as a design of networks with fixed costs and, to exemplify the manual resolution of the Steiner problem, some relatively simple examples will be presented for its implementation with the intention of being able to apply it to more complex problems that require some difficulty. and computational speed.

The version with rectilinear distances will also be commented where the connections will be restricted, they can only be horizontal or vertical.

**Keywords:** Combinatorial Optimization, Integer Programming, Graph Theory.





# ÍNDICES



## Tabla de contenido

<b>ÍNDICES</b> .....	9
Índice de Figuras .....	13
Índice de Tablas.....	17
Índice de Ilustraciones .....	19
<b>INTRODUCCIÓN Y OBJETIVOS</b> .....	21
Contextualización del tema. ....	23
Motivación y objetivos del trabajo.....	23
<b>DESARROLLO</b> .....	25
1. ELEMENTOS BÁSICOS DE UN GRAFO.....	27
2. ELEMENTOS BÁSICOS DE UN ÁRBOL.....	29
3. EL PROBLEMA DE STEINER.....	30
3.1. Algoritmos de Kruskal y Prim para obtener el MST de un grafo. ....	31
3.1.1. Algoritmo de Kruskal.....	31
3.1.2. Algoritmo de Prim.....	33
4. FORMULACIÓN DEL PROBLEMA DE STEINER.....	39
4.1. Modelización del problema de Steiner. ....	40
4.2. Formato del archivo que usa el software como input.....	42
4.3. Comandos especiales para la creación de gráficos en XPRESS MOSEL	43
5. ALGUNOS EJEMPLOS RESUELTOS DE GRAFOS.....	44
5.1. Grafo ponderado con trece nodos y veinticuatro aristas. ....	44
5.2. Grafo ponderado con dieciocho nodos y treinta y seis aristas. ....	49
5.3. Grafo ponderado con quince nodos y veinticinco aristas.....	56
6. EL PROBLEMA DE STEINER EUCLÍDEO .....	62
6.1. El problema de Steiner con distancias euclídeas. ....	62
6.1.1. Escenario uno: Las líneas que conectan los nodos terminales sólo pueden tocar los propios nodos del conjunto P. ....	63
6.1.2. Escenario dos: Las líneas se pueden cortar con nodos que no están en el conjunto P.....	75
6.2. El problema de Steiner Euclídeo con distancias Manhattan. ....	87
6.2.1. Escenario 1: Sin nodos de Steiner.....	87
6.2.2. Escenario 2: Con nodos de Steiner.....	90

7. APLICACIONES DE LA OBTENCIÓN DEL ÁRBOL DE EXPANSIÓN MÍNIMO EUCLÍDEO.....	93
7.1. Aplicación en un caso real.....	94
7.1.1. Presentación del problema .....	94
7.1.2. Resolución.....	97
7.1.3. Limitaciones del problema.....	99
<b>CONCLUSIONES.....</b>	<b>101</b>
<b>BIBLIOGRAFÍA.....</b>	<b>105</b>

## Índice de Figuras

Figura 1: Esquema de grafo con dos aristas paralelas, un lazo y un vértice aislado.....	27
Figura 2: Grafo simple ponderado con cinco nodos y seis aristas.....	28
Figura 3: Subgrafo derivado del grafo simple ponderado con cinco nodos y seis aristas. ....	29
Figura 4: Árbol derivado del grafo simple ponderado con cinco nodos y seis aristas. ....	30
Figura 5: Grafo ponderado con cinco nodos y diez aristas para la aplicación del algoritmo de Kruskal.....	32
Figura 6: Solución obtenida con el algoritmo de Kruskal para el ejemplo de grafo ponderado con cinco nodos y diez aristas. ....	33
Figura 7: Ejemplo de corte en grafos.....	34
Figura 8: Ejemplo de grafo ponderado con cinco nodos y diez aristas para la aplicación del algoritmo de Prim.....	36
Figura 9: Cortes en el ejemplo de grafo ponderado con cinco nodos y diez aristas en la aplicación del algoritmo de Prim.....	37
Figura 10: Solución para el ejemplo de grafo ponderado con cinco nodos y diez aristas obtenida con el algoritmo de Prim. ....	38
Figura 11: Grafo de veinticinco nodos y cuarenta y ocho aristas para la aplicación del algoritmo de Prim.....	38
Figura 12: Árbol mínimo de expansión obtenido del grafo de veinticinco nodos y cuarenta y ocho aristas.....	39
Figura 13: Grafo ponderado con trece nodos y veinticuatro aristas.....	44
Figura 14: Solución manual propuesta para el grafo ponderado con trece nodos y veinticuatro aristas.....	45
Figura 15: Solución óptima del grafo ponderado con trece nodos y veinticuatro aristas obtenida con el programa Xpress Mosel.....	46
Figura 16: Ejemplo de grafo ponderado con dieciocho nodos y treinta y seis aristas. ....	50
Figura 17: Solución manual propuesta al grafo ponderado con dieciocho nodos y treinta y seis aristas. ....	51

Figura 18: Solución óptima del grafo ponderado con dieciocho nodos y treinta y seis aristas, obtenida con el programa Xpress Mosel. ....	51
Figura 19: Ejemplo de grafo ponderado con quince nodos y veinticinco aristas. ....	56
Figura 20: Solución manual propuesta para el grafo ponderado con quince nodos y veinticinco aristas. ....	57
Figura 21: Solución óptima para el grafo ponderado con quince nodos y veinticinco aristas obtenida con el programa Xpress Mosel. ....	58
Figura 22: Enunciado del ejemplo 0 en el apartado 6.1.1. con cuatro puntos en el espacio euclídeo. ....	63
Figura 23: Solución manual para el ejemplo 0 en el apartado 6.1.1. con cuatro puntos en el espacio euclídeo.....	65
Figura 24: Solución del software para el ejemplo 0 en el apartado 6.1.1. con cuatro puntos en el espacio euclídeo.....	66
Figura 25: Enunciado del ejemplo 1 en el apartado 6.1.1. con diez puntos en el espacio euclídeo. ....	68
Figura 26: Solución manual para el ejemplo 1 del apartado 6.1.1. con diez puntos en el espacio euclídeo.....	68
Figura 27: Solución del software para el ejemplo 1 del apartado 6.1.1. con diez puntos en el espacio euclídeo.....	69
Figura 28: Enunciado del ejemplo 2 en el apartado 6.1.1. con veinticuatro puntos en el espacio euclídeo.....	72
Figura 29: Solución manual para el ejemplo 2 del apartado 6.1.1. con veinticuatro puntos en el espacio euclídeo.....	73
Figura 30: Solución del software para el ejemplo 2 del apartado 6.1.1. con veinticuatro puntos en el espacio euclídeo.....	74
Figura 31: Árbol de Steiner derivado del ejemplo 0 con cuatro nodos en el espacio euclídeo del apartado 6.1.1.....	76
Figura 32: Enunciado del ejemplo 1 en el apartado 6.1.2. con diez puntos en el espacio euclídeo. ....	78
Figura 33: Solución manual para la obtención del árbol de expansión mínimo euclídeo (gráfico superior) y el árbol de Steiner (gráfico inferior) para el enunciado del ejemplo 1 del apartado 6.1.2.....	79

Figura 34: Solución del software para la obtención del árbol de expansión mínimo euclídeo en el ejemplo 1 del apartado 6.1.2. con diez puntos en el espacio euclídeo. ....	80
Figura 35: Enunciado del ejemplo 2 en el apartado 6.1.2. con quince puntos en el espacio euclídeo. ....	83
Figura 36: Solución manual para la obtención del árbol de expansión mínimo euclídeo en el ejemplo 2 del apartado 6.1.2. ....	84
Figura 37: Solución manual para la obtención del árbol de Steiner en el ejemplo 2 del apartado 6.1.2. ....	84
Figura 38: Solución del software para la obtención del árbol de expansión mínimo euclídeo en el ejemplo 2 del apartado 6.1.2. con quince puntos en el espacio euclídeo. ....	86
Figura 39: Enunciado y solución manual propuesta con distancias rectilíneas a un problema de Steiner de cinco nodos. ....	88
Figura 40: Enunciado y solución manual propuesta con distancias rectilíneas a un problema de Steiner de nueve nodos. ....	88
Figura 41: Conjunto de sesenta puntos para la obtención del árbol mínimo expandido. ....	89
Figura 42: Árbol mínimo expandido para un conjunto de sesenta nodos. ....	90
Figura 43: Posible solución del problema de Steiner de cinco nodos con un árbol mínimo rectilíneo de Steiner (usando nodos de Steiner). ....	91
Figura 44: Posible solución del problema de Steiner de nueve nodos con un árbol mínimo rectilíneo de Steiner (usando nodos de Steiner). ....	92
Figura 45: Posible solución del problema de Steiner de sesenta nodos con un árbol mínimo rectilíneo de Steiner (usando nodos de Steiner). ....	93
Figura 46: Solución obtenida con software con el árbol de expansión mínimo euclídeo para el caso real del apartado 7.1. ....	98





## Índice de Tablas

Tabla 1: Arcos y sus correspondientes costos fijos para el grafo ponderado con trece nodos y veinticuatro aristas.....	45
Tabla 2: Arcos y sus correspondientes costos fijos para el grafo ponderado con dieciocho nodos y treinta y seis aristas.....	50
Tabla 3: Arcos y sus correspondientes costos fijos para el grafo ponderado con quince nodos y veinticinco aristas.....	57
Tabla 4: Coordenadas de los nodos del ejemplo 0 del apartado 6.1.1.....	64
Tabla 5: Distancia entre nodos del ejemplo 0 del apartado 6.1.1.....	64
Tabla 6: Tramos de la solución obtenida con software para el ejemplo 0 del apartado 6.1.1. ....	65
Tabla 7: Coordenadas de los nodos del ejemplo 1 del apartado 6.1.1.....	67
Tabla 8: Distancia entre nodos del ejemplo 1 del apartado 6.1.1.....	67
Tabla 9: Tramos de la solución obtenida con software para el ejemplo 1 del apartado 6.1.1. ....	69
Tabla 10: Coordenadas de los nodos del ejemplo 2 del apartado 6.1.1.....	70
Tabla 11: Distancia entre nodos del ejemplo 2 del apartado 6.1.1. ....	71
Tabla 12: Tramos de la solución obtenida con software para el ejemplo 2 del apartado 6.1.1. ....	74
Tabla 13: Coordenadas de los nodos del ejemplo 1 del apartado 6.1.2.....	77
Tabla 14: Distancia entre nodos del ejemplo 1 del apartado 6.1.2. ....	77
Tabla 15: Tramos de la solución obtenida con software para el ejemplo 1 del apartado 6.1.2. ....	80
Tabla 16: Coordenadas de los nodos del ejemplo 2 del apartado 6.1.2.....	81
Tabla 17: Distancia entre nodos del ejemplo 2 del apartado 6.1.2. ....	82
Tabla 18: Tramos de la solución obtenida con software para el ejemplo 2 del apartado 6.1.2. ....	85
Tabla 19: Datos de las treinta y seis ubicaciones en la ciudad de Valladolid.	95
Tabla 20: Tramos de la solución del caso real del apartado 7.1. ....	97



## Índice de Ilustraciones

Ilustración 1: Un árbol mínimo euclídeo de Steiner (SMT) para un conjunto de puntos que representan 532 ciudades en los Estados Unidos. Fuente: (Zachariasen, 2009). .....	87
Ilustración 2: Mapa de las ubicaciones utilizadas para el caso real en la ciudad de Valladolid (partes superior e inferior de la ciudad). Fuente: Google Maps.	96



# INTRODUCCIÓN Y OBJETIVOS



### Contextualización del tema.

“En el siglo XIX se empezó a formular el problema de Steiner (antes llamado problema de Fermat) fruto de la curiosidad de algunos matemáticos, como Fermat, que querían obtener: puntos estratégicos en el espacio que permitiesen minimizar la distancia a otros puntos dados“ (Martínez Lacañina, 2018).

Cuando llegó la revolución industrial el problema de Steiner cobró gran importancia, ya que todo lo referente a la optimización de caminos como pueden ser redes ferroviarias, eléctricas o de otras infraestructuras se relacionaban con aplicaciones del problema de Steiner. Fue entonces cuando Steiner, que estaba inmerso en el problema, hizo grandes avances y fue por eso que se le dio su nombre al mismo.

Dependiendo de las aplicaciones en diferentes industrias, el problema del árbol de Steiner se puede extender a otros espacios métricos. Entre los más populares se pueden encontrar:

- El árbol euclídeo de Steiner (el árbol de Steiner en el plano euclídeo).
- El árbol rectilíneo de Steiner (el árbol de Steiner en el plano rectilíneo).
- El árbol de Steiner en grafos (el árbol de Steiner en una red no dirigida).

Éstos, según Du (2001), son reconocidos como los más importantes y han recibido mucha atención. Se llaman los problemas clásicos del árbol de Steiner.

También según Du (2001), en los años setenta y ochenta se demostró que los problemas clásicos del árbol de Steiner son NP-Completo. Esto significa que es improbable que se obtengan soluciones óptimas eficientes en tiempo polinomial, de ahí que desde entonces se haya investigado para obtener soluciones aceptables para la obtención del árbol mínimo de Steiner.

### Motivación y objetivos del trabajo.

El presente trabajo fin de grado, a partir de ahora TFG, consiste en el planteamiento y resolución del problema del árbol de Steiner. Se va a tratar desde la definición de los elementos básicos de un grafo hasta la obtención de la solución óptima de un problema complejo del árbol de Steiner utilizando el software Xpress Mosel.

A lo largo del Grado he cursado diversas asignaturas que van desde la química hasta contabilidad, no obstante, en asignaturas como Métodos cuantitativos I y II, he podido ver una gran utilidad a la hora de manejar diferentes herramientas para resolver problemas y optimizar los recursos disponibles. Estas asignaturas desglosan algunos problemas de partida, definiendo adecuadamente las variables de decisión, una vez formulado el modelo, se utilizan algunas herramientas para obtener un resultado óptimo. Uno de esos

solvers es el programa Xpress Mosel, el cual se utiliza para resolver problemas de optimización y será el escogido para resolver los problemas en este trabajo. Resulta interesante ver cómo, una vez planteado un problema, se pueden formular y escribir las instrucciones en un código asequible. Esto permitirá resolver el problema planteado y sus posibles variantes en poco tiempo con la ayuda de un ordenador.

A las asignaturas mencionadas anteriormente hay que sumarle una asignatura optativa que he cursado, llamada Métodos Matemáticos, la cual está relacionada con la temática de este TFG. En ella se exponen los aspectos matemáticos fundamentales de la optimización y se sientan las bases de las técnicas de resolución de problemas de optimización tales como los métodos metaheurísticos de optimización o métodos basados en la teoría de grafos. Realmente, la parte que más me cautivó fue la teoría de grafos ya que permite resolver multitud de problemas de optimización con un planteamiento matricial muy directo y visual.

Por otro lado, haciendo una revisión de la bibliografía sobre el tema de mi TFG, he podido descubrir que el problema del árbol de Steiner tiene muchas aplicaciones en el diseño de circuitos o redes, algo que está a la orden del día en la optimización de tiempos u otros recursos valiosos.

Por tanto, en este TFG, con el problema del árbol de Steiner mínimo, se aúna algo de todas estas asignaturas. Hay que añadir también que la mayoría de la investigación existente trata este tema con un número de nodos más bien reducido. En este trabajo se va a ampliar esas investigaciones y se va a trabajar con un número mayor de nodos, lo que puede ser una aportación en un ámbito en el que no hay tanta proliferación científica.

Para realizar el trabajo primero se va a exponer brevemente el marco teórico del problema, definiendo los conceptos necesarios para la comprensión del desarrollo posterior. A continuación, se van a presentar la formulación del problema y se van a exponer ejemplos de grafos sencillos con los que se va a buscar el árbol de Steiner mínimo junto con su solución. Después, se introducirá el problema de Steiner en el plano euclídeo y se presentarán algunas variantes de éste. Por último, en la parte final del trabajo, se tratarán las posibles aplicaciones del problema de Steiner y se resolverá un ejemplo con datos reales.



# DESARROLLO



## 1. ELEMENTOS BÁSICOS DE UN GRAFO

Los grafos que se van a tratar son no dirigidos y se van a denominar como  $G = (N, A)$ . Cada grafo estará formado por un conjunto no vacío de  $N = \{1, \dots, n\}$  **nodos**, también llamados **vértices**, unidos por un conjunto de  $A = \{1, \dots, a\}$  **aristas**, también llamados **arcos**, donde cada arco corresponde al par  $(i, j)$  de nodos  $i, j \in N$ . Asociado a cada arco  $(i, j) \in A$  hay una variable de flujo con valores  $0 \leq x_{ij} \leq u_{ij}$ . Cada arista puede unirse con un par no ordenado de vértices. Si una arista  $a \in A$  está unida a dos nodos  $(u, v)$ , se escribe  $a = (u, v)$  o  $a = (v, u)$  (Johnsonbaugh, 2005). Las aristas van a representar una relación simétrica entre los vértices y no van a tener un sentido definido.

Puede haber diferentes tipos de aristas. En la *Figura 1* se representa un grafo de cuatro nodos  $(n_1, \dots, n_4)$  y cuatro aristas  $(a_1, \dots, a_4)$ . Las aristas  $a_1$  y  $a_2$ , ambas conectan el nodo  $n_1$  con el nodo  $n_2$  o viceversa; a estas aristas se les denomina **aristas paralelas**. También puede haber una arista que tenga como origen y destino el mismo vértice como  $a_3$ , a este tipo de aristas se les llama **lazos**. Un grafo sin lazos ni aristas paralelas se llama grafo simple (Johnsonbaugh, 2005).

El **grado** de un nodo viene dado por la cantidad de aristas que inciden en él. También pueden existir nodos sin aristas incidentes como  $n_4$  en la *Figura 1*, a estos nodos se les llaman **nodos aislados** y tienen grado 0. Por otro lado, el nodo  $n_4$ , al tener una arista incidente, tiene grado 1.

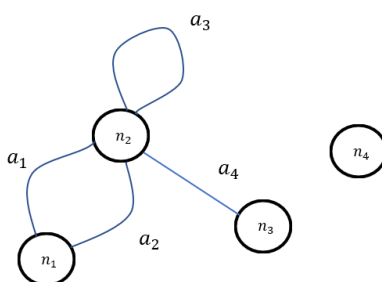
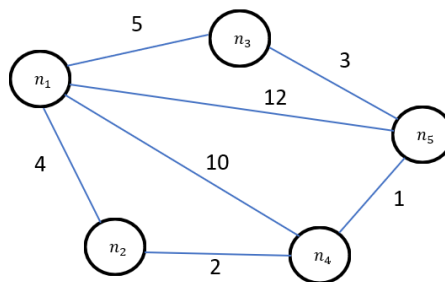


Figura 1: Esquema de grafo con dos aristas paralelas, un lazo y un vértice aislado.

Si se viaja de un nodo  $n_1$  al nodo  $n_2$  y así sucesivamente hasta el nodo  $n_n$  el viaje completo se llama **trayectoria o ruta**. La trayectoria tiene una longitud  $n$  que es igual al número de aristas que componen dicha trayectoria. Si en una trayectoria no hay nodos repetidos, se trata de una **trayectoria simple**. Por ejemplo, en la *Figura 2* más adelante, en la que se representa un grafo con cinco nodos ( $n_1, \dots, n_5$ ) y siete aristas, se escoge una trayectoria que empieza en  $n_1$  pasa por  $n_3$  y acaba en  $n_5$  la trayectoria se designará  $(n_1, n_3, n_5)$ .

Un **ciclo** se define como una trayectoria de longitud distinta de cero desde un nodo  $n$  al mismo nodo  $n$  sin aristas repetidas. Un ciclo será simple cuando no tenga nodos repetidos a excepción del nodo de inicio y fin.

Los grafos que se van a tratar de ahora en adelante van a ser **grafos ponderados**. En estos grafos las aristas tendrán asociado un valor o peso. Por ejemplo, en la *Figura 2*, el peso de la arista  $(n_1, n_2)$  es 4. En estos grafos se llama **longitud de la ruta** a la suma de los pesos asociados a las aristas en dicha ruta (Johnsonbaugh, 2005). Por ejemplo, la longitud de la ruta que comienza en  $n_1$ , pasa por  $n_3$  y termina en  $n_5$  es igual a 8.



*Figura 2: Grafo simple ponderado con cinco nodos y seis aristas.*

Un grafo  $G$  será **conexo** si en cualquier par de vértices o nodos pertenecientes a  $G$  existe una trayectoria. Aquellos grafos que no sean conexos van a constar de varios **subgrafos** y éstos se van a llamar **componentes**. Los vértices y aristas que se seleccionan del grafo original  $G$  para formar parte de un subgrafo  $G'$  tienen una restricción. Si se selecciona una arista  $e$  cualquiera de  $G$  para formar parte de  $G'$ , y que incide en los vértices  $u$  y  $v$ , entonces los vértices  $u$  y  $v$  tendrán que incluirse también en  $G'$  (Johnsonbaugh, 2005).

Dado un grafo  $G = (N, A)$ .  $(N', A')$  es un subgrafo de  $G$  si

- $N' \subseteq N$  y  $A' \subseteq A$ .
- Y también para toda arista  $a' \in A'$ , si  $a'$  incide en  $u'$  y  $v'$ , entonces  $u'$  y  $v' \in N'$ .

Por ejemplo, en la Figura 2 se muestra un grafo conexo ya que entre dos nodos cualesquiera existe una trayectoria. En cambio, en la Figura 3, podemos observar un subgrafo derivado del grafo de la Figura 2 con cinco nodos y cuatro aristas, el cual no es conexo ya que no existe una trayectoria entre  $n_1$  y  $n_3$ , el nodo  $n_3$  está aislado.

Un **subgrafo** será **de expansión** si  $N' = N$ . Por tanto, los nodos o vértices presentes en el grafo original lo estarán en el subgrafo de expansión. El grafo de la Figura 3 es un subgrafo de expansión de la Figura 2.

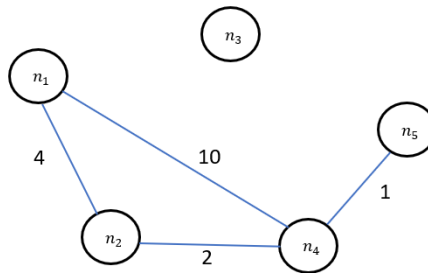


Figura 3: Subgrafo derivado del grafo simple ponderado con cinco nodos y seis aristas.

## 2. ELEMENTOS BÁSICOS DE UN ÁRBOL

Dado un grafo  $G = (N, A)$ , simple y no dirigido.  $G$  será un **árbol** si es conexo y no tiene ciclos. Entre dos vértices cualesquiera pertenecientes a un árbol existirá una única trayectoria simple que los conecta (Alvarez Nuñez, 2013).

Un **árbol de expansión** de un grafo  $G = (N, A)$ , será un subgrafo de expansión que también es un árbol, como se puede ver en la Figura 4, la cual representa un árbol de expansión del grafo de la Figura 2 con los mismos cinco nodos, pero sólo cuatro aristas. Un árbol de expansión contiene todos los vértices del grafo del que proviene. Un **árbol de expansión mínimo** será un árbol de expansión cuya suma de los pesos de las aristas sea mínimo, el grafo de la Figura 4 corresponde al árbol de expansión mínimo de la Figura 2.

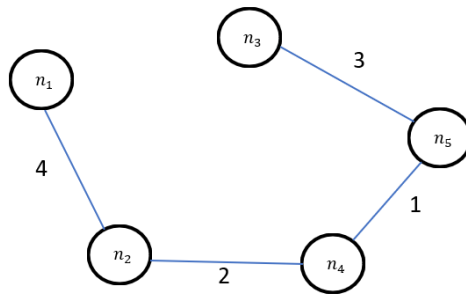


Figura 4: Árbol derivado del grafo simple ponderado con cinco nodos y seis aristas.

Para un árbol  $G = (N, A)$  el cual es un grafo simple no dirigido, las siguientes condiciones son equivalentes, es decir, si se verifica una de ellas, se verifica el resto:

- $G$  es conexo y si se elimina cualquier arista de  $A$ , queda desconectado en dos subgrafos que son árboles.
- $G$  no tiene ciclos y  $|N| = |A| + 1$
- $G$  es conexo y  $|N| = |A| + 1$
- $G$  no tiene ciclos y, si se añade una arista, se forma un grafo con un ciclo.

### 3. EL PROBLEMA DE STEINER

En el problema de Steiner se tiene un grafo no dirigido  $G = (N, A)$ , el cual es ponderado, (teniendo  $u_{ij}$ ,  $(i, j) \in A$  pesos en las aristas) y, además, tiene un subconjunto de nodos,  $T \subseteq N$ , llamados **nodos terminales**. El resto de los nodos no terminales se denominan nodos de Steiner.

