Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías Industriales

Propuesta de algoritmo para sincronización semafórica en una ciudad.

Autor: José María Jiménez Gómez

Tutor: Jesús Muñuzuri Sanz

Dep. Organización Industrial y Gestión de Empresas II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Propuesta de algoritmo para sincronización semafórica en una ciudad.

Autor:

José María Jiménez Gómez

Tutor:

Jesús Muñuzuri Sanz

Profesor titular

Dep. de Organización Industrial y Gestión de Empresas II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Trabajo Fin de Grado: Propuesta de algoritmo para sincronización semafórica en una ciudad.

Autor: José María Jiménez Gómez

Tutor: Jesús Muñuzuri Sanz

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

Resumen

En la actualidad, el número de ventas de vehículos se ha disparado hasta unos niveles desorbitados. Este hecho, junto con la mala programación en las ciudades, ocasiona graves retenciones.

Los semáforos aportan una solución para la regulación del tráfico, asignando prioridad de paso a las vías para que se produzca una circulación eficiente. Sin embargo, esta medida no es suficiente y las grandes retenciones, a veces en horas punta y otras continuas, se siguen produciendo en las ciudades con grandes flujos de circulación de vehículos.

El objetivo de este trabajo es realizar una propuesta para la regulación del tráfico. Realizaremos la implementación de una Onda Verde, la cual se aplicará a vías y semáforos para que el funcionamiento de éstos sea más efectivo y evitar las retenciones antes citadas y los largos tiempos de viaje.

Abstract

Currently, the number of vehicle sales has soared to levels exorbitant. This fact, coupled with poor programming in cities causes serious delays.

Semaphores provide a solution for traffic control, giving priority to passing ways for an efficient movement to occur. However, this measure is not enough and large retentions, sometimes at peak times and other continuous, are still occurring in cities with large flows of vehicular traffic.

The objective of this project is to make a proposal for traffic regulation. We will implement a Green Wave, which apply to roads and semaphores to make the operation of these more effective and avoid the aforementioned retentions and large travel times.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE IMÁGENES

Notación

A^*	Conjugado
c.t.p.	En casi todos los puntos
c.q.d.	Como queríamos demostrar
∎	Como queríamos demostrar
e.o.c.	En cualquier otro caso
e	número e
IRe	Parte real
IIm	Parte imaginaria
sen	Función seno
tg	Función tangente
arctg	Función arco tangente
sen	Función seno
$\sin^x y$	Función seno de *x* elevado a *y*
$\cos^x y$	Función coseno de *x* elevado a *y*
Sa	Función sampling
sgn	Función signo
rect	Función rectángulo
Sinc	Función sinc
$\partial y\ \partial x$	Derivada parcial de *y* respecto
x°	Notación de grado, *x* grados.
$\Pr(A)$	Probabilidad del suceso *A*
SNR	Signal-to-noise ratio
MSE	Minimum square error
:	Tal que
<	Menor o igual
>	Mayor o igual
\	Backslash
⇔	Si y sólo si

1 INTRODUCCIÓN

Cuando dos o más vías se cruzan a un mismo nivel y la intensidad de tráfico es demasiado grande como para controlarla con una regulación de preferencia de paso, la utilización de semáforos es un medio seguro, económico y eficaz para regularización del tráfico. Entre los inconvenientes encontramos las detenciones de los vehículos que se producen en las intersecciones, que tienen como consecuencia el aumento del tiempo de circulación de los vehículos.

Los semáforos permiten la entrada de los vehículos desde los distintos accesos que confluyen a la intersección durante un tiempo determinado. Este suceso es concedido ordenadamente y se repite a lo largo del tiempo cíclicamente. Es decir, una vez terminada la sucesión programada de cambios de luces en los semáforos, ésta vuelve a empezar desde su inicio. Cuando los movimientos de los vehículos son paralelos pueden realizarse al mismo tiempo aunque sean de distintos accesos. Sin embargo, no podemos decir lo mismo cuando son convergentes o "se cruzan".

