

PROYECTO FIN DE CARRERA

Autor: Daniel Guzmán Moreno
Tutor: José Manuel García Sánchez

DEPARTAMENTO: ORGANIZACIÓN
INDUSTRIAL Y GESTIÓN DE EMPRESAS

ANÁLISIS DE ESTRATEGIAS DE MODELADO PARA LA RESOLUCIÓN DEL PROBLEMA DE STEINER EN GRAFOS



INDICE

OBJETO DEL PROYECTO.....	2
PROBLEMA DE STEINER GENERALIZADO.	4
Aplicaciones del problema de Steiner.....	7
MODELOS MATEMÁTICOS EMPLEADOS PARA LA RESOLUCIÓN.....	11
Estrategia 1. Formulación de Flujo.....	13
Modelo A.....	17
Modelo B.....	18
Estrategia 2. Formulación Ciclos.....	19
Estrategia 3 Formulación ciclos resolución iterativa.....	22
IMPLEMENTACIÓN DE LOS MODELOS EN LINGO CON VBA.....	28
Estrategia 1. Formulación de Flujo.....	29
Estrategia 2. Formulación de Ciclos.....	31
Estrategia 3 Formulación ciclos resolución iterativa.....	33
LINGO.....	35
RESULTADOS EXPERIMENTALES E INTERPRETACIÓN.....	40
Características de los problemas.....	40
Resultado de los modelos.....	46
Estrategia 1. Formulación de flujo.....	46
Estrategia 2. Formulación ciclos.....	50
Estrategia 3. Formulación ciclos resolución iterativa.....	52
CONCLUSIONES.....	58
BIBLIOGRAFÍA.....	60
ANEXOS.....	61
ANEXO 1 Ejemplo de un fichero de Entrada.....	61
ANEXO 2 Archivos de entrada LINGO.....	62
Estrategia 1. Modelo A.....	62
Estrategia 1. Modelo B.....	63
Estrategia 2. Formulación Ciclos.....	65
Estrategia 3. Formulación Ciclos resolución iterativa.....	67
Anexo 3. Modulo de VBA.....	71

OBJETO DEL PROYECTO

El objeto de este proyecto será realizar un análisis comparativo de los diferentes modelos matemáticos para resolver un mismo problema, en concreto estudiaremos el problema clásico en el campo de la optimización conocido como problema de Steiner generalizado.

La ventaja de resolver el modelo matemático es la obtención de la solución exacta del modelo.

En la actualidad, también se usan métodos heurísticos o metaheurísticos para la resolución del problema de Steiner, debido a que los tiempos de resolución de dicho problema crecen de forma exponencial, a medida que aumenta el número de variables de nuestro problema. Con los métodos aproximados se consigue un compromiso entre exactitud y tiempos de resolución aceptables.

Para llevar a cabo este estudio se plantearán diferentes modelos matemáticos del problema de Steiner, usando una batería de problemas de una famosa librería de la Universidad de Londres OR-Library Imperial College of London [6], con la ayuda de Visual Basic "VBA" se automatizará la toma de datos de la biblioteca de problemas de la OR-Library y se le dará un formato de salida apto para que posteriormente resolverlo con una conocida librería de optimización llamado LINGO.

En el primer apartado el problema de Steiner generalizado, se comenzará por explicar en que consiste el problema de Steiner, las condiciones particulares del problema y veremos el campo de aplicación del problema de Steiner donde se comentarán algunos ejemplos prácticos aplicados a la realidad.

En el siguiente apartado, modelos matemáticos para la resolución, se expondrán los modelos matemáticos desarrollados para cada una de las estrategias de resolución. Se abordarán tres estrategias de resolución para el problema de Steiner, desarrollando un modelo matemático por cada estrategia.

Posteriormente, se verá la implementación de los modelos matemáticos, usando VBA para la leer los archivos de entrada con los grafos de partida de nuestra biblioteca de problemas, darles un formato de salida adecuado y posteriormente usar la biblioteca de optimización llamada LINGO para la resolución de dichos problemas. Todo esto se vera en la sección de Implementación de los modelos en LINGO usando VBA.

En el apartado resultados experimentales e interpretación, se realizará un análisis de la batería de problema y se mostrarán los resultados obtenidos de la resolución de los problemas de la librería de optimización para cada una de las estrategias planteadas.

Se presentarán comparativas de las diferentes estrategias de resolución adoptadas.

Por último, se expondrán las conclusiones del resultado del análisis de las estrategias planteadas en este proyecto.

PROBLEMA DE STEINER GENERALIZADO.

El problema de Steiner fue propuesto hace más de un siglo por el matemático Alemán Jakob Steiner, es uno de los problemas más conocidos y estudiados dentro del campo de la optimización combinatoria.

Este problema está clasificado dentro de una lista de problemas denominados NP-completo, lo que significa, que es de los problemas más difíciles, ya que no tienen solución en un tiempo polinómico.

A raíz de esto actualmente el problema se aborda por dos caminos: Heurísticas o metaheurísticas y la resolución exacta del modelo.

Las heurísticas o metaheurísticas son algoritmos, que de manera general, resuelven el problema de forma aproximada en tiempos de ejecución bastante moderados en comparación con los métodos exactos.

Nuestro estudio se centra en la resolución exacta del modelo, esto se obtiene resolviendo el modelo matemático planteado aplicado a un problema concreto. Esta manera de abordar el problema se puede tornar en una misión imposible a medida que crece el número de variables. Debido a esto, muchas veces se opta por métodos aproximados y no exactos.

El problema de Steiner en redes considera la conexión de un subconjunto de nodos en un grafo $G=(N, A)$, llamados nodos terminales, pudiendo o no utilizar para ello el otro subconjunto de nodos denominados Steiner. Dicho problema busca establecer la red de mínimo coste que interconecta el subconjunto de nodos terminales del grafo G .

Dado un grafo no-dirigido, $G=(N, A)$ con un conjunto de n nodos, otro de m arcos, un coste asociado a cada arco C_m , una partición de los nodos en un subconjunto $T \in N$ denominados nodos Terminales, con p nodos y quedando el resto de nodos $n-p$ caracterizados como nodos Steiner o subconjunto $S \in (T-N)$.

Dicho grafo no tiene por qué tener una representación en dos dimensiones, puede tener arcos cruzados lo que implica, que posee tres dimensiones espaciales.

La representación de dicho problema en dos dimensiones es un caso particular del problema de Steiner generalizado, el cual tiene representación gráfica en el plano.

El estudio se centrará en la resolución del Steiner más general.

Se puede observar un grafo constituido por nodos Terminales y nodos Steiner en la fig1.1.

Como se puede observar en dicha representación, hay arcos cruzados. Lo cuál indica que estamos en el caso más general.

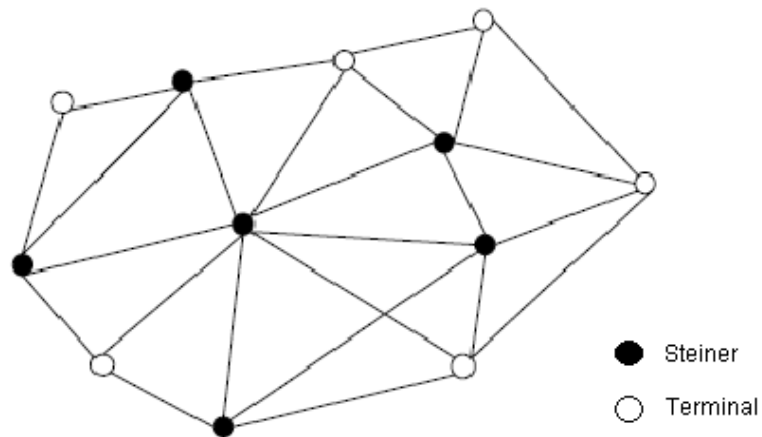


Fig2.1 Grafo de Steiner

Fuente: Elaboración propia

La solución de dicha red es un subgrafo, en el que todos los nodos terminales están conectados y los nodos Steiner, pueden, o no estar en dicha red. Una posible solución a dicha red en la fig2.2.

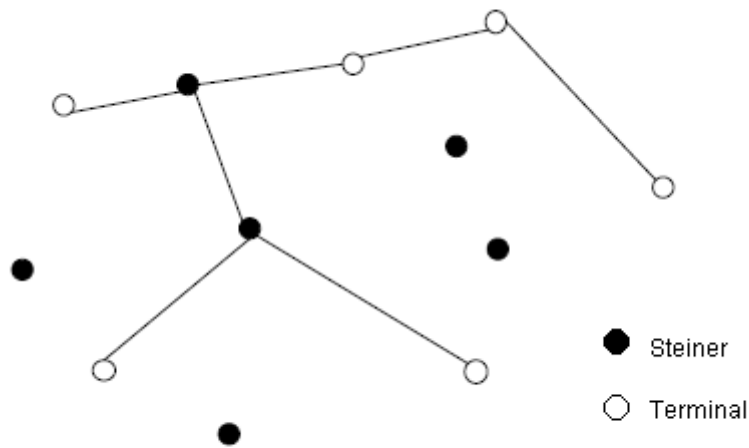


Fig. 2.2 Solución al grafo de Steiner

Fuente: Elaboración propia

La solución $G_s = (N_s, V_s)$ se divide en dos conjuntos. El primero es el subconjunto N_s solución del problema. En el que todos los nodos terminales forman parte de la red y otro subconjunto formado por los nodos Steiner. Los cuáles, no forman parte de la solución.

A continuación veremos las diferentes soluciones admisibles de un mismo grafo.

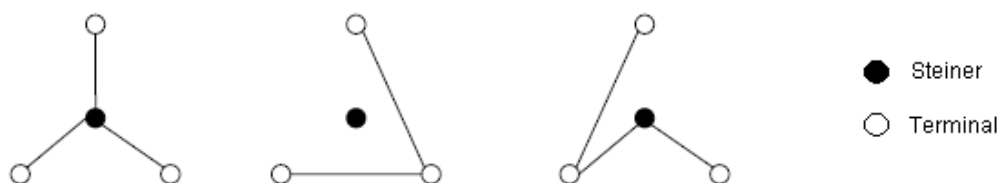


Fig. 2.3 Soluciones Alternativas

Fuente Elaboración Propia

La solución óptima de dicha red siempre forma un árbol, es decir, tiene $N-1$ arcos conectados, siendo N el número de nodos terminales más el número de nodos Steiner que forman parte de la solución, donde los arcos suman el coste mínimo de la red.

Aplicaciones del problema de Steiner

Un caso particular de nuestro problema es el STP Steiner Tree Problem.

La diferencia entre STP y el problema generalizado es que en este caso podemos hacer una representación del grafo en dos dimensiones ya que no hay cortes entre arcos del grafo.

Dicho problema encuentra su aplicación en muchos problemas de localización, logística y transporte, así como de planificación.

En el campo de las telecomunicaciones, es un caso particularmente rico en problemas que han desembocado en el tradicional problema de Steiner. Entre ellos se encuentran el problema del diseño de red jerárquica, Dynamic Predicate Stasching Copy Problem, que constituye una aplicación para la entrega de información y distribución de sistema de computadores.

En la actualidad estamos rodeados de redes de todo tipo, como red de agua, gas, electricidad, carreteras... Por eso este tipo de problemas son muy interesantes a la hora de optimizar recursos.

Ejemplo 1

La red de carreteras de España representa un ejemplo de este modelo en el que los nodos terminales corresponderían a las ciudades que queremos conectar. Donde el coste de las carreteras es uno de los factores principales que condicionan el diseño de dicha red, ya que existe multitud de caminos para ir de una ciudad a otra pero dependiendo de la zona por la que vallamos a realizar el trazado de la carretera los costes variarán, no siendo la solución más económica la línea recta que une dichos puntos terminales. [7]



Fig. 2.4 Mapa de carreteras de Andalucía

Fuente: <http://www.juntadeandalucia.es/institutodeestadisticaycartografia/guiacarreteras/gc/mapa.html>

Así trasladando esta información al problema de Steiner, todos los posibles caminos que engloban los puntos a conectar formarían el grafo inicial. Identificando una serie de puntos intermedios de interés, como núcleos de población secundarios, estos constituirían los nodos Steiner y los puntos a conectar como nodos Steiner.

Ejemplo 2

Otro posible escenario, se puede ver representado en la red de internet. Donde los nodos terminales serían los ordenadores conectados a la red y los nodos Steiner los servidores que dan cobertura a los mismos.

A continuación en la fig1.5 vemos Mapa parcial de Internet representado como un grafo.

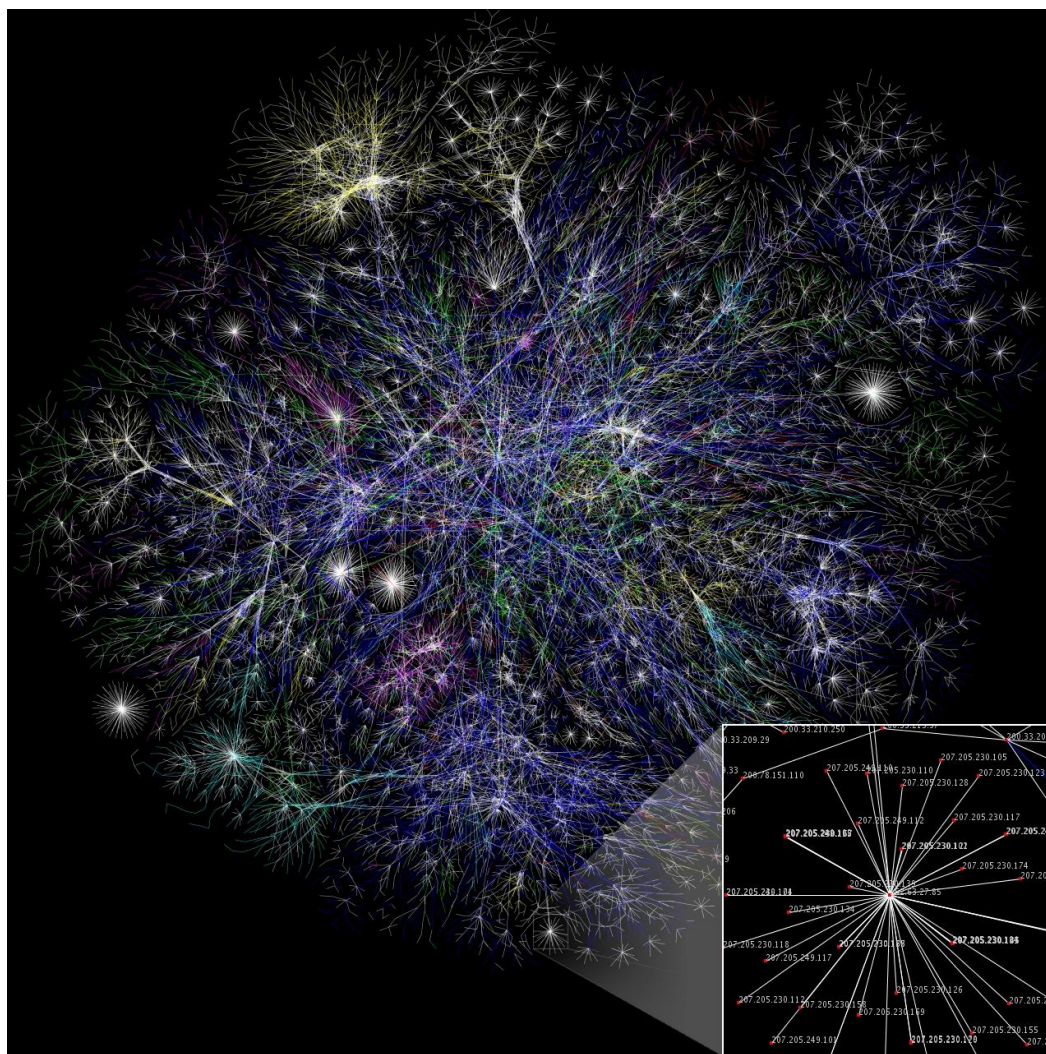


Fig2.5 Mapa de Internet

Fuente: https://es.wikipedia.org/wiki/Internet#/media/File:Internet_map_1024.jpg

Este está basado en la información obtenida del sitio opte.org el 15 de enero de 2005. Cada línea dibujada entre dos nodos representa el enlace entre dos direcciones IP. La longitud de las líneas es proporcional al tiempo de espera

entre los nodos. La imagen representa 30% de las redes tipo C accesibles al programa de colección de datos de 2005.

El color de las líneas corresponde a su clasificación RFC 1918. [8]

MODELOS MATEMÁTICOS EMPLEADOS PARA LA RESOLUCIÓN

En primer lugar, explicaremos de manera escueta que es un modelo de optimización.