Si el grafo tiene  $N$  nodos, todo árbol expandido tiene exactamente  $N-1$  aristas.

El problema se basa en encontrar un subgrafo que conecte los nodos terminales entre sí y la suma de sus pesos sea mínima (Du D. Z., 2000). Normalmente los pesos son positivos, en cuyo caso la solución tendrá forma de árbol y éste será conocido como árbol de Steiner.

Hay dos casos especiales:

1. Si  $T = N$ , entonces el problema se reduce a un árbol mínimo expandido o MST (*Minimal Spanning Tree*).
2. Si el número de nodos terminales es dos, entonces el problema se reduce a encontrar el camino más corto (o con menores pesos) entre los dos nodos terminales.

Mientras que para el problema del árbol mínimo expandido existen algoritmos polinómicos que lo solucionan como el de Kruskal o Prim (Du & Hu, 2008), los cuales se van a detallar a continuación, para el problema del árbol de Steiner o SMT (*Steiner Minimum Tree*) estos algoritmos no lo solucionan.

En el siguiente apartado se presentan y explican los algoritmos de Kruskal y Prim, los cuales sirven para solucionar el problema del árbol mínimo expandido (MST).

### 3.1. Algoritmos de Kruskal y Prim para obtener el MST de un grafo.

Se asume que se tiene un grafo simple, conexo, no dirigido y ponderado con pesos positivos. Los dos algoritmos que se van a presentar para conseguir el árbol mínimo expandido del grafo son ambiciosos, esto supone que los algoritmos buscan la mejor solución en cada momento con la esperanza de que una serie de soluciones óptimas locales conlleven la obtención de una solución óptima global (Cormen, 2022).

#### 3.1.1. Algoritmo de Kruskal.

Hay que recalcar que el algoritmo de Kruskal es una **solución manual** y no puede ser programada para que sea utilizada por un software, es por esto que el algoritmo de Prim será el seleccionado para problemas utilizando software más adelante. El grafo  $G = (V, E)$  va a tener un conjunto de  $V$  vértices y  $E$  aristas. Cada arista  $e_i$  tiene un peso  $P_{e_i}$ . Las aristas que se escojan en cada paso pertenecen al conjunto  $A$ , del que se retirarán las aristas elegidas al final de cada paso. Una vez elegidas las aristas, éstas se van moviendo del conjunto  $A$  al conjunto  $T$ , el cual al final del proceso contendrá las aristas que forman el árbol mínimo expandido.

El proceso va a tener  $n - 1$  pasos, donde  $n = |V|$ . Procedimiento:

1. Inicialización:  $A = E ; T = \emptyset$
2. Paso que se repite  $n - 1$  veces:
  - a. Selección de la arista  $e_i \in A$  que tenga peso mínimo y no forme ciclo con las aristas de  $T$ . Si hay más de una arista con el peso mínimo, se escoge arbitrariamente.
  - b. Actualización de los conjuntos:  $A \leftarrow A - \{e_i\}; T \leftarrow T \cup \{e_i\}$

Más adelante se va a detallar un ejemplo de aplicación del algoritmo de Kruskal (Figura 5):

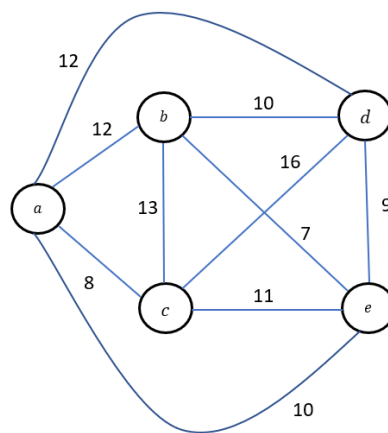


Figura 5: Grafo ponderado con cinco nodos y diez aristas para la aplicación del algoritmo de Kruskal.

El grafo del ejemplo se compone de 5 vértices ( $a, \dots, e$ ) y 10 aristas. Por lo tanto, el algoritmo consistirá en  $5 - 1 = 4$  pasos.

#### Inicialización

$$A = E = \{a, b; a, c; b, c; b, e; c, d; b, d; c, e; d, e; a, d; a, e\}; T = \emptyset$$

#### Paso 1

$$\text{Arista de peso mínimo: } e_1 = \{e, b\}; P_{e_1} = 7$$

$$A \leftarrow E - \{e_1\}; T = \{e_1\}$$

#### Paso 2

$$\text{Arista de peso mínimo: } e_2 = \{a, c\}; P_{e_2} = 8$$

$$A \leftarrow E - \{e_1; e_2\}; T = \{e_1; e_2\}$$



### Paso 3

Arista de peso mínimo:  $e_3 = \{e, d\}$ ;  $P_{e_3} = 9$

$A \leftarrow E - \{e_1; e_2; e_3\}$ ;  $T = \{e_1; e_2; e_3\}$

### Paso 4

Arista de peso mínimo:  $e_4 = \{b, d\}$  ó  $e_4 = \{a, e\}$ ;  $P_{e_4} = 10$

Se escoge  $\{b, d\}$  arbitrariamente, sin embargo, esta arista formaría ciclo en el conjunto T. Por lo tanto,  $e_4 = \{a, e\}$ .

$A \leftarrow E - \{e_1; e_2; e_3; e_4\}$ ;  $T = \{e_1; e_2; e_3; e_4\}$

La solución gráfica se muestra a continuación en la Figura 6:

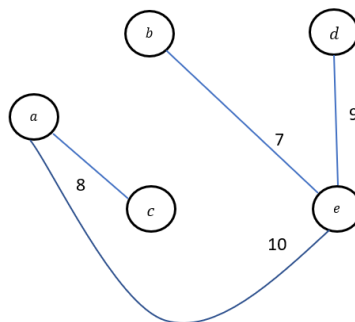


Figura 6: Solución obtenida con el algoritmo de Kruskal para el ejemplo de grafo ponderado con cinco nodos y diez aristas.

### 3.1.2. Algoritmo de Prim.

Para explicar el funcionamiento del algoritmo de Prim, primero se va a detallar una solución manual y, a continuación, se mostrará una solución conseguida con programación. Este algoritmo va a ser el que se va a usar a la hora de conseguir el árbol de Steiner con el programa y así poder resolver el problema que atañe a este trabajo. La principal razón por la que se escoge este algoritmo y no el de Kruskal es por la dificultad a la hora de programar la comprobación de la formación de ciclos en el algoritmo de Kruskal.

Antes de continuar con el ejemplo es necesario introducir el concepto de **corte** en grafos: un corte  $[S, \bar{S}]$  en un grafo  $G = (V, E)$  divide el conjunto de vértices  $V$  en dos subconjuntos ( $S$  y  $\bar{S}$ ), donde  $\bar{S} = V - S$  y  $(\bar{S} \cup S = V)$ . El corte consistirá en todas las aristas de  $E$  con un extremo en  $S$  y otro en  $\bar{S}$ . A continuación, se muestra un ejemplo:

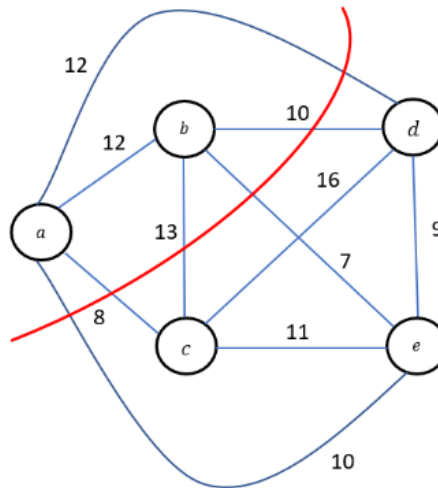


Figura 7: Ejemplo de corte en grafos.

En la Figura 7, los subconjuntos que crea el corte que aparece como una línea roja serían los siguientes:

$$S = \{a, b\}; \quad \bar{S} = \{c, d, e\}$$

$$[S, \bar{S}] = \{ad, ac, ae, bd, be, bc\}$$

Se puede ver como la línea roja del corte deja a un lado los nodos  $a$  y  $b$  y al otro los nodos  $c, d$  y  $e$ . Como se ha explicado anteriormente, las aristas que se corten con la línea roja van a ser las que compongan el conjunto del corte  $[S, \bar{S}]$ .

Se va a utilizar el mismo grafo  $G = (V, E)$  que se ha usado en el apartado en el que se explica el algoritmo de Kruskal. El grafo del ejemplo va a tener un conjunto de  $V$  vértices y  $E$  aristas, cada arista  $e_i$  tiene un peso  $P_{e_i}$ . Para este algoritmo se van a tener varios conjuntos que al comienzo del proceso tienen los siguientes valores:

- $N = V$  Donde  $N$  es el conjunto de nodos seleccionables y  $V$  es el conjunto de nodos que componen el grafo.
- $P = \emptyset$  Donde  $P$  es el conjunto de nodos que se han seleccionado en los pasos de la etapa 2.
- $T = \emptyset$  Donde  $T$  es el conjunto de aristas que se van eligiendo para formar parte de la solución.

El proceso va a tener  $n - 1$  pasos, donde  $n = |V|$ . Procedimiento:

- Etapa 1, Inicialización: se elige  $v_1 \in V$  arbitrariamente,  $N = V$ ;  $P = \emptyset$ ;  $T = \emptyset$ .

En esta etapa se establecen todos los conjuntos y se elige un nodo de forma arbitraria para comenzar con la siguiente la cual va a constituir el proceso de selección de las aristas.

- Etapa 2, procedimiento que se repite  $n - 1$  veces comenzando por  $i = 2$ :
  - 1º Actualización de los conjuntos:  $N \leftarrow N - \{v_i\}$ ;  $P \leftarrow P \cup \{v_i\}$ .

En la primera fase de la etapa se actualizan los conjuntos  $N$  y  $P$  restando y sumando respectivamente el vértice seleccionado en la inicialización o en el paso anterior.
  - 2º Selección de arista:  $e_i = \{x, y\} \in [P, N]$  (donde  $x \in P$ ,  $y \in N$ ) de coste mínimo.  $T \leftarrow T \cup \{e_i\}$ ,  $v_{i+1} = y$ .

En la segunda fase se elige la arista de coste mínimo que se encuentra entre las disponibles. Las aristas disponibles serán aquellas que compongan el corte  $[P, N]$ .

Por último, se añade la arista elegida al conjunto  $T$  y se designa el nodo del conjunto  $N$  de la arista elegida como el nodo que ha de actualizarse en los conjuntos  $P$  y  $N$  en el siguiente paso.

Los vértices ( $y$ ) que se escojan en cada paso en consecuencia de elegir las aristas formadas por  $\{x, y\}$  pertenecen al conjunto  $N$ , del que se retirarán los vértices ( $y$ ) elegidos al final de cada paso. Es decir, en cada paso los vértices ( $y$ ) se van moviendo del conjunto  $N$  al conjunto  $P$ . Las aristas formadas por  $\{x, y\}$  en cada paso se añadirán al conjunto  $T$  el cual al final del proceso contendrá las aristas que forman el árbol mínimo expandido.

A continuación, se va a explicar un ejemplo de aplicación del algoritmo de Prim. Se añadirán comentarios a la inicialización y el primer paso de la etapa 2, los pasos restantes van a desarrollarse de la misma forma (el grafo va a ser el mismo que se ha utilizado para la explicación del algoritmo de Kruskal).

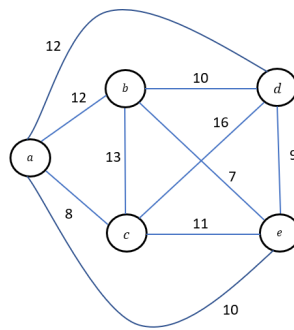


Figura 8: Ejemplo de grafo ponderado con cinco nodos y diez aristas para la aplicación del algoritmo de Prim.

El grafo del ejemplo se compone de 5 vértices ( $a, \dots, e$ ) y 10 aristas. Por lo tanto, el algoritmo consistirá en  $5 - 1 = 4$  pasos.

### Inicialización

$$N = \{a, b, c, d, e\}; P = \emptyset; T = \emptyset; v_1 = e$$

El conjunto  $N$  se compone en el comienzo de todos los vértices que componen el grafo del enunciado. Los conjuntos  $P$  y  $T$  se encuentran vacíos al no haberse elegido ningún vértice o arista.

Se escoge arbitrariamente el vértice  $e$  para comenzar el paso 1 de la etapa 2.

### Paso 1

$$1^\circ \text{ Actualización: } N = \{a, b, c, d\}; P = \{e\}$$

El vértice  $e$  escogido en la actualización se resta del conjunto  $N$  y se suma al conjunto  $P$ .

$$2^\circ \text{ Selección de arista: } [P, N] = \{\{e, c\}, \{e, b\}, \{e, a\}, \{e, d\}\} \rightarrow e_1 = \{e, b\} \rightarrow T = \{\{e, b\}\}; v_2 = b$$

El corte  $[P, N]$  se compone de las aristas elegibles, en la Figura 9 se puede ver como en este paso el corte (línea roja) rodea al nodo  $e$ , al ser el único del conjunto  $P$ .

Se selecciona la arista  $\{e, b\}$  al ser la arista con menor coste (coste 7), esto determina que el siguiente nodo que se va a tratar en la actualización va a ser el nodo  $b$ .

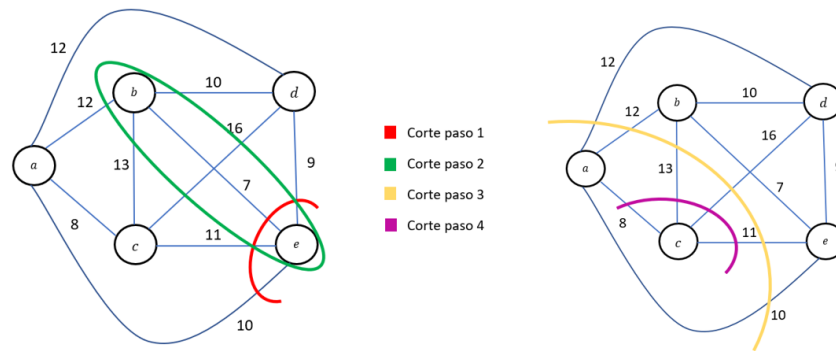


Figura 9: Cortes en el ejemplo de grafo ponderado con cinco nodos y diez aristas en la aplicación del algoritmo de Prim.

### Paso 2

Actualización:  $N = \{a, c, d\}$ ;  $P = \{e, b\}$ .

Selección de arista:  $[P, N] = \{\{b, d\}, \{b, c\}, \{b, a\}, \{e, d\}, \{e, a\}, \{e, c\}\} \rightarrow e_2 = \{e, d\} \rightarrow T = \{\{e, b\}, \{e, d\}\}$ ;  $v_3 = d$

### Paso 3

Actualización:  $N = \{a, c\}$ ;  $P = \{e, b, d\}$

Selección de arista:  $[P, N] = \{\{d, a\}, \{b, a\}, \{b, c\}, \{d, c\}, \{e, c\}, \{e, a\}\} \rightarrow e_3 = \{e, a\} \rightarrow T = \{\{e, b\}, \{e, d\}, \{e, a\}\}$ ;  $v_4 = a$

### Paso 4

Actualización:  $N = \{c\}$ ;  $P = \{e, b, d, a\}$

Selección de arista:  $[P, N] = \{\{c, a\}, \{c, b\}, \{c, d\}, \{c, e\}\} \rightarrow e_3 = \{a, c\} \rightarrow T = \{\{e, b\}, \{e, d\}, \{e, a\}, \{a, c\}\}$ ;  $v_5 = c$

Por último, se muestra en la Figura 10 el árbol que representa la solución del ejemplo anterior y que está compuesto por las aristas del conjunto  $T$  final, la solución va a ser la misma que se obtuvo con el algoritmo de Kruskal.

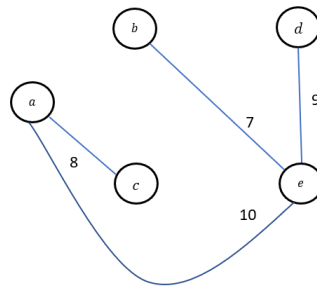


Figura 10: Solución para el ejemplo de grafo ponderado con cinco nodos y diez aristas obtenida con el algoritmo de Prim.

Una vez se ha resuelto un caso sencillo con la aplicación del algoritmo de Prim, se va a proceder con un caso más complejo. La figura que se muestra a continuación muestra el grafo del que se va a obtener el árbol mínimo expandido aplicando el algoritmo de Prim, el grafo consta de veinticinco nodos y cuarenta y ocho aristas.

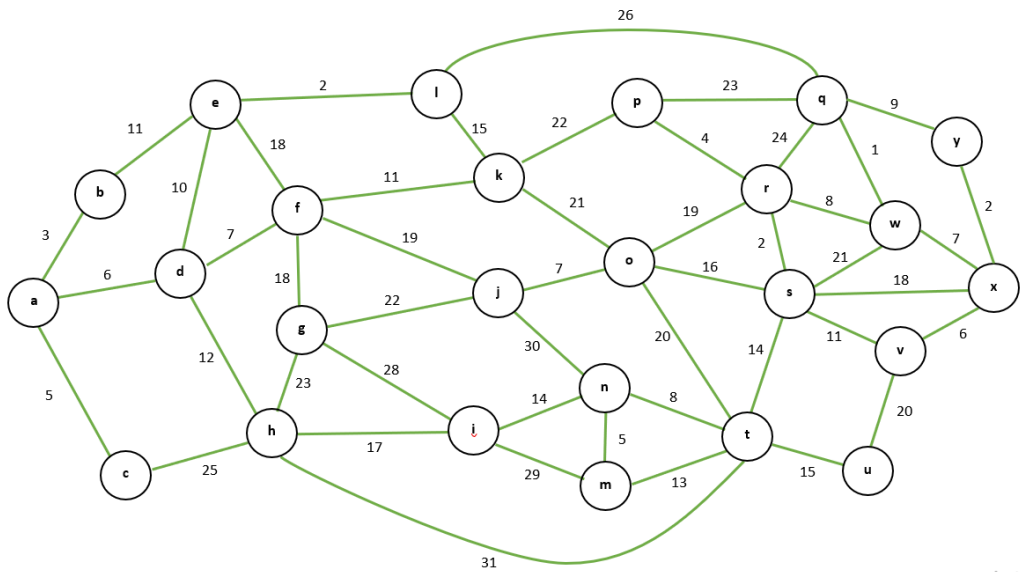


Figura 11: Grafo de veinticinco nodos y cuarenta y ocho aristas para la aplicación del algoritmo de Prim.

Una vez expuesto el grafo con todos sus nodos y aristas, se procede a aplicar el algoritmo de Prim eligiendo primero el primer nodo de forma arbitraria, el primer nodo será el nodo a. A continuación, en la Figura 12, se muestra el árbol mínimo de expansión obtenido tras aplicar el algoritmo de Prim sobre el grafo de la Figura 11.

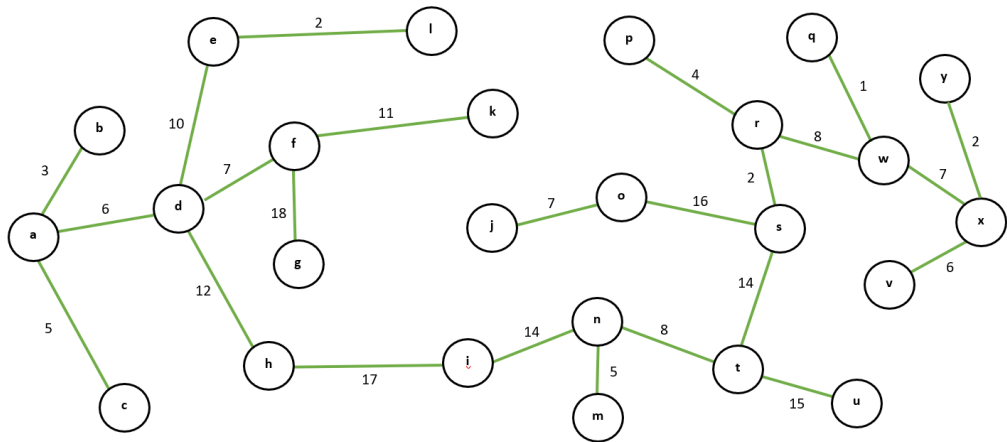


Figura 12: Árbol mínimo de expansión obtenido del grafo de veinticinco nodos y cuarenta y ocho aristas.

## 4. FORMULACIÓN DEL PROBLEMA DE STEINER

En el presente apartado, se van a explicar los pasos para formular el problema de Steiner y así poder abordarlo de forma matemática. Como se explicó anteriormente, los algoritmos de Kruskal o Prim no solucionan el problema de Steiner. Por esto, se va a explicar a continuación la forma en que se va a resolver el problema.

El problema del mínimo árbol de Steiner se puede formular como un **problema de flujo en redes con costos fijos**, con las siguientes características:

1. Se convierte el grafo en uno dirigido, transformando cada arista  $(i, j)$  en dos aristas dirigidas  $(i, j)$  y  $(j, i)$ .
2. Los costos fijos  $f_{ij} = w_{ij}$  serán los únicos existentes ya que **no habrá costos variables** para toda arista  $(i, j) \in A$ .
3. Se designa arbitrariamente a un nodo terminal como raíz,  $r \in T$  y se toman como ofertas/demandas:

$$b_r = |T| - 1$$

$$b_i = -1 \text{ si } i \in T, i \neq r$$

$$b_i = 0 \text{ para todo } i \notin T$$

$b_r$  va a ser la oferta/demanda del nodo terminal raíz, que será igual al número de nodos terminales menos uno.

$b_i$  va a ser la oferta/demanda del resto de nodos. Para los nodos terminales que no son raíz, la oferta/demanda va a ser menos uno (al ser negativo será demanda). En el caso de los nodos que no son terminales la oferta/demanda será cero.

4. Las capacidades de los arcos se toman igual al flujo total, que es igual a  $|T| - 1$ .

Una vez formulado el problema, en el siguiente apartado se van a definir las variables, función objetivo y restricciones del problema.

#### 4.1. Modelización del problema de Steiner.

En el presente apartado se va a presentar el modelo del problema de Steiner, explicando de forma matemática el proceso para llegar a la solución. Para ello se van a describir tanto las variables de decisión como la función objetivo y sus restricciones.

##### 1. Variables de decisión del problema

Se definen las variables de decisión del problema:

$$x_{ij}, (ij) \in A \quad x_{ij} \geq 0$$

Donde  $x_{ij}$  es una variable continua que representa el flujo que circula a través del arco  $(ij)$  el cual pertenece al conjunto de aristas  $(A)$  del grafo considerado.

Las variables  $y_{ij}$  son las variables de diseño, éstas son las que dan la configuración o el diseño final de la red. Tienen asociado en la función objetivo un coste fijo  $f_{ij}$  (por ejemplo, el costo de construcción del tramo o el mantenimiento de la ruta).

$$y_{ij} (ij) \in \{0, 1\} \quad , \quad (ij) \in A$$

Donde  $y_{ij}$  es una variable binaria que indica si el arco  $(i, j)$  se incluye en el diseño o no. En el primer caso toma valor 1 y en el segundo caso toma valor 0. Es decir, cuando  $x_{ij} \geq 0$ .



## 2. Función objetivo

La función objetivo pretende minimizar el sumatorio de los costos fijos de aquellos arcos que forman parte del modelo:

$$\min \sum_{(ij) \in A} f_{ij} y_{ij}$$

La función objetivo está sujeta a dos bloques de restricciones del siguiente tipo:

*s. a.*

$$\sum_{j \in N} x_{ij} - \sum_{j \in N} x_{ji} = b_i, \quad (ij) \in A, \quad (ji) \in A, \quad \forall i \in N$$

$$x_{ij} \leq u_{ij} \cdot y_{ij}, \forall i \in A$$

El primer tipo de restricción corresponde a las ecuaciones de balance o de conservación del flujo de redes (ecuación de Kirchhoff). Para cada nodo  $i \in N$ , se tiene la oferta o demanda de ese nodo  $b_i$ , con la convención de que si  $b_i > 0$  es un nodo de oferta, y si  $b_i < 0$  es un nodo de demanda. La diferencia entre el sumatorio de los flujos entrantes en los nodos y el sumatorio de los flujos salientes tiene que ser igual a la oferta o demanda de dicho nodo. Se supone que la red está balanceada, es decir, que  $\sum_{i \in N} b_i = 0$ . Al final del bucle, la oferta del nodo raíz tiene que ser compensada por las demandas del resto de nodos terminales. Los nodos de Steiner, como se ha explicado en apartados anteriores, van a tener  $b_i = 0$ , por tanto, la diferencia de sumatorios de los correspondientes flujos tiene que ser igual a 0.