En una intersección podemos regular la circulación de vehículos y de peatones, pero también podemos dejar a un lado a estos últimos y centrarnos en regular la circulación de vehículos. Esto implica dejar la circulación de peatones a merced de la regularización prevista para los vehículos.

Normalmente, la regularización semafórica se realiza en zonzas urbanas, donde tenemos situaciones tales como un cruce con tres vías de entrada o salida. Este hecho nos obliga a dar prioridad a una o varias de estas vías. También podemos aplicar la regularización en carretera, pero debido a la menor cantidad de cruces e intersecciones, esta regularización sería menos notable. Se aplicará sobre todo cerca de poblaciones y en intersecciones peligrosas, donde los semáforos siempre deben de ir acompañados de una señal fija de preaviso de semáforo.

Para controlar estos semáforos instalados en un cruce se utiliza un regulador, el cual está conectado a los semáforos y se encarga de ordenar los cambios de luces de éstos. Dichos reguladores pueden ser electromecánicos, electrónicos o de microprocesador. Por tanto, intervienen dos partes distintas y complementarias a la vez para llevar a cabo esta tarea. La ingeniería de tráfico determina teóricamente la manera en que los semáforos han de funcionar. La eléctrica se encarga de garantizar el plan teórico formulado, encendiendo y apagando con precisión los semáforos. Este encendido y apagado de las lámparas se realiza de acuerdo con el plan de tráfico, el cual determina los movimientos de tráfico autorizados que se realizan en la intersección, su orden y el tiempo de duración.

A continuación, se definen una serie de conceptos que nos ayudarán a entender la regularización semafórica:

- El ciclo es el tiempo transcurrido desde el inicio de una secuencia de cambio de luces hasta la repetición de dicha situación en los semáforos conectados a un mismo regulador.

- Las fases son cada una de las divisiones del ciclo, los colores de todos los grupos semafóricos permanece invariable.

- El despeje es el tiempo necesario para que los vehículos que han accedido a la intersección salgan de la zona de ésta y la dejen totalmente libre para que entren nuevos vehículos sin que se produzca o ninguna colisión o accidente.

- El reparto del ciclo es la división del tiempo total de ciclo entre cada una de las fases que integran la estructura de regulación propia del plan de tráfico operativo.

La regulación de tráfico tiene como primer objetivo la solución de un problema unitario y localizado, generalmente en un cruce de vías o intersección, sin tener que preocuparse por el problema que pueda existir

aguas arriba o aguas abajo de la intersección o a su derecha o izquierda. Una vez solucionado el problema local pueden darse interferencias no deseadas entre ellos que es preciso solucionar.
Es conveniente sincronizar los reguladores entre varias intersecciones cercanas, para coordinar los semáforos y así un vehículo podría circular por una vía en un sentido concreto y a una velocidad determinada, encontrando el menor número de semáforos en rojo y minimizando los tiempos de parada. Si invertimos esta afirmación, tenemos por objetivo lograr que un vehículo en circulación encuentre el mayor número de semáforos en verde posible.

Como hemos explicado, la regulación semafórica se refiere a la interacción de un regulador con los semáforos que tiene asignados en una intersección concreta. A diferencia de ésta, encontramos la coordinación o sincronización, la cual se encarga de determinar el modo en que estas intersecciones están relacionadas entre sí. El objetivo es asegurar que un vehículo que circula de una intersección a otra evite todas las paradas semafóricas posibles. Es decir, consiste en programar el encendido de las luces de los semáforos de tal forma que los vehículos puedan atravesar la vía, de extremo a extremo sin detenerse. Por tanto, la coordinación forma parte del plan de ingeniería de tráfico y determina cómo han de funcionar los semáforos para conseguir el efecto práctico deseado.

Cuando conseguimos que se produzca este hecho, hemos conseguido crear lo que se denomina una Onda Verde. Así pues, ésta no es más que una sucesión de intersecciones conectadas y sincronizadas entre sí, que permiten circular de unas a otras logrando no detenerse en ningún semáforo y lograr una continuidad en el flujo de vehículos.