Un modelo de optimización lineal se caracteriza por tener un sistema de ecuaciones llamadas restricciones, maximizando o minimizando una función lineal, llamada función objetivo.

$$\text{Max/Min F.O}$$

$$\text{sujeto a: } X_1 + X_2 + \dots + X_n \leq B_n$$

Definición de las Variables

El problema de Steiner considera la conexión de un subconjunto de nodos en un grafo $G=(N, A)$. Siendo N el número de nodos que componen nuestro grafo y A todos los arcos del problema.

Hay dos tipos de nodos. Nodos terminales los cuáles tienen que estar todos conectados por al menos un arco y nodos Steiner que no tienen por qué estar conectados.

El problema de Steiner busca establecer la red de mínimo coste que conecta un subconjunto de nodos terminales del grafo G .

DADO: Un grafo no-dirigido, $G(N, A)$, con n nodos, m arcos, con un coste asociado a cada arco C_m , una partición de los nodos en un subconjunto $I \in N$ con p nodos Terminales y quedando los restantes nodos, $n-p$ caracterizados como nodos Steiner.

ENCONTRAR: Una subred G_1 de G , tal que exista un camino posible entre todo par de nodos pertenecientes al subconjunto I , es decir, entre todo par de nodos Terminales, de forma que la suma del coste de los arcos asociados a G_1 sea mínima.

En la fig. 3.1 se puede ver una representación del grafo de partida del problema de Steiner.

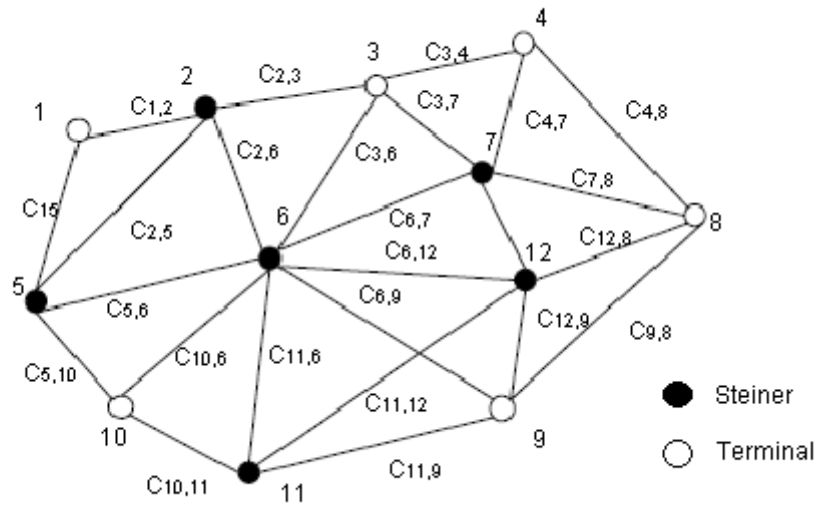


Fig. 3.1 Grafo de Partida

Fuente: Elaboración propia

El problema de Steiner tiene diversas formas de modelarse matemáticamente. En la bibliografía se puede encontrar formulación de flujo, formulación de cobertura, formulación con restricciones anti-bucle, formulación lagrangiana y programación dinámica. [3]

La primera estrategia de resolución se basará en una formulación de flujo del problema de Steiner y algunas modificaciones en sus restricciones darán lugar a diferentes variantes del problema dentro de esta forma de modelar el problema.

La segunda estrategia de resolución se basará en restricciones anti-bucle.

La tercera estrategia de resolución será una formulación con las restricciones anti-bucle que iremos incorporando al modelo.

Estrategia 1. Formulación de Flujo

En principio el problema de Steiner presenta un problema de un grafo con arcos no dirigidos, como hemos visto anteriormente, pero para esta formulación se dará una dirección a todos los arcos.

Se comenzarán por definir los datos y variables del problema.

DATOS:

C_{ij} : Coste asociado a cada arco.

N : Conjunto de nodos del grafo.

T : Subconjunto de nodos Terminales.

$N-T$: Nodos Steiner.

VARIABLES:

X_{ij} : Variable entera que representa la cantidad de flujo que circula del nodo "i" al nodo "j".

Para cada arco no dirigido del problema X_i inicial se crearán dos arcos: uno de salida X_{ij} y otro de entrada, X_{ji} . Teniendo definidas dos direcciones por cada arco. En la fig. 3.2 podemos ver este hecho gráficamente.

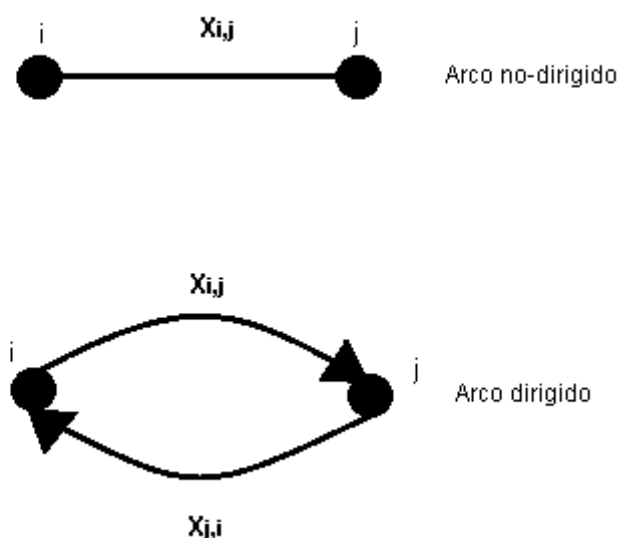


Fig. 3.2 Gráfico de arcos orientados

Fuente: Elaboración propia

y_{ij} : Variable binaria, es decir, solo puede tomar valores "0" o "1". Tomará valor 0 cuando el arco y_{ij} , no forme parte de la solución del problema y 1 cuando éste forme parte de la red.

e : Es el primer nodo terminal.

Así el problema se modelará de la siguiente forma:

$$\text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij}$$

$$\text{sa: } X(j, N) - X(N, j) \leq 1 \text{ si } (i, j) \in T$$

$$X(j, N) - X(N, j) \leq 0 \text{ si } (i, j) \in (N - T) \text{ } j \neq e \text{ } j \in N$$

$$X(j, N) - X(N, j) = -|T| + 1 \text{ si } (i, j) = e$$

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

$$X_{ij} \geq 0$$

En la fig3.3 se observa la red formulada.

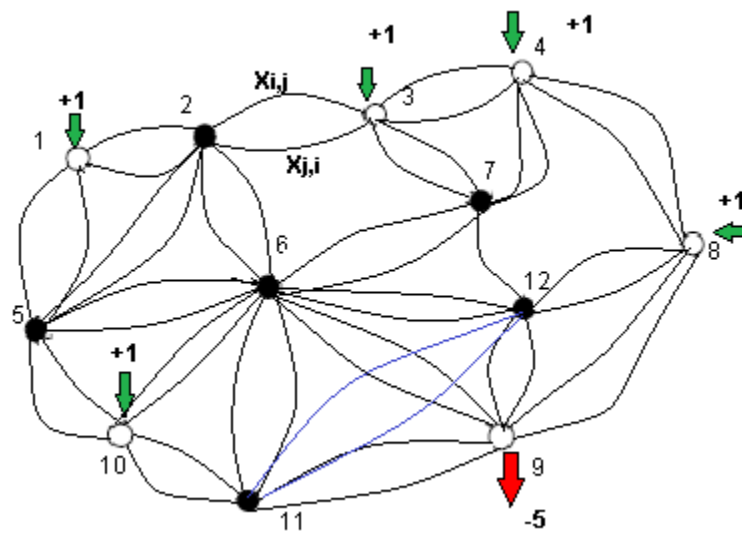


Fig. 3.3 Grafo con Arcos Orientados Formulación de Flujo

Fuente: Elaboración propia

Dónde $Min \sum_{(i,j) \in A} C_{ij} * y_{ij}$ es la función objetivo del problema, la cuál tratará de que el coste del grafo solución sea mínimo.

El problema estará sujeto a las siguientes restricciones, las cuáles representan las reglas por la que se rige el modelo.

Las restricciones no son más que un balance de todos los arcos que entran y salen de un nodo para cada uno de los nodos del problema.

Veamos cada una de las restricciones por separado.

1. $X(j, N) - X(N, j) = 1$ si $(i, j) \in T$

La primera restricción hace que para todos los nodos Terminales, la suma de los arcos que entran y salen tiene que ser uno, es decir se inyecta una unidad de flujo en ese terminal, asegurando que para todos los nodos terminales llegue algún arco, o lo que es lo mismo, que este conectado.

2. $X(j, N) - X(N, j) = 0$ si $(i, j) \in (N - T)$ $j \neq e$ $j \in N$

La segunda restricción hace un balance de flujo para los nodos Steiner, es decir, la suma de los arcos que entran o salen del nodo Steiner tiene que ser igual a 0.

3. $X(j, N) - X(N, j) = -|T| + 1$ si $(i, j) = e$

La tercera restricción inyecta en el primer nodo terminal de la red una cantidad de flujo $-|T| + 1$, y de esta forma, se cumple que las unidades de flujo que se inyectan en la red son igual número de unidades que la red demanda.

Sólo queda por ver la definición que se ha dado de las variables:

- $y_{ij} \in (0,1)$ Definición de variable binaria.
- $X_{ij} \geq 0$ Variable que tiene que ser mayor que uno dándole al flujo el mismo sentido que por el arco por el que circula.

Así el problema queda modelado con una formulación de flujo.

Este mismo problema, pero sin nodos Steiner representaría el problema de flujo máximo a coste mínimo.

Todavía nos haría falta relacionar la variable x con y para que la función objetivo haga su trabajo.

Esto se puede hacer de dos formas, una de ellas añadiendo a las restricciones una adicional. Veamos como quedan los problemas añadiendo dos versiones de una misma restricción.

Modelo A

El modelo A presenta una restricción agregada.

- $X_{ij} + X_{ji} \leq (N * T - 1) * y_{ij}$
Esta restricción relaciona la variable X_{ij} con Y_{ij} y hace que, si circula alguna unidad de flujo por el arco bidireccional, se active la variable binaria.

Así la formulación de flujo del modelo A quedará de la siguiente manera:

MODELO A

$$\text{Min } \sum_{(i,j) \in A} C_{ij} * y_{ij}$$

$$\text{sa: } X(j, N) - X(N, j) \leq 1 \text{ si } (i, j) \in T$$

$$X(j, N) - X(N, j) \leq 0 \text{ si } (i, j) \in (N - T) \text{ } j \neq e \text{ } j \in N$$

$$X(j, N) - X(N, j) = -|T| + 1 \text{ si } (i, j) = e$$

$$X_{ij} + X_{ji} \leq (N * T - 1) * y_{ij}$$

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

$$X_{ij} \geq 0$$

Modelo B

El modelo B presenta una restricción desagregada, o lo que es lo mismo, dos restricciones en vez de una como en el caso anterior.

- $X_{ij} \leq (N * T - 1) * y_{ij}$
- $X_{ji} \leq (N * T - 1) * y_{ij}$

Este par de restricciones funcionan igual que la restricción anterior, aunque se verá más adelante que aunque funcionen del mismo modo computacionalmente no sean lo mismo.

Así el modelo B quedará de la siguiente forma:

MODELO B

$$\text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij}$$

$$\text{sa: } X(j, N) - X(N, j) \leq 1 \text{ si } (i, j) \in T$$

$$X(j, N) - X(N, j) \leq 0 \text{ si } (i, j) \in (N - T) \text{ } j \neq e \text{ } j \in N$$

$$X(j, N) - X(N, j) = -|T| + 1 \text{ si } (i, j) = e$$

$$X_{ij} \leq (N * T - 1) * y_{ij}$$

$$X_{ji} \leq (N * T - 1) * y_{ij}$$

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

$$X_{ij} \geq 0$$

Estrategia 2. Formulación Ciclos

La formulación ciclos que vamos a desarrollar estará compuesta por variables binarias, esto quiere decir, que ahora el grafo no es dirigido, a priori, esto es una forma de simplificar el problema con respecto al problema anterior, ya que se manejarán la mitad de variables al no darle dirección a los arcos.

Los datos de partida del problema al igual que en el caso anterior son:

DATOS:

C_{ij} : Coste asociado a cada arco.

N : Conjunto de nodos del grafo.

T : Subconjunto de nodos terminales.

$N-T$: Nodos Steiner.

Sin embargo, ahora las variables serán sensiblemente diferentes.

VARIABLES:

Y_{ij} : Variable binaria, que representa si el arco que va de "i" a "j" forma parte de la red tomando valor uno o cero si no forma parte de la red.

Así el problema se modelará de la siguiente forma:

$$\begin{aligned} \text{Min} \quad & \sum_{(i,j) \in A} C_{ij} * y_{ij} \\ & \sum_{\forall (i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T \\ & \sum_{\forall (i,j)} Y_{ij} \leq 0 \quad \text{o} \quad \sum_{\forall (i,j)} Y_{ij} \geq 2 \quad \forall (i,j) \in (N - T) \\ & \sum_{(i,j) \in N} Y_{ij} \leq |H| - 1 \quad \forall \text{Conjunto } |H| \geq 3 \\ & y_{ij} \in (0,1) \end{aligned}$$

Dónde $Min \sum_{(i,j) \in A} C_{ij} * y_{ij}$ es la función objetivo del problema, la cuál tratará de que el coste del grafo solución sea mínimo.

Se verá el significado de cada restricción para comprender el modelado propuesto.

1. $\sum_{\forall(i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T$

Esta restricción hace un balance de arcos que llegan a un nodo, siendo al menos un arco el que se tiene que conectar para todos los nodos terminales.

2. $\sum_{\forall(i,j)} Y_{ij} \leq 0 \quad o \quad \sum_{\forall(i,j)} Y_{ij} \geq 2 \quad \forall (i,j) \in (N - T)$

Esta restricción hace el mismo balance para todo los nodos Steiner del grafo, siendo el número de arcos a conectar o ninguno o más de dos, siendo esta la definición de nodo Steiner del problema.

3. $\sum_{(i,j) \in N} Y_{ij} \leq |H| - 1 \quad \forall \text{Conjunto } |H| \geq 3$

Siendo H un subconjunto de N del grafo compuesto por al menos 3 nodos colindantes, la suma de los arcos que unen estos 3 o más nodos no puede ser mayor que el número de nodos de este subconjunto menos 1.

Esta restricción básicamente impide que se formen ciclos en la solución óptima dando así, cómo resultado, un árbol de expansión mínima.

Esta restricción es la que hace que este problema sea NP-completo, ya que para resolverlo antes habrá que calcular todos las posibles subrutras que se pueden formar en el grafo.

Para implementar esto en LINGO se reformulará la segunda restricción del problema de la siguiente forma:

$$\sum_{\forall(i,j)} Y_{ij} \leq 0 \quad o \quad \sum_{\forall(i,j)} Y_{ij} \geq 2 \quad \forall (i,j) \in (N - T).$$

Se sustituirá por la siguiente expresión:

1. $N_i = \sum_{\forall i \in (N-T)} Y_{ij}$,
2. $N_i \geq 2 - M * (1 - \beta_i) \quad \forall i \in (N - T)$
3. $N_i \leq M * \beta_i \quad \forall i \in (N - T)$

Se ha introducido una variable binaria auxiliar, se asigna una por cada nodo Steiner del problema. Así tomará valor uno cuando el nodo Steiner se active haciendo que la segunda restricción quede como $N_i \geq 2$ y la tercera como $N_i \leq M$, siendo M el número de arcos conectados a nuestro nodo Steiner quedando definido $N_i \in (2, M)$.

Si esta variable tomase valor "0" entonces la 2ª restricción, queda $N_i \geq (2-M)$ y la 3ª queda $N_i \leq 0$, solo pudiendo ser $N_i = 0$.

Así, el modelo a implementar será:

MODELO FORMULACIÓN CICLOS.

$$\text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij}$$

$$\sum_{\forall (i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T$$

$$N_i = \sum_{\forall i \in (N-T)} Y_{ij}$$

$$N_i \geq 2 - M * (1 - \beta_i) \quad \forall i \in (N - T)$$

$$N_i \leq M * \beta_i \quad \forall i \in (N - T)$$

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

$$\beta \in (0,1)$$

Estrategia 3 Formulación ciclos resolución iterativa

Los datos de partida del problema igual que en el caso anterior son:

DATOS:

C_{ij} : Coste asociado a cada arco.

N : Conjunto de nodos del grafo.

T : Subconjunto de nodos terminales.

$N-T$: Nodos Steiner.

VARIABLES:

Y_{ij} : Variable binaria, que representa si el arco que va de “i” a “j” forma parte de la red tomando valor uno o cero si no forma parte de la red.