El segundo tipo de restricción está relacionado con el flujo máximo que puede circular por los arcos. Lógicamente el flujo máximo tendrá que ser menor o igual a la capacidad de cada arco que está implicado en el modelo. Para que se cumpla la relación entre las variables, es necesario que estén presentes estas restricciones. Así, cuando  $x_{ij} \geq 0$ , respectivamente,  $y_{ij} = 1$ . Para asociar las variables de flujo  $x_{ij} \geq 0$  con las variables binarias  $y_{ij} \in \{0, 1\}$  se puede considerar  $u_{ij} = N - 1$ , donde  $u_{ij}$  será el flujo máximo que puede circular por los arcos y será igual al número de nodos terminales menos uno.

## 4.2. Formato del archivo que usa el software como input.

Este archivo va a contener los datos que describen el grafo que se quiere optimizar. Contiene cuatro bloques principales:

- El vector  $b$  contiene los nodos terminales. Van a aparecer en una lista tantos números como nodos, no obstante, sólo los nodos que sean terminales van a tener asignado un número distinto de 0. El primer nodo terminal que aparezca (el nodo raíz) tendrá un valor igual a la suma de nodos terminales menos 1. El resto de los nodos terminales tendrán un valor igual a -1. De este modo se pondrá la primera condición del modelo.
- El vector arco es el encargado de crear todos los arcos existentes entre los nodos tanto en un sentido como en el contrario. Por ejemplo, para el arco que conecta dos nodos 1 y 2 en el archivo aparecería como (1, 2)1 y justo debajo (2, 1)1. Siempre habrá presente un 1 después del paréntesis.
- El vector costo fijo es el que otorga el peso a los arcos, tiene un formato parecido al vector arco, pero con la diferencia de que van a cambiar los valores posteriores al paréntesis, estos valores van a ser los que designen el peso de cada arco. Por ejemplo, el arco que hay entre dos nodos 1 y 2, el cual tiene un peso de valor cinco, aparecería de la siguiente forma: (1, 2)5 y justo debajo (2, 1)5.
- El vector cota establece los valores para las cotas superiores de los arcos. La cota superior se puede igualar al flujo máximo que puede circular en los arcos. Como se ha descrito en la modelización del problema, la cota superior ( $u_{ij}$ ) será igual al número de nodos terminales menos uno. Los elementos de este vector serán parecidos a los de los vectores arco y costo fijo, pero en este caso todos los elementos tendrán un valor igual a la cota superior a la derecha del paréntesis que contiene el arco. En el problema que se trata, el valor de la cota superior va a ser igual para todos los arcos. Por ejemplo, en un grafo que tiene 5 nodos terminales, el arco que existe entre dos nodos 1 y 2 aparecería de la siguiente forma: (1, 2)4 y justo debajo (2, 1)4.

### 4.3. Comandos especiales para la creación de gráficos en XPRESS MOSEL

En este apartado se detallan los comandos especiales para crear gráficos en Xpress Mosel, usados para la búsqueda del árbol mínimo expandido de los ejemplos de los apartados 6.1 y 7.1.

#### **procedure IVEerase**

Elimina todos los gráficos y resetea el gráfico de usuario (gráfico construido por el usuario durante la ejecución de Mosel).

#### **procedure IVEzoom (x1: real, y1: real, x2:real, y2:real)**

Escala el gráfico de usuario. El área visible está determinada por sus esquinas inferior izquierda y superior derecha.

#### **function IVEaddplot (name: string, color:integer): integer**

Inserta un nuevo gráfico en el gráfico de usuario. Un gráfico se identifica por su nombre y se puede mostrar u ocultar utilizando la casilla de verificación de su leyenda correspondiente. El número máximo actual de gráficos distintos está limitado a 20. Sin embargo, cada gráfico puede contener un número ilimitado de puntos, líneas, flechas y etiquetas.

#### **procedure IVEdrawline (handle:integer, x1: real, y1:real, x2:real, y2:real)**

Agrega una línea a un gráfico existente. La línea conecta los dos puntos cuyas coordenadas se dan como parámetros.

#### **procedure IVEdrawpoint (handle: integer, x: real, y: real)**

Agrega un pequeño cuadrado para marcar un punto en las coordenadas dadas.

#### **procedure IVEdrawlabel (handle: integer, x: real, y: real, text: string)**

Agrega un cuadro de texto a un gráfico existente. El cuadro se centrará horizontalmente justo encima del punto dado.

## 5. ALGUNOS EJEMPLOS RESUELTOS DE GRAFOS

Para ejemplificar la resolución del problema de Steiner, se van a exponer algunos ejemplos de grafos sencillos. Después de cada ejemplo se presentarán las soluciones conseguidas de forma manual y mediante el programa Xpress Mosel, mostrando el árbol mínimo expandido de cada uno de los ejemplos.

### 5.1. Grafo ponderado con trece nodos y veinticuatro aristas.

En la Figura 13 se puede observar un grafo no dirigido ponderado sencillo con trece nodos, de los cuales cuatro son nodos terminales  $N = \{1, 4, 7, 11\}$  y están marcados con un cuadrado rojo. El resto son nodos Steiner, están marcados con un círculo negro. Estos nodos están conectados por veinticuatro aristas o arcos. Tanto para este grafo como para los siguientes ejemplos del apartado 5, se va a buscar el árbol mínimo expandido tanto de forma manual como con el uso de software, comparando ambas soluciones.

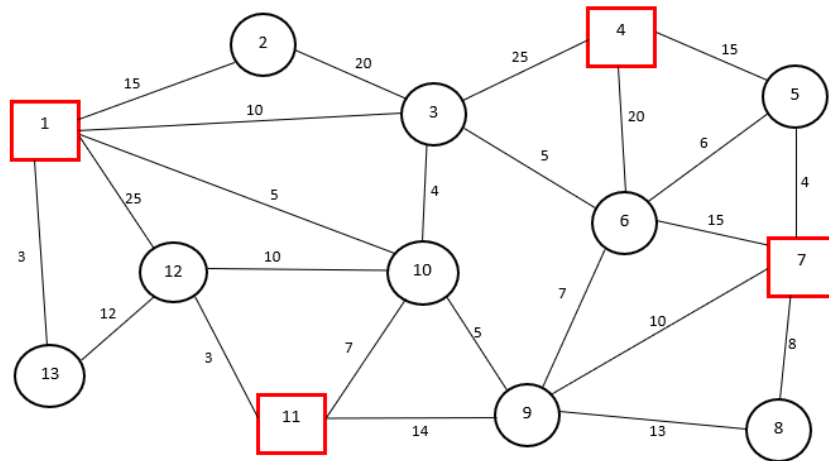


Figura 13: Grafo ponderado con trece nodos y veinticuatro aristas.

En la Tabla 1 se muestran los correspondientes costos fijos para cada arco, los costos están también señalados al lado de las aristas del grafo. Para conseguir una posible solución de forma manual, se van a ir recorriendo caminos de forma visual entre nodos terminales, escogiendo los caminos con la menor suma de costos fijos.

Tabla 1: Arcos y sus correspondientes costos fijos para el grafo ponderado con trece nodos y veinticuatro aristas.

ARCOS		COSTO FIJO
(1, 2)	(2, 1)	15
(2, 3)	(3, 2)	20
(3, 4)	(4, 3)	25
(4, 5)	(5, 4)	15
(5, 6)	(6, 5)	6
(6, 7)	(7, 6)	15
(7, 8)	(8, 7)	8
(8, 9)	(9, 8)	13
(9, 10)	(10, 9)	5
(10, 11)	(11, 10)	7
(11, 12)	(12, 11)	3
(12, 13)	(13, 12)	12
(1, 13)	(13, 1)	3
(1, 12)	(12, 1)	25
(1, 10)	(10, 1)	5
(1, 3)	(3, 1)	10
(10, 12)	(12, 10)	10
(3, 10)	(10, 3)	4
(3, 6)	(6, 3)	5
(6, 4)	(4, 6)	20
(6, 9)	(9, 6)	7
(5, 7)	(7, 5)	4
(7, 9)	(9, 7)	10
(9, 11)	(11, 9)	14

Después de varios intentos, se obtiene la solución manual propuesta en la Figura 14. Con esta solución manual propuesta, sin usar herramientas informáticas, se obtiene una suma de costos fijos igual a 56 con 3 nodos Steiner.

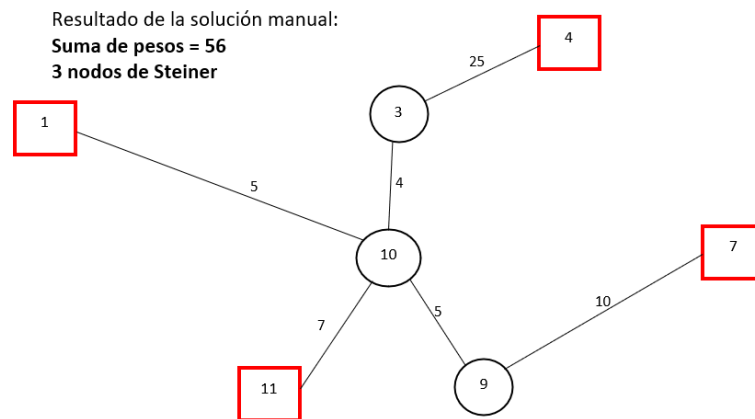


Figura 14: Solución manual propuesta para el grafo ponderado con trece nodos y veinticuatro aristas.

Más adelante en la Figura 15, se presenta la solución óptima obtenida con el software de optimización Xpress Mosel. Esta solución presenta una suma de pesos igual a 46 y tiene también 3 nodos Steiner. Se puede observar que el software esta vez ha podido mejorar la solución al obtener un valor menor de la función objetivo.

Solución óptima: Peso = 46 con 3 nodos de Steiner

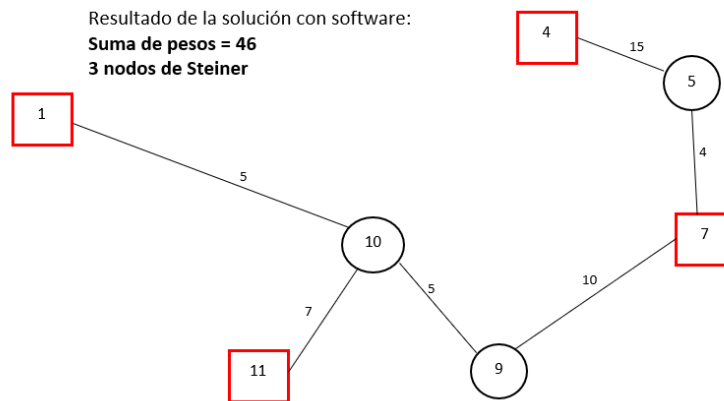


Figura 15: Solución óptima del grafo ponderado con trece nodos y veinticuatro aristas obtenida con el programa Xpress Mosel.

A continuación, se presenta la función objetivo necesaria para obtener la solución de la Figura 15 y las correspondientes restricciones del problema.

Función objetivo para el grafo de la Figura 13:

$$\begin{aligned}
 \min & 15 y_{1,2} + 10 y_{1,3} + 5 y_{1,10} + 25 y_{1,12} + 3 y_{1,12} + 15 y_{2,1} + 20 y_{2,3} + \\
 & 10 y_{3,1} + 20 y_{3,2} + 25 y_{3,4} + 5 y_{3,6} + 4 y_{3,10} + 25 y_{4,3} + 15 y_{4,5} + \\
 & 20 y_{4,6} + 15 y_{5,4} + 6 y_{5,6} + 4 y_{5,7} + 5 y_{6,3} + 20 y_{6,4} + 6 y_{6,5} + \\
 & 15 y_{6,7} + 7 y_{6,9} + 4 y_{7,5} + 15 y_{7,6} + 8 y_{7,8} + 10 y_{7,9} + 8 y_{8,7} + \\
 & 13 y_{8,9} + 7 y_{9,6} + 10 y_{9,7} + 13 y_{9,8} + 5 y_{9,10} + 14 y_{9,11} + 5 y_{10,1} + \\
 & 4 y_{10,3} + 5 y_{10,9} + 7 y_{10,11} + 10 y_{10,12} + 14 y_{11,9} + 7 y_{11,10} + \\
 & 3 y_{11,12} + 25 y_{12,1} + 10 y_{12,10} + 3 y_{12,11} + 12 y_{12,13} + 3 y_{13,1} + 12 y_{13,12}
 \end{aligned}$$

Restricciones para el grafo de la Figura 13:

*sujeto a:*

1. Ecuaciones de balance:

$$x_{1,2} + x_{1,3} + x_{1,10} + x_{1,12} + x_{1,13} - x_{2,1} - x_{3,1} - x_{10,1} - x_{12,1} - x_{13,1} = 3$$

$$-x_{1,2} + x_{2,1} + x_{2,3} - x_{3,2} = 0$$

$$-x_{1,3} - x_{2,3} + x_{3,1} + x_{3,2} + x_{3,4} + x_{3,6} + x_{3,10} - x_{4,3} - x_{6,3} - x_{10,3} = 0$$

$$-x_{3,4} + x_{4,3} + x_{4,5} + x_{4,6} - x_{5,4} - x_{6,4} = -1$$

$$-x_{4,5} + x_{5,4} + x_{5,6} + x_{5,7} - x_{6,5} - x_{7,5} = 0$$

$$-x_{3,6} - x_{4,6} - x_{5,6} + x_{6,3} + x_{6,4} + x_{6,5} + x_{6,7} + x_{6,9} - x_{7,6} - x_{9,6} = 0$$

$$-x_{5,7} - x_{6,7} + x_{7,5} + x_{7,6} + x_{7,8} + x_{7,9} - x_{8,7} - x_{9,7} = -1$$

$$-x_{7,8} + x_{8,7} + x_{8,9} - x_{9,8} = 0$$

$$-x_{6,9} - x_{7,9} - x_{8,9} + x_{9,6} + x_{9,7} + x_{9,8} + x_{9,10} + x_{9,11} - x_{10,9} - x_{11,9} = 0$$

$$-x_{1,10} - x_{3,10} - x_{9,10} + x_{10,1} + x_{10,3} + x_{10,9} + x_{10,11} + x_{10,12} - x_{11,10} - x_{12,10} = 0$$

$$-x_{9,11} - x_{10,11} + x_{11,9} + x_{11,10} + x_{11,12} - x_{12,11} = -1$$

$$-x_{1,12} - x_{10,12} - x_{11,12} + x_{12,1} + x_{12,10} + x_{12,11} + x_{12,13} - x_{13,12} = 0$$

$$-x_{1,13} - x_{12,13} + x_{13,1} + x_{13,12} = 0$$

2. Cotas:

$$x_{13,12} - 3 y_{13,12} \leq 0$$

$$x_{13,1} - 3 y_{13,1} \leq 0$$

$$x_{12,13} - 3 y_{12,13} \leq 0$$

$$x_{12,11} - 3 y_{12,11} \leq 0$$

$$x_{12,10} - 3 y_{12,10} \leq 0$$

$$x_{12,1} - 3 y_{12,1} \leq 0$$

$$x_{11,12} - 3 y_{11,12} \leq 0$$

$$x_{11,10} - 3 y_{11,10} \leq 0$$

$$x_{11,9} - 3 y_{11,9} \leq 0$$

$$x_{10,12} - 3 y_{10,12} \leq 0$$

$$\begin{aligned}x_{10,11} - 3 y_{10,11} &\leq 0 \\x_{10,9} - 3 y_{10,9} &\leq 0 \\x_{10,3} - 3 y_{10,3} &\leq 0 \\x_{10,1} - 3 y_{10,1} &\leq 0 \\x_{9,11} - 3 y_{9,11} &\leq 0 \\x_{9,10} - 3 y_{9,10} &\leq 0 \\x_{9,8} - 3 y_{9,8} &\leq 0 \\x_{9,7} - 3 y_{9,7} &\leq 0 \\x_{9,6} - 3 y_{9,6} &\leq 0 \\x_{8,9} - 3 y_{8,9} &\leq 0 \\x_{8,7} - 3 y_{8,7} &\leq 0 \\x_{7,9} - 3 y_{7,9} &\leq 0 \\x_{7,8} - 3 y_{7,8} &\leq 0 \\x_{7,6} - 3 y_{7,6} &\leq 0 \\x_{7,5} - 3 y_{7,5} &\leq 0 \\x_{6,9} - 3 y_{6,9} &\leq 0 \\x_{6,7} - 3 y_{6,7} &\leq 0 \\x_{6,5} - 3 y_{6,5} &\leq 0 \\x_{6,4} - 3 y_{6,4} &\leq 0 \\x_{6,3} - 3 y_{6,3} &\leq 0 \\x_{5,7} - 3 y_{5,7} &\leq 0 \\x_{5,6} - 3 y_{5,6} &\leq 0 \\x_{5,4} - 3 y_{5,4} &\leq 0 \\x_{4,6} - 3 y_{4,6} &\leq 0 \\x_{4,5} - 3 y_{4,5} &\leq 0 \\x_{4,3} - 3 y_{4,3} &\leq 0 \\x_{3,10} - 3 y_{3,10} &\leq 0 \\x_{3,6} - 3 y_{3,6} &\leq 0\end{aligned}$$



$$x_{3,4} - 3 y_{3,4} \leq 0$$

$$x_{3,2} - 3 y_{3,2} \leq 0$$

$$x_{3,1} - 3 y_{3,1} \leq 0$$

$$x_{2,3} - 3 y_{2,3} \leq 0$$

$$x_{2,1} - 3 y_{2,1} \leq 0$$

$$x_{1,13} - 3 y_{1,13} \leq 0$$

$$x_{1,12} - 3 y_{1,12} \leq 0$$

$$x_{1,10} - 3 y_{1,10} \leq 0$$

$$x_{1,3} - 3 y_{1,3} \leq 0$$

$$x_{1,2} - 3 y_{1,2} \leq 0$$

3. No negatividad de las variables de flujo:

$$x_{1,2}, x_{1,3}, x_{1,10}, \dots, x_{13,12} \geq 0$$

4. Variable  $y$  binaria:

$$y_{1,2}, y_{1,3}, y_{1,10}, \dots, y_{13,12} \in \{0, 1\}$$

## 5.2. Grafo ponderado con dieciocho nodos y treinta y seis aristas.

En la Figura 16 se considera un caso similar más complejo. El número de aristas y nodos de Steiner es superior y ahora se presentan treinta y seis aristas y dieciocho nodos, de los cuales cinco son nodos terminales  $N = \{2, 4, 6, 16, 18\}$ . Es importante resaltar que el nodo número 1 no tiene por qué ser un nodo terminal, de hecho, en este ejemplo, el primer nodo terminal sería el 2.

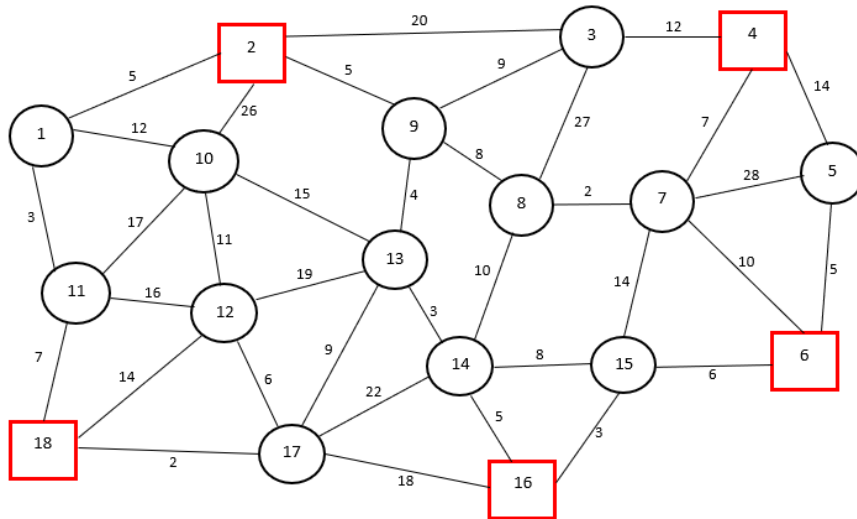


Figura 16: Ejemplo de grafo ponderado con dieciocho nodos y treinta y seis aristas.

En la Tabla 2 se muestran los correspondientes costos fijos para cada arco:

Tabla 2: Arcos y sus correspondientes costos fijos para el grafo ponderado con dieciocho nodos y treinta y seis aristas.

ARCOS		COSTO FIJO	ARCOS		COSTO FIJO
(1, 2)	(2, 1)	5	(8, 3)	(3, 8)	27
(2, 3)	(3, 2)	20	(8, 14)	(14, 8)	10
(3, 4)	(4, 3)	12	(9, 2)	(2, 9)	5
(4, 5)	(5, 4)	14	(9, 13)	(13, 9)	4
(5, 6)	(6, 5)	5	(2, 10)	(10, 2)	26
(6, 7)	(7, 6)	10	(10, 12)	(12, 10)	11
(7, 8)	(8, 7)	2	(10, 13)	(13, 10)	15
(8, 9)	(9, 8)	8	(11, 18)	(18, 11)	7
(10, 11)	(11, 10)	17	(12, 17)	(17, 12)	6
(11, 12)	(12, 11)	16	(12, 18)	(18, 12)	14
(12, 13)	(13, 12)	19	(13, 17)	(17, 13)	9
(1, 10)	(10, 1)	12	(14, 13)	(13, 14)	3
(1, 11)	(11, 1)	3	(14, 15)	(15, 14)	8
(3, 9)	(9, 3)	9	(14, 16)	(16, 14)	5
(5, 7)	(7, 5)	28	(14, 17)	(17, 14)	22
(6, 15)	(15, 6)	6	(15, 16)	(16, 15)	3
(7, 4)	(4, 7)	7	(17, 16)	(16, 17)	18
(7, 15)	(15, 7)	14	(17, 18)	(18, 17)	2

Como en el ejemplo anterior, se presenta una solución obtenida de forma manual. La solución obtenida sin usar el programa, la cual se puede ver en la Figura 17, tiene un costo total igual a 81 con 8 nodos de Steiner.

Resultado de la solución manual:  
**Suma de pesos = 81**  
**8 nodos de Steiner**

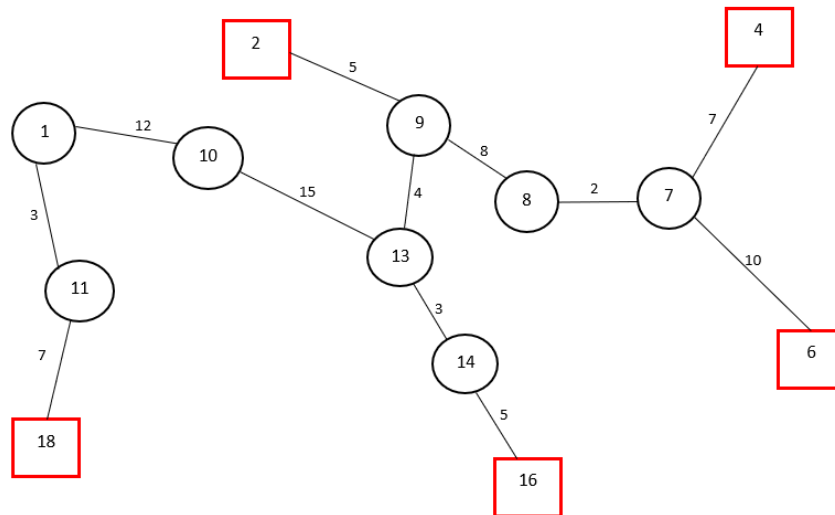


Figura 17: Solución manual propuesta al grafo ponderado con dieciocho nodos y treinta y seis aristas.

En la Figura 18 se muestra la solución óptima obtenida con software:

Resultado de la solución con software:  
**Suma de pesos = 54**  
**7 nodos de Steiner**

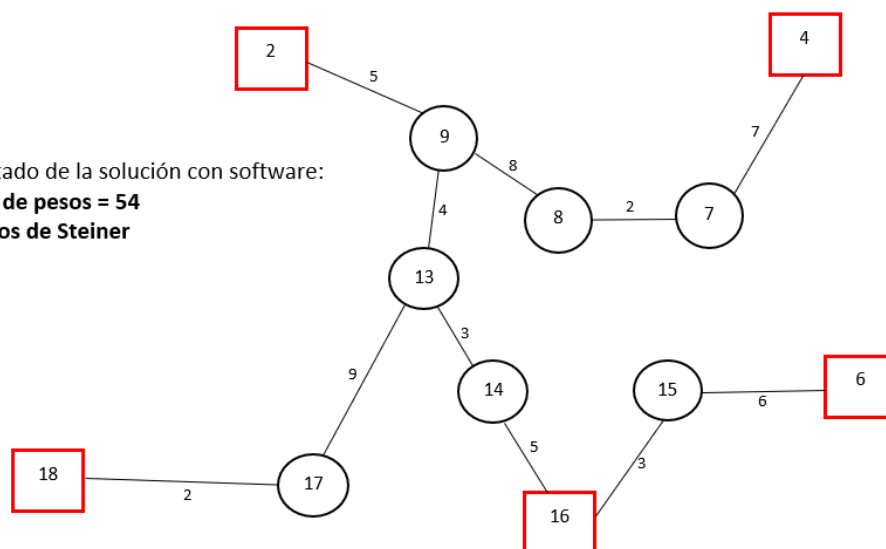


Figura 18: Solución óptima del grafo ponderado con dieciocho nodos y treinta y seis aristas, obtenida con el programa Xpress Mosel.