En la actualidad, el aumento exponencial del volumen del tráfico y del número de cruces regulados por semáforos hace necesaria una coordinación cada vez más precisa y cercana a la saturación que se encuentre la vía, ya que pequeños errores de cálculo conducirían al bloqueo de la misma con el consiguiente aumento de los tiempos de recorrido.

2 OBJETIVOS Y PLAN DE TRABAJO

2.1 Objetivos

Este trabajo pretende una revisión de los métodos para la coordinación de intersecciones reguladas por semáforos, con el fin de obtener las intersecciones y vías que deberán formar una Onda Verde y por las que se podrá circular sin necesidad de parar en semáforos mientras se produzca ésta.

Vamos a obviar la regulación de cada intersección unitaria, concentrándonos en cuales de éstas deben estar sincronizadas entre sí y así tener más prioridad de circulación ante las otras. Del mismo modo, vamos a obviar la existencia de peatones, no teniéndolos en cuenta a la hora de la sincronización. Así pues, éstos se verán subordinados a las decisiones que tomemos para la circulación de vehículos, que es nuestra prioridad.

Añadir que este trabajo sólo tiene sentido si es aplicado en una zona urbana, en la cual tenemos diversas intersecciones conectadas entre ellas. La sincronización apenas sería útil en una zona de carretera, ya que generalmente con una simple regulación de la intersección podemos evitar los problemas de tráfico.

Para ello, crearemos un método para elegir las intersecciones que formarán la Onda Verde de una red de prueba inicial. Una vez obtenido el método para la elección de la Onda Verde, vamos a implantar nuestro sistema en una zona urbana concreta, obteniendo una "Propuesta Semafórica para Sevilla", lo que hace referencia al título del proyecto.

Se ha elegido la ciudad de Sevilla debido a la gran cantidad de intersecciones semaforizadas que ofrece y a los grandes flujos de vehículos que circulan a diario por la capital hispalense. Todo ello tiene como resultado que esta ciudad no tenga aplicada una sincronización efectiva. Es por ello la realización de este proyecto, el cual intentará aportar soluciones a este problema.

2.2 Plan de trabajo

En primer lugar, debemos resolver el modelo matemático que nos describe las características y restricciones que debemos de tomar a la hora de elegir las intersecciones y vías. Para ello utilizaremos una red de prueba. Una vez resuelto el modelo, debemos verificar que hemos obtenido un "árbol de expansión máxima". Para ello debemos asegurarnos de que cumple ciertas propiedades.

La red de Sevilla es demasiado grande para utilizar un software de optimización. La cantidad de variables y restricciones agotarían la capacidad de éste. Por ello, en segundo lugar debemos programar una heurística que nos ofrezca una solución óptima o cercana a ésta.

En tercer lugar, probaremos las posibles heurísticas con la red de prueba y, una vez elegida la más adecuada para solucionar nuestro problema, la aplicaremos a la red de la ciudad de Sevilla.

Aplicado el algoritmo a la Red de Sevilla, para finalizar analizaremos los resultados obtenidos y los compararemos con otra propuesta de sincronización semafórica.

3 ÁRBOL DE EXPANSIÓN MÁXIMA

A continuación, vamos a exponer una serie de características que tenemos que tener en cuenta a la hora de elegir las intersecciones que van a formar la Onda Verde. Estas condiciones coinciden con las que cumple un árbol de expansión.

- En los arboles de expansión todos los nodos están interconectados entre sí mediante arcos pero siempre teniendo en cuenta que no se pueden formar ciclos.

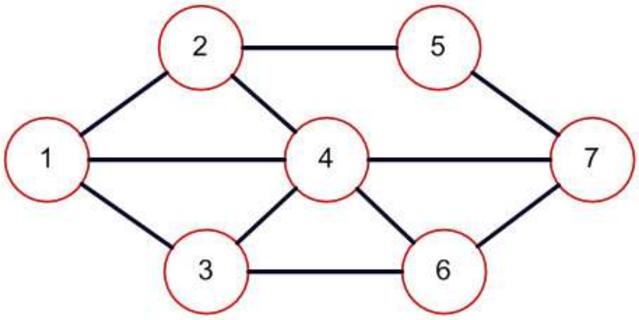


Imagen 1. Red 1

Como podemos apreciar en la Red 1, los diferentes nodos están unidos mediante arcos. Para nuestro proyecto, vamos a tomar nuestras intersecciones como nodos, y las vías que unen éstas como arcos. Nuestra misión será elegir los arcos que nos garanticen las condiciones de un árbol de expansión.