Así el problema se modelará de la siguiente forma:

$$\begin{aligned} & \text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij} \\ & \sum_{\forall (i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T \\ & \sum_{\forall (i,j)} Y_{ij} \leq 0 \quad \text{o} \quad \sum_{\forall (i,j)} Y_{ij} \geq 2 \quad \forall (i,j) \in (N - T) \\ & \sum_{\forall i \in (N-T)} \beta_i + \sum_{\forall (i,j) \in T} y_{ij} = N - 1 \\ & y_{ij} \in (0,1) \end{aligned}$$

Dónde $\text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij}$ es la función objetivo del problema, la cuál tratará de que el coste del grafo solución sea mínimo.

1. $\sum_{\forall (i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T$

Esta restricción hace un balance de arcos que llegan a un nodo, siendo al menos un arco el que se tiene que conectar para todos los nodos terminales.

$$2. \sum_{\forall(i,j)} Y_{ij} \leq 0 \text{ o } \sum_{\forall(i,j)} Y_{ij} \geq 2 \quad \forall (i,j) \in (N - T)$$

Esta restricción hace el mismo balance para todo los nodos Steiner del grafo, siendo el número de arcos a conectar o ninguno o más de dos.

$$3. \sum_{\forall i \in (N-T)} \beta_i + \sum_{\forall(i,j) \in T} y_{ij} = N - 1$$

Esta restricción impide que la solución óptima este formada por más arcos que nodos contenga la red.

El modelo que se resolverá al principio será el mismo, pero quitando la restricción anti-bucles y añadiendo una restricción adicional para que no se forme una red formada por más arcos que nodos contenga la red.

A priori, de esta forma no se tienen que calcular todos los posibles de ciclos que se pueden formar dentro del grafo.

Observamos que la 3ª restricción del anterior problema o restricción anti-bucle, es la que es responsable de que el problema sea NP-Completo, debido a que calcular todos los posibles ciclos que se pueden formar dentro del grafo, cuando hay un número bastante grande de nodos hace que los tiempos que empleamos en el cálculo de esta restricción sean bastante grandes. Incluso para determinados problemas, como se verá en la parte de los resultados, imposible.

De manera análoga al caso anterior se reformula la restricción:

$$\sum_{\forall(i,j)} Y_{ij} \leq 0 \text{ o } \sum_{\forall(i,j)} Y_{ij} \geq 2 \quad \forall (i,j) \in (N - T)$$

Y se sustituirá por:

$$N_i = \sum_{\forall i \in (N-T)} Y_{ij}$$

$$N_i \geq 2 - M * (1 - \beta_i) \quad \forall i \in (N - T)$$

$$N_i \leq M * \beta_i \quad \forall i \in (N - T)$$

Quedando el modelo de la siguiente manera.

$$\begin{aligned} & \text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij} \\ & \sum_{\forall (i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T \\ & N_i = \sum_{\forall i \in (N-T)} Y_{ij} \\ & N_i \geq 2 - M * (1 - \beta_i) \quad \forall i \in (N - T) \\ & N_i \leq M * \beta_i \quad \forall i \in (N - T) \\ & \sum_{\forall i \in (N-T)} \beta_i + \sum_{\forall (i,j) \in T} y_{ij} = N - 1 \\ & y_{ij} \in (0,1) \\ & \beta_i \in (0,1) \end{aligned}$$

La solución de este modelo es una solución parcial del modelo de Steiner, a esta solución compuesta por los arcos: y_{ij}^* , β_i^* .

A este subgrafo obtenido se le calcularán todos los posibles ciclos que se han formado y se incluirán estas restricciones anti-bucle en el modelo inicial. Donde se volverá a resolver estableciendo un proceso iterativo de resolución que terminará cuando no haya más ciclos en las soluciones parciales. [5]

Así se obtiene el siguiente modelo matemático:

MODELO FORMULACION CICLOS RESOLUCION ITERATIVA

$$\text{Min} \sum_{(i,j) \in A} C_{ij} * y_{ij}$$

$$\sum_{\forall (i,j)} Y_{ij} \leq 1 \quad \forall (i,j) \in T$$

$$N_i = \sum_{\forall i \in (N-T)} Y_{ij}$$

$$N_i \geq 2 - M * (1 - \beta_i) \quad \forall i \in (N - T)$$

$$N_i \leq M * \beta_i \quad \forall i \in (N - T)$$

$$\sum_{\forall i \in (N-T)} \beta_i + \sum_{\forall (i,j) \in T} y_{ij} = N - 1$$

$$\sum_{(i,j) \in N^o} Y_{ij}' \leq |H'| - 1 \quad \forall \text{Conjunto } |H'| \geq 3$$

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

$$\beta_i \in (0,1)$$

- $\sum_{(i,j) \in N^o} Y_{ij}' \leq |H'| - 1 \quad \forall \text{Conjunto } |H'| \geq 3$

Siendo H' todos los subconjuntos de nodos conectados en los grafos de la soluciones parciales.

Como se ha comentado, estableceremos un proceso iterativo de resolución. Resolviendo el problema inicial, calculando los ciclos que contengan dicha solución e incorporando dichas restricciones a nuestro modelo.

El proceso terminará cuando en alguna de las soluciones parciales obtenidas no se encuentre ningún ciclo, obteniendo la solución óptima del modelo. Entonces se tendrá un modelo compuesto por las restricciones anteriores, así como las restricciones anti-bucles críticas para el problema.

De esta forma se ahorrará calcular todos los posibles ciclos que se pueden formar en la red.

Las soluciones parciales obtenidas siempre tienen un valor de la función objetivo menor que el óptimo.

Se verá un ejemplo gráfico de la estrategia de resolución propuesta de un grafo cualquiera. Se observa en la fig3.4 el grafo de la primera solución parcial obtenida.

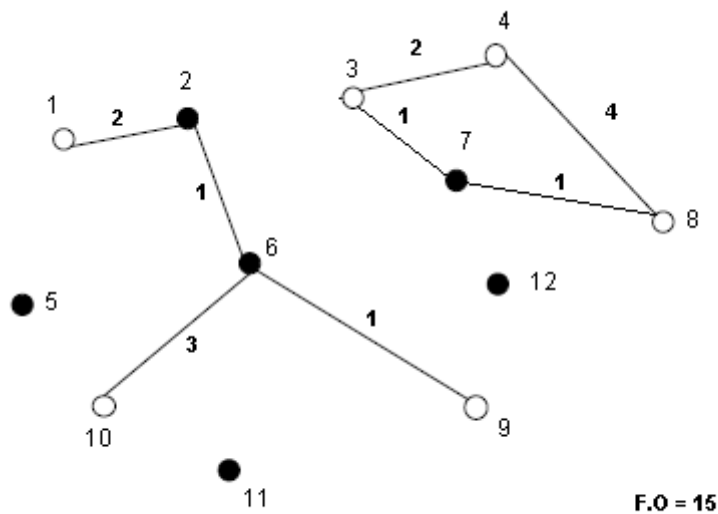


Fig. 3.4 Grafo solución parcial

Fuente: Elaboración propia

A dicha solución se le calcularán los ciclos formados en dicho grafo, y se incluirán en el modelo.

En este caso, sería el ciclo formado por los nodos "3", "4", "7" y "8". Añadiendo esta restricción y volviendo a resolver se obtiene el siguiente grafo de la fig3.5.

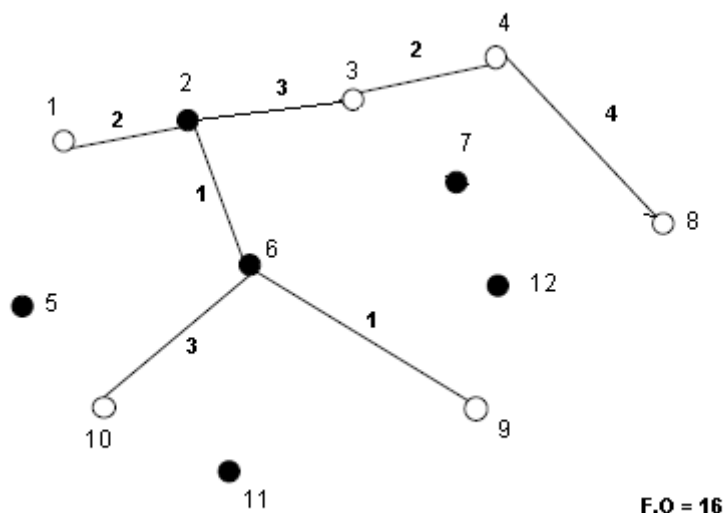


Fig. 3.5 Grafo solución óptima Iteración 1ª

Fuente: Elaboración propia

Esta solución parcial obtenida sería la solución óptima del problema al no contener ningún ciclo. Dando por terminado el proceso iterativo de resolución. [5]

Como se puede observar el valor de la función objetivo aumenta, es decir que una sola iteración da una primera aproximación con un valor de la función igual o menor al óptimo.

IMPLEMENTACIÓN DE LOS MODELOS EN LINGO CON VBA

Los grafos de partida vendrán dados con un formato que se puede ver en el ANEXO 1.

Estos dan como datos de partida: el número de nodos Steiner, el número de nodos terminales, número de arcos que componen la red, así como el inicio y final de los arcos que componen la red y vienen enumerados los nodos terminales.

En este apartado se presenta el diseño funcional de la aplicación desarrollada para la resolución del problema de Steiner con la librería de optimización LINGO.

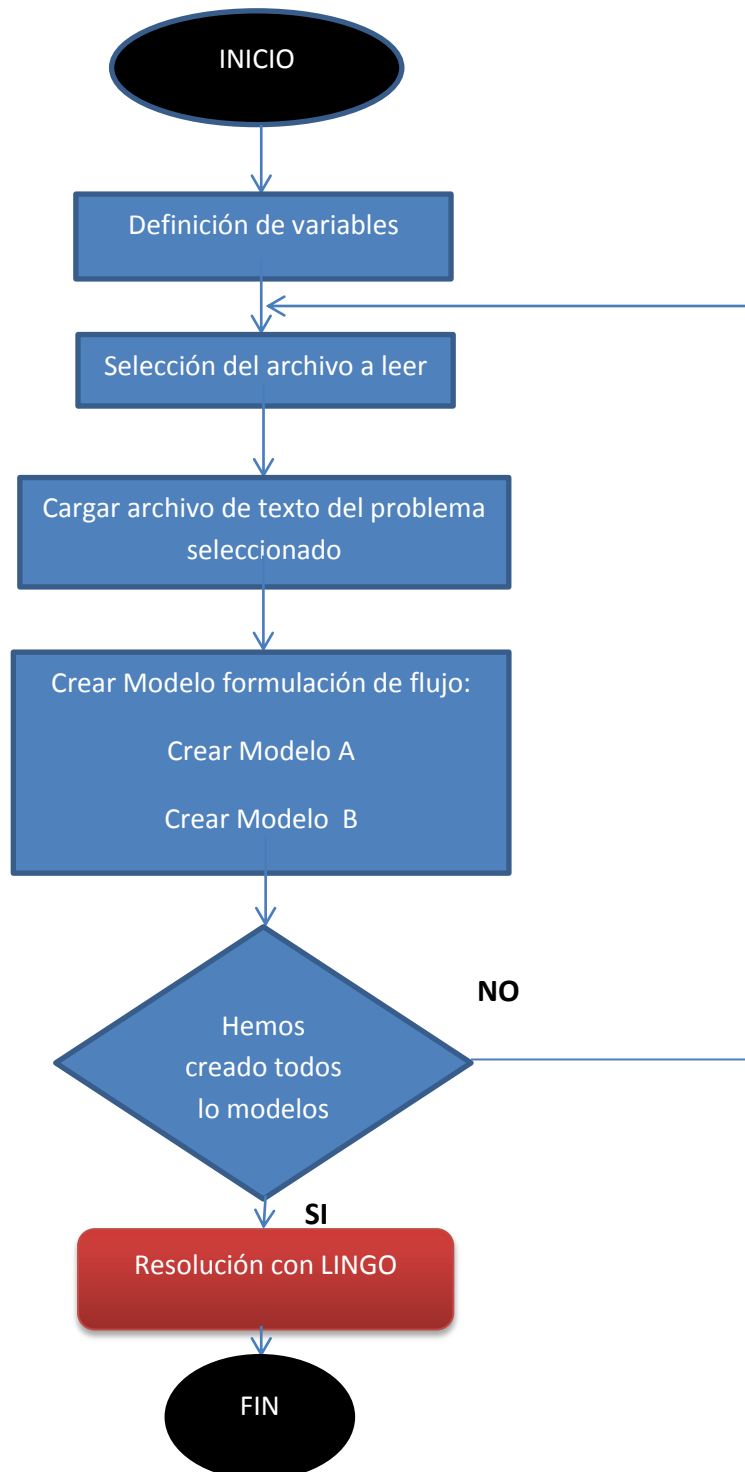
Para una presentación clara de las diferentes funciones usadas en Visual Basic "VBA", se hará uso de diagrama de flujo de datos, los cuáles permitirán representar gráficamente la programación usada. [1]

En el Anexo 3 aparece una descripción de las diferentes funciones creadas.

Estrategia 1. Formulación de Flujo

Se procederá a explicar la programación usada para generar los archivos de entrada de LINGO y se verá un ejemplo del archivo de entrada para cada uno de los modelos desarrollados en la sección anterior.

Seguidamente, se muestra el diagrama de flujo de como funciona la aplicación utilizada para generar los modelos de formulación de flujo para el Modelo A y el Modelo B.



Cuando iniciamos el programa lo primero que se hace es definir todas las variables que hacen falta para almacenar la información del grafo.

A continuación se selecciona el problema que vamos a procesar.

Con el modulo Cargar archivo del problema se lee el archivo .txt, donde se almacena la información del grafo y se carga esa información en las variables del programa.

Con el modulo Crear Modelo de formulación de flujo, se escribe el modelo matemático en un fichero .lgr que posteriormente resolverá LINGO.

Un ejemplo de los archivos de entrada generados está en el Anexo 2 Modelo A y Modelo B.

Se repetirán todos estos pasos hasta escribir todos estos modelos y por último, se procederá a resolver todos los modelos generados.

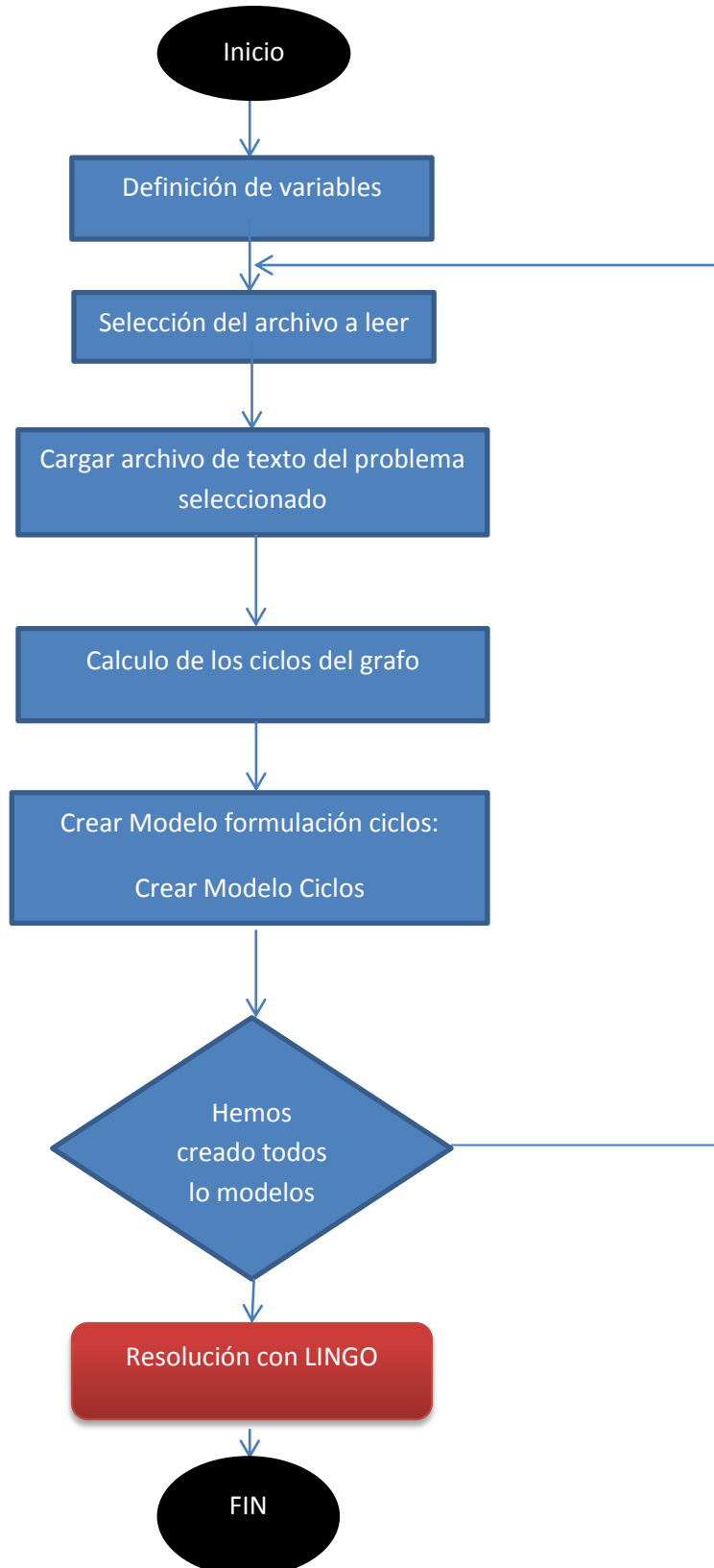
La programación utilizada se puede encontrar en el Anexo 3 de este documento, en el apartado de programación estrategia 1.