Otra vez, la solución óptima obtenida con software va a mejorar la solución propuesta conseguida a mano. La solución óptima de la Figura 18 tiene un peso total de 54 con 7 nodos de Steiner, mejorando en 27 la suma de pesos total de la solución manual.

Como se ha hecho para el grafo 1, se presenta la función objetivo necesaria para obtener la solución óptima de la Figura 16 y las correspondientes restricciones.

Función objetivo para el grafo de la Figura 16:

$$\begin{aligned}
 \min & 5y_{1,2} + 12y_{1,10} + 3y_{1,11} + 5y_{2,1} + 20y_{2,3} + 5y_{2,9} + 26y_{2,10} + \\
 & 20y_{3,2} + 12y_{3,4} + 27y_{3,8} + 9y_{3,9} + 12y_{4,3} + 14y_{4,5} + 7y_{4,7} + \\
 & 14y_{5,4} + 5y_{5,6} + 28y_{5,7} + 5y_{6,5} + 10y_{6,7} + 6y_{6,15} + 7y_{7,4} + \\
 & 28y_{7,5} + 10y_{7,6} + 2y_{7,8} + 14y_{7,15} + 27y_{8,3} + 2y_{8,7} + 8y_{8,9} + \\
 & 10y_{8,14} + 5y_{9,2} + 9y_{9,3} + 8y_{9,8} + 4y_{9,13} + 12y_{10,1} + 26y_{10,2} + \\
 & 17y_{10,11} + 11y_{10,12} + 15y_{10,13} + 3y_{11,1} + 17y_{11,10} + 16y_{11,12} + \\
 & 7y_{11,18} + 11y_{12,10} + 16y_{12,11} + 19y_{12,13} + 6y_{12,17} + 14y_{12,18} + \\
 & 4y_{13,9} + 15y_{13,10} + 19y_{13,12} + 3y_{13,14} + 9y_{13,17} + 10y_{14,8} + \\
 & 3y_{14,13} + 8y_{14,15} + 5y_{14,16} + 22y_{14,17} + 6y_{15,6} + 14y_{15,7} + \\
 & 8y_{15,14} + 3y_{15,16} + 5y_{16,14} + 3y_{16,15} + 18y_{16,17} + 6y_{17,12} + \\
 & 9y_{17,13} + 22y_{17,14} + 18y_{17,16} + 2y_{17,18} + 7y_{18,11} + 14y_{18,12} + \\
 & 2y_{18,17}
 \end{aligned}$$

Restricciones para el grafo de la Figura 16:

*sujeto a:*

1. Ecuaciones de balance:

$$x_{1,2} + x_{1,10} + x_{1,11} - x_{2,1} - x_{10,1} - x_{11,1} = 0$$

$$-x_{1,2} + x_{2,1} + x_{2,3} + x_{2,9} + x_{2,10} - x_{3,2} - x_{9,2} - x_{10,2} = 4$$

$$-x_{2,3} + x_{3,2} + x_{3,4} + x_{3,8} + x_{3,9} - x_{4,3} - x_{8,3} - x_{9,3} = 0$$

$$-x_{3,4} + x_{4,3} + x_{4,5} + x_{4,7} - x_{5,4} - x_{7,4} = -1$$

$$-x_{4,5} + x_{5,4} + x_{5,6} + x_{5,7} - x_{6,5} - x_{7,5} = 0$$

$$-x_{5,6} + x_{6,5} + x_{6,7} + x_{6,15} - x_{7,6} - x_{15,6} = -1$$

$$\begin{aligned}
-x_{4,7} - x_{5,7} - x_{6,7} + x_{7,4} + x_{7,5} + x_{7,6} + x_{7,8} + x_{7,15} - x_{8,7} - x_{15,7} &= 0 \\
-x_{3,8} - x_{7,8} + x_{8,3} + x_{8,7} + x_{8,9} + x_{8,14} - x_{9,8} - x_{14,8} &= 0 \\
-x_{2,9} - x_{3,9} - x_{8,9} + x_{9,2} + x_{9,3} + x_{9,8} + x_{9,13} - x_{13,9} &= 0 \\
-x_{1,10} - x_{2,10} + x_{10,1} + x_{10,2} + x_{10,11} + x_{10,12} + x_{10,13} - x_{11,10} - x_{12,10} - x_{13,10} &= 0 \\
-x_{1,11} - x_{10,11} + x_{11,1} + x_{11,10} + x_{11,12} + x_{11,18} - x_{12,11} - x_{18,11} &= 0 \\
-x_{10,12} - x_{11,12} + x_{12,10} + x_{12,11} + x_{12,13} + x_{12,17} + x_{12,18} - x_{13,12} - x_{17,12} - x_{18,12} \\
&= 0 \\
-x_{9,13} - x_{10,13} - x_{12,13} + x_{13,9} + x_{13,10} + x_{13,12} + x_{13,14} + x_{13,17} - x_{14,13} - x_{17,13} &= 0 \\
-x_{8,14} - x_{13,14} + x_{14,8} + x_{14,13} + x_{14,15} + x_{14,16} + x_{14,17} - x_{15,14} - x_{16,14} - x_{17,14} &= 0 \\
-x_{6,15} - x_{7,15} - x_{14,15} + x_{15,6} + x_{15,7} + x_{15,14} + x_{15,16} - x_{16,15} &= 0 \\
-x_{14,16} - x_{15,16} + x_{16,14} + x_{16,15} + x_{16,17} - x_{17,16} &= -1 \\
-x_{12,17} - x_{13,17} - x_{14,17} - x_{16,17} + x_{17,12} + x_{17,13} + x_{17,14} + x_{17,16} + x_{17,18} - x_{18,17} &= 0 \\
-x_{11,18} - x_{12,18} - x_{17,18} + x_{18,11} + x_{18,12} + x_{18,17} &= -1
\end{aligned}$$

## 2. Cotas:

$$\begin{aligned}
x_{18,17} - 4 y_{18,17} &\leq 0 \\
x_{18,12} - 4 y_{18,12} &\leq 0 \\
x_{18,11} - 4 y_{18,11} &\leq 0 \\
x_{17,18} - 4 y_{17,18} &\leq 0 \\
x_{17,16} - 4 y_{17,16} &\leq 0 \\
x_{17,14} - 4 y_{17,14} &\leq 0 \\
x_{17,13} - 4 y_{17,13} &\leq 0 \\
x_{17,12} - 4 y_{17,12} &\leq 0 \\
x_{16,17} - 4 y_{16,17} &\leq 0 \\
x_{16,15} - 4 y_{16,15} &\leq 0 \\
x_{16,14} - 4 y_{16,14} &\leq 0 \\
x_{15,16} - 4 y_{15,16} &\leq 0 \\
x_{15,14} - 4 y_{15,14} &\leq 0 \\
x_{15,7} - 4 y_{15,7} &\leq 0
\end{aligned}$$

$$\begin{aligned}
x_{15,6} - 4 y_{15,6} &\leq 0 \\
x_{14,17} - 4 y_{14,17} &\leq 0 \\
x_{14,16} - 4 y_{14,16} &\leq 0 \\
x_{14,15} - 4 y_{14,15} &\leq 0 \\
x_{14,13} - 4 y_{14,13} &\leq 0 \\
x_{14,8} - 4 y_{14,8} &\leq 0 \\
x_{13,17} - 4 y_{13,17} &\leq 0 \\
x_{13,14} - 4 y_{13,14} &\leq 0 \\
x_{13,12} - 4 y_{13,12} &\leq 0 \\
x_{13,10} - 4 y_{13,10} &\leq 0 \\
x_{13,9} - 4 y_{13,9} &\leq 0 \\
x_{12,18} - 4 y_{12,18} &\leq 0 \\
x_{12,17} - 4 y_{12,17} &\leq 0 \\
x_{12,13} - 4 y_{12,13} &\leq 0 \\
x_{12,11} - 4 y_{12,11} &\leq 0 \\
x_{12,10} - 4 y_{12,10} &\leq 0 \\
x_{11,18} - 4 y_{11,18} &\leq 0 \\
x_{11,12} - 4 y_{11,12} &\leq 0 \\
x_{11,10} - 4 y_{11,10} &\leq 0 \\
x_{11,1} &\leq 0 \\
x_{10,13} - 4 y_{10,13} &\leq 0 \\
x_{10,12} - 4 y_{10,12} &\leq 0 \\
x_{10,11} - 4 y_{10,11} &\leq 0 \\
x_{10,2} - 4 y_{10,2} &\leq 0 \\
x_{10,1} - 4 y_{10,1} &\leq 0 \\
x_{9,13} - 4 y_{9,13} &\leq 0 \\
x_{9,8} - 4 y_{9,8} &\leq 0 \\
x_{9,3} - 4 y_{9,3} &\leq 0
\end{aligned}$$

$$x_{9,2} - 4 y_{9,2} \leq 0$$

$$x_{8,14} - 4 y_{9,14} \leq 0$$

$$x_{8,9} - 4 y_{8,9} \leq 0$$

$$x_{8,7} - 4 y_{8,7} \leq 0$$

$$x_{8,3} - 4 y_{8,3} \leq 0$$

$$x_{7,15} - 4 y_{7,5} \leq 0$$

$$x_{7,8} - 4 y_{7,8} \leq 0$$

$$x_{7,6} - 4 y_{7,6} \leq 0$$

$$x_{7,5} - 4 y_{7,5} \leq 0$$

$$x_{7,4} - 4 y_{7,4} \leq 0$$

$$x_{6,15} - 4 y_{6,15} \leq 0$$

$$x_{6,7} - 4 y_{6,7} \leq 0$$

$$x_{6,5} - 4 y_{6,5} \leq 0$$

$$x_{5,7} - 4 y_{5,7} \leq 0$$

$$x_{5,6} - 4 y_{5,6} \leq 0$$

$$x_{5,4} - 4 y_{5,4} \leq 0$$

$$x_{4,7} - 4 y_{4,7} \leq 0$$

$$x_{4,5} - 4 y_{4,5} \leq 0$$

$$x_{4,3} - 4 y_{4,3} \leq 0$$

$$x_{3,9} - 4 y_{3,9} \leq 0$$

$$x_{3,8} - 4 y_{3,8} \leq 0$$

$$x_{3,4} - 4 y_{3,4} \leq 0$$

$$x_{3,2} - 4 y_{3,2} \leq 0$$

$$x_{2,10} - 4 y_{2,10} \leq 0$$

$$x_{2,9} - 4 y_{2,9} \leq 0$$

$$x_{2,3} - 4 y_{2,3} \leq 0$$

$$x_{2,1} - 4 y_{2,1} \leq 0$$

$$x_{1,11} \leq 0$$

$$x_{1,10} - 4 y_{1,10} \leq 0$$

$$x_{1,2} - 4 y_{1,2} \leq 0$$

3. No negatividad de las variables de flujo:

$$x_{1,2}, x_{1,10}, x_{1,11}, \dots, x_{13,12} \geq 0$$

4. Variable ( $y$ ) binaria:

$$y_{1,2}, y_{1,3}, y_{1,10}, \dots, y_{18,17} \in \{0, 1\}$$

### 5.3. Grafo ponderado con quince nodos y veinticinco aristas.

En este apartado se presenta el último ejemplo, disponible en la Figura 19. Se presentan veinticinco aristas y quince nodos, de los cuales cuatro son nodos terminales  $N = \{1, 5, 13, 15\}$ .

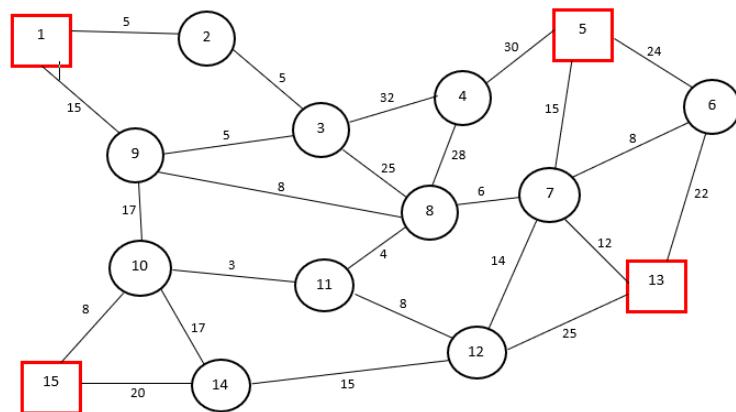


Figura 19: Ejemplo de grafo ponderado con quince nodos y veinticinco aristas.



En la Tabla 3 se muestran los correspondientes costos fijos para cada arco:

Tabla 3: Arcos y sus correspondientes costos fijos para el grafo ponderado con quince nodos y veinticinco aristas.

ARCOS		COSTO FIJO	ARCOS		COSTO FIJO
(1, 2)	(2, 1)	5	(7, 13)	(13, 7)	12
(2, 3)	(3, 2)	5	(5, 7)	(7, 5)	15
(3, 4)	(4, 3)	32	(7, 12)	(12, 7)	14
(4, 5)	(5, 4)	30	(4, 8)	(8, 4)	28
(5, 6)	(6, 5)	24	(8, 11)	(11, 8)	4
(6, 7)	(7, 6)	8	(3, 8)	(8, 3)	25
(7, 8)	(8, 7)	6	(9, 3)	(3, 9)	5
(8, 9)	(9, 8)	8	(1, 9)	(9, 1)	15
(9, 10)	(10, 9)	17	(14, 12)	(12, 14)	15
(10, 11)	(11, 10)	3	(10, 15)	(15, 10)	8
(11, 12)	(12, 11)	8	(10, 14)	(14, 10)	17
(12, 13)	(13, 12)	25	(14, 15)	(15, 14)	20
(13, 6)	(6, 13)	22			

La solución propuesta sin usar el programa tiene un peso total de 71 con 5 nodos de Steiner, se puede ver en la Figura 20.

Resultado de la solución manual:  
**Suma de pesos = 71**  
**5 nodos de Steiner**

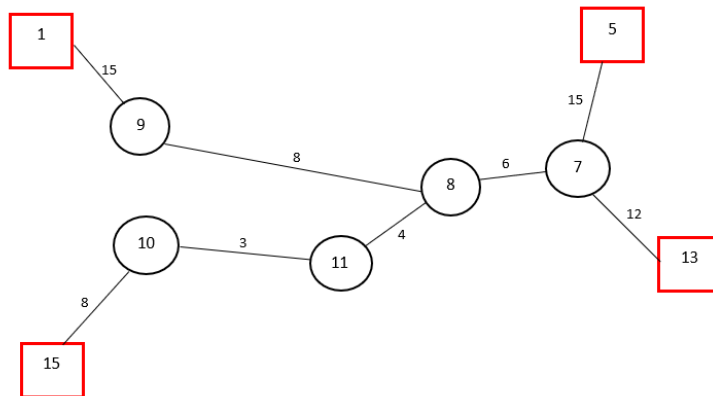


Figura 20: Solución manual propuesta para el grafo ponderado con quince nodos y veinticinco aristas.

A continuación, se muestra la solución óptima conseguida con software:

Resultado de la solución con software:

**Suma de pesos = 71**

**7 nodos de Steiner**

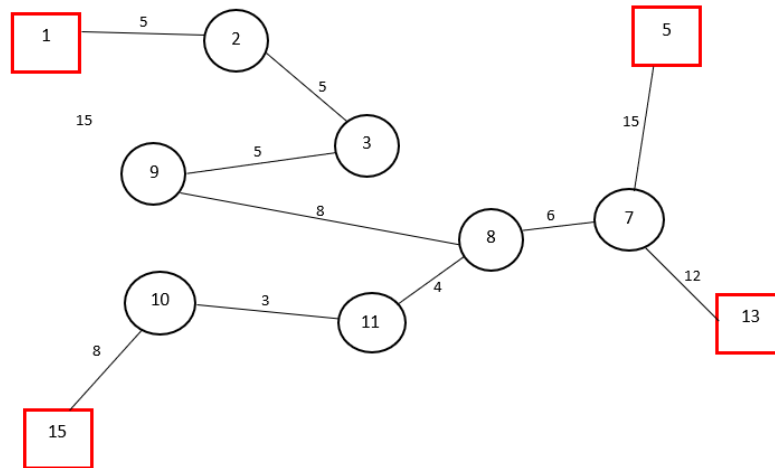


Figura 21: Solución óptima para el grafo ponderado con quince nodos y veinticinco aristas obtenida con el programa Xpress Mosel.

Esta vez, la solución óptima obtenida con software no va a mejorar la solución propuesta hecha a mano, el peso total es también 71 pero esta vez con 7 nodos de Steiner. De forma que ya se había obtenido o alcanzado la solución óptima con el procedimiento manual.

Como se ha hecho para los grafos anteriores, se presenta la función objetivo necesaria para obtener la solución óptima de la Figura 19 y las correspondientes restricciones.

Función objetivo para el grafo de la Figura 19:

$$\begin{aligned}
 \min & 5 y_{1,2} + 15 y_{1,9} + 5 y_{2,1} + 5 y_{2,3} + 5 y_{3,2} + 32 y_{3,4} + 25 y_{3,8} \\
 & + 5 y_{3,9} + 32 y_{4,3} + 30 y_{4,5} + 28 y_{4,8} + 30 y_{5,4} + 24 y_{5,6} \\
 & + 15 y_{5,7} + 24 y_{6,5} + 8 y_{6,7} + 22 y_{6,13} + 15 y_{7,5} + 8 y_{7,6} \\
 & + 6 y_{7,8} + 14 y_{7,12} + 12 y_{7,13} + 25 y_{8,3} + 28 y_{8,4} + 6 y_{8,7} \\
 & + 8 y_{8,9} + 4 y_{8,11} + 15 y_{9,1} + 5 y_{9,3} + 8 y_{9,8} + 17 y_{9,10} \\
 & + 17 y_{10,9} + 3 y_{10,11} + 17 y_{10,14} + 8 y_{10,15} + 4 y_{11,8} \\
 & + 3 y_{11,10} + 8 y_{11,12} + 14 y_{12,7} + 8 y_{12,11} + 25 y_{12,13} \\
 & + 15 y_{12,14} + 22 y_{13,6} + 12 y_{13,7} + 25 y_{13,12} + 17 y_{14,10} \\
 & + 15 y_{14,12} + 20 y_{14,15} + 8 y_{15,10} + 20 y_{15,14}
 \end{aligned}$$

Restricciones para el grafo de la Figura 19:

*sujeto a:*

1. Ecuaciones de balance:

$$x_{1,2} + x_{1,9} - x_{2,1} - x_{9,1} = 3$$

$$-x_{1,2} + x_{2,1} + x_{2,3} - x_{3,2} = 0$$

$$-x_{2,3} + x_{3,2} + x_{3,4} + x_{3,8} + x_{3,9} - x_{4,3} - x_{8,3} - x_{9,3} = 0$$

$$-x_{3,4} + x_{4,3} + x_{4,5} + x_{4,8} - x_{5,4} - x_{8,4} = 0$$

$$-x_{4,5} + x_{5,4} + x_{5,6} + x_{5,7} - x_{6,5} - x_{7,5} = -1$$

$$-x_{5,6} + x_{6,5} + x_{6,7} + x_{6,13} - x_{7,6} - x_{13,6} = 0$$

$$-x_{5,7} - x_{6,7} + x_{7,5} + x_{7,6} + x_{7,8} + x_{7,12} + x_{7,13} - x_{8,7} - x_{12,7} - x_{13,7} = 0$$

$$-x_{3,8} - x_{4,8} - x_{7,8} + x_{8,3} + x_{8,4} + x_{8,7} + x_{8,9} + x_{8,11} - x_{9,8} - x_{11,8} = 0$$

$$-x_{1,9} - x_{3,9} - x_{8,9} + x_{9,1} + x_{9,3} + x_{9,8} + x_{9,10} - x_{10,9} = 0$$

$$-x_{9,10} + x_{10,9} + x_{10,11} + x_{10,14} + x_{10,15} - x_{11,10} - x_{14,10} - x_{15,10} = 0$$

$$-x_{8,11} - x_{10,11} + x_{11,8} + x_{11,10} + x_{11,12} - x_{12,11} = 0$$

$$-x_{7,12} - x_{11,12} + x_{12,7} + x_{12,11} + x_{12,13} + x_{12,14} - x_{13,12} - x_{14,12} = 0$$

$$-x_{6,13} - x_{7,13} - x_{12,13} + x_{13,6} + x_{13,7} + x_{13,12} = -1$$

$$-x_{10,14} - x_{12,14} + x_{14,10} + x_{14,12} + x_{14,15} - x_{15,14} = 0$$

$$-x_{10,15} - x_{14,15} + x_{15,10} + x_{15,14} = -1$$

2. Cotas:

$$x_{15,14} - 3y_{15,14} \leq 0$$

$$x_{15,10} - 3y_{15,10} \leq 0$$

$$x_{14,15} - 3y_{14,15} \leq 0$$

$$x_{14,12} - 3y_{14,12} \leq 0$$

$$\begin{aligned}
x_{14,10} - 3 y_{14,10} &\leq 0 \\
x_{13,12} - 3 y_{13,12} &\leq 0 \\
x_{13,7} - 3 y_{13,7} &\leq 0 \\
x_{13,6} - 3 y_{13,6} &\leq 0 \\
x_{12,14} - 3 y_{12,14} &\leq 0 \\
x_{12,13} - 3 y_{12,13} &\leq 0 \\
x_{12,11} - 3 y_{12,11} &\leq 0 \\
x_{12,7} - 3 y_{12,7} &\leq 0 \\
x_{11,12} - 3 y_{11,12} &\leq 0 \\
x_{11,10} - 3 y_{11,10} &\leq 0 \\
x_{11,8} - 3 y_{11,8} &\leq 0 \\
x_{10,15} - 3 y_{10,15} &\leq 0 \\
x_{10,14} - 3 y_{10,14} &\leq 0 \\
x_{10,11} - 3 y_{10,11} &\leq 0 \\
x_{10,9} - 3 y_{10,9} &\leq 0 \\
x_{9,10} - 3 y_{9,10} &\leq 0 \\
x_{9,8} - 3 y_{9,8} &\leq 0 \\
x_{9,3} - 3 y_{9,3} &\leq 0 \\
x_{9,1} - 3 y_{9,1} &\leq 0 \\
x_{8,11} - 3 y_{8,11} &\leq 0 \\
x_{8,9} - 3 y_{8,9} &\leq 0 \\
x_{8,7} - 3 y_{8,7} &\leq 0 \\
x_{8,4} - 3 y_{8,4} &\leq 0 \\
x_{8,3} - 3 y_{8,3} &\leq 0 \\
x_{7,13} - 3 y_{7,13} &\leq 0
\end{aligned}$$

$$x_{7,12} - 3 y_{7,12} \leq 0$$

$$x_{7,8} - 3 y_{7,8} \leq 0$$

$$x_{7,6} - 3 y_{7,6} \leq 0$$

$$x_{7,5} - 3 y_{7,5} \leq 0$$

$$x_{6,13} - 3 y_{6,13} \leq 0$$

$$x_{6,7} - 3 y_{6,7} \leq 0$$

$$x_{6,5} - 3 y_{6,5} \leq 0$$

$$x_{5,7} - 3 y_{5,7} \leq 0$$

$$x_{5,6} - 3 y_{5,6} \leq 0$$

$$x_{5,4} - 3 y_{5,4} \leq 0$$

$$x_{4,8} - 3 y_{4,8} \leq 0$$

$$x_{4,5} - 3 y_{4,5} \leq 0$$

$$x_{4,3} - 3 y_{4,3} \leq 0$$

$$x_{3,9} - 3 y_{3,9} \leq 0$$

$$x_{3,8} - 3 y_{3,8} \leq 0$$

$$x_{3,4} - 3 y_{3,4} \leq 0$$

$$x_{3,2} - 3 y_{3,2} \leq 0$$

$$x_{2,3} - 3 y_{2,3} \leq 0$$

$$x_{2,1} - 3 y_{2,1} \leq 0$$

$$x_{1,9} - 3 y_{1,9} \leq 0$$

$$x_{1,2} - 3 y_{1,2} \leq 0$$

3. No negatividad de las variables de flujo:

$$x_{1,2}, x_{1,9}, x_{2,1}, \dots, x_{15,14} \geq 0$$

4. Variable ( $y$ ) binaria:

$$y_{1,2}, y_{1,9}, y_{2,1}, \dots, y_{15,14} \in \{0, 1\}$$

## 6. EL PROBLEMA DE STEINER EUCLÍDEO

En el siguiente paso se resuelve el problema de Steiner buscando una red de longitud total mínima (árbol mínimo expandido) que interconecte un conjunto finito dado de  $P$  puntos en el plano euclidiano (Brazil, 2014). Como se va a trabajar con  $n = 2$  dimensiones del espacio euclídeo, se hablará de aquí en adelante del plano euclídeo. El objetivo vuelve a ser conectar un número de puntos de tal forma que la longitud total de las líneas que los unen sea mínima.