- Otra característica importante que debemos cumplir es evitar a toda costa los ciclos. Un ciclo es una cadena de nodos interconectados con arcos que une uno de los nodos consigo mismo. Adaptando esta característica a nuestro proyecto, si se produjesen ciclos, un vehículo que pasa por una intersección y sigue una Onda Verde, podría volver otra vez a la misma intersección pero el tiempo que el semáforo se mantiene en verde ya se ha agotado y, por tanto, forzaría una detención.

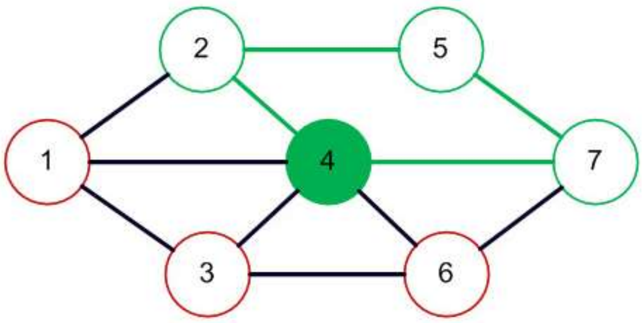


Imagen 2. Red 2

- La siguiente característica de un árbol de expansión es, simplemente, que todos los nodos deben estar interconectados por arcos activados. Decimos que un arco está activado cuando está dentro de nuestra selección para formar parte del árbol de expansión. Añadir un arco que conecte a un nodo que no esté conectado aún a los demás nunca va a suponernos un problema a la hora de evitar ciclos en ese nodo en concreto.

Finalmente obtendremos un árbol parecido al de la Red 3, donde podemos observar la inexistencia de ciclos y todos los nodos conectados entre sí.

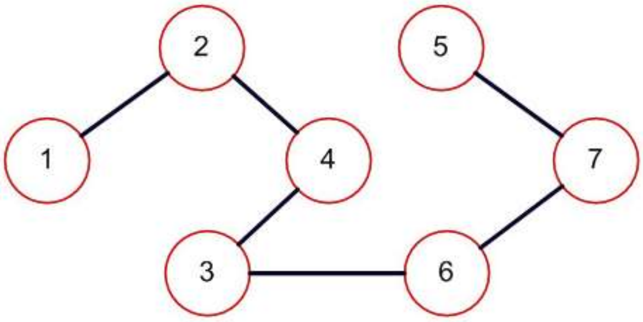


Imagen 3. Red 3

Una árbol de expansión nos proporciona los nodos que deben estar conectados y los arcos por donde deben de hacerlo pero, ¿Atendiendo a qué criterio? En nuestro caso real, queremos que las vías que donde se implante la Onda Verde sean las vías más transitadas de la ciudad. Por ello, introducimos el concepto de “flujo de vehículos”, el cual nos indica la cantidad de vehículos que circulan por una vía. En nuestra red, este dato significará el peso que tiene asignado cada arco.

En el presente trabajo necesitamos averiguar el árbol de expansión con el cual consigamos integrar el mayor número de vehículos posibles circulando en la Onda Verde. El árbol que nos ofrece estas características es el “Árbol de Expansión Máxima” el cual elige los tramos que conectan todas las intersecciones eligiendo el conjunto de vías por las que más vehículos circulan. Es decir, seleccionará el conjunto de arcos, que cumpliendo con las condiciones, nos ofrezca mayor suma total de sus pesos. Análogamente, el “Árbol de Expansión Mínima” conecta los nodos eligiendo el conjunto de arcos que nos ofrezca menor suma total de sus pesos.