Como se puede ver hay un cambio de notación con respecto a la formulación indicada a la hora de desarrollar el modelo, para nombrar los arcos y los costes de estos. Antes representábamos las variables de los arcos como y_{ij} , desde “i” hasta “j”, ahora simplemente los nombraremos como y_i siendo $i=1$ hasta n , siendo n el número de arcos a representar. Aunque no se perderá el sentido físico de esta variable.

De esta forma también hay un cambio de notación en los costes asociados a los arcos C_{ij} , que ahora pasará a ser C_i .

Estrategia 2. Formulación de Ciclos

A continuación, se muestra el diagrama de flujo de como funciona la aplicación utilizada para generar los modelos de formulación de ciclos.



Como se observa, la diferencia entre esta manera de programar y la manera que se utilizó en la programación de flujo, estriba en la función que crea el modelo y en la adicción de un modulo llamado cálculo de ciclos.

El módulo cálculo de ciclos llama a la función que realiza el cálculo de los ciclos del grafo sometido a estudio y grabará los ciclos calculados en las variables globales del programa previo al modulo de generación del modelo.

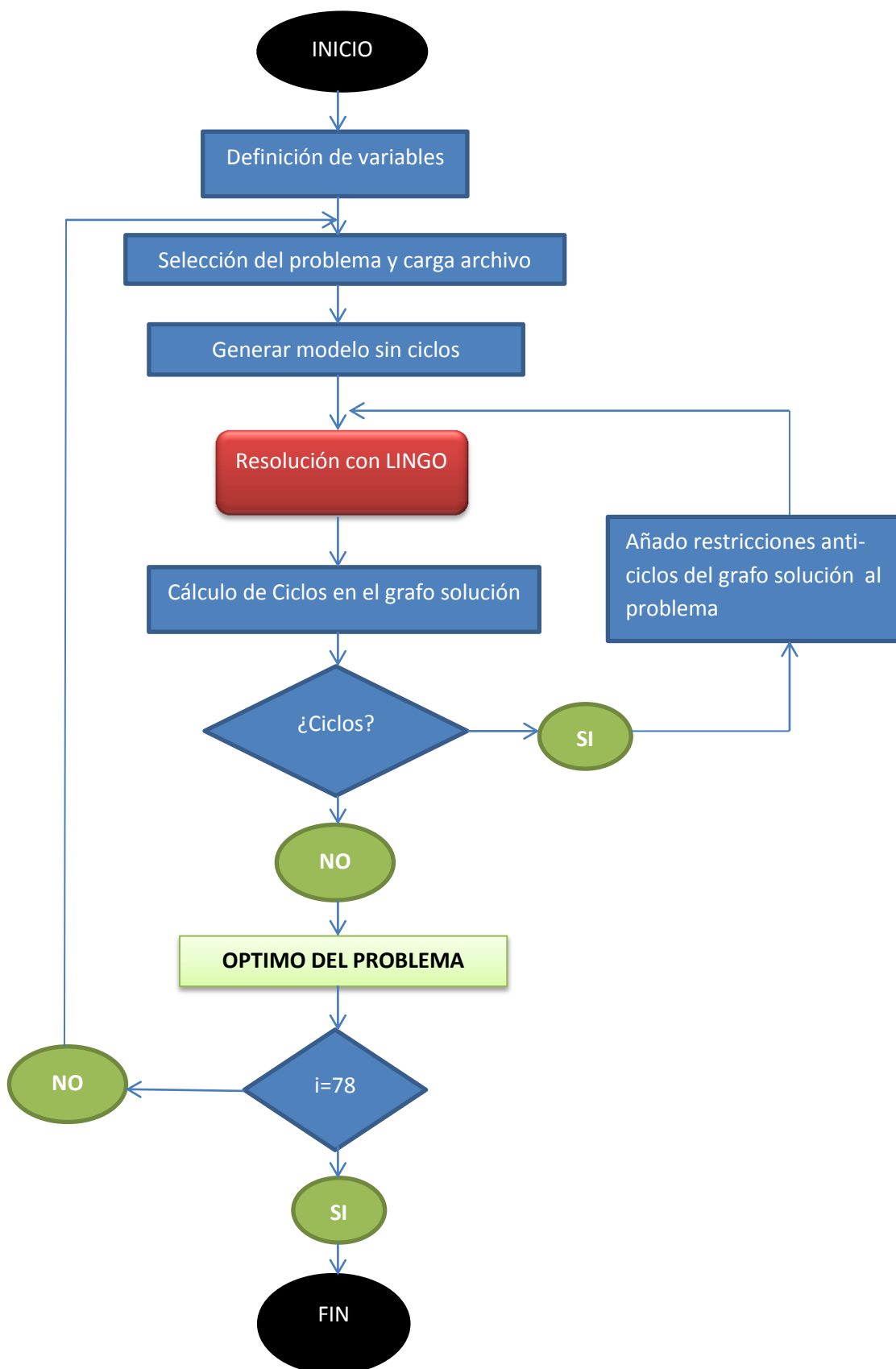
La función Crear Modelo Ciclos creará los archivos de entrada de LINGO con los modelos plantados para la estrategia 2, un ejemplo de los archivos de entrada generados se puede encontrar en el Anexo 2, modelo de ciclos.

Por último, se resolverán los archivos generados con LINGO.

La programación utilizada se puede encontrar en el Anexo 3 de este documento, en el apartado de programación estrategia 2.

Estrategia 3 Formulación ciclos resolución iterativa

Se muestra el diagrama de flujo de como funciona la aplicación utilizada para generar los modelos de formulación de ciclos resuelto de forma iterativa.



Cuando se inicia el programa, lo primero a realizar es definir todas las variables que nos hacen falta para almacenar la información del grafo.

A continuación se selecciona el problema que se va a procesar.

Cargar archivo del problema, realiza una lectura del archivo .txt dónde se almacena la información del grafo y se carga esa información en las variables del programa.

Se genera el modelo sin restricciones anti-bucle con el módulo generar modelo sin ciclos y se resuelve este modelo con LINGO.

Con el módulo Cálculo de Ciclos en el grafo solución, se calculan los ciclos de la solución del modelo anterior.

Si hay ciclos, se incorporan dichas restricciones al modelo inicial y se inicia el proceso de nuevo desde el módulo de resolución con LINGO.

Si no los hay, se habrá hallado la solución óptima de ese problema y se pasará al siguiente problema de la lista.

En el ANEXO 2 modelo de formulación ciclos iterativos, encontramos el archivo que se genera al terminar toda la secuencia de programación. Donde ya están agregadas todas las restricciones anti-bucle.

En el Anexo 3, estrategia 3 de resolución modelos de ciclos iterativos, se encuentra la programación desarrollada con VBA.

A continuación se explica como funciona la librería de optimización llamada LINGO.

LINGO

Introducción a LINGO

LINGO es una potente herramienta que sirve para resolver problemas lineales y no lineales extensos, de forma breve y analizar las solución.

En la siguiente imagen se observa la ventana de entrada al programa cuando se abre por primera vez.

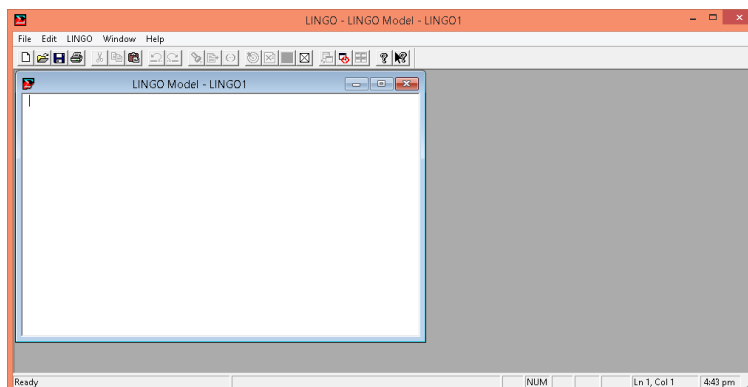


Fig4.1 Ventana de Entrada LINGO

Fuente: Elaboración propia.

La ventana grande es del programa, en la parte superior se ven las diferentes opciones que ofrece el programa.

La ventana pequeña con titulo LINGO-1 es una plantilla en blanco donde se escribe el modelo a resolver.

Implementación del modelo en LINGO

Lo primero que se debe hacer es escribir la función objetivo, la cuál será de maximizar o minimizar. Se escribirá MAX o MIN respectivamente en mayúscula seguido de la función objetivo como en la Fig. 1.7.

$$\text{MIN} = 41 + 7*\text{Beta1} + 1*\text{Beta2} + 7*\text{Beta3} + 2*\text{Beta4} + 2*\text{Beta5} + 2*\text{Beta6}$$

Fig4.2 F.O LINGO

Fuente: Elaboración propia.

A continuación se escriben las restricciones.

Cuando se finaliza de escribir una restricción siempre se ha de terminar con “;”. Ya que si no LINGO no detectará la restricción y dará problemas a la hora de resolver el modelo.

En la imagen de la Fig. 1.8 se observa un ejemplo de escritura de restricciones.

```
X20 + X3 + X12 = X1 + X22 + X31 ;
X9 + X11 + X13 = X28 + X30 + X32 ;
X21 + X22 + X7 = X2 + X3 + X26 ;
7 + X23 + X5 = X4 + X24 ;
X1 + X6 = X20 + X25 + 1 ;
X24 + X25 + X26 + X14 = X5 + X6 + X7 + X33 + 1 ;
X10 + X18 = X29 + X37 + 1 ;
X4 + X27 = X23 + X8 + 1 ;
X2 + X28 + X29 + X19 = X21 + X9 + X10 + X38 ;
X8 + X30 + X15 = X27 + X11 + X34 + 1 ;
X31 + X32 + X16 = X12 + X13 + X35 + 1 ;
X33 + X34 + X35 + X36 + X37 = X14 + X15 + X16 + X17 + X18 ;
X17 + X38 = X36 + X19 + 1 ;
```

Fig4.3 Imagen de las restricciones LINGO

Fuente: Elaboración propia

Por último, se da la definición de las variables.

Si no se indica nada, LINGO asumirá que las variables pueden tomar cualquier valor dentro de los números reales. Si no tendremos que hacer la definición de variables.

Se puede observar en la Fig1.9 las variables que se definen son binarias, es decir, solo pueden tomar valores 0 o 1.

```
@BIN(Beta1) ;
@BIN(Beta2) ;
@BIN(Beta3) ;
@BIN(Beta4) ;
@BIN(Beta5) ;
@BIN(Beta6) ;
@BIN(Beta7) ;
@BIN(Beta8) ;
@BIN(Beta9) ;
@BIN(Beta10) ;
@BIN(Beta11) ;
@BIN(Beta12) ;
@BIN(Beta13) ;
@BIN(Beta14) ;
@BIN(Beta15) ;
@BIN(Beta16) ;
@BIN(Beta17) ;
@BIN(Beta18) ;
@BIN(Beta19) ;
```

Fig4.4 Restricciones LINGO

Fuente: Elaboración propia

Resolución del modelo con LINGO

Una vez escrito todo el problema, es hora de resolver el modelo presionando el botón de Solve. Una vez hecho esto LINGO cargará el modelo y el programa determinará si toda la sintaxis que hemos hecho es correcta o no.

Si no es correcta la sintaxis que hemos hecho, el programa nos mostrará una ventana mostrando el error que se ha cometido, así como de que tipo es, usando un código interno del programa.

Si se ha escrito el modelo de forma correcta, LINGO comenzará a resolverlo.

En la siguiente imagen se puede ver la ventana que muestra el programa cuando está resolviendo o ha resuelto el problema.



Fig4.5 Restricciones LINGO

Fuente: Elaboración propia

En esta ventana muestra un apartado llamado Solver Status que da información del problema.

En la siguiente imagen se muestra un esquema de la información aportada por LINGO.

CAMPO	DESCRIPCION
<i>Model Class</i>	Muestra la clasificación del modelo. Las clases son: "LP", "QP", "ILP", "IQP", "PILP", "PIQP", "NLP" Y "PILP".
<i>State</i>	Da información del estado de la solución, los posibles estados son "Global Optimun", "Local Optimum", "Feasible", "Infisible"

	,"Unbounden", "Interrupted", and "Undetermined".
<i>Objective</i>	Nos muestra el valor actual de la función objetivo
<i>Infeasibility</i>	El valor por el cual no estamos cumpliendo alguna de las restricciones.
<i>Iteration</i>	Numero de Iteraciones hechas por el programa.

Campo model Class nos encontramos las propiedades del modelo, las cuales se encuentran listadas en la siguiente tabla:

ABREVIACION	CLASE	DESCRIPCION
LP	Programación Lineal	Todas las expresiones son lineales y el modelo contiene en las restricciones variables no enteras
QP	Programación Cuadrática	Todas las expresiones son lineales o cuadráticas, el modelo es convexo y no contiene restricciones con variables no enteras
ILP	Programación Lineal Entera	Todas las expresiones son lineales y varias expresiones de las restricciones contienen variables enteras
IQP	Programación Entera cuadrática	Todas las expresiones son lineales o cuadráticas, el modelo es convexo y las restricciones contienen algunas variables enteras.
PILP	Programación lineal pura entera	Todas las expresiones son lineales y todas las variables que forman parte de las restricciones son valores enteros
PIQP	Programación cuadrática entera pura	Todas las expresiones son lineales o cuadráticas, el modelo es convexo y todas las variables de las restricciones son enteras
NLP	Programación No-Lineal	Al menos una de las relaciones en el modelo es no-lineal con respecto a las variables
INLP	Programación Entera No-Lineal	Al menos una de las expresiones del modelo es no lineal y un conjunto de variables de las restricciones son enteras.
PINLP	Programación No-Lineal entera pura	Al menos una expresión del modelo es no Lineal y todas las variables que forman parte de las restricciones son enteras

Solución LINGO

Cuando LINGO termina de iterar, o lo detenemos, el programa presenta un informe de la solución encontrada, cómo se muestra en la siguiente imagen de la fig1.10:

```
Global optimal solution found.
Objective value:                104.0000
Extended solver steps:          2
Total solver iterations:        762

Model Title: MBPM steinb8.txt

      Variable      Value      Reduced Cost
      X1             1.000000      6.000000
      X2             1.000000      1.000000
      X3             0.000000      4.000000
      X4             1.000000      9.000000
      X5             1.000000      3.000000
      X6             1.000000      3.000000
      X7             0.000000      6.000000
      X8             0.000000      7.000000
      X9             0.000000      2.000000
      X10            1.000000      9.000000
      X11            1.000000      3.000000
      X12            0.000000      8.000000
      X13            0.000000      9.000000
      X14            1.000000      2.000000
      X15            1.000000      5.000000
```

Fig4.5 Informe LINGO

Fuente: Elaboración propia

El programa muestra si ha encontrado una solución óptima o no, seguido del valor de la función objetivo, las iteraciones totales del problema y también se muestra para cada variable en una tabla su valor y su coste reducido.

El coste reducido se puede interpretar de dos formas de diferentes, una de ellas sería el coste que le cuesta a la función objetivo que esa variable entre a formar parte de la solución del problema. [2]

RESULTADOS EXPERIMENTALES E INTERPRETACIÓN

Para el análisis y comprobación de los modelos propuestos, se han utilizado los problemas que aparecen en la OR library [7] mantenida por J.Beasley, en la pagina web del departamento de investigación operativa, del Imperial College de Londres.

Características de los problemas

Estos problemas se clasifican en función de su complejidad y dimensión, en las clases "b","c","d" y "e", siendo estos últimos los más complejos y por consiguiente los que mayor tiempo de resolución requerirán.

En la siguiente tabla de la fig5.1 se puede observar la lista de problemas que se van a someter a estudio, así como algunos parámetros del problema: Nº de nodos, nº de arcos y nº de nodos terminales.