### 6.1. El problema de Steiner con distancias euclídeas.

El problema del árbol mínimo euclídeo de Steiner (“Minimum Euclidean Steiner Tree”) trata de resolver el problema de Steiner en espacios métricos con distancias euclídeas.

La distancia euclídea entre dos puntos  $x$  e  $y$  en un espacio euclídeo es la longitud de la línea recta que los une. Si  $x = (x_1, x_2, \dots, x_n)$  y  $y = (y_1, y_2, \dots, y_n)$ , entonces la distancia euclídea entre  $x$  e  $y$  es dada por la expresión que sigue (Soothill, 2010):

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

En nuestro caso vamos a considerar distancias euclídeas cuando  $n = 2$ , por tanto, la distancia vendrá dada por:  $d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$ . Por ejemplo, se tiene más adelante en la Figura 22 dos nodos,  $P1$  y  $P2$ ,  $P1$  situado en el punto  $(2, 3)$  y  $P2$  en el  $(8, 3)$ , la distancia euclídea será:  $d(P1, P2) = \sqrt{(2 - 8)^2 + (3 - 3)^2} = 6$ .

El problema consistirá en resolver el problema de Steiner para un conjunto  $P$  puntos que son coplanares en el plano euclídeo. Al principio, no existen arcos de unión entre los nodos terminales por lo que pueden surgir dos escenarios que se describen a continuación.

6.1.1. Escenario uno: Las líneas que conectan los nodos terminales sólo pueden tocar los propios nodos del conjunto P.

El problema se convierte ahora en la búsqueda de un árbol mínimo expandido con  $P$  puntos o nodos y  $A$  aristas, las cuales tendrán distancias euclídeas.

Se van a presentar varios ejemplos que van creciendo en dificultad. Primero se van a resolver de forma manual, que será de forma visual y aproximada en ocasiones ante la gran cantidad de nodos y distancias a medir. Posteriormente se hará uso del software específico Xpress Mosel programando el algoritmo de Prim para obtener la solución óptima.

#### 6.1.1.1. Ejemplo 0.

En la Figura 22 se presenta un ejemplo muy básico para presentar el proceso de forma clara. El ejemplo consta de cuatro puntos (nodos), cada uno tiene sus respectivas coordenadas. El objetivo del problema es conseguir el árbol mínimo de expansión que conecte los cuatro puntos.

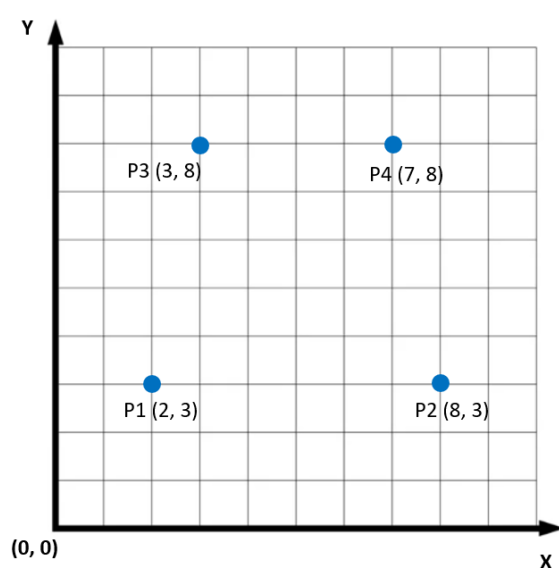


Figura 22: Enunciado del ejemplo 0 en el apartado 6.1.1. con cuatro puntos en el espacio euclídeo.

En las siguientes tablas se muestran las coordenadas de los puntos y las distancias entre ellos:

Tabla 4: Coordenadas de los nodos del ejemplo 0 del apartado 6.1.1.

Coordenadas		
Puntos	Coordenada X	Coordenada Y
P1	2	3
P2	8	3
P3	3	8
P4	7	8

Tabla 5: Distancia entre nodos del ejemplo 0 del apartado 6.1.1.

Distancia entre nodos	
P1-P2	6
P1-P3	5,01
P1-P4	7,07
P2-P3	7,07
P2-P4	5,01
P3-P4	4

Para obtener una solución, que consistirá en un grafo, se calculan las distancias euclídeas entre los nodos más cercanos. Además, se obtiene el árbol mínimo expandido, con la ayuda de algoritmos como pueden ser los de Kruskal o Prim. Si el grafo tiene por ejemplo  $P = 4$  nodos, el árbol tendrá  $P - 1 = 3$  aristas.

En la Figura 23 se encuentra la solución con el árbol mínimo expandido para la Figura 22, esta solución se ha elaborado de forma manual. Las distancias euclídeas entre nodos del grafo se muestran en color rojo, éstas se han calculado usando la formula correspondiente explicada en el comienzo del apartado 6.1. La longitud total de las aristas suma  $l = 14,20 u. m$  (unidades de medida).



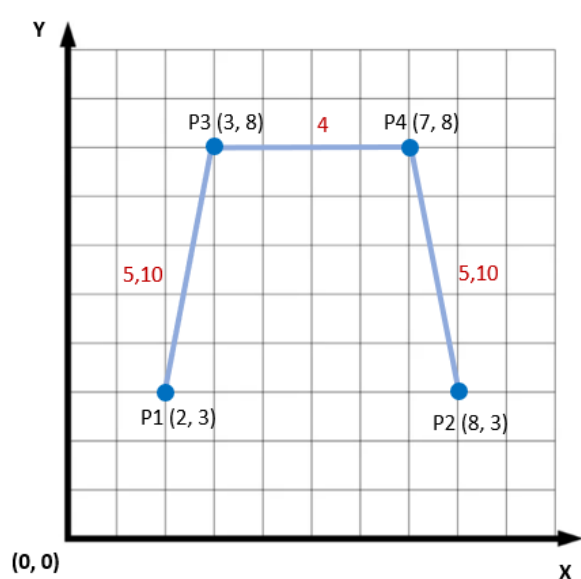


Figura 23: Solución manual para el ejemplo 0 en el apartado 6.1.1. con cuatro puntos en el espacio euclídeo.

Por último, en la Figura 24 se presenta la solución óptima obtenida con software para el mismo enunciado. Al final del ejemplo se compararán la solución manual y la obtenida con software. También se muestran los diferentes tramos con los que está formado el grafo.

Tramos (aristas) del árbol mínimo de expansión obtenido:

Tabla 6: Tramos de la solución obtenida con software para el ejemplo 0 del apartado 6.1.1.

Tramos de la solución		
Tramos (aristas)	Punto de origen	Punto de destino
Tramo 1	1	3
Tramo 2	3	4
Tramo 3	4	2

Distancia total de la solución óptima obtenida con software:  $l = 14,20 \text{ u. m.}$

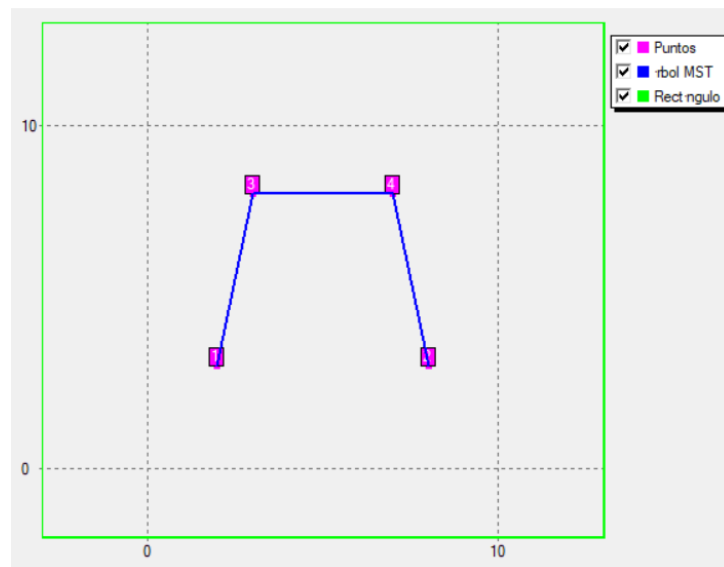


Figura 24: Solución del software para el ejemplo 0 en el apartado 6.1.1. con cuatro puntos en el espacio euclídeo.

En este ejemplo, al ser tan sencillo, las soluciones manual y óptima tienen la misma longitud total y cuentan con los mismos tramos. En ejemplos más complejos, ambas soluciones podrán ser diferentes ya que no se podrá calcular todas las distancias entre nodos al haber un gran número de nodos.

#### 6.1.1.2. Ejemplo 1.

Para el ejemplo 1 se va a subir la dificultad, esta vez se presenta en la Figura 25 y tablas 7 y 8 un conjunto de diez nodos terminales con sus coordenadas y respectivas distancias entre sí.

Tabla 7: Coordenadas de los nodos del ejemplo 1 del apartado 6.1.1.

Coordenadas		
Puntos	Coordenada X	Coordenada Y
P1	2	3
P2	3	1
P3	7	1
P4	6	4
P5	3	5
P6	4	6
P7	8	5
P8	7	7
P9	1	7
P10	4	9

Tabla 8: Distancia entre nodos del ejemplo 1 del apartado 6.1.1.

Distancia entre nodos			
P1-P2	2,24	P4-P5	3,16
P1-P3	5,39	P4-P6	2,83
P1-P4	4,12	P4-P7	2,24
P1-P5	2,24	P4-P8	3,16
P1-P6	3,61	P4-P9	5,83
P1-P7	6,32	P4-P10	5,39
P1-P8	6,4	P5-P6	1,41
P1-P9	4,12	P5-P7	5
P1-P10	6,32	P5-P8	4,47
P2-P3	4	P5-P9	2,83
P2-P4	4,24	P5-P10	4,12
P2-P5	4	P6-P7	4,12
P2-P6	5,1	P6-P8	3,16
P2-P7	6,4	P6-P9	3,16
P2-P8	7,21	P6-P10	3
P2-P9	6,32	P7-P8	2,24
P2-P10	8,06	P7-P9	7,28
P3-P4	3,16	P7-P10	5,66
P3-P5	5,66	P8-P9	6
P3-P6	5,83	P8-P10	3,61
P3-P7	4,12	P9-P10	3,61
P3-P8	6		
P3-P9	8,49		
P3-P10	8,54		

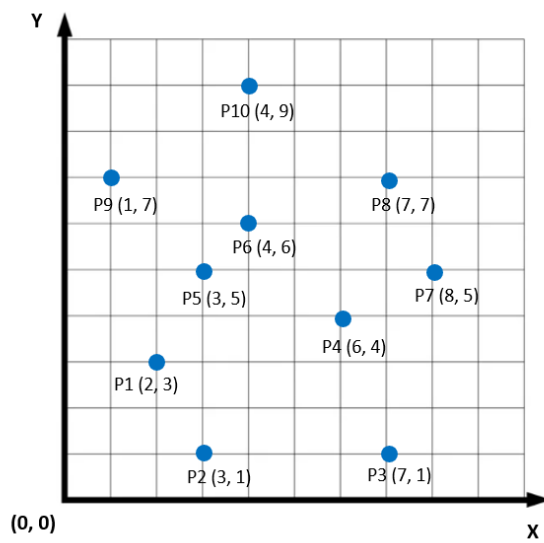


Figura 25: Enunciado del ejemplo 1 en el apartado 6.1.1. con diez puntos en el espacio euclídeo.

Como en el ejemplo anterior, se presenta el árbol mínimo de expansión calculado de forma manual. Las distancias euclídeas entre nodos del grafo se muestran en color rojo. La distancia total de la solución es  $l = 24,2 \text{ u.m.}$

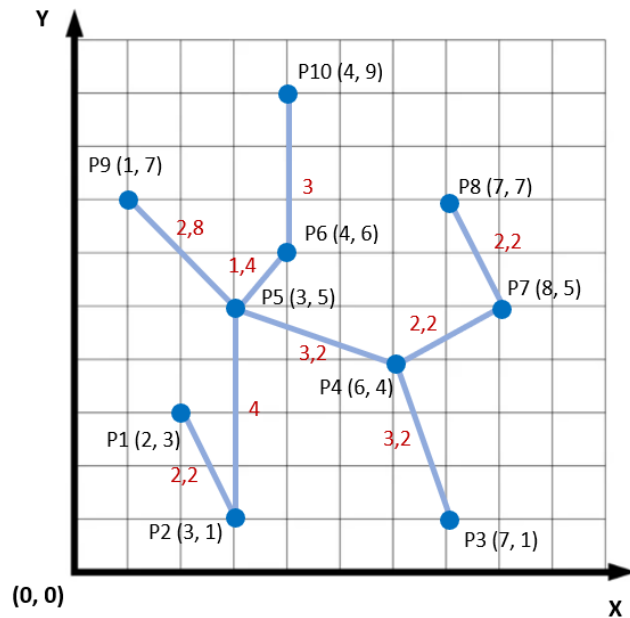


Figura 26: Solución manual para el ejemplo 1 del apartado 6.1.1. con diez puntos en el espacio euclídeo.

Por último, los tramos y el grafo de la solución óptima obtenida con software:

Tabla 9: Tramos de la solución obtenida con software para el ejemplo 1 del apartado 6.1.1.

Tramos de la solución		
Tramos (aristas)	Punto de origen	Punto de destino
Tramo 1	1	2
Tramo 2	1	5
Tramo 3	4	3
Tramo 4	4	7
Tramo 5	5	6
Tramo 6	5	9
Tramo 7	6	4
Tramo 8	6	10
Tramo 9	7	8

Distancia total de la solución óptima obtenida con software:  $l = 22,2 \text{ u. m.}$

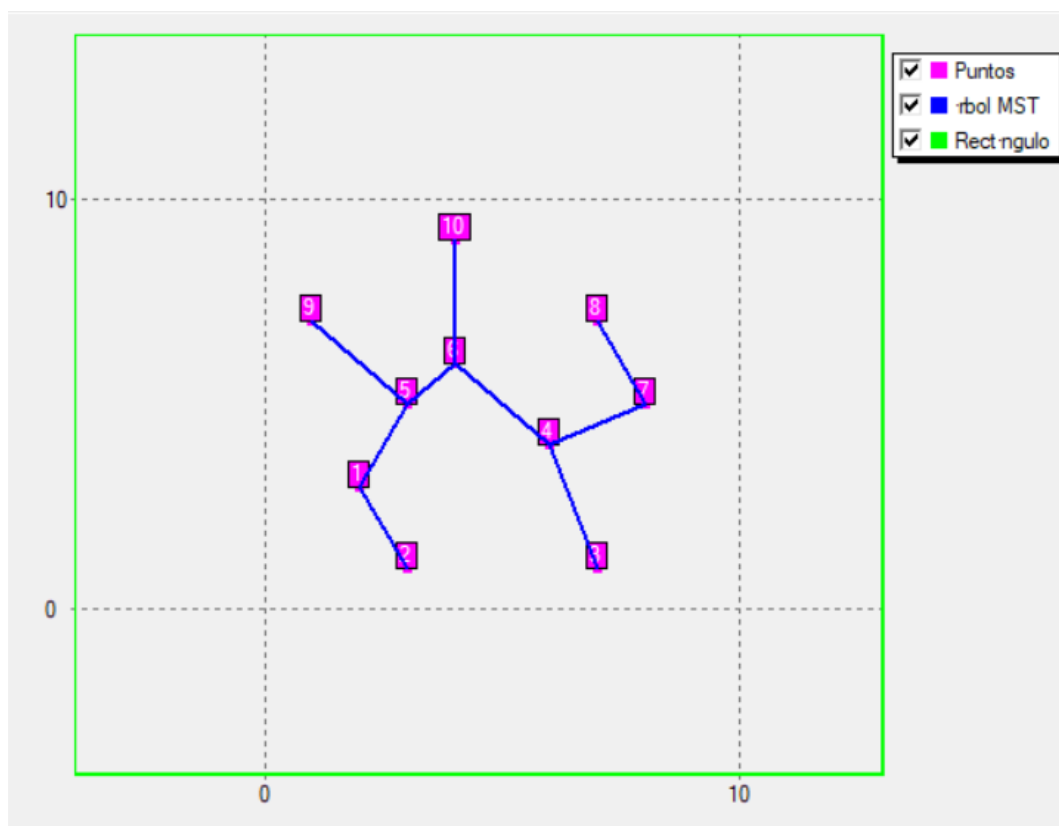


Figura 27: Solución del software para el ejemplo 1 del apartado 6.1.1. con diez puntos en el espacio euclídeo.

Comparando ambas soluciones, los tramos 2 (el cual une los nodos 1 y 5) y 7 (el cual une los nodos 6 y 4) de la solución óptima no están en la solución manual. La elección de dos tramos no óptimos en la solución manual causa una diferencia en distancia total de  $24,2 - 22,2 = 2 u.m$  con la solución óptima.

### 6.1.1.3. Ejemplo 2.

Como último ejemplo, en la figura siguiente se muestra un caso más complejo, esta vez el grafo dispone de veinticuatro nodos. A continuación, se pueden ver los diferentes nodos con sus respectivas coordenadas en la Tabla 10.

Tabla 10: Coordenadas de los nodos del ejemplo 2 del apartado 6.1.1.

Coordenadas		
Puntos	Coordenada X	Coordenada Y
P1	4	4
P2	2	8
P3	6	10
P4	4	16
P5	10	20
P6	4	24
P7	10	34
P8	12	28
P9	14	24
P10	18	20
P11	16	14
P12	16	5
P13	28	6
P14	26	10
P15	20	16
P16	26	16
P17	30	12
P18	36	16
P19	30	20
P20	35	24
P21	25	23
P22	30	28
P23	22	30
P24	28	36

En la Tabla 11 se muestran algunas distancias entre nodos. Se omiten la mayoría ya que, al haber veinticuatro nodos, habría que calcular demasiadas distancias entre nodos como para mostrarlas en una tabla.

Tabla 11: Distancia entre nodos del ejemplo 2 del apartado 6.1.1.

Distancia entre nodos	
P1-P2	4.47
P1-P3	6.32
P1-P4	12
P1-P5	17.09
P1-P6	20
P1-P7	30.59
P1-P8	25.3
P1-P9	22.36
P1-P10	21.26
P1-P11	15.62
P1-P12	12.04
...	...
P20-P21	10.05
P20-P22	6.4
P20-P23	14.32
P20-P24	13.89
P21-P22	7.07
P21-P23	7.62
P21-P24	13.34
P22-P23	8.25
P22-P24	8.25
P23-P24	8.49

En la Figura 28 se muestran los diferentes nodos en el espacio euclídeo con sus respectivas coordenadas.

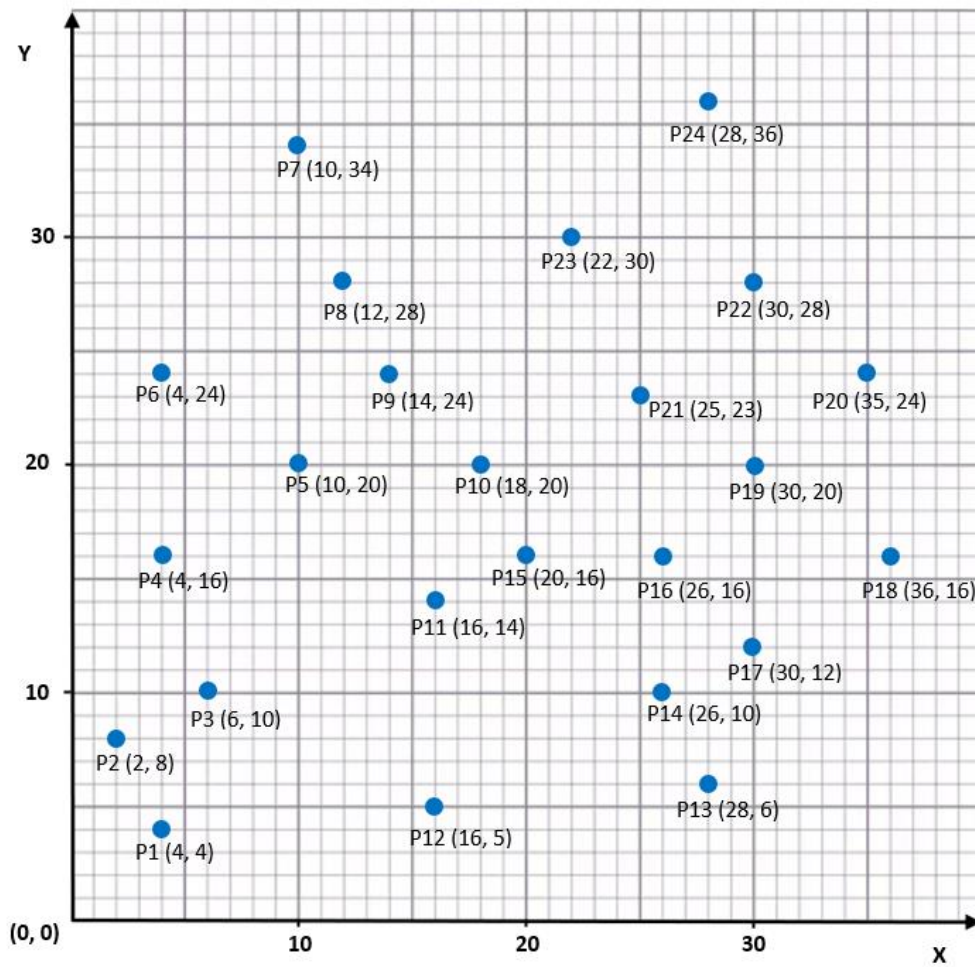


Figura 28: Enunciado del ejemplo 2 en el apartado 6.1.1. con veinticuatro puntos en el espacio euclídeo.

A continuación, se muestra el árbol mínimo expandido como solución al problema de Steiner euclídeo que se plantea en la figura anterior. El árbol es una posible solución manual. Las distancias euclídeas entre nodos del grafo se muestran en color rojo. La longitud total de las aristas suma  $l = 141 u. m.$



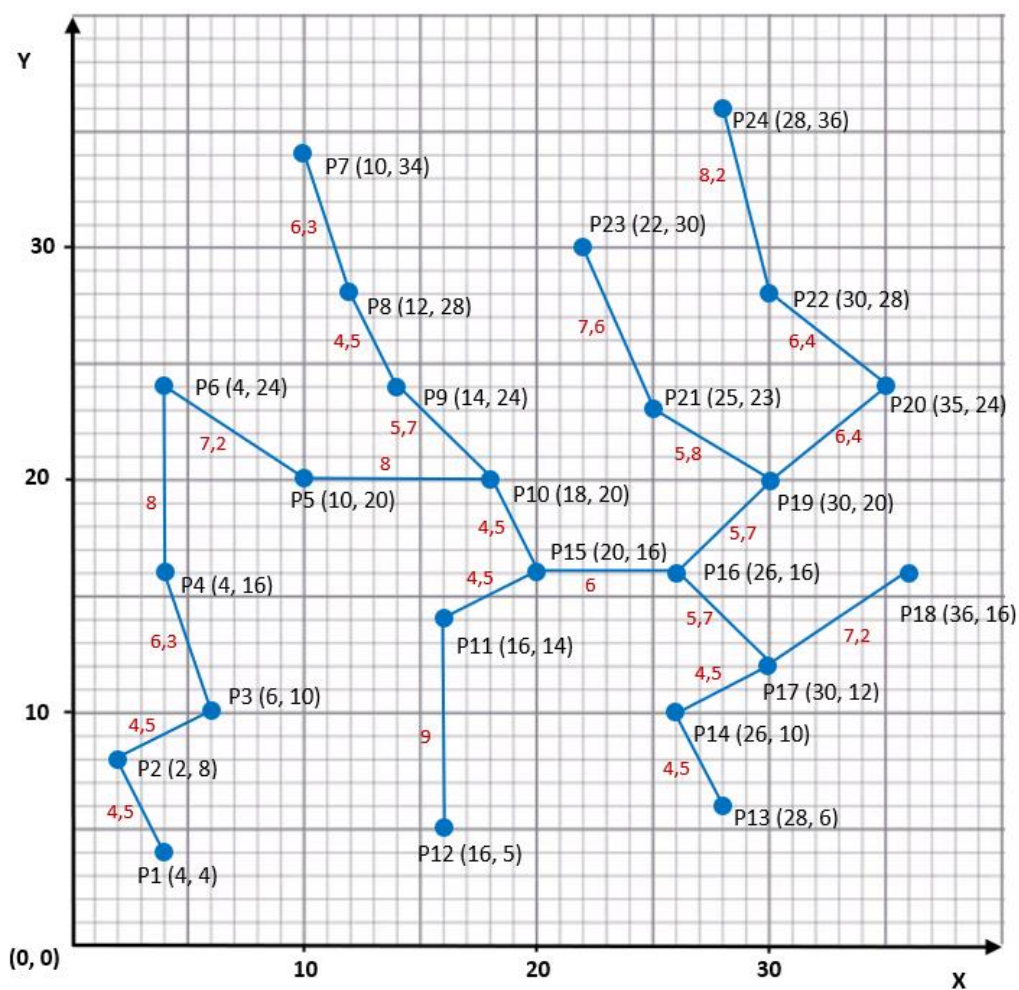


Figura 29: Solución manual para el ejemplo 2 del apartado 6.1.1. con veinticuatro puntos en el espacio euclídeo.