4 MODELO

Una de las opciones para encontrar el árbol de expansión máxima de una red es resolviendo el modelo que lo describe con un software de optimización. Hablamos del llamado "Modelo de Jerarquización de Rutas Semafóricas".

A continuación, una descripción del modelo.

Datos:
w_{ij}: *Peso del arco que une el nodo i con el nodo j.*
n: *Constante suficientemente grande.*

Variables:
x_{ij}: *Variable binaria que vale* 1 *cuando el arco que va del nodo i al j se activa*
y_i: *Variable auxiliar asociada al nodo i que tomará valores no negativos.*

Modelo:

Función Objetivo:
$$Max \sum_{(i,j)\in A} (x_{ij}+x_{ji}) \quad (1)$$

Sujeto a:
$$x_{ij} + x_{ji} \leq 1 \qquad \forall (i,j) \in A \quad (2)$$

$$\sum_{j\in N} x_{ij} \leq 1 \qquad \forall i \in N \quad (3)$$

$$y_i - y_j + n(x_{ij} + x_{ji}) \leq n - 1 \qquad \forall (i,j) \in A \quad (4)$$

$$x_{ij} binario, y_i \geq 0 \quad (5)$$

1. La función objetivo maximiza la suma de los pesos de cada arco por las variables de jerarquización de los semáforos, donde el peso wij se puede tomar como la intensidad media diaria que circula por el arco.

2. En cada arco (i, j), el nodo i está sometido al j, o el j está sometido al nodo i. No puede darse el caso de que estén sometidos los dos a la vez.

3. Cada nodo de la red sólo puede estar sometido a otro. Esto implica que cada intersección está subordinada a otra.

4. Estas restricciones impiden que se formen ciclos de jerarquización, siendo n una constante suficientemente grande. Este número debe ser del orden superior al de los pesos. La restricción va asignando valores de a las variables y_i de cada nodo de forma que el valor de y del nodo siguiente debe ser mayor al anterior. Esta acción evita que volvamos de nuevo un nodo anterior, evitando así que se forme un ciclo.

5. Definen los valores posibles para las variables. Las variables xij son variables binarias que valen 1 si el nodo j está sometido al nodo i y 0 en caso contrario. Las variables yi son variables auxiliares no negativas.

Este modelo representa al problema de encontrar el árbol de expansión máxima en una red de N nodos y A arcos.

5 RED DE PRUEBA

Para realizar una propuesta para la sincronización semafórica de Sevilla, debemos tener un método para hallar las vías e intersecciones que serán seleccionadas para dicha sincronización.

Debemos facilitar esta tarea, por ello vamos a basarnos en cálculos realizados sobre una “Red de Prueba”. Esta red será mucho más pequeña que la red viaria de la ciudad de Sevilla, por lo que será posible encontrar una solución con un software de optimización. Además, a la hora de aplicar la heurística sobre ella, vamos a obtener resultados de una forma casi inmediata. De este modo podremos comprobar fácilmente si la heurística funciona y en caso contrario, observar dónde falla. La heurística debe ser genérica y funcionar para cualquier red, no adaptándose a la red de prueba ya que el objetivo es verificar que funciona en la de prueba para después utilizarla en la red de Sevilla o en cualquier otra.

Como ejemplo de red de prueba, hemos tomado una red de 16 nodos y 24 arcos, que se extienden a lo largo del plano formando una estructura cuadrada de 4x4 nodos.