Problema	Nº nodos	Nº Arcos	Nº Terminales
Steinb1	50	63	9
Steinb2	50	63	13
Steinb3	50	63	25
Steinb4	50	100	9
Steinb5	50	100	13
Steinb6	50	100	25
Steinb7	75	94	13
Steinb8	75	94	19
Steinb9	75	94	38
Steinb10	75	150	13
Steinb11	75	150	19
Steinb12	75	150	38
Steinb13	100	125	17
Steinb14	100	125	25
Steinb15	100	125	50
Steinb16	100	200	17
Steinb17	100	200	25
Steinb18	100	200	50
Steinc1	500	625	5
Steinc2	500	625	10
Steinc3	500	625	83
Steinc4	500	625	125
Steinc5	500	625	250
Steinc6	500	1000	5
Steinc7	500	1000	10
Steinc8	500	1000	83

ANÁLISIS DE ESTRATEGIAS DE MODELADO PARA LA RESOLUCIÓN DEL PROBLEMA DE STEINER EN GRAFOS

Steinc9	500	1000	125
Steinc10	500	1000	250
Steinc11	500	2500	5
Steinc12	500	2500	10
Steinc13	500	2500	83
Steinc14	500	2500	125
Steinc15	500	2500	250
Steinc16	500	12500	5
Steinc17	500	12500	10
Steinc18	500	12500	83
Steinc19	500	12500	125
Steinc20	500	12500	250
Steind1	1000	1250	5
Steind2	1000	1250	10
Steind3	1000	1250	167
Steind4	1000	1250	250
Steind5	1000	1250	500
Steind6	1000	2000	5
Steind7	1000	2000	10
Steind8	1000	2000	167
Steind9	1000	2000	250
Steind10	1000	2000	500
Steind11	1000	5000	5
Steind12	1000	5000	10
Steind13	1000	5000	167
Steind14	1000	5000	250
Steind15	1000	5000	500
Steind16	1000	25000	5
Steind17	1000	25000	10
Steind18	1000	25000	167
Steind19	1000	25000	250
Steind20	1000	25000	500
Steine1.txt	2500	3125	5
Steine2.txt	2500	3125	10
Steine3.txt	2500	3125	417
Steine4.txt	2500	3125	625
Steine5.txt	2500	3125	1250
Steine6.txt	2500	5000	5
Steine7.txt	2500	5000	10
Steine8.txt	2500	5000	417
Steine9.txt	2500	5000	625
Steine10.txt	2500	5000	1250

Steine11.txt	2500	12500	5
Steine12.txt	2500	12500	10
Steine13.txt	2500	12500	417
Steine14.txt	2500	12500	625
Steine15.txt	2500	12500	1250
Steine16.txt	2500	62500	5

Fig5.1 Tabla de problemas Steiner

Fuente: Elaboración propia

En la gráfica de la fig5.2 se representan las características del problema.

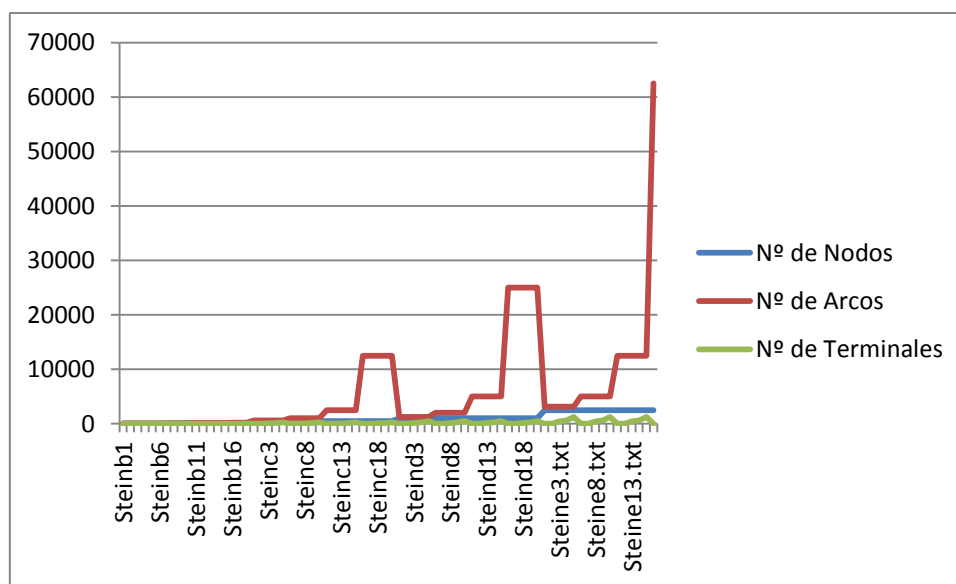


Fig5.2 Grafica de propiedades del problema

Fuente: Elaboración propia

Como puede observarse no siguen una relación lineal, aunque existe una clara tendencia al alza en el número de nodos, número arcos y número de nodos terminales a medida que avanzamos hacia la derecha en el eje de abscisas de la gráfica.

En la gráfica de la fig1.12 se ha representado el número de nodos totales y el número de nodos terminales.

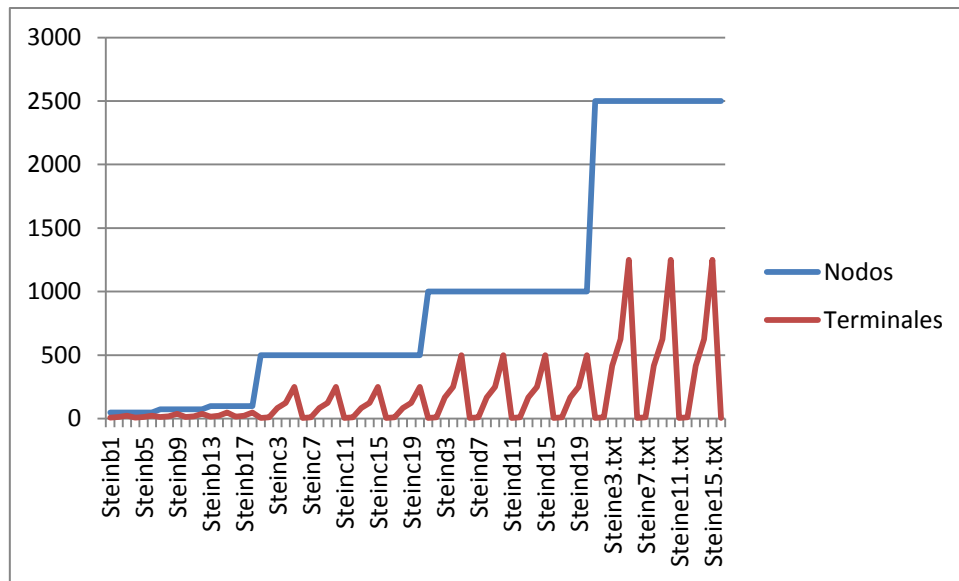


Fig5.3 Grafica del número de Nodos y número de Terminales.

Fuente: Elaboración propia

El número de nodos terminales sigue una gráfica con forma de sierra, mientras el número de nodos del problema crece de forma escalonada, donde hay un escalón por cada tipo de problema, aumentando el número de nodos a medida que nos movemos hacia la derecha según el eje abscisas. Estos dos hechos hacen que la densidad de nodos terminales aumente y disminuya de un problema a otro de forma repetitiva.

La biblioteca de problemas se someterá a un proceso de reducción, con el propósito de hacer los tiempos de resolución lo mas óptimos posibles, al disminuir el número de variables de los problemas.

En la tabla de la fig5.5 se muestran los datos de los problemas reducidos: nº nodos, nº de arcos y nº de terminales.

PROBLEMA	Nº Nodos	Nº Arcos	Nº Terminales
steinb1.txt	13	19	8
steinb2.txt	15	21	11
steinb3.txt	20	25	15
steinb4.txt	40	80	9
steinb5.txt	39	80	12
steinb6.txt	45	87	25
steinb7.txt	22	33	11
steinb8.txt	26	38	15
steinb9.txt	27	35	23
steinb10.txt	55	121	13

ANÁLISIS DE ESTRATEGIAS DE MODELADO PARA LA RESOLUCIÓN DEL PROBLEMA DE STEINER EN GRAFOS

steinb11.txt	63	129	19
steinb12.txt	63	125	36
steinb13.txt	36	56	14
steinb14.txt	42	65	21
steinb15.txt	48	69	38
steinb16.txt	77	166	17
steinb17.txt	74	153	23
steinb18.txt	82	166	45

Fig5.5 Tabla de problemas Steinb reducidos.

Fuente: Elaboración propia

En la gráfica de la fig5.6, puede observarse la relación entre número nodos y número nodos terminales del problema reducido, que ya no sigue una tendencia escalonada como en el problema sin reducir.

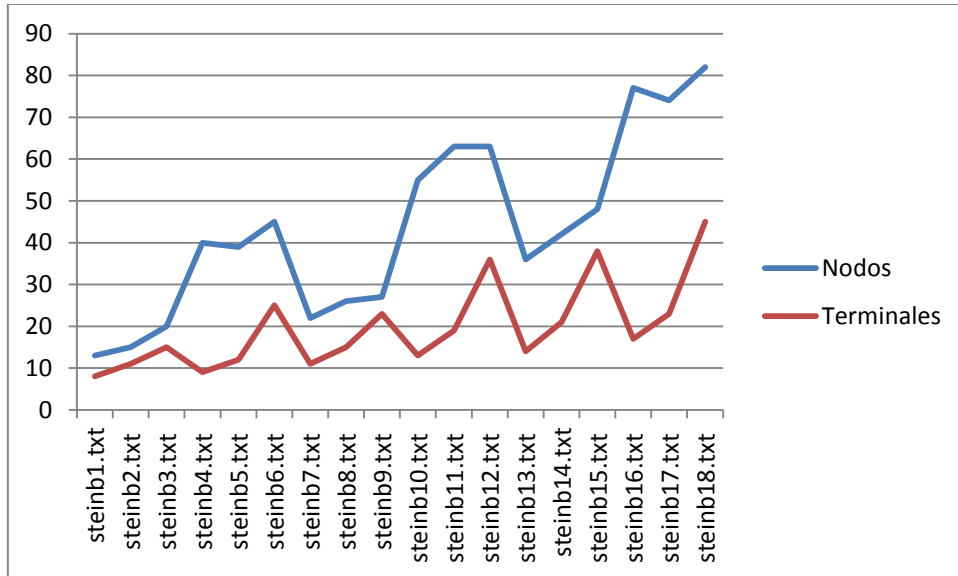


Fig5.6 Gráfica del número de Nodos y número de Terminales del problema reducido.

Fuente: Elaboración propia

En la gráfica de la fig5.7 están representados el número de nodos del problema original y el número de nodos del problema reducido.

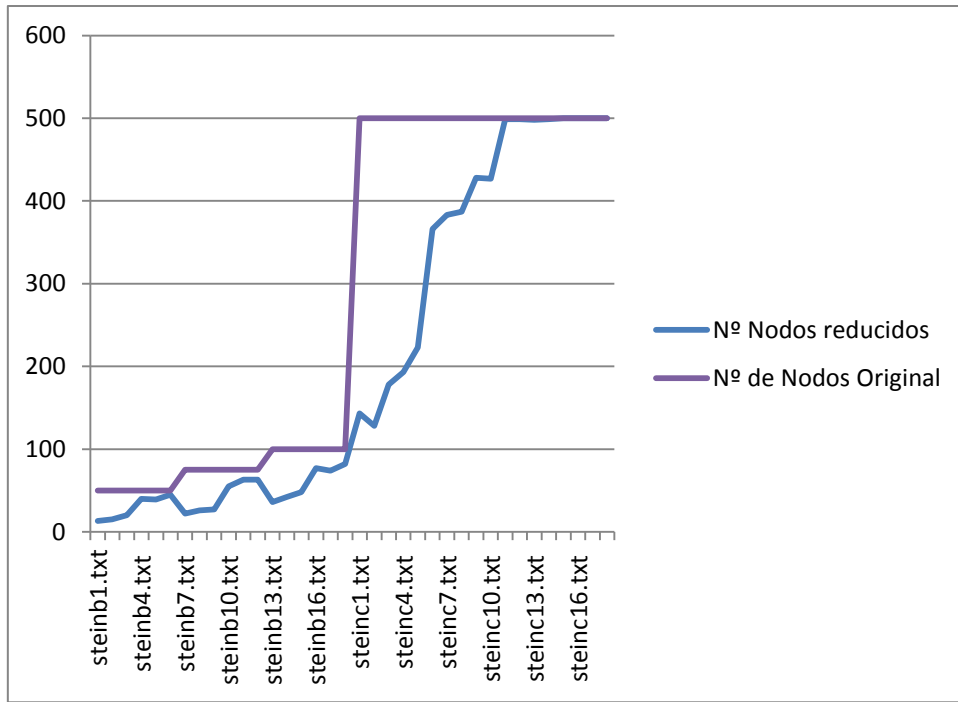


Fig5.7 Gráfica de nº de nodos del problema original y del problema reducido.

Fuente: Elaboración propia

En la gráfica de la fig5.7 están representados el número de arcos del problema reducido y el número de arcos del problema original.

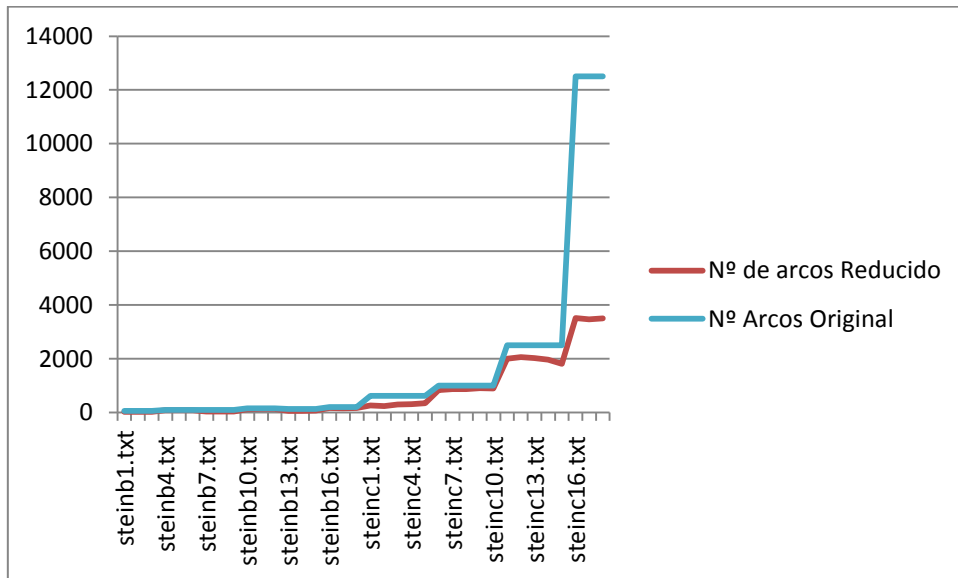


Fig5.8 Gráfica del nº de arcos del problema original y reducido.

Fuente: Elaboración propia

Resultado de los modelos

En este apartado se muestran los resultados obtenidos por LINGO para las diferentes estrategias de resolución explicadas.

Debido al carácter del problema, las simulaciones de la biblioteca de optimización las dejaremos un tiempo máximo de 24 horas por cada problema de librería de optimización OR-Library [7].

Los datos que se muestran para cada problema son: el valor de la función objetivo, las iteraciones que ha tenido que hacer LINGO para resolver el problema, los pasos extensos de resolución y el tiempo de resolución.

Se comenzarán por mostrar los resultados para los modelos de formulación de flujo, modelo A y modelo B.

Estrategia 1. Formulación de flujo

Modelo A

Los resultados obtenidos para la estrategia de resolución formulación de flujo modelo A, se muestran en la tabla de la fig5.8.

MA_stein	Objetive value:	Extended solver steps:	Total solver iterations:	time
steinb1.txt	82	119	3051	1
steinb2.txt	83	69	9007	3
steinb3.txt	138	675	35075	9
steinb4.txt	59	6482	294460	78
steinb5.txt	61	70426	2936347	432
steinb6.txt				
steinb7.txt	111	436	29835	6
steinb8.txt	104	2550	102384	19
steinb9.txt	222	4040	131012	31
steinb10.txt	86	1101590	82057263	8117

Fig5.8 Tabla de resultados estrategia 1 Modelo A

Fuente: Elaboración propia

Del problema steinb1 al steinb9 los tiempos de resolución son del orden de minutos, con la excepción del steinb6, el cuál no obtuvo solución óptima tras las 24 horas de tiempo de cálculo máximo. El problema steinb10 obtuvo un tiempo de 12 horas aproximadamente.

A partir del steinb10 no se obtuvo una solución de ningún problema más.