Por último, los tramos y el grafo de la solución óptima obtenida con software en la Tabla 12 y Figura 30 respectivamente.

Tabla 12: Tramos de la solución obtenida con software para el ejemplo 2 del apartado 6.1.1.

Tramos de la solución		
Tramos (aristas)	Punto de origen	Punto de destino
Tramo 1	1	2
Tramo 2	2	3
Tramo 3	3	4
Tramo 4	4	5
Tramo 5	5	6
Tramo 6	5	9
Tramo 7	8	7
Tramo 8	9	8
Tramo 9	9	10
Tramo 10	10	15
Tramo 11	11	12
Tramo 12	14	13
Tramo 13	15	11
Tramo 14	15	16
Tramo 15	16	17
Tramo 16	16	19
Tramo 17	17	14
Tramo 18	17	18
Tramo 19	19	20
Tramo 20	19	21
Tramo 21	20	22
Tramo 22	21	23
Tramo 23	22	24

Distancia total de la solución óptima obtenida con software:  $l = 137,7$  u. m.

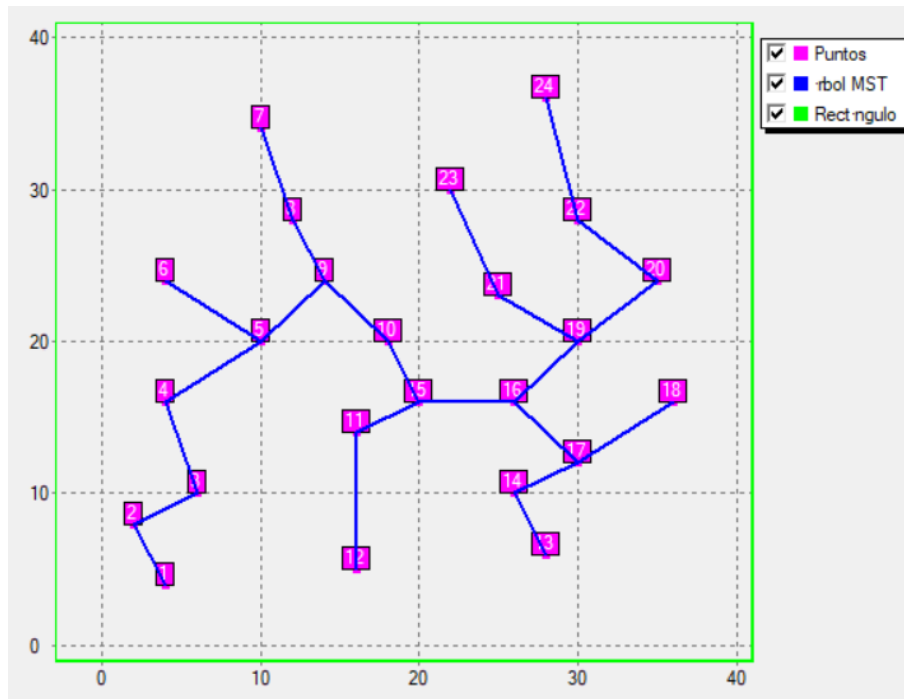


Figura 30: Solución del software para el ejemplo 2 del apartado 6.1.1. con veinticuatro puntos en el espacio euclídeo.

Como en el ejemplo 1, la solución óptima obtenida con software tiene una distancia total más baja ya que en la solución manual se han escogido tramos no óptimos para formar el árbol mínimo de expansión. Tramos 4 y 6 Concretamente los tramos 4 (el cual une los nodos 4 y 5) y 6 (el cual une los nodos 5 y 9) de la solución óptima no están en la solución manual. La diferencia de longitud total de la solución manual con la solución óptima es  $141 - 137,7 = 3,3 \text{ u. m.}$

#### 6.1.2. Escenario dos: Las líneas se pueden cortar con nodos que no están en el conjunto P.

Estos puntos distintos del conjunto P, el cual contiene los nodos terminales, son llamados nodos de Steiner y otorgarán más flexibilidad al problema consiguiendo mejores resultados. El resultado del problema será un **árbol mínimo de Steiner** que conectará un conjunto  $P' \supset P$ . Donde P será el conjunto que componen los nodos terminales iniciales y  $P'$  estará compuesto por los nodos terminales y nodos de Steiner.

Las nuevas soluciones, utilizando los nodos de Steiner, permitirán conseguir una reducción en la longitud total de las aristas, esto se debe al uso del punto de Fermat a la hora de posicionar un punto de Steiner ya que minimiza las distancias con otros nodos terminales con los que forma triángulos. Para poder resolver el problema euclidiano de Steiner y efectuar una minimización óptima de los árboles de recubrimiento se deben cumplir las siguientes condiciones (López, 2005):

- Los puntos de Steiner agregados deben tener grado tres.
- Las aristas incidentes en éstos deben formar ángulos de 120 grados o más.
- No deben existir cruces de aristas.

Con estas premisas, se puede deducir que un árbol de Steiner puede tener un máximo de  $P - 2$  nodos de Steiner, donde P es el número de nodos terminales.

La resolución de los siguientes ejemplos se va a hacer de forma manual y usando el software como en el apartado 6.1.1. A continuación, se van a introducir nodos de Steiner de forma manual donde sea posible y se va a comparar el resultado con el de otras soluciones. El objetivo principal es demostrar que efectivamente la inclusión de nodos de Steiner acorta la distancia total de las soluciones sin estos nodos.

### 6.1.2.1. Ejemplo 0.

La siguiente figura va a mostrar el mismo caso 0 del apartado 6.1.1 pero esta vez usando nodos de Steiner. Este ejemplo básico ayudará a ver de forma sencilla y directa el efecto de la inclusión de los nodos de Steiner.

En este ejemplo, el conjunto  $P$  lo forman 4 nodos terminales, los mismos que para el ejemplo 0 del apartado 6.1.1.

El conjunto  $P'$  lo forman 6 nodos. Este conjunto estará formado por los cuatro nodos terminales del conjunto  $P$  y la incorporación de dos nodos de Steiner ( $S1$  y  $S2$ ) con coordenadas:  $S1 (5, 7)$  y  $S2 (5, 4)$ . Los nodos de Steiner son los que aparecen en la figura con un contorno amarillo.

En la siguiente figura se muestra el nuevo árbol de Steiner con el que se ha obtenido una longitud total de las aristas de  $l = 13,8 u.m.$  Las distancias euclídeas entre nodos del grafo se muestran en color rojo. Si comparamos este árbol de Steiner con el árbol mínimo de expansión que se obtuvo en el ejemplo 0 del apartado 6.1.1 la diferencia es de  $14,2 - 13,8 = 0,4 u.m.$  Esta reducción en la distancia total se debe a la inclusión de los dos nodos de Steiner.

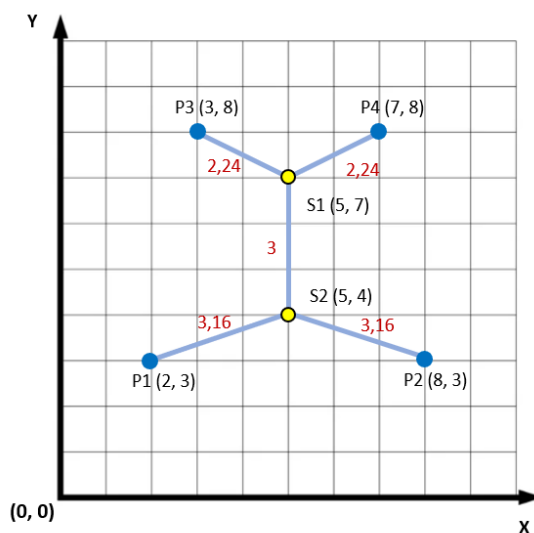


Figura 31: Árbol de Steiner derivado del ejemplo 0 con cuatro nodos en el espacio euclídeo del apartado 6.1.1.

### 6.1.2.2. Ejemplo 1.

A continuación, se muestra una figura con un conjunto de diez nodos terminales para los que se va a obtener el árbol de expansión mínimo euclídeo (EMST: *Euclidean Minimum Spanning Tree*) y el árbol de Steiner, ambos de forma manual, y por último, se van a comparar ambas soluciones.

Primero se muestran la Tabla 13 con las coordenadas:

Tabla 13: Coordenadas de los nodos del ejemplo 1 del apartado 6.1.2.

Coordenadas		
Puntos	Coordenada X	Coordenada Y
P1	2	1
P2	4	2
P3	5	6
P4	8	1
P5	9	7
P6	11	3
P7	14	4
P8	14	7
P9	17	5
P10	18	3

En la Tabla 14 se muestran las distancias entre nodos:

Tabla 14: Distancia entre nodos del ejemplo 1 del apartado 6.1.2.

Distancia entre nodos			
P1-P2	2.24	P4-P5	6.08
P1-P3	5.83	P4-P6	3.61
P1-P4	6	P4-P7	6.71
P1-P5	9.22	P4-P8	8.49
P1-P6	9.22	P4-P9	9.85
P1-P7	12.37	P4-P10	10.2
P1-P8	13.42	P5-P6	4.47
P1-P9	15.52	P5-P7	5.83
P1-P10	16.12	P5-P8	5
P2-P3	4.12	P5-P9	8.24
P2-P4	4.12	P5-P10	9.85
P2-P5	7.07	P6-P7	3.16
P2-P6	7.07	P6-P8	5
P2-P7	10.2	P6-P9	6.32
P2-P8	11.18	P6-P10	7
P2-P9	13.34	P7-P8	3
P2-P10	14.04	P7-P9	3.16
P3-P4	5.83	P7-P10	4.12
P3-P5	4.12	P8-P9	3.61
P3-P6	6.71	P8-P10	5.66
P3-P7	9.22	P9-P10	2.24
P3-P8	9.06		
P3-P9	12.04		
P3-P10	13.34		

En la Figura 32 se muestra la disposición de los diez nodos en el espacio euclídeo:

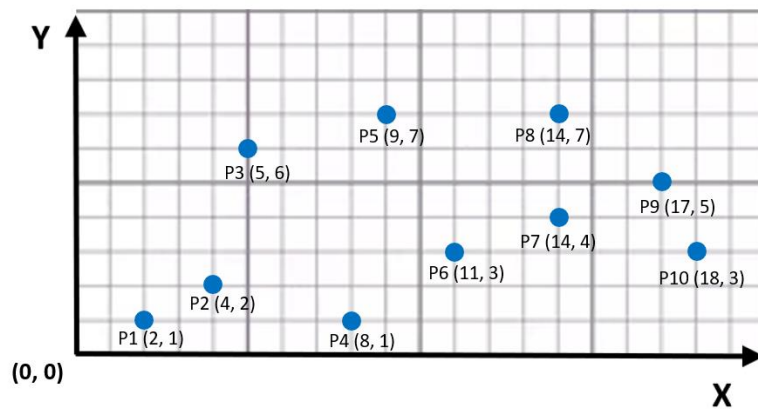


Figura 32: Enunciado del ejemplo 1 en el apartado 6.1.2. con diez puntos en el espacio euclídeo.

Seguidamente se muestra la Figura 33 con el árbol mínimo de expansión (parte superior) y el árbol de Steiner (parte inferior) para el conjunto de diez nodos mostrado anteriormente. Las distancias euclídeas entre nodos del grafo se muestran en color rojo.

La distancia total de las aristas del árbol mínimo de expansión de la parte superior es de 30,5 u. m. mientras que la distancia total del árbol de Steiner de la parte inferior es de 29 u.m. Como en el ejemplo anterior, se reduce la distancia total añadiendo nodos de Steiner: nodos S1 con coordenadas (6, 4), S2 (8, 4), S3 (9, 3) y S4 (15, 5). Esta vez la distancia que se ha podido recortar es de:  $30,5 - 29 = 1,5$  u. m.

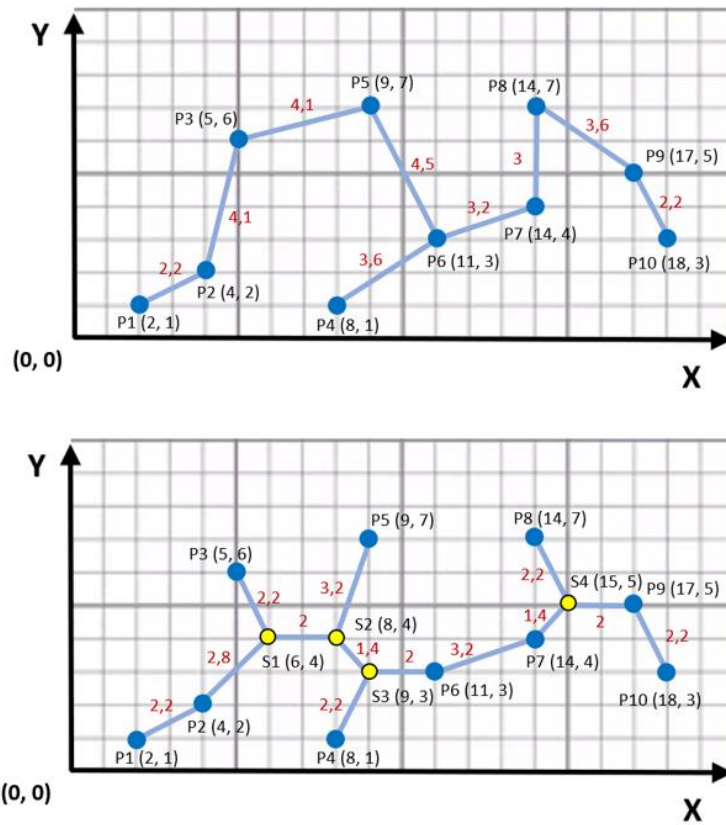


Figura 33: Solución manual para la obtención del árbol de expansión mínimo euclídeo (gráfico superior) y el árbol de Steiner (gráfico inferior) para el enunciado del ejemplo 1 del apartado 6.1.2.

Por último, se muestra en la Figura 34 la solución óptima obtenida con software para compararla con el árbol mínimo de expansión conseguido de forma manual en la figura anterior. En la Figura 15 se muestran los tramos de la solución.

La distancia total de la solución óptima obtenida con software es de:  $l = 29,8 \text{ u. m.}$

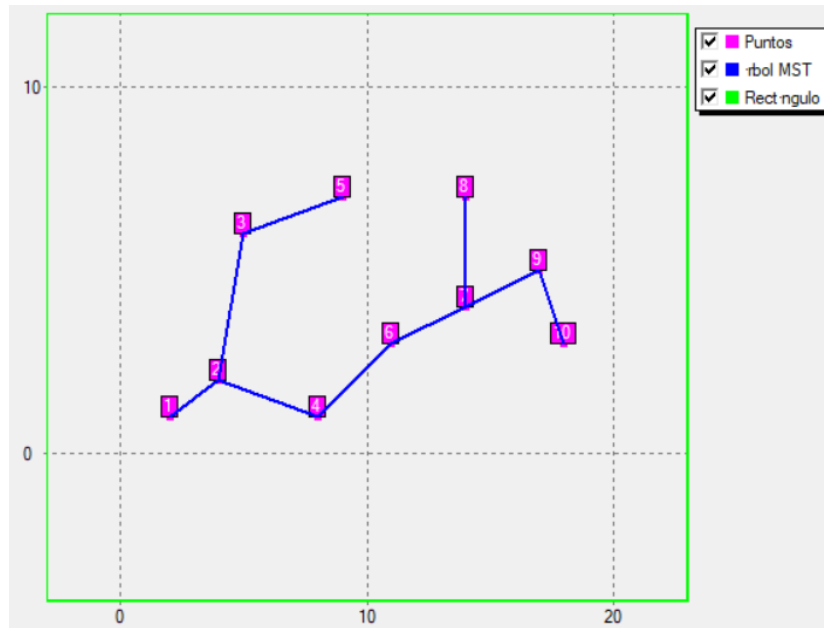


Figura 34: Solución del software para la obtención del árbol de expansión mínimo euclídeo en el ejemplo 1 del apartado 6.1.2. con diez puntos en el espacio euclídeo.

Tabla 15: Tramos de la solución obtenida con software para el ejemplo 1 del apartado 6.1.2.

Tramos de la solución		
Tramos (aristas)	Punto de origen	Punto de destino
Tramo 1	1	2
Tramo 2	2	3
Tramo 3	2	4
Tramo 4	3	5
Tramo 5	4	6
Tramo 6	6	7
Tramo 7	7	8
Tramo 8	7	9
Tramo 9	9	10

La solución óptima obtenida con software tiene una distancia total más baja que la solución manual. Los tramos 3 (el cual une los nodos 2 y 4) y 8 (el cual une los nodos 7 y 9) de la solución óptima no están en la solución manual. La diferencia de longitud total de la solución manual con la solución óptima es  $30,5 - 29,8 = 60,3 \text{ u.m.}$

Comparando la solución óptima obtenida con software (sin usar nodos de Steiner) y el árbol de Steiner obtenido de forma manual, el árbol de Steiner tiene una distancia total menor de 29 u.m.



### 6.1.2.3. Ejemplo 2.

Por último, se muestra un ejemplo más complejo con un conjunto de quince nodos terminales para los que se va a obtener el árbol de expansión mínimo euclídeo (EMST) y el árbol de Steiner, ambos de forma manual. Después se van a comparar ambas soluciones.

Primero se muestran la tabla con las coordenadas de los nodos terminales (Tabla 16) y la tabla con las distancias entre nodos (Tabla 17):

Tabla 16: Coordenadas de los nodos del ejemplo 2 del apartado 6.1.2.

Coordenadas		
Puntos	Coordenada X	Coordenada Y
P1	3	7
P2	5	13
P3	8	15
P4	6	21
P5	9	20
P6	13	21
P7	12	17
P8	17	17
P9	21	13
P10	16	12
P11	12	11
P12	9	8
P13	19	7
P14	14	3
P15	7	3

Tabla 17: Distancia entre nodos del ejemplo 2 del apartado 6.1.2.

Distancia entre nodos					
P1-P2	6.32	P4-P5	3.16	P8-P9	5.66
P1-P3	9.43	P4-P6	7	P8-P10	5.1
P1-P4	14.32	P4-P7	7.21	P8-P11	7.81
P1-P5	14.32	P4-P8	11.7	P8-P12	12.04
P1-P6	17.2	P4-P9	17	P8-P13	10.2
P1-P7	13.45	P4-P10	13.45	P8-P14	14.32
P1-P8	17.2	P4-P11	11.66	P8-P15	17.2
P1-P9	18.97	P4-P12	13.34	P9-P10	5.1
P1-P10	13.93	P4-P13	19.1	P9-P11	9.22
P1-P11	9.85	P4-P14	19.7	P9-P12	13
P1-P12	6.08	P4-P15	18.03	P9-P13	6.32
P1-P13	16	P5-P6	4.12	P9-P14	12.21
P1-P14	11.7	P5-P7	4.24	P9-P15	17.2
P1-P15	5.66	P5-P8	8.54	P10-P11	4.12
P2-P3	3.61	P5-P9	13.89	P10-P12	8.06
P2-P4	8.06	P5-P10	10.63	P10-P13	5.83
P2-P5	8.06	P5-P11	9.49	P10-P14	9.22
P2-P6	11.31	P5-P12	12	P10-P15	12.73
P2-P7	8.06	P5-P13	16.4	P11-P12	4.24
P2-P8	12.65	P5-P14	17.72	P11-P13	8.06
P2-P9	16	P5-P15	17.12	P11-P14	8.25
P2-P10	11.05	P6-P7	4.12	P11-P15	9.43
P2-P11	7.28	P6-P8	5.66	P12-P13	10.05
P2-P12	6.4	P6-P9	11.31	P12-P14	7.07
P2-P13	15.23	P6-P10	9.49	P12-P15	5.39
P2-P14	13.45	P6-P11	10.05	P13-P14	6.4
P2-P15	10.2	P6-P12	13.6	P13-P15	12.65
P3-P4	6.32	P6-P13	15.23	P14-P15	7
P3-P5	5.1	P6-P14	18.03		
P3-P6	7.81	P6-P15	18.97		
P3-P7	4.47	P7-P8	5		
P3-P8	9.22	P7-P9	9.85		
P3-P9	13.15	P7-P10	6.4		
P3-P10	8.54	P7-P11	6		
P3-P11	5.66	P7-P12	9.49		
P3-P12	7.07	P7-P13	12.21		
P3-P13	13.6	P7-P14	14.14		
P3-P14	13.42	P7-P15	14.87		
P3-P15	12.04				

En la Figura 35 se muestra el gráfico con el posicionamiento de los nodos:

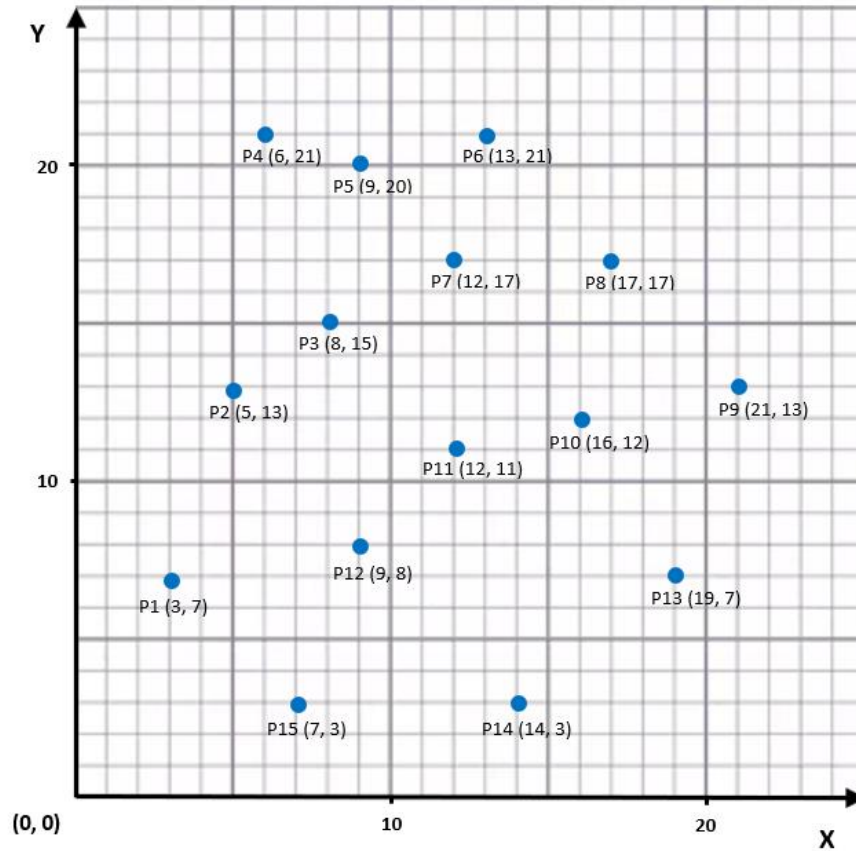


Figura 35: Enunciado del ejemplo 2 en el apartado 6.1.2. con quince puntos en el espacio euclídeo.

En las siguientes figuras se muestran primero el árbol mínimo euclídeo de expansión (Figura 36), y, segundo, el árbol mínimo de Steiner para el conjunto de nodos que se ha mostrado en la figura anterior (Figura 37). Las distancias euclídeas entre nodos del grafo se muestran en color rojo.