Esto es lo normal debido al carácter de este problema, NP-Completo.

Modelo B

Los resultados obtenidos para la estrategia de resolución formulación de flujo modelo B, se muestran en la tabla de la fig5.9.

MB_Stein	Objective value:	Extended solver steps:	Total solver iterations:	time
steinb1.txt	82	114	2540	1
steinb2.txt	83	645	6543	1
steinb3.txt	138	576	14042	2
steinb4.txt	59	9238	386934	63
steinb5.txt	61	84354	3625780	787
steinb6.txt				
steinb7.txt	111	1511	27305	4
steinb8.txt	104	2014	94680	22
steinb9.txt	220	16415	338505	39
steinb10.txt	86	4193312	452729095	44784

Fig5.9 Tabla de resultados estrategia 1 modelo B

Fuente: Elaboración propia

Del problema steinb1 al steinb9 los tiempos de resolución son del orden de minutos, con la excepción del steinb6, el cual no obtuvo solución óptima tras las 24 horas de tiempo de cálculo máximo. El problema steinb10 obtuvo un tiempo de alrededor de 12 horas. A partir del steinb10 no se obtuvo una solución de ningún problema más al igual que en el caso anterior.

A continuación se verá en la gráfica de la fig5.10, una comparativa de tiempos entre los dos modelados propuestos.

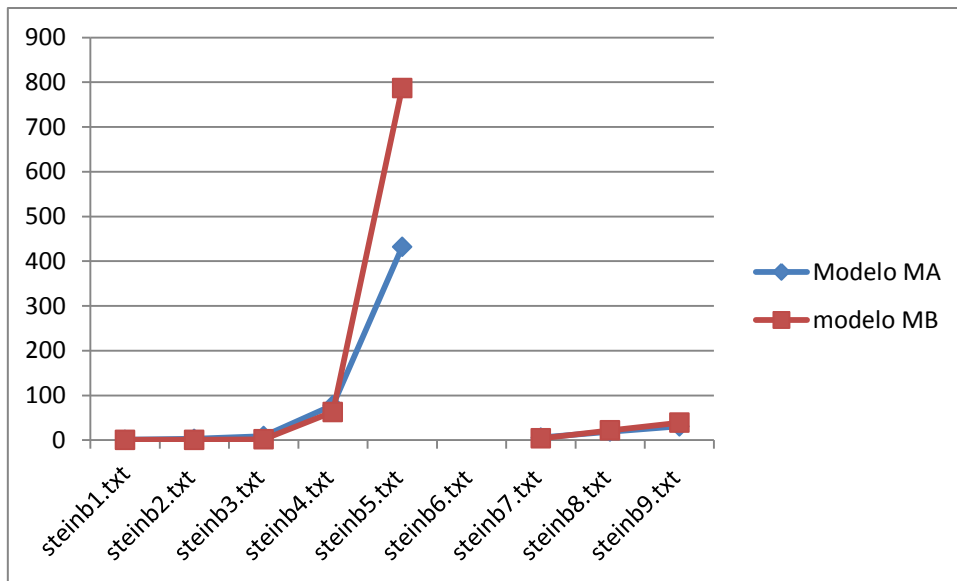


Fig5.10 Gráfico de los resultados para los modelos MA y MB

Fuente: Elaboración propia

Tanto el modelo A, como el modelo B tienen tiempos de resolución similares, aunque puede observarse que el modelado “B”, es más lento que el modelo “A” para los problemas steinb7 en adelante, sin embargo, para los problemas del steinb1 al steinb4 sucede lo contrario. A priori no era de esperar ya que el modelo B tiene el doble de restricciones, al tenerlas desagregada.

Resaltar que este modelado de forma general es poco efectivo, por qué a medida que aumenta la complejidad del problema deja de dar una respuesta en un tiempo razonable.

A continuación se verá la representación gráfica del resultado del problema steinb10, que ha sido representado aparte para no perder el orden de magnitud en el gráfico de la fig. 5.10.

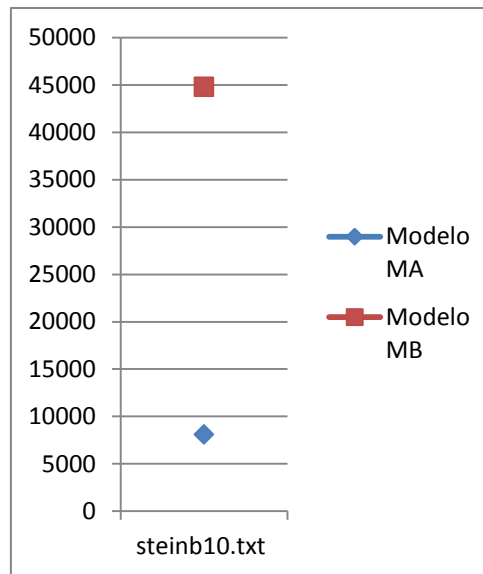


Fig5.11 Gráfico de los tiempos para el problema steinb10.

Fuente: Elaboración propia

Como puede observarse los tiempos de resolución son de 2:30:00 para el modelo A y de 12:00:00 para el modelo B.

Recordemos el modelado de las restricciones de los arcos de flujo para cada uno de los modelos representados.

Modelo A

- $X_{ij} + X_{ji} \leq (N * T - 1) * y_{ij}$

Modelo B

- $X_{ij} \leq (N * T - 1) * y_{ij}$
- $X_{ji} \leq (N * T - 1) * y_{ij}$

Como era de esperar, el modelo B añade complejidad al modelo al tener el doble de restricciones de este tipo, que el modelo A. Sin embargo para los problemas con un número de nodos bajo correspondiente a los problemas del steinb1 al steinb4, el tener el doble de restricciones hace que los tiempos de resolución sean más bajos.

Estrategia 2. Formulación ciclos

En este apartado se mostrarán los resultados obtenidos para la estrategia 2, formulación de ciclos.

Al probar con el modelo de ciclos, se puede comprobar que a partir del tercer problema del primer bloque, obtener el modelo del problema para implementarlo en LINGO es casi imposible, por qué al tener que calcular todos los ciclos que puede contener el grafo, el cálculo de esta restricción hace que VBA se atasque y no podamos seguir sacando los modelos de los problemas propuestos.

En la tabla de la fig5.12, se puede observar los tiempos de resolución para la estrategia 2.

PROBLEMA	FORMULACION DE FLUJO
stein_b1	3
stein_b2	6
stein_b3	11

Fig5.12 Tabla de tiempos estrategia 2 Modelo de ciclos.

Fuente: Elaboración propia

A partir del problema steinb4 no se pudo obtener una solución.

En la fig5.12 se puede observar la representación gráfica de los tiempos de resolución para los problemas resueltos mediante la estrategia 2.

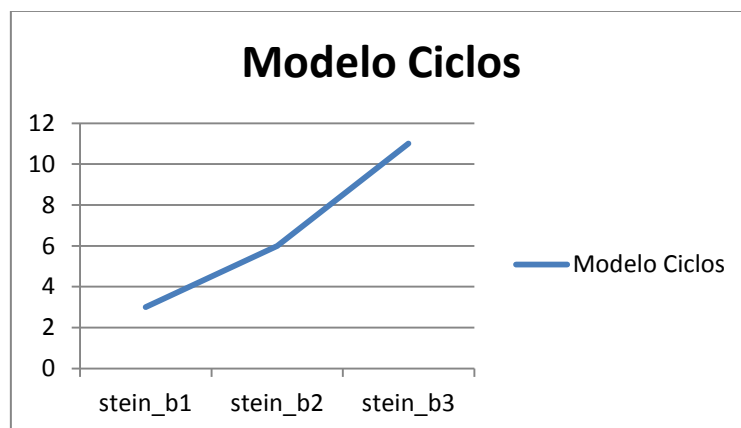


Fig5.12 Gráfico de los tiempos para modelo de ciclos.

Fuente: Elaboración propia

Esta estrategia de resolución es ineficaz para problemas con una cierta dimensión, además es más lenta que la estrategia 1 de resolución para estos tres problemas resueltos.

Estrategia 3. Formulación ciclos resolución iterativa

En este apartado se mostrarán los resultados obtenidos por LINGO para la estrategia 3, formulación ciclos resolución iterativa.

En la siguiente tabla de la fig5.13, se muestran los resultados obtenidos para el primer bloque de problemas.

Problema	Densidad STEINER	nº de iteraciones	tiempos de resolución
Steinb1	0,82	0	0
Steinb2	0,74	0	0
Steinb3	0,5	1	0
Steinb4	0,82	1	0
Steinb5	0,74	1	4
Steinb6	0,5	14	7
Steinb7	0,826666667	1	0
Steinb8	0,746666667	7	7
Steinb9	0,493333333	1	1
Steinb10	0,826666667	0	0
Steinb11	0,746666667	15	18
Steinb12	0,493333333	0	0
Steinb13	0,83	1	2
Steinb14	0,75	2	2
Steinb15	0,5	10	14
Steinb16	0,83	1	0
Steinb17	0,75	23	20
Steinb18	0,5	24	26
Steinc1	0,99	1	0
Steinc2	0,98	1	3
Steinc3	0,834	754	21781
Steinc4	0,75		
Steinc5	0,5		
Steinc6	0,99	4	298
Steinc7	0,98	0	6
Steinc8	0,834		
Steinc9	0,75		
Steinc10	0,5		
Steinc11	0,99	2	381
Steinc12	0,98	6	366
Steinc13	0,834		
Steinc14	0,75		
Steinc15	0,5		

ANÁLISIS DE ESTRATEGIAS DE MODELADO PARA LA RESOLUCIÓN DEL PROBLEMA DE STEINER EN GRAFOS

Steinc16	0,99		
Steinc17	0,98		
Steinc18	0,834		
Steinc19	0,75		
Steinc20	0,5		
Steind1	0,995	1	3
Steind2	0,99	0	1
Steind3	0,833		
Steind4	0,75		
Steind5	0,5		
Steind6	0,995	0	205
Steind7	0,99	2	410
Steind8	0,833		
Steind9	0,75		
Steind10	0,5		
Steind11	0,995		
Steind12	0,99		
Steind13	0,833		
Steind14	0,75		
Steind15	0,5		
Steind16	0,995		
Steind17	0,99		
Steind18	0,833		
Steind19	0,75		
Steind20	0,5		
Steine1.txt	0,998	3	26
Steine2.txt	0,996	0	71

Fig5.13 Tabla de resultados del modelo de formulación implícita resuelto de forma iterativa

Fuente: Elaboración propia

En la gráfica de la fig5.14, se muestra para cada problema: tiempo de resolución, número de iteraciones y densidad de nodos Steiner.

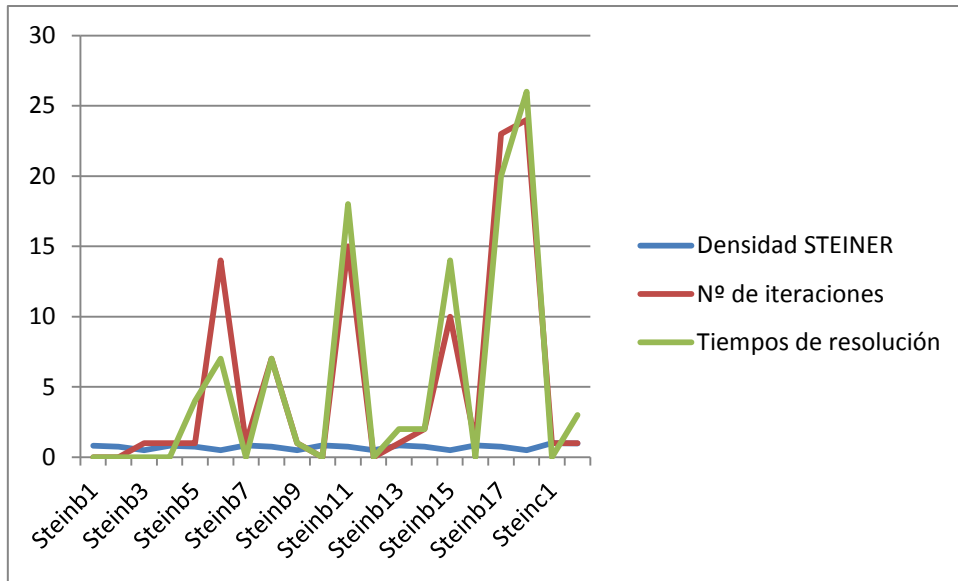


Fig5.14 Gráfico de los tiempos para el bloque de problema steinb.

Fuente: Elaboración propia

Para los problemas del Steinb1 al steinb18 los tiempos de resolución obtenidos no superan los 30 segundos, a partir de del steinb18 algunos problemas dan una solución y otros no para simulaciones de hasta 12 horas.

Notar que los tiempos de resolución mas elevados corresponden a densidad baja de nodos Steiner, por tanto para analizar el resto de bloques “c”, “d”, “e”, se representará en una gráfica la densidad de Steiner, y si se ha encontrado solución.

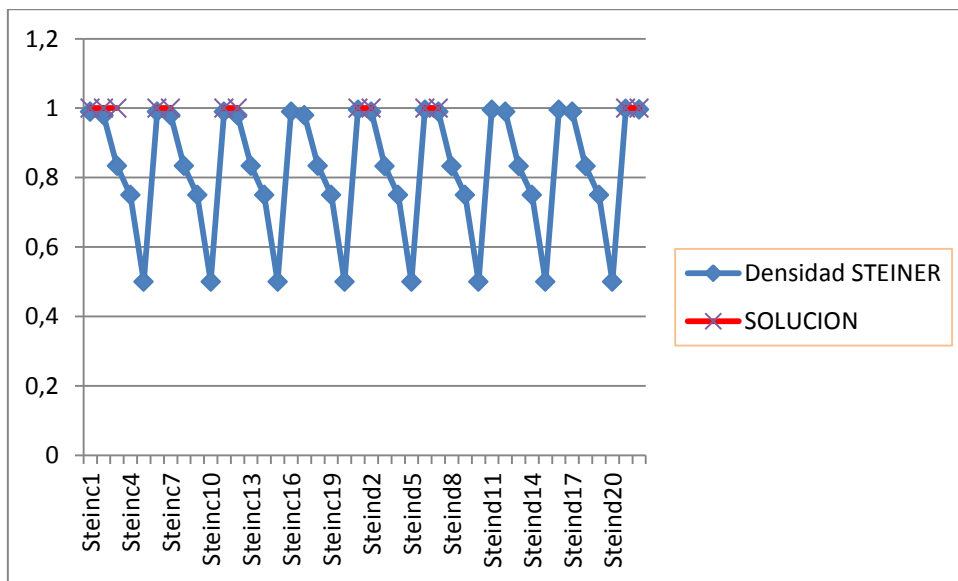


Fig5.15 Gráfico de los tiempos para el bloque de problema steinb.

Fuente: Elaboración propia

Como puede observarse cuando la densidad de nodos Steiner es alta, 90% o superior esta estrategia de resolución, es un método que da soluciones en tiempos de ejecución razonables.

Sin embargo cuando esta densidad baja del 90%, los tiempos de resolución del problema aumentan. Como el caso del problema steinc3, con una densidad de nodos Steiner del 84% y un tiempo de resolución de unas 6 horas aproximadamente.

De forma que se pueden clasificar los problemas según la densidad de nodos Steiner con respecto a los tiempos de resolución:

- Densidad de nodos Steiner 90% o superior:
Problemas de resolución rápida, del orden de minutos
- Densidad de nodos Steiner desde 75% hasta el 90%:
Problemas con tiempos de resolución moderados, del orden de horas
- Densidad de nodos Steiner desde 50 % hasta 75%:
Problemas con tiempos de resolución elevados del orden de días o semanas.

Pasaremos a analizar los tiempos de resolución en la primera iteración de cada problema, con esto se logra resolver el problema de forma parcial consiguiendo una solución aproximada.

En la siguiente gráfica de la fig5.16 podemos observar los tiempos de resolución para la primera iteración.

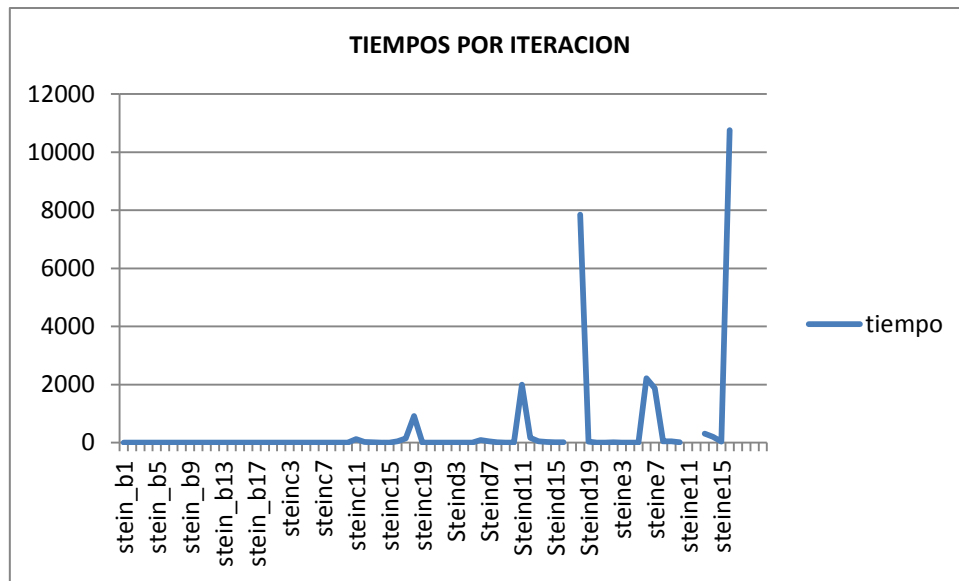


Fig5.16 Grafico de los tiempos para el problema steinb10.

Fuente: Elaboración propia

Los tiempos de resolución por iteración no son demasiado elevados en la mayoría de los casos aunque en ciertos problemas este tiempo se dispara incluso a horas.

En la gráfica de la fig5.17 se verán los errores cometidos para cada problema con la primera iteración.

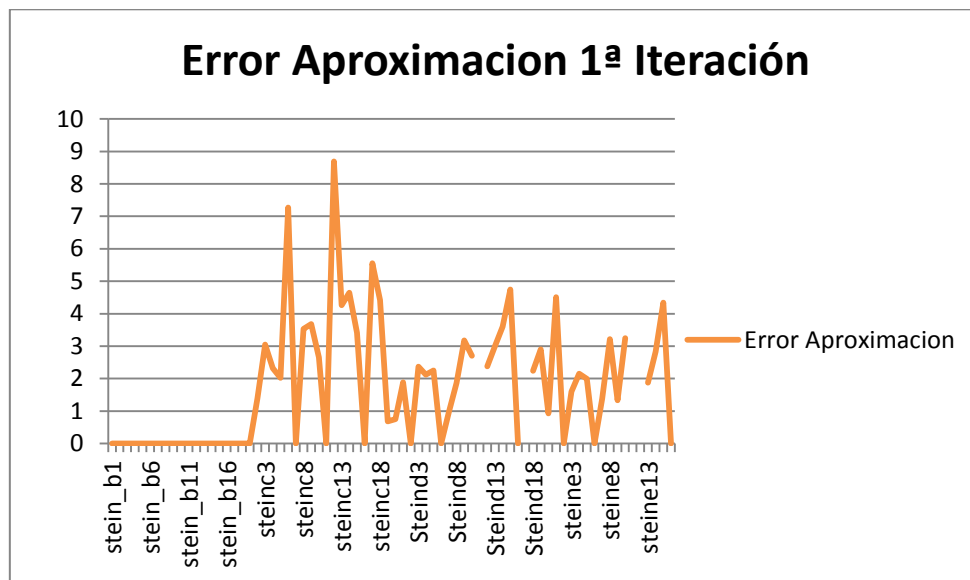


Fig5.17 Gráfico propiedades de los problemas.

Fuente: Elaboración propia

Los errores cometidos no son mayores a un 10% en el peor de los casos y la media del porcentaje de error esta en torno a un 3%.

El valor de la función objetivo para la solución parcial obtenida, es siempre menor o igual a la de la función objetivo en el óptimo del problema.

CONCLUSIONES

Con este proyecto se ha intentado abordar la resolución del problema de Steiner desde un punto de vista diferente a la mayoría de estudios que existen, ya que la mayoría de estudios se centran en métodos heurísticos.

De las tres estrategias presentadas: estrategia 1, formulación de flujo, estrategia 2, formulación de ciclos y estrategia 3, formulación ciclos resolución iterativa. La que mejores resultados presenta es la tercera estrategia de resolución.

La estrategia 1, modelado de flujo presenta tiempos de resolución elevados, ya que los bloques “c”, “d”, “e” tienen tiempos de resolución inoperativos al exceder las 24 horas de tiempo de resolución. Dentro de las variantes planteadas modelo A y modelo B, las dos presentan tiempos similares aunque con algunas discordancias. El modelo A al tener las restricciones agregadas y presentar la mitad de restricciones lo usual sería pensar que tendría tiempos de resolución mas cortos en todos los problemas, aunque esto no pasa para los problemas con un bajo número de nodos.

Con respecto estrategia 2, formulación de ciclos, a partir del problema steinb3 del primer bloque, fue imposible generar el modelo, debido al elevado número de ciclos que se pueden formar en redes de dicho tamaño y que tenemos que calcular antes de empezar a resolver el modelo. Por tanto, podemos concluir que esta estrategia de resolución es poco práctica, ya que solo tiene interés a nivel teórico.

La estrategia 3, formulación de ciclos de resolución iterativa tiene mejores resultados que las estrategias planteadas anteriormente.

Para el primer bloque de problemas “steinb” la estrategia 3, proporciona tiempos de resolución bastante buenos ya que no se excede de los dos minutos de tiempo de resolución. Para los problemas de los bloques “c”, “d”, “e” solo se resuelven de forma rápida aquellos problemas con una densidad de nodos Steiner elevada, donde podemos establecer una clasificación de los problemas según la densidad de nodos Steiner y los tiempos de resolución.

Comentar el hecho de que cuanto más densidad de nodos terminales tiene nuestro problema más se aproxima al problema del mínimo árbol expandido, donde todos los nodos son terminales. Sin embargo cuanto más nos aproximamos a este problema, mayores son los tiempos resolución. El problema

del mínimo árbol expandido o MST (Minimum Expansion Tree) no es un problema NP-Completo y tiene resolución en tiempo polinómico.

La primera iteración de la estrategia 3, es una solución parcial del problema que se podría utilizar para conocer una estimación del problema de Steiner, cometiendo un pequeño error, con un valor de la función objetivo menor o igual al óptimo y tiempos de resolución cortos. Aunque esto no era el objeto inicial del proyecto.

Añadir que estas estrategias no son las únicas que se han investigado en la realización de este proyecto, pero no han tenido cabida en el proyecto al no presentar resultados significativos.

Comentar que para la estrategia 1, formulación de flujo se probaron distintas variaciones de las restricciones, como las vistas entre los modelos A y B que no aportaban tiempos de resolución más cortos, ni información adicional sobre el tratamiento del problema.

Se intentaron proponer otras estrategias basadas en otros modelos matemáticos, como la formulación de cobertura de redes, que no se pudo llevar a cabo debido esta formulación esta orientada a problemas de grafos que se pueden representar gráficamente.

Por último concluir, que dependiendo del modelado que se haga de un problema los resultados pueden variar enormemente para un problema concreto y por tanto, se debe seguir investigando en los modelos matemáticos para resolver este tipo de problemas.

BIBLIOGRAFÍA

1. Visual Basic
Autor: José Eduardo maluf de Carvalho
Editorial: McGraw-Hill
2. LINGO user's guide
Editorial: LINDO SYSTEM INC.
3. Ingeniería de Organización: modelos y aplicaciones.
Autores: Pablo Cortés, Luis Omieva
Editorial: Ediciones Díaz de Santos
4. The Steiner connectivity problema
Autores: Ralf Borndörfer, Marika Karbstein, Marc E. Pfetsch
Editorial: Springer
5. European Journal of Operational Research

Exact approaches for solving robust prize-collecting Steiner tree Problems
Autores: Eduardo Álvarez-Miranda, Ivana Ljubic, Paolo Toth
Año de la publicación: 2012

The Steiner Traveling Salesman Problem with online edge blockages
Autores: HuiliZhang, WeitianTong, YinfengXu, GuohuiLin
Año de la publicación: 2014

Compact formulations of the Steiner Traveling Salesman Problem and related problems
Autores: Adam N. Letchford, D. Nasiri , Dirk Oliver Theis
Año de publicación: 2013
6. <http://www.brunel.ac.uk/~mastjib/jeb/info.html> OR-Library J E Beasley
7. <http://www.iuntadeandalucia.es/institutodeestadisticaycartografia>
8. <https://es.wikipedia.org>

ANEXOS

ANEXO 1 Ejemplo de un fichero de Entrada

Problema Steinb1

13 19 "T=Numero de nodos Terminales", "N=Numero de nodos del Problema"

15 7 "Coste del arco", "i", "j"

3 9 1

3 1 7

4 8 2

6 4 2

6 5 2

6 3 8

8 10 4

9 2 8

9 7 5

10 2 6

11 1 8

11 2 14

12 6 8

12 10 9

12 11 2

12 13 7

12 7 15

13 9 11

8 "Nº de nodos Steiner"

4 5 6 7 8 10 11 13 "Nodos Steiner"

41 "Coste inicial fijo"

ANEXO 2 Archivos de entrada LINGO

Estrategia 1. Modelo A

```
TITLE MBPM steinbl.txt;
MIN = 41 + 7*Beta1 + 1*Beta2 + 7*Beta3 + 2*Beta4 + 2*Beta5 + 2*Beta6 +
8*Beta7 + 4*Beta8 + 8*Beta9 + 5*Beta10 + 6*Beta11 + 8*Beta12 +
14*Beta13 + 8*Beta14 + 9*Beta15 + 2*Beta16 + 7*Beta17 + 15*Beta18 +
11*Beta19;
X20 + X3 + X12 = X1 + X22 + X31 + 1;
X9 + X11 + X13 = X28 + X30 + X32 + 1;
X21 + X22 + X7 = X2 + X3 + X26 + 1;
13 + X23 + X5 = X4 + X24 + 1;
X1 + X6 = X20 + X25 + 1;
X24 + X25 + X26 + X14 = X5 + X6 + X7 + X33 + 1;
X10 + X18 = X29 + X37 + 1;
X4 + X27 = X23 + X8 + 1;
X2 + X28 + X29 + X19 = X21 + X9 + X10 + X38 + 1;
X8 + X30 + X15 = X27 + X11 + X34 + 1;
X31 + X32 + X16 = X12 + X13 + X35 + 1;
X33 + X34 + X35 + X36 + X37 = X14 + X15 + X16 + X17 + X18 + 1;
X17 + X38 = X36 + X19 + 1;
X1 + X20 <= 12*Beta1;
X2 + X21 <= 12*Beta2;
X3 + X22 <= 12*Beta3;
X4 + X23 <= 12*Beta4;
X5 + X24 <= 12*Beta5;
X6 + X25 <= 12*Beta6;
X7 + X26 <= 12*Beta7;
X8 + X27 <= 12*Beta8;
X9 + X28 <= 12*Beta9;
X10 + X29 <= 12*Beta10;
X11 + X30 <= 12*Beta11;
X12 + X31 <= 12*Beta12;
X13 + X32 <= 12*Beta13;
X14 + X33 <= 12*Beta14;
X15 + X34 <= 12*Beta15;
X16 + X35 <= 12*Beta16;
X17 + X36 <= 12*Beta17;
X18 + X37 <= 12*Beta18;
X19 + X38 <= 12*Beta19;
@BIN (Beta1);
@BIN (Beta2);
@BIN (Beta3);
@BIN (Beta4);
@BIN (Beta5);
@BIN (Beta6);
@BIN (Beta7);
@BIN (Beta8);
@BIN (Beta9);
@BIN (Beta10);
@BIN (Beta11);
@BIN (Beta12);
@BIN (Beta13);
@BIN (Beta14);
@BIN (Beta15);
@BIN (Beta16);
@BIN (Beta17);
@BIN (Beta18);
@BIN (Beta19);
```

Estrategia 1. Modelo B

```
TITLE MBPM steinb1.txt;
MIN = 41 + 7*Beta1 + 1*Beta2 + 7*Beta3 + 2*Beta4 + 2*Beta5 + 2*Beta6 +
8*Beta7 + 4*Beta8 + 8*Beta9 + 5*Beta10 + 6*Beta11 + 8*Beta12 +
14*Beta13 + 8*Beta14 + 9*Beta15 + 2*Beta16 + 7*Beta17 + 15*Beta18 +
11*Beta19;
X20 + X3 + X12 = X1 + X22 + X31 + 1;
X9 + X11 + X13 = X28 + X30 + X32 + 1;
X21 + X22 + X7 = X2 + X3 + X26 + 1;
13 + X23 + X5 = X4 + X24 + 1;
X1 + X6 = X20 + X25 + 1;
X24 + X25 + X26 + X14 = X5 + X6 + X7 + X33 + 1;
X10 + X18 = X29 + X37 + 1;
X4 + X27 = X23 + X8 + 1;
X2 + X28 + X29 + X19 = X21 + X9 + X10 + X38 + 1;
X8 + X30 + X15 = X27 + X11 + X34 + 1;
X31 + X32 + X16 = X12 + X13 + X35 + 1;
X33 + X34 + X35 + X36 + X37 = X14 + X15 + X16 + X17 + X18 + 1;
X17 + X38 = X36 + X19 + 1;
X1 <= 12*Beta1;
X2 <= 12*Beta2;
X3 <= 12*Beta3;
X4 <= 12*Beta4;
X5 <= 12*Beta5;
X6 <= 12*Beta6;
X7 <= 12*Beta7;
X8 <= 12*Beta8;
X9 <= 12*Beta9;
X10 <= 12*Beta10;
X11 <= 12*Beta11;
X12 <= 12*Beta12;
X13 <= 12*Beta13;
X14 <= 12*Beta14;
X15 <= 12*Beta15;
X16 <= 12*Beta16;
X17 <= 12*Beta17;
X18 <= 12*Beta18;
X19 <= 12*Beta19;
X20 <= 12*Beta1;
X21 <= 12*Beta2;
X22 <= 12*Beta3;
X23 <= 12*Beta4;
X24 <= 12*Beta5;
X25 <= 12*Beta6;
X26 <= 12*Beta7;
X27 <= 12*Beta8;
X28 <= 12*Beta9;
X29 <= 12*Beta10;
X30 <= 12*Beta11;
X31 <= 12*Beta12;
X32 <= 12*Beta13;
X33 <= 12*Beta14;
X34 <= 12*Beta15;
X35 <= 12*Beta16;
X36 <= 12*Beta17;
X37 <= 12*Beta18;
X38 <= 12*Beta19;
@BIN (Beta1);
@BIN (Beta2);
@BIN (Beta3);
@BIN (Beta4);
```


@BIN (Beta5);
@BIN (Beta6);
@BIN (Beta7);
@BIN (Beta8);
@BIN (Beta9);
@BIN (Beta10);
@BIN (Beta11);
@BIN (Beta12);
@BIN (Beta13);
@BIN (Beta14);
@BIN (Beta15);
@BIN (Beta16);
@BIN (Beta17);
@BIN (Beta18);
@BIN (Beta19);

Estrategia 2. Formulación Ciclos

```

TITLE MBPM steinb1.txt;
MIN = 41 + 7*X1 + 1*X2 + 7*X3 + 2*X4 + 2*X5 + 2*X6 + 8*X7 + 4*X8 +
8*X9 + 5*X10 + 6*X11 + 8*X12 + 14*X13 + 8*X14 + 9*X15 + 2*X16 + 7*X17
+ 15*X18 + 11*X19 ;
X1 + X3 + X12 = N1;
N1 >= 2 - 3 *(1 - beta1);
N1 <= 3*beta1;
X9 + X11 + X13 = N2;
N2 >= 2 - 3 *(1 - beta2);
N2 <= 3*beta2;
X2 + X3 + X7 = N3;
N3 >= 2 - 3 *(1 - beta3);
N3 <= 3*beta3;
X4 + X5 = N4;
N4 >= 2 - 2 *(1 - beta4);
N4 <= 2*beta4;
X1 + X6 >= 1;
X5 + X6 + X7 + X14 >= 1;
X10 + X18 >= 1;
X4 + X8 >= 1;
X2 + X9 + X10 + X19 = N9;
N9 >= 2 - 4 *(1 - beta9);
N9 <= 4*beta9;
X8 + X11 + X15 >= 1;
X12 + X13 + X16 >= 1;
X14 + X15 + X16 + X17 + X18 = N12;
N12 >= 2 - 5 *(1 - beta12);
N12 <= 5*beta12;
X17 + X19 >= 1;
X1+X6+X7+X3<=3;
X11+X15+X16+X13<=3;
X10+X19+X17+X18<=3;
X1+X6+X14+X16+X12<=4;
X3+X2+X9+X13+X12<=4;
X9+X10+X18+X15+X11<=4;
X2+X10+X18+X14+X7<=4;
X4+X8+X15+X14+X5<=4;
X3+X2+X10+X18+X16+X12<=5;
X9+X2+X7+X14+X15+X11<=5;
X1+X6+X7+X2+X9+X13+X12<=6;
X3+X2+X9+X11+X15+X16+X12<=6;
X9+X2+X7+X5+X4+X8+X11<=6;
X11+X8+X4+X5+X14+X16+X13<=6;
X1+X6+X5+X4+X8+X11+X13+X12<=7;
X3+X2+X10+X18+X15+X11+X13+X12<=7;
X9+X10+X18+X14+X5+X4+X8+X11<=7;
X11+X15+X17+X19+X10+X18+X16+X13<=7;
X2+X10+X18+X15+X8+X4+X5+X7<=7;
X1+X6+X5+X4+X8+X11+X9+X2+X3<=8;
X3+X2+X10+X18+X17+X19+X9+X13+X12<=8;
X9+X2+X7+X5+X4+X8+X15+X16+X13<=8;
X11+X15+X14+X7+X2+X10+X18+X16+X13<=8;
X4+X8+X15+X17+X19+X10+X18+X14+X5<=8;
X1+X6+X5+X4+X8+X15+X17+X19+X2+X3<=9;
X3+X2+X9+X11+X8+X4+X5+X14+X16+X12<=9;
X9+X2+X7+X14+X17+X19+X10+X18+X15+X11<=9;
X11+X8+X4+X5+X7+X2+X10+X18+X16+X13<=9;
X1+X6+X5+X4+X8+X11+X9+X10+X18+X16+X12<=10;
X3+X2+X9+X11+X15+X17+X19+X10+X18+X16+X12<=10;
X9+X10+X18+X17+X19+X2+X7+X5+X4+X8+X11<=10;