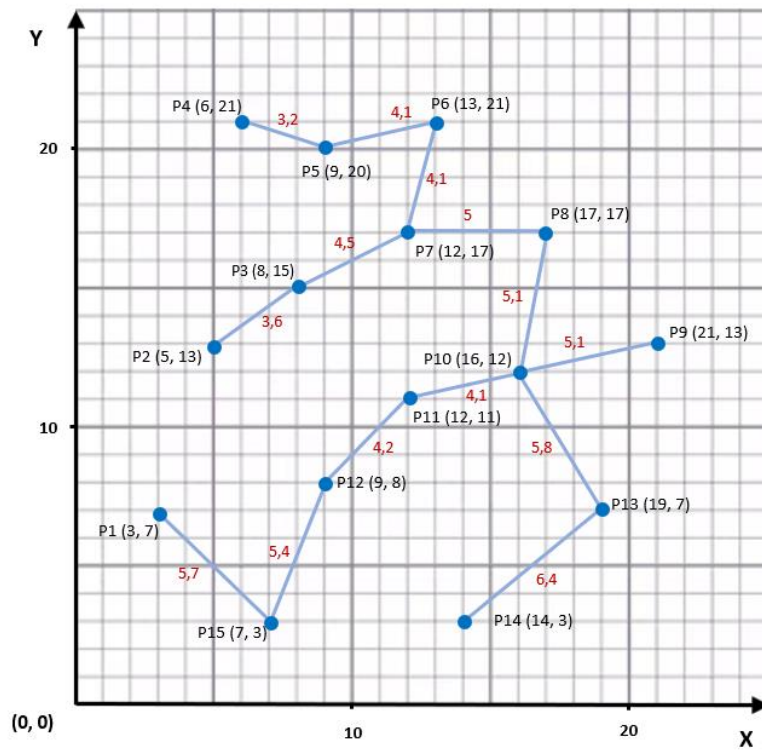


Figura 36: Solución manual para la obtención del árbol de expansión mínimo euclídeo en el ejemplo 2 del apartado 6.1.2.

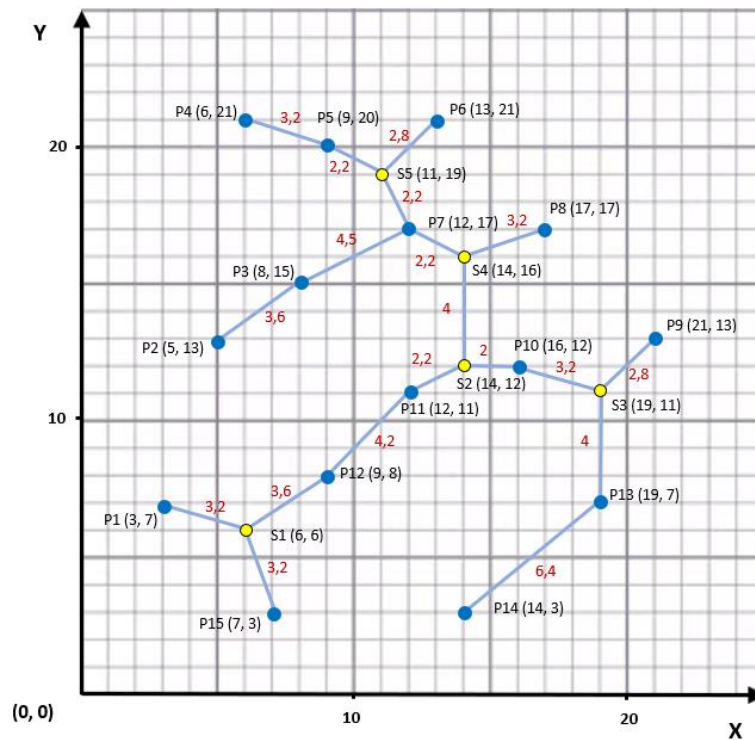


Figura 37: Solución manual para la obtención del árbol de Steiner en el ejemplo 2 del apartado 6.1.2.

El árbol mínimo euclídeo de expansión suma una distancia total de  $l = 66,3 \text{ u. m.}$  Por otro lado, el árbol mínimo de Steiner suma una distancia total de  $l = 62,7 \text{ u. m.}$  La inclusión de nodos de Steiner ha conseguido recortar una distancia total de  $66,3 - 62,7 = 3,6 \text{ u. m.}$

En la Tabla 18 y Figura 38 siguientes se presentan los tramos y el grafo del árbol mínimo expandido obtenido con software para el mismo conjunto de nodos.

Tabla 18: Tramos de la solución obtenida con software para el ejemplo 2 del apartado 6.1.2.

Tramos de la solución		
Tramos (aristas)	Punto de origen	Punto de destino
Tramo 1	1	15
Tramo 2	3	2
Tramo 3	5	4
Tramo 4	6	5
Tramo 5	7	3
Tramo 6	7	6
Tramo 7	8	7
Tramo 8	10	8
Tramo 9	10	9
Tramo 10	10	13
Tramo 11	11	10
Tramo 12	12	11
Tramo 13	13	14
Tramo 14	15	12

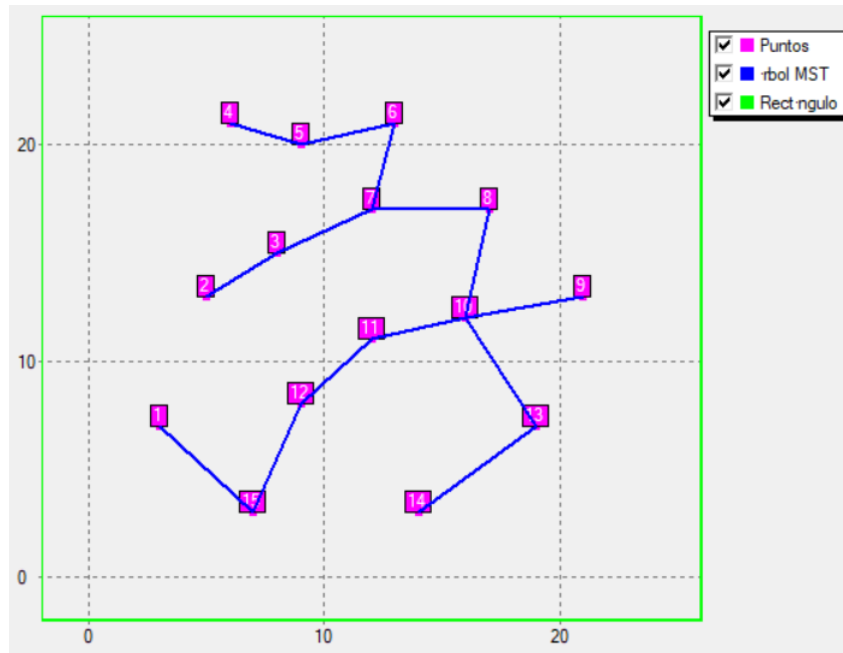


Figura 38: Solución del software para la obtención del árbol de expansión mínimo euclídeo en el ejemplo 2 del apartado 6.1.2. con quince puntos en el espacio euclídeo.

La distancia total de la solución óptima conseguida con software es  $l = 66,3 \text{ u.m.}$  Esta vez, la solución manual del árbol mínimo expandido para el conjunto de nodos tiene la misma distancia total que la solución óptima conseguida con software. Aun así, el árbol mínimo de Steiner ofrece una solución con menor distancia total.

#### 6.1.2.4. Casos con más nodos.

Por último, se muestra una idea de cómo puede ser una solución del problema de Steiner euclídeo pudiendo usar nodos de Steiner, pero para casos con muchos más nodos. En la figura se puede apreciar que el SMT euclídeo es el resultado de árboles completos de Steiner (FSTs o *full Steiner trees*) los cuales son árboles localmente mínimos que tienen una topología completa de Steiner. En la Ilustración 1 se puede ver que la mayoría de los árboles completos de Steiner abarcan 2, 3 y 4 terminales. La imagen está sacada de (Zachariasen, 2009).

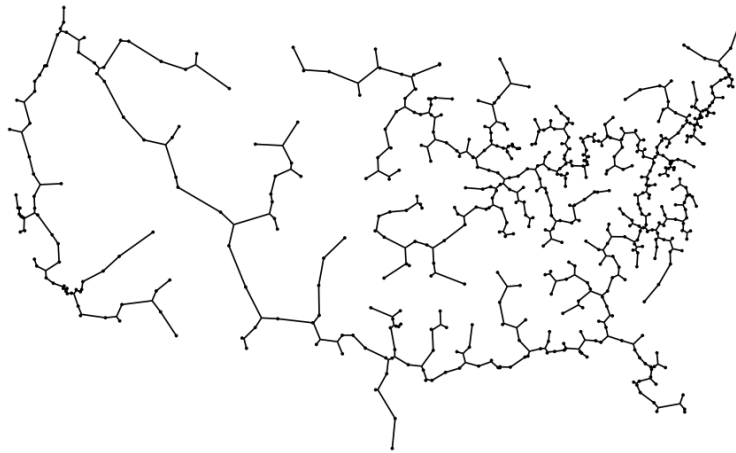


Ilustración 1: Un árbol mínimo euclídeo de Steiner (SMT) para un conjunto de puntos que representan 532 ciudades en los Estados Unidos. Fuente: (Zachariasen, 2009).

## 6.2. El problema de Steiner Euclídeo con distancias Manhattan.

El problema de Steiner en el plano euclídeo tiene otra versión, esta vez con distancias rectilíneas. Las conexiones estarán restringidas, sólo podrán ser horizontales o verticales. Las distancias rectilíneas que hay entre nodos (también llamadas distancias Manhattan) se medirán de la siguiente forma: en un espacio afín de dimensión “d”, la distancia entre dos puntos  $x: (x_1, \dots, x_d)$  e  $y: (y_1, \dots, y_d)$  se define como  $\sum_{i=1}^d |x_i - y_i|$  (Cieslik, 1998).

Una vez más, el problema puede tener dos variantes, una con nodos de Steiner y otra sin ellos, buscando alternativamente un árbol de expansión mínimo rectilíneo o un **árbol mínimo rectilíneo de Steiner**:

### 6.2.1. Escenario 1: Sin nodos de Steiner.

Como en apartados anteriores, las aristas sólo podrán incidir en nodos terminales por lo que restringirá la forma del árbol mínimo expandido. Los ejemplos a continuación se realizarán de forma manual.

#### 6.2.1.1. Ejemplo 0

En la siguiente figura se muestra un ejemplo sencillo para un conjunto de cinco nodos en la parte izquierda y la solución manual propuesta encontrando el árbol de expansión mínimo rectilíneo en la parte derecha. La longitud total de las aristas es  $l = 16 u. m.$

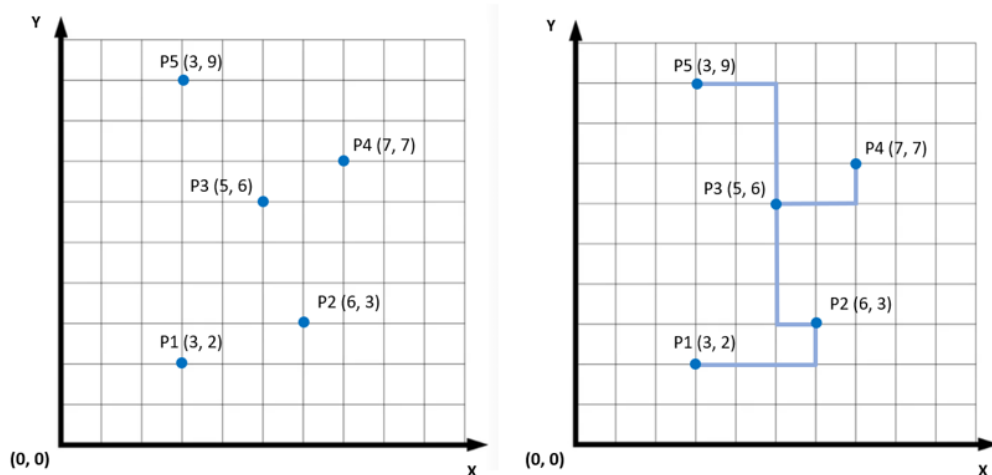


Figura 39: Enunciado y solución manual propuesta con distancias rectilíneas a un problema de Steiner de cinco nodos.

### 6.2.1.2. Ejemplo 1

En la Figura 40 se muestra el enunciado y una posible solución para la obtención del árbol mínimo expandido con distancias rectilíneas, esta vez para un conjunto de nueve puntos, la longitud total de las aristas de la solución es  $l = 27$  u. m.

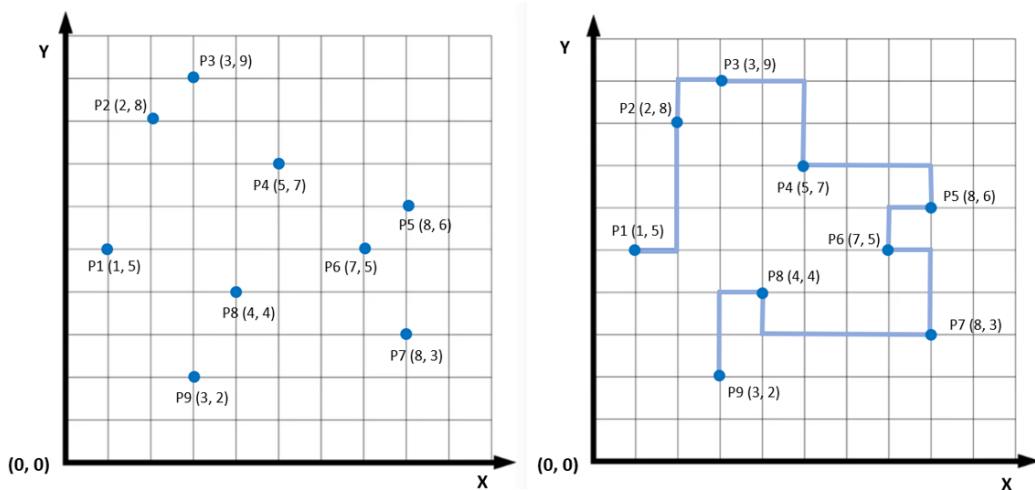


Figura 40: Enunciado y solución manual propuesta con distancias rectilíneas a un problema de Steiner de nueve nodos.



### 6.2.1.3. Ejemplo 3.

Por último, se muestra a continuación en la Figura 41 otro ejemplo más complejo de obtención del árbol mínimo expandido con distancias rectilíneas para un conjunto de sesenta nodos. La solución propuesta, en la Figura 42, se basa en un árbol que conecta los sesenta nodos, la distancia resultante de las aristas es  $l = 264 \text{ u.m.}$

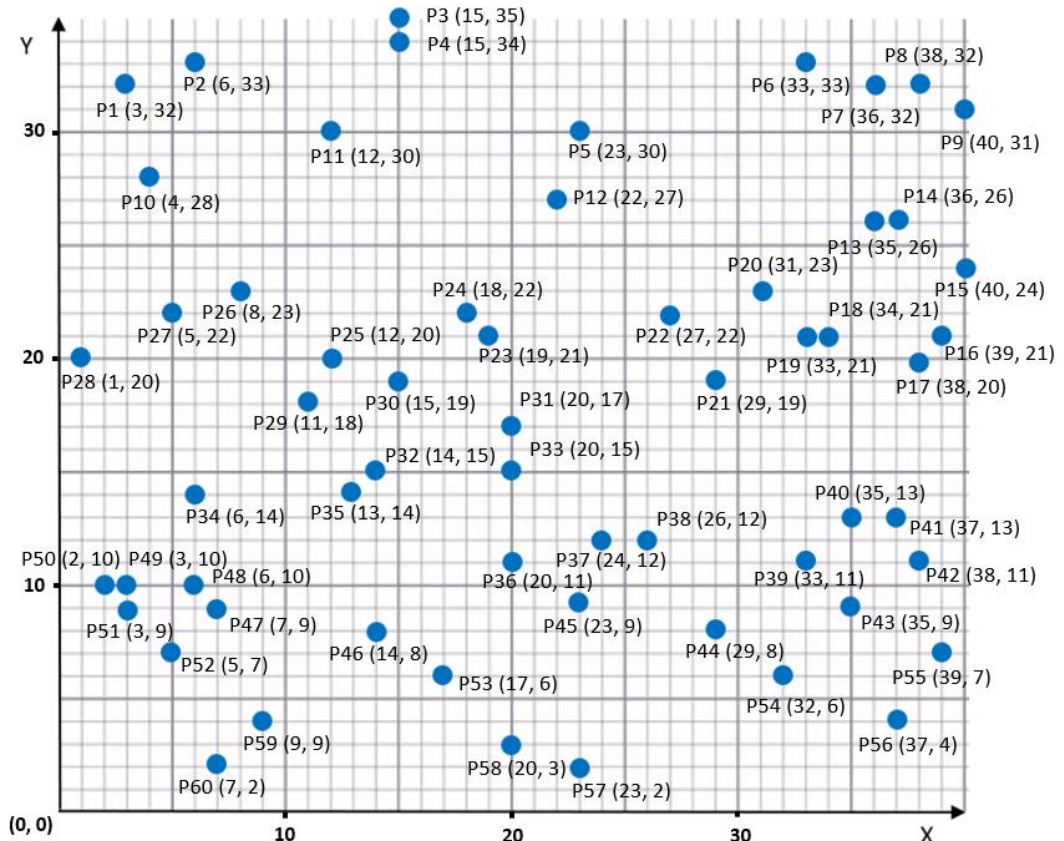


Figura 41: Conjunto de sesenta puntos para la obtención del árbol mínimo expandido.

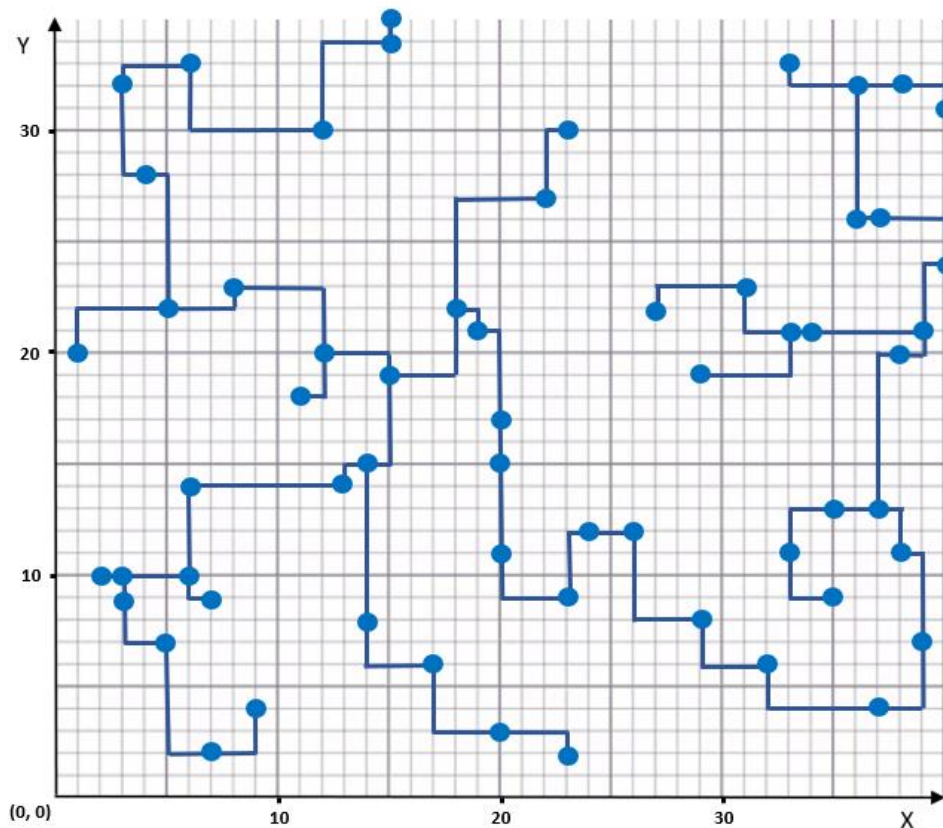


Figura 42: Árbol mínimo expandido para un conjunto de sesenta nodos.

### 6.2.2. Escenario 2: Con nodos de Steiner.

En esta ocasión se podrán añadir nodos de Steiner de forma estratégica de forma que se pueda recortar la longitud total del árbol obtenido. Al buscar un el árbol mínimo rectilíneo de Steiner debemos considerar, como posibles puntos de Steiner, sólo las esquinas desocupadas de los rectángulos que determinan cada par de nodos dados (Hanan, 1966). Por lo tanto, si se trazan líneas horizontales y verticales por los nodos, se forma un entramado, y se puede ver fácilmente que los puntos de Steiner se encuentran en las intersecciones.

Para los siguientes ejemplos se van a tomar las soluciones obtenidas en el apartado 6.2.1 y se van a comparar con las obtenidas introduciendo nodos de Steiner. Como en el apartado 6.1.2, se podrá observar que el posicionamiento de nodos de Steiner de forma estratégica puede acortar la distancia total de las soluciones.

### 6.2.2.1. Ejemplo 0.

En la Figura 43 se muestra un ejemplo sencillo de uso de los nodos de Steiner para el mismo conjunto de cinco nodos del apartado anterior y así, poder comparar la longitud total de las aristas obtenida con ambos métodos. Los puntos amarillos con contorno negro son nodos de Steiner. La longitud total de las aristas es  $l = 14$  u. m. Una longitud inferior en dos unidades a la conseguida sin usar nodos de Steiner.

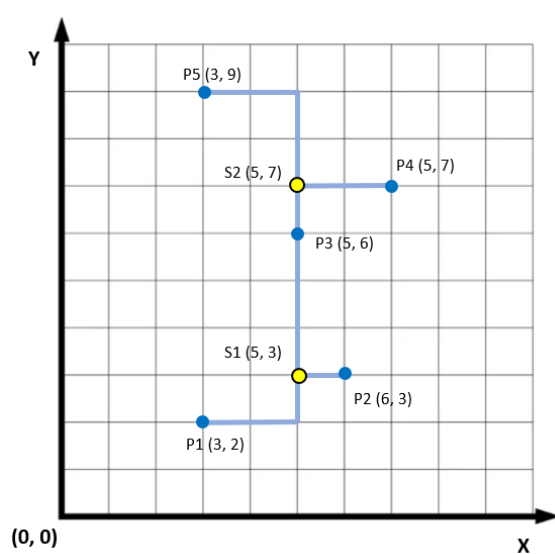


Figura 43: Posible solución del problema de Steiner de cinco nodos con un árbol mínimo rectilíneo de Steiner (usando nodos de Steiner).

### 6.2.2.2. Ejemplo 1.

En la Figura 44 se muestra una posible solución al enunciado de nueve nodos del ejemplo 1 del apartado anterior, pero esta vez obteniendo un árbol mínimo rectilíneo de Steiner, la longitud total de las aristas de la solución es  $l = 21$  u. m. Una longitud inferior en seis unidades a la conseguida sin usar nodos de Steiner.

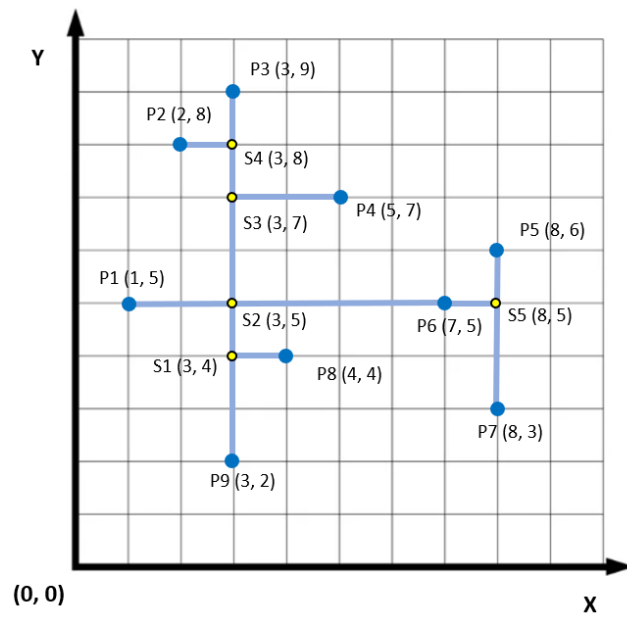


Figura 44: Posible solución del problema de Steiner de nueve nodos con un árbol mínimo rectilíneo de Steiner (usando nodos de Steiner).

### 6.2.2.3. Ejemplo 2.

Como último ejemplo se va a obtener el árbol mínimo expandido usando nodos de Steiner para el conjunto de sesenta nodos del ejemplo 2 del apartado anterior. La solución aparece a continuación, la distancia total de las aristas es  $l = 242 u. m$ , reduciendo de nuevo la distancia respecto a la solución sin el uso de los nodos de Steiner, esta vez se ha reducido veintidós unidades.

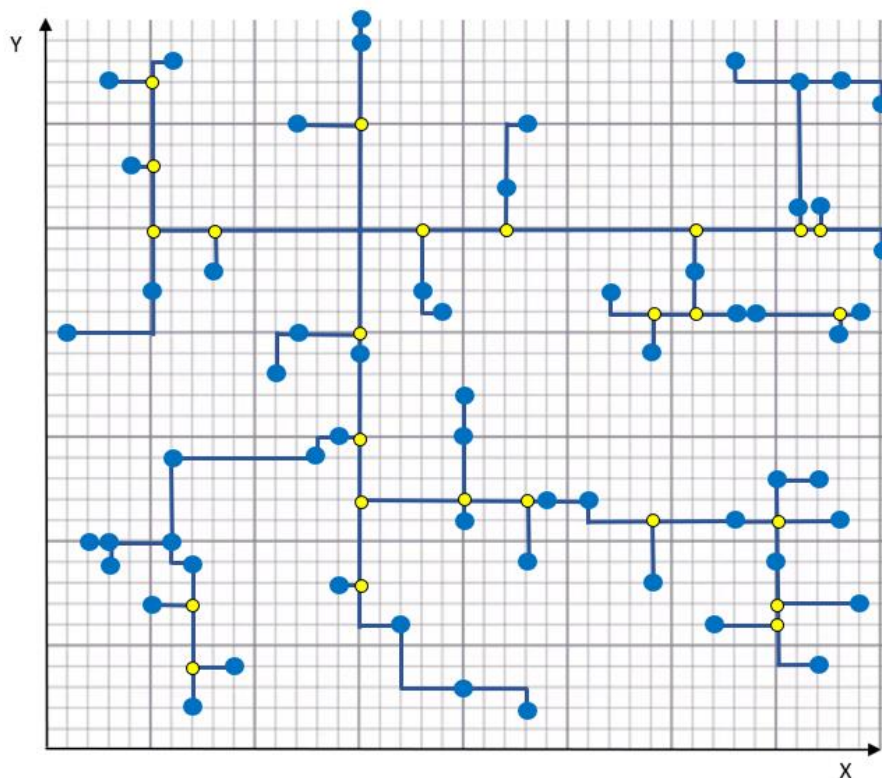


Figura 45: Posible solución del problema de Steiner de sesenta nodos con un árbol mínimo rectilíneo de Steiner (usando nodos de Steiner).

## 7. APLICACIONES DE LA OBTENCIÓN DEL ÁRBOL DE EXPANSIÓN MÍNIMO EUCLÍDEO.

El problema del árbol de Steiner se puede aplicar a muchos casos reales que afectan a bastantes industrias. Cualquier escenario en el que haya que enlazar un conjunto de puntos, los cuales pueden representar ciudades, centrales eléctricas, conectores en una red de telecomunicaciones, una red ferroviaria, una red de minas, etc. se puede convertir en un problema de Steiner en el que hay que unir dichos puntos o nodos cubriendo ciertas condiciones. Una de las condiciones puede ser conseguir el árbol mínimo de expansión de los nodos terminales obteniendo la mínima distancia total de las aristas. Dependiendo del caso la distancia de las aristas puede representar distancia entre ciudades, coste logístico, conectividad, etc. Por tanto, se puede resumir que, **en cualquier problema de red, ya sea de transmisión de información digital, fluidos, energía, etc. está presente el problema del árbol de Steiner.**

## 7.1. Aplicación en un caso real.

### 7.1.1. Presentación del problema

En este apartado se va a exponer un conjunto de nodos en el plano euclídeo como en apartados anteriores sólo que esta vez se van a utilizar datos reales. El propósito es el de obtener el árbol de expansión mínimo euclídeo de dichos nodos con el software Xpress Mosel. De este modo se va a poder ver la utilidad real de la resolución del problema de Steiner euclídeo en un caso real.

Los diferentes nodos van a representar diferentes ubicaciones reales en la ciudad de Valladolid (España). Cada nodo va a disponer de una latitud y una longitud (datos obtenidos de Google Maps). El objetivo del problema va a ser conectar todos los nodos mediante un árbol de expansión mínimo euclídeo. De tal forma podremos saber cuál es la solución óptima para conectar todas las ubicaciones con la menor distancia total posible.

A continuación, se muestra una tabla con las coordenadas (longitud y latitud) en decimal. Las coordenadas de longitud van a ser el equivalente a las coordenadas del eje x en nuestro programa, las coordenadas de latitud serán equivalentes a las coordenadas en el eje y.

Tabla 19: Datos de las treinta y seis ubicaciones en la ciudad de Valladolid.

Número de Nodo	Nombre de Ubicación	Longitud	Latitud
1	Estadio José Zorrilla	-4.762317533	41.64613686
2	Auditorio Miguel Delibes	-4.755210058	41.64438253
3	Polideportivo Pisuerga	-4.744188532	41.63185947
4	Centro comercial Vallsur	-4.750576616	41.62349929
5	Museo de la Ciencia	-4.745278851	41.6376053
6	Centro comercial El Corte Inglés	-4.740303372	41.63736482
7	Hospital Río Hortega	-4.714090826	41.62960256
8	Parque Campo Grande	-4.730237491	41.64606894
9	Plaza de Toros de Valladolid	-4.736988221	41.63968939
10	Feria de Valladolid	-4.738937778	41.65606661
11	Museo Casa de Cervantes	-4.727136078	41.6482032
12	Estación de trenes Campo Grande	-4.727139356	41.64206452
13	Estación de Autobuses Valladolid	-4.732157958	41.64146217
14	Plaza de Zorrilla	-4.729233417	41.6480533
15	Plaza de España	-4.725149658	41.6489489
16	Plaza Circular	-4.717113333	41.64730618
17	Plaza Mayor de Valladolid	-4.728282524	41.65209805
18	Plaza del Poniente	-4.731332013	41.65322992
19	Museo Patio Herreriano	-4.730283981	41.65452444
20	Plaza del Viejo Coso	-4.728426059	41.65664932
21	Palacio Real	-4.725924949	41.65665973
22	Iglesia de San Pablo	-4.724631195	41.65701618
23	Museo Nacional de Escultura	-4.723286744	41.65764358
24	Mercado del Val	-4.728462272	41.65370646
25	Playa de las Moreras	-4.733633266	41.65732267
26	Parque Ribera de Castilla	-4.722248492	41.66556406
27	Teatro Calderón	-4.724494168	41.65402551
28	Catedral de Nuestra Señora de la Asunción	-4.723253782	41.65286508
29	Iglesia de Santa María de la Antigua	-4.722858325	41.65387282
30	Plaza de la Universidad	-4.722232728	41.65295338
31	Facultad de Derecho	-4.721759578	41.65262778
32	Escuela de Ingenierías Industriales	-4.710773165	41.6571156
33	Campus Universitario Miguel Delibes	-4.704504035	41.66431333
34	Factoría Michelin Valladolid	-4.711946352	41.67592039
35	Factoría Renault Valladolid	-4.722382158	41.60197773
36	Factoría Iveco Valladolid	-4.709390303	41.63395009

En la Ilustración 2, a continuación, se muestra una foto de Valladolid con vista de satélite con las ubicaciones que se quieren enlazar en nuestro caso real.

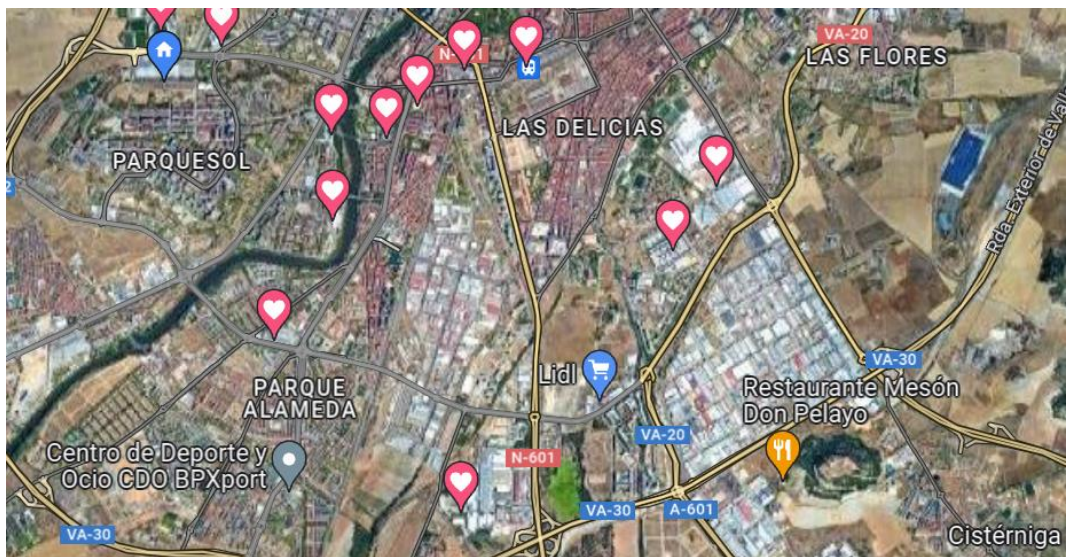
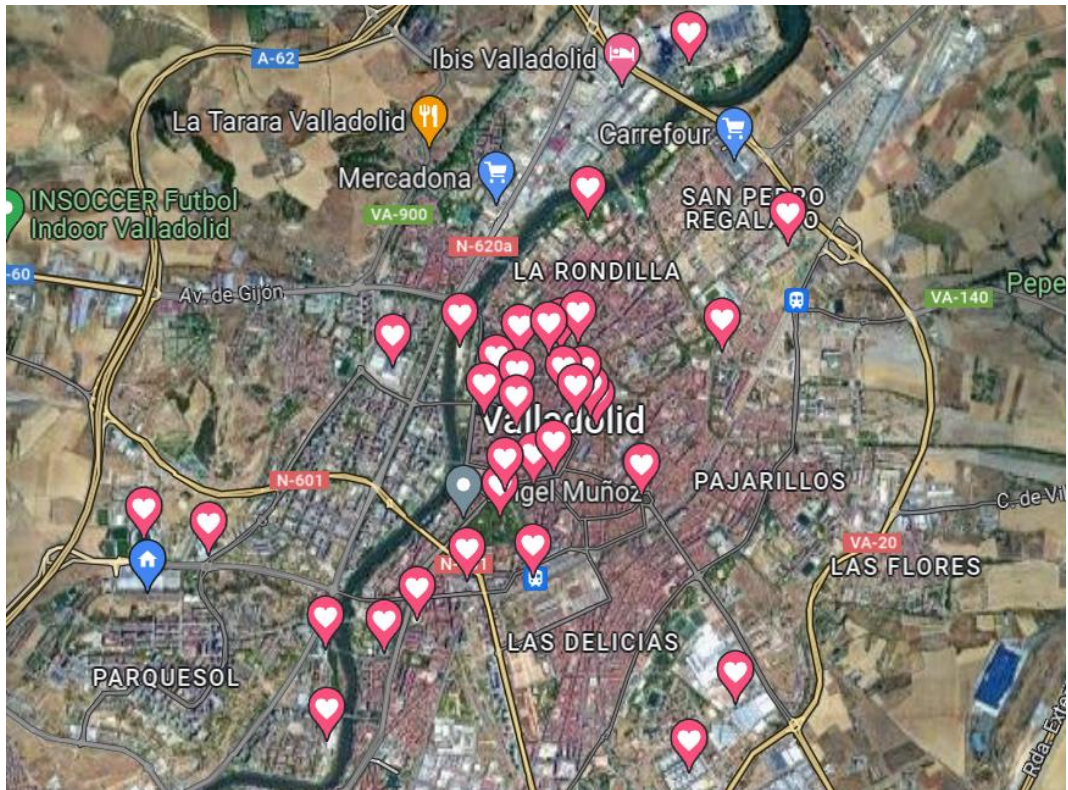


Ilustración 2: Mapa de las ubicaciones utilizadas para el caso real en la ciudad de Valladolid (partes superior e inferior de la ciudad). Fuente: Google Maps.



Si se observan las diferentes ubicaciones descritas en la Tabla 19, en la Ilustración 2 se pueden encontrar marcadas con un símbolo de un corazón. Por ejemplo, al noreste de la ciudad podemos encontrar las ubicaciones 32, 33 y 34 correspondientes a Escuela de Ingenierías Industriales, el Campus Universitario Miguel Delibes y la Factoría Michelín respectivamente. Si se observan las ubicaciones más al oeste se pueden encontrar las ubicaciones 1, 2, 3 y 4 correspondientes a el Estadio José Zorrilla, Auditorio Miguel Delibes, Polideportivo Pisuerga y el Centro Comercial Vallsur respectivamente.

Para la resolución del caso, nuestro programa, en Mosel, va ejecutando el algoritmo de Prim, calculando las posibles distancias entre nodos y escogiendo las distancias mínimas. Según se van eligiendo nodos éstos se marcan, el proceso termina cuando están todos los nodos marcados.

### 7.1.2. Resolución.

En la Tabla 20 y Figura 46 se presentan los tramos y el grafo del árbol mínimo expandido obtenido con software para conjunto de nodos presentado en el apartado anterior 7.1.1.

Tabla 20: Tramos de la solución del caso real del apartado 7.1.

Tramos de la solución					
Tramos (aristas)	Punto de origen	Punto de destino	Tramos (aristas)	Punto de origen	Punto de destino
Tramo 1	1	2	Tramo 19	19	20
Tramo 2	2	5	Tramo 20	19	25
Tramo 3	3	4	Tramo 21	20	21
Tramo 4	5	3	Tramo 22	21	22
Tramo 5	5	6	Tramo 23	22	23
Tramo 6	6	9	Tramo 24	22	27
Tramo 7	7	35	Tramo 25	23	26
Tramo 8	8	14	Tramo 26	24	19
Tramo 9	9	13	Tramo 27	25	10
Tramo 10	11	15	Tramo 28	27	29
Tramo 11	11	17	Tramo 29	28	30
Tramo 12	13	8	Tramo 30	29	28
Tramo 13	13	12	Tramo 31	30	31
Tramo 14	14	11	Tramo 32	31	16
Tramo 15	16	32	Tramo 33	32	33
Tramo 16	16	36	Tramo 34	33	34
Tramo 17	17	24	Tramo 35	36	7
Tramo 18	19	18			

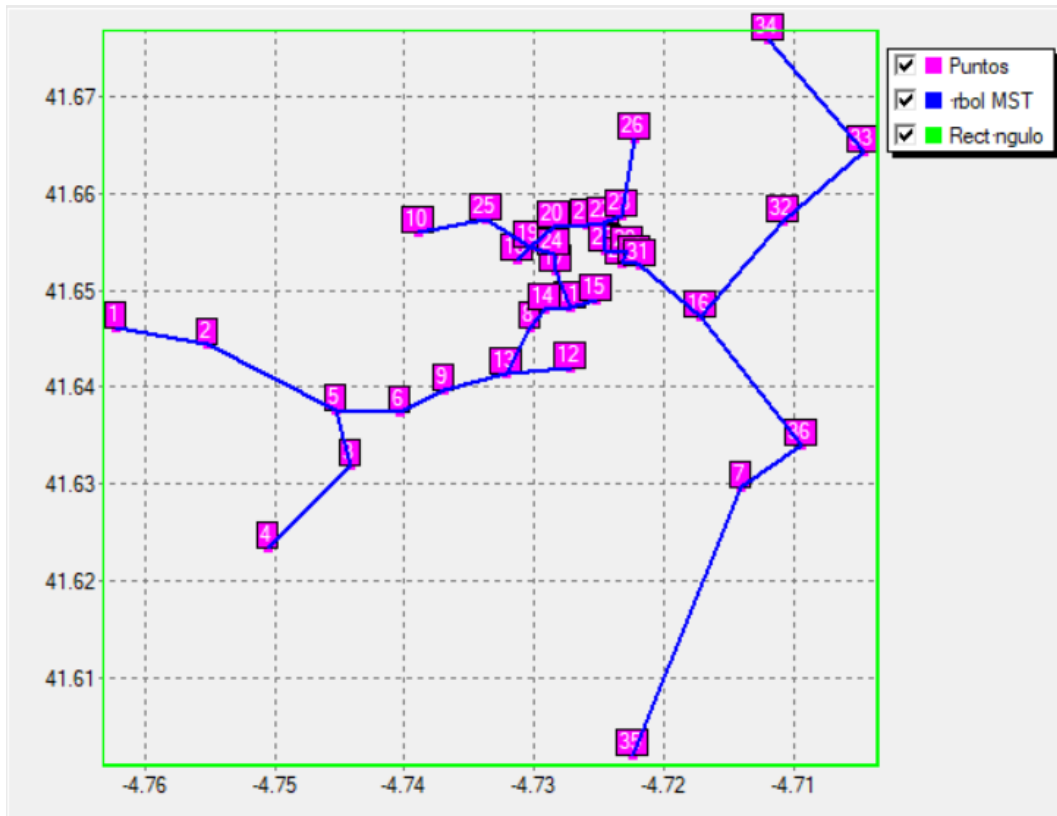


Figura 46: Solución obtenida con software con el árbol de expansión mínimo euclídeo para el caso real del apartado 7.1.

Como se puede observar en la Figura 46 la mayoría de las ubicaciones se sitúan en la zona centro (ubicaciones de 8 a 31) y son conectadas por aristas que cubren poca distancia de la solución final. Después se pueden apreciar varias ramificaciones importantes en la solución obtenida con el software. Primero se puede ver una unión de aristas que se despega del conjunto de ubicaciones del centro de la ciudad hacia el oeste, ésta acaba en la ubicación 5, de donde surgen otras dos ramificaciones hacia las ubicaciones 1 y 4, aquí se sitúan los lugares más situados hacia el oeste (barrio de Parquesol) como pueden ser el estadio José Zorrilla o el Museo de la Ciencia. También hay otras dos ramificaciones importantes al este que se separan del centro, ambas parten de la ubicación 16 (Plaza Circular), una parte hacia el norte acabando en la ubicación 34 (Factoría Michelín) y otra hacia el sur acabando en la ubicación 35 (Factoría Renault).

El programa propone que la solución óptima tiene una distancia total de 0.2017471694. No obstante, hay que convertir esta medida decimal en grados DMS (grados minutos y segundos) para poder así convertirla en kilómetros y visualizar la solución con una unidad conocida. Se tienen las siguientes equivalencias:

- Un grado equivale aproximadamente a 111,1 Km.
- Un minuto equivale a 1,8 Km.
- Un segundo equivale a 31 m.

Si convertimos la solución de distancia total (0.2017471694) a grados DMS, el resultado da 12' 6" (12 minutos y 6 segundos). Con este resultado y las equivalencias anteriores la **distancia total en kilómetros es 21,786 km.**

### 7.1.3. Limitaciones del problema.

Hay que puntualizar que nuestro problema tiene ciertas limitaciones, ya que el programa que se utiliza no tiene en cuenta las restricciones de la naturaleza que puede presentar el caso en cuestión. En condiciones normales para ir de un nodo a otro habría que valorar las calles que hay que tomar, la circulación, el tiempo de espera en semáforos, etc. Para la resolución en este caso no las tendremos en cuenta y se permite unir los nodos con líneas rectas.

También hay una limitación a la hora de introducir nodos de Steiner, el programa va a tomar todos los nodos introducidos como nodos terminales. Por tanto, todos los nodos van a representar ubicaciones por las que se quiere pasar, sin poder añadir puntos intermedios que puedan servir para acortar el camino.

En futuras extensiones de este trabajo, el programa se podría completar añadiendo restricciones de la naturaleza o habilitando los nodos de Steiner, de tal forma que se asemeje más al caso real.



# CONCLUSIONES



Para comenzar, se ha presentado un problema que lleva llamando la atención de los matemáticos y científicos desde hace siglos, como es el entonces llamado problema de Fermat, ahora llamado problema de Steiner. Los principales avances en este problema se han dado en las últimas décadas ya que hasta entonces no se disponía de la suficiente capacidad de cálculo para resolver casos complejos.

Al ser problemas NP-Completo, tanto la obtención del árbol mínimo de expansión como el árbol de Steiner necesitan algoritmos y una capacidad de cálculo suficiente que permitan obtener una solución óptima aceptable en tiempo polinomial. Es por esto por lo que se han explicado algoritmos como el de Kruskal o Prim, este último de especial importancia ya que va a ser el utilizado en nuestro programa para obtener las soluciones.

Se han abordado los diferentes conceptos y elementos relacionados con la teoría de grafos que afectan al problema de Steiner. Estos conceptos van desde los más básicos en los grafos hasta el concepto de árbol, clave a la hora de resolver el problema de Steiner.

Después de modelizar y formular el problema de Steiner para poder crear un programa con el que pueda trabajar el software Xpress Mosel, se han resuelto varios ejemplos de diversas dificultades tanto de forma manual como con el software. Comparando ambos métodos de resolución, se demuestra que no sólo el software permite resolver los problemas de forma infinitamente más rápida (cuestión de décimas de segundo), sino que se alcanzan las soluciones óptimas.

Además, en este trabajo se han expuesto las variantes del problema de Steiner en el plano euclídeo, usando tanto distancias euclídeas como distancias Manhattan. También se han comparado las soluciones obtenidas introduciendo nodos de Steiner o sin ellos para los mismos ejemplos, y se ha observado la reducción en la distancia total de los árboles al introducir éstos. El problema de Steiner euclídeo cobra gran importancia en la naturaleza y es el tipo de problema que se resuelve en el último apartado usando datos reales.

En el último apartado se muestran algunas posibles aplicaciones de la resolución del problema de Steiner en diferentes industrias, poniendo en valor los avances en este campo. La utilidad de la resolución del problema de Steiner se demuestra usando el programa que se ha desarrollado en este trabajo resolviendo un problema real como es unir varias ubicaciones en una ciudad con la mínima distancia recorrida.

Este trabajo da pie a futuras extensiones y ampliaciones, por ejemplo, el programa puede modificarse para incluir nodos de Steiner, aristas preestablecidas u otras restricciones de la naturaleza como se pudo ver en el caso real resuelto. A pesar de esto, con el ejemplo real utilizado se puede observar el potencial de los programas desarrollados en este ámbito dada su posible aplicación en materias relacionadas con la estadística espacial, donde los datos georreferenciados son analizados según su correlación (como puede ser la distancia entre nodos) para sacar valiosos resultados, incluso haciendo uso de programas como R, SAGA o QGIS.



# BIBLIOGRAFÍA



- Alvarez Nuñez, M. F. (2013). *Teoría de grafos*. Universidad del Bío-Bío.
- Brazil, M. G. (2014). *On the history of the Euclidean Steiner tree problem*. (Vol. 68). Archive for history of exact sciences.
- Cieslik, D. (1998). *Steiner Minimal Trees*. NY: Springer US.
- Cormen, T. H. (2022). *Introduction to algorithms*. MIT press.
- Du, D. C. (2001). *Steiner Trees in Industry*. Kluwer Academic Publishers.
- Du, D. Z. (2000). *Advances in Steiner trees* (Vol. 6). Springer Science & Business Media.
- Du, D., & Hu, X. (2008). *Steiner tree problems in computer communication networks*. World Scientific.
- Frank K. Hwang, D. S. (1992). The Steiner Tree Problem. En *Annals of Discrete Mathematics*. (Vol. 53).
- Hanan, M. (1966). *On Steiner's problem with rectilinear distance*. (Vol. 14). SIAM Journal on Applied mathematics.
- Johnsonbaugh, R. (2005). *Matemáticas discretas*. Pearson Educación.
- Julstrom, B. A. (April de 1994). Seeding the population: improved performance in a genetic algorithm for the rectilinear steiner problem. *Proceedings of the 1994 ACM symposium on Applied computing*, 222-226.
- López, P. D. (2005). Metodos computacionales orientados a la obtencion del arbol EMST: un aporte al problema de Steiner. *Revista de investigación de Sistemas e Informática.*, 38-42.
- Martínez Lacañina, V. (2018). *Aplicaciones en la industria del problema de Steiner y su resolución mediante algoritmos genéticos*. Sevilla.
- Nesmachnow, S. (2004). *Algoritmos genéticos paralelos y su aplicación al diseño de redes de comunicaciones confiables*.
- Prömel, H. J. (2002). *The Steiner Tree Problem A Tour through Graphs, Algorithms, and Complexity*. vieweg.
- Robledo Amoza, F. R. (2000). *Diseño topológico de redes: casos de estudio: "The generalized Steiner problem" and "The Steiner 2-edge-connected subgraph problem"*.

Soothill, G. (2010). *The euclidean steiner problem*. Report, Department of Mathematical Science, Durham University.

Zachariasen, M. (2009). *Fixed orientation interconnection problems: Theory, algorithms and applications*.