```

Restricciones Anti-bucle

```
X11+X8+X4+X5+X14+X17+X19+X10+X18+X16+X13<=10;
X1+X6+X5+X4+X8+X11+X9+X2+X7+X14+X16+X12<=11;
X3+X7+X5+X4+X8+X15+X17+X19+X10+X18+X16+X12<=11;
-beta1-beta2-beta3-beta4-beta9-beta12-
X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16+X17+X18+X19>=6;
@BIN (beta1);
@BIN (beta2);
@BIN (beta3);
@BIN (beta4);
@BIN (beta9);
@BIN (beta12);
@BIN (X1);
@BIN (X2);
@BIN (X3);
@BIN (X4);
@BIN (X5);
@BIN (X6);
@BIN (X7);
@BIN (X8);
@BIN (X9);
@BIN (X10);
@BIN (X11);
@BIN (X12);
@BIN (X13);
@BIN (X14);
@BIN (X15);
@BIN (X16);
@BIN (X17);
@BIN (X18);
@BIN (X19);
```

Estrategia 3. Formulación Ciclos resolución iterativa

```

TITLE MBPM steinb5.txt;
MIN = 5 + 2*X1 + 3*X2 + 10*X3 + 2*X4 + 1*X5 + 5*X6 + 1*X7 + 9*X8 +
10*X9 + 3*X10 + 8*X11 + 2*X12 + 6*X13 + 9*X14 + 6*X15 + 5*X16 + 8*X17
+ 9*X18 + 9*X19 + 10*X20 + 10*X21 + 6*X22 + 7*X23 + 7*X24 + 5*X25 +
5*X26 + 3*X27 + 1*X28 + 1*X29 + 10*X30 + 2*X31 + 5*X32 + 6*X33 + 1*X34
+ 7*X35 + 5*X36 + 7*X37 + 8*X38 + 2*X39 + 8*X40 + 1*X41 + 3*X42 +
2*X43 + 10*X44 + 5*X45 + 4*X46 + 10*X47 + 8*X48 + 9*X49 + 1*X50 +
9*X51 + 4*X52 + 3*X53 + 4*X54 + 7*X55 + 6*X56 + 8*X57 + 4*X58 + 5*X59
+ 8*X60 + 2*X61 + 6*X62 + 4*X63 + 3*X64 + 9*X65 + 6*X66 + 3*X67 +
6*X68 + 4*X69 + 4*X70 + 2*X71 + 6*X72 + 2*X73 + 10*X74 + 4*X75 + 2*X76
+ 1*X77 + 4*X78 + 1*X79 + 7*X80 ;
X1 + X2 + X32 + X49 + X60 = N1;
N1 >= 2 - 5 *(1 - beta1);
N1 <= 5*beta1;
X3 + X4 + X6 + X20 = N2;
N2 >= 2 - 4 *(1 - beta2);
N2 <= 4*beta2;
X5 + X33 + X37 + X52 >= 1;
X6 + X7 + X8 + X9 + X47 = N4;
N4 >= 2 - 5 *(1 - beta4);
N4 <= 5*beta4;
X10 + X27 + X39 + X78 >= 1;
X11 + X12 + X13 + X14 + X61 = N6;
N6 >= 2 - 5 *(1 - beta6);
N6 <= 5*beta6;
X1 + X73 >= 1;
X54 + X56 + X70 = N8;
N8 >= 2 - 3 *(1 - beta8);
N8 <= 3*beta8;
X15 + X16 + X17 + X62 = N9;
N9 >= 2 - 4 *(1 - beta9);
N9 <= 4*beta9;
X18 + X19 + X28 >= 1;
X11 + X20 + X21 + X22 + X23 = N11;
N11 >= 2 - 5 *(1 - beta11);
N11 <= 5*beta11;
X23 + X40 >= 1;
X3 + X24 + X63 >= 1;
X21 + X25 + X26 + X45 = N14;
N14 >= 2 - 4 *(1 - beta14);
N14 <= 4*beta14;
X2 + X12 + X27 + X28 + X29 = N15;
N15 >= 2 - 5 *(1 - beta15);
N15 <= 5*beta15;
X18 + X30 + X57 + X69 = N16;
N16 >= 2 - 4 *(1 - beta16);
N16 <= 4*beta16;
X5 + X30 + X31 + X76 >= 1;
X13 + X15 + X24 + X25 + X32 + X33 + X34 + X35 = N18;
N18 >= 2 - 8 *(1 - beta18);
N18 <= 8*beta18;
X36 + X42 + X48 + X64 >= 1;
X7 + X37 + X38 + X55 >= 1;
X16 + X39 + X40 + X41 + X42 = N21;
N21 >= 2 - 5 *(1 - beta21);
N21 <= 5*beta21;
X14 + X43 + X44 = N22;
N22 >= 2 - 3 *(1 - beta22);
N22 <= 3*beta22;
X45 + X46 + X65 = N23;

```

```

N23 >= 2 - 3 *(1 - beta23);
N23 <= 3*beta23;
X41 + X47 + X50 + X68 = N24;
N24 >= 2 - 4 *(1 - beta24);
N24 <= 4*beta24;
X43 + X48 >= 1;
X49 + X50 + X51 = N26;
N26 >= 2 - 3 *(1 - beta26);
N26 <= 3*beta26;
X36 + X52 + X53 = N27;
N27 >= 2 - 3 *(1 - beta27);
N27 <= 3*beta27;
X22 + X79 >= 1;
X17 + X51 + X54 + X55 + X71 + X74 = N29;
N29 >= 2 - 6 *(1 - beta29);
N29 <= 6*beta29;
X8 + X34 + X53 + X56 + X57 + X58 >= 1;
X59 + X75 + X80 = N31;
N31 >= 2 - 3 *(1 - beta31);
N31 <= 3*beta31;
X19 + X58 + X60 = N32;
N32 >= 2 - 3 *(1 - beta32);
N32 <= 3*beta32;
X61 + X62 + X63 + X64 + X65 + X66 + X67 + X72 = N33;
N33 >= 2 - 8 *(1 - beta33);
N33 <= 8*beta33;
X66 + X68 + X69 = N34;
N34 >= 2 - 3 *(1 - beta34);
N34 <= 3*beta34;
X38 + X59 + X70 + X71 + X72 = N35;
N35 >= 2 - 5 *(1 - beta35);
N35 <= 5*beta35;
X26 + X29 + X73 + X74 + X75 + X77 = N36;
N36 >= 2 - 6 *(1 - beta36);
N36 <= 6*beta36;
X4 + X76 + X77 = N37;
N37 >= 2 - 3 *(1 - beta37);
N37 <= 3*beta37;
X10 + X44 + X46 = N38;
N38 >= 2 - 3 *(1 - beta38);
N38 <= 3*beta38;
X9 + X31 + X35 + X67 + X78 + X79 + X80 = N39;
N39 >= 2 - 7 *(1 - beta39);
N39 <= 7*beta39;
X27+X29+X77+X76+X31+X78<=5;
X27+X29+X77+X76+X31+X67+X64+X42+X39<=8;
X1+X73+X29+X2<=3;
-beta1-beta2-beta4-beta6-beta8-beta9-beta11-beta14-beta15-beta16-
beta18-beta21-beta22-beta23-beta24-beta26-beta27-beta29-beta31-beta32-
beta33-beta34-beta35-beta36-beta37-beta38-beta39 +
X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16+X17+X18+X19+X20
+X21+X22+X23+X24+X25+X26+X27+X28+X29+X30+X31+X32+X33+X34+X35+X36+X37+X
38+X39+X40+X41+X42+X43+X44+X45+X46+X47+X48+X49+X50+X51+X52+X53+X54+X55
+X56+X57+X58+X59+X60+X61+X62+X63+X64+X65+X66+X67+X68+X69+X70+X71+X72+X
73+X74+X75+X76+X77+X78+X79+X80 = 11;
@BIN (beta1);
@BIN (beta2);
@BIN (beta3);
@BIN (beta4);
@BIN (beta6);
@BIN (beta8);

```

@BIN (beta9);
@BIN (beta11);
@BIN (beta14);
@BIN (beta15);
@BIN (beta16);
@BIN (beta18);
@BIN (beta21);
@BIN (beta22);
@BIN (beta23);
@BIN (beta24);
@BIN (beta26);
@BIN (beta27);
@BIN (beta29);
@BIN (beta31);
@BIN (beta32);
@BIN (beta33);
@BIN (beta34);
@BIN (beta35);
@BIN (beta36);
@BIN (beta37);
@BIN (beta38);
@BIN (beta39);
@BIN (X1);
@BIN (X2);
@BIN (X3);
@BIN (X4);
@BIN (X5);
@BIN (X6);
@BIN (X7);
@BIN (X8);
@BIN (X9);
@BIN (X10);
@BIN (X11);
@BIN (X12);
@BIN (X13);
@BIN (X14);
@BIN (X15);
@BIN (X16);
@BIN (X17);
@BIN (X18);
@BIN (X19);
@BIN (X20);
@BIN (X21);
@BIN (X22);
@BIN (X23);
@BIN (X24);
@BIN (X25);
@BIN (X26);
@BIN (X27);
@BIN (X28);
@BIN (X29);
@BIN (X30);
@BIN (X31);
@BIN (X32);
@BIN (X33);
@BIN (X34);
@BIN (X35);
@BIN (X36);
@BIN (X37);
@BIN (X38);
@BIN (X39);

@BIN (X40) ;
@BIN (X41) ;
@BIN (X42) ;
@BIN (X43) ;
@BIN (X44) ;
@BIN (X45) ;
@BIN (X46) ;
@BIN (X47) ;
@BIN (X48) ;
@BIN (X49) ;
@BIN (X50) ;
@BIN (X51) ;
@BIN (X52) ;
@BIN (X53) ;
@BIN (X54) ;
@BIN (X55) ;
@BIN (X56) ;
@BIN (X57) ;
@BIN (X58) ;
@BIN (X59) ;
@BIN (X60) ;
@BIN (X61) ;
@BIN (X62) ;
@BIN (X63) ;
@BIN (X64) ;
@BIN (X65) ;
@BIN (X66) ;
@BIN (X67) ;
@BIN (X68) ;
@BIN (X69) ;
@BIN (X70) ;
@BIN (X71) ;
@BIN (X72) ;
@BIN (X73) ;
@BIN (X74) ;
@BIN (X75) ;
@BIN (X76) ;
@BIN (X77) ;
@BIN (X78) ;
@BIN (X79) ;
@BIN (X80) ;

Anexo 3. Modulo de VBA

CrearModeloCiclos

Funcion que crea el modelo de Ciclos

CrearModelo1A

Función que crea el modelo 1A y su variante, la modificación que sufre el código de un modelo a otro esta entrecomillada en el código.

CrearModelo1B

Crea el modelo 1B y su variante.

Obtener Cadena

Obtiene nombre del archivo .txt. del modelo de entrada.

Cargar archivo de texto reducido

Carga a las variables del archivo .txt en VBA

Calcular adyacencias

Calcula para el nodo "i" todos los nodos adyacentes "j"

AdjCalcular ciclos pequeños

Calculo de ciclos de con $N \leq 1000$

Calcular ciclos arcos

Calcula los arcos que componen el ciclo

Calculo Recursivo

Calculo Recursivo de los Ciclos del problema

Guardar Ciclo

Guarda en memoria el ciclo calculado

Adjuntar calculo Ciclo Tamaño

Calcula el tamaño del ciclo

Adjuntar Calculo ciclo tamaño Pequeño

AdjConstruyeCadena

Construye la cadena o nombre del archivo de salida de LINGO que será un archivo con extensión .lg

NoEnLista

Comprueba que no esta listado el ciclo

AdjFinalizarCadena

Adjunta al modelo el ciclo que contiene en esa cadena

Buscagualdad

Busca si hay algún nodo igual a otro

OrdenarArcos

Ordena los arcos de mayor a menor coste

Mezcla

Hace un traspaso de las variables de VBA que contienen el grafo por las de otro.

CalcularNodosSteiner

Calcula el número de nodos Steiner

ObtenerOptimos

IncluirNodos

Incluye el nodo en el ciclo que esta calculando

Incluir Restriccion

Incluye la restricción en el modelo

PreTraducirCiclos

Hace un precalculo de la conversión de ciclos de la solución del modelo, para incluirlos en el inicial.

TraducirCiclos

Hace el calculo de la conversión de ciclos de la solución del modelo, para incluirlos en el inicial.

CrearModelo1A Cadena

Crea el modelo de entrada para LINGO del Modelo 1A

CrearModelo1B Cadena

Crea el modelo de entrada para LINGO del Modelo 1B

CrearModeloCiclos

Crea el modelo de entrada para LINGO del Modelo Ciclos

CrearModeloCiclos_Iter

Crea el modelo de entrada para LINGO del Modelo Ciclos añadiendo las restricciones de ciclos calculadas de la solución parcial obtenida.

Programación Estrategia 1

Sub Main()

Dim I As Long

Path_Entrada = App.Path

For I = 1 To 78

NumeroProblema = I

Cadena = ObtenerCadena(I)

Cargar_Archivo_Texto_Reducido Cadena

CrearModelo1A Cadena

CrearModelo1B Cadena

Next

End Sub

Programación Estrategia 2

Sub Main()

Dim I As Long

Path_Entrada = App.Path

For I = 1 To 78

NumeroProblema = I

Cadena = ObtenerCadena(I)

Cargar_Archivo_Texto_Reducido Cadena

CalcularCiclosArcos Cadena

CrearModeloCiclos Cadena

Next

End Sub

Programación Estrategia 3

Sub Main()

Dim i As Long

Dim K As Integer

Dim Valores() As Long

Dim Fichero As String

Dim Pasadas As Long

Path_Entrada = App.Path

For i = 1 To 78

NumeroProblema = i

Cadena = ObtenerCadena(i)

Cargar_Archivo_Texto_Reducido Cadena

' AdjCalcularCiclosPequeños

TiempoInicio = Now

CalcularCiclosArcos

CrearModeloCiclos Cadena, Fichero

ReDim Valores(NumArcos)

ResolverLingoInicio Fichero, Valores()

' 'parar para resolver y guardar archivo solucion

' K = K

CrearGrafo Cadena, Valores()

Cargar_Archivo_Texto_Reducido "Post_" & Cadena

Encontrado = False

NumCiclos = 0

CalcularCiclosArcos

Pasadas = 0

```
While NumCiclos <> 0

DoEvents

PreTraducirCiclos

Cargar_Archivo_Texto_Reducido Cadena

TraducirCiclos

CrearModeloCiclos_Iter Cadena, Fichero, Pasadas

ReDim Valores(NumArcos)

ResolverLingolterativo Fichero, Valores()

' 'parar para resolver y guardar archivo solucion

NumCiclos = 0

Encontrado = False

CrearGrafo Cadena, Valores()

Cargar_Archivo_Texto_Reducido "Post_" & Cadena

'AdjCalcularCiclos

CalcularCiclosArcos

Pasadas = Pasadas + 1

Wend

TiempoFin = Now

Tiempo = CDate(TiempoFin - Tiempolnicio)

ImprimeResultado Cadena, ValorFO, Tiempo, Pasadas

Next

'CerrarResultados

End Sub
```