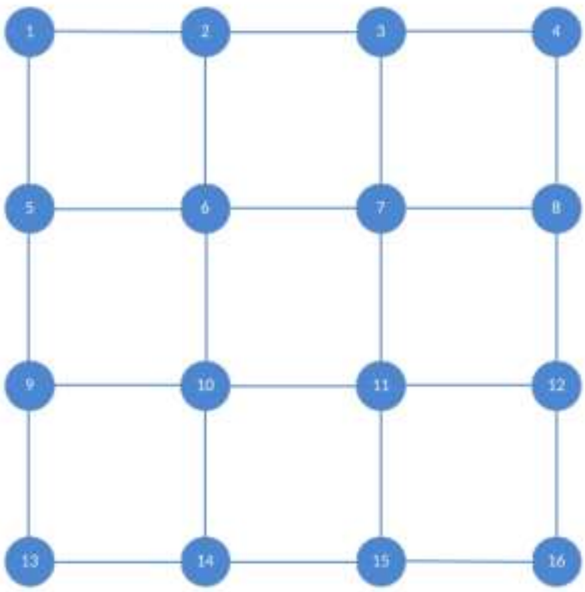


Imagen 4. Red de Prueba 1

Cada arco tiene asignado un peso entre 0 y 100. Como puede apreciarse en la imagen, los arcos de nuestra red de prueba no son arcos dirigidos. Esto quiere decir que el peso del arco al unir un nodo con otro es el mismo que de unir este segundo con el primero. En el caso de tener un peso para cada sentido del arco, consideraremos siempre elegir el mayor de estos dos valores. El hecho de elegir arcos no direccionados simplifica mucho el proceso tanto a la hora de comprobar si nos encontramos ante el árbol de expansión máxima como a la hora de crear un algoritmo en el caso de utilizar alguna heurística.

En la siguiente tabla se detalla el peso de cada arco con los nodos a los cuales une.

Tabla 1. Arcos Red de Prueba

NODO	NODO	PESO
1	2	96
1	5	59.2
2	3	70.4
2	6	82.2
3	4	91.1
3	7	64.4
4	8	38.4
5	6	39.6
5	9	49.3
6	7	44.2
6	10	20.4
7	8	92.4
7	11	34.5
8	12	94.3
9	10	86.6
9	13	85.7
10	11	79.9
10	14	51.2
11	12	61.6
11	15	69.5
12	16	63.9
13	14	94.8
14	15	57.6
15	16	79.8

6 PROBLEMÁTICA CON REDES GRANDES

Teóricamente podríamos resolver un grafo mucho mayor al de la red de prueba, en cuanto a nodos y arcos se refiere. Sin embargo, a la hora de llevarlo a cabo de forma práctica empiezan a surgir ciertos problemas.

- En primer lugar, por cada nodo que agregamos al grafo debemos añadir una restricción (véase restricción 3 del modelo) y por cada arco nuevo que agregamos debemos añadir dos restricciones más al modelo (véanse restricciones 2 y 4 del modelo). Esto puede suponer una gran pérdida de tiempo debido a que estas restricciones aplicadas a para cada grafo son distintas y no podemos preestablecer una forma genérica para escribirlas.

- La segunda complicación reside en el software que vamos a utilizar para resolver el modelo en la red ya que éstos suelen tener limitación de restricciones y variables.

En nuestro problema debemos hallar el árbol de expansión máxima para la red de Sevilla. Incluso haciendo una reducción a las calles principales para esta red, la obtenida es bastante mayor que la red de prueba. Para solventar este problema, vamos a utilizar un método distinto para la obtención del árbol de expansión máxima. Vamos a utilizar un método heurístico.

En mi experiencia de búsqueda de un método para hallar árboles de expansión máxima, he utilizado dos procedimientos con sus ventajas e inconvenientes. En los siguientes apartados se explican los detalles, la implementación y los resultados obtenidos en cada uno de ellos.

7 RESOLUCIÓN DEL MODELO

Una vez tenemos el modelo debemos aplicar la función objetivo y las restricciones a la red de prueba donde lo vamos a resolver. Esta red es la explicada en el apartado anterior y, debido a su reducido tamaño, no debería suponer ningún problema para el software de optimización a la hora de resolverla.

Para esta tarea, vamos a utilizar el software Solver. Es una herramienta integrada en Microsoft Excel que, entre otras funcionalidades, sirve para resolver problemas de programación lineal utilizando el método Simplex. Nuestro modelo es lineal, ya que su función objetivo y restricciones lo son, por lo tanto, podemos utilizar Solver.

La implementación de las ecuaciones y resolución del modelo se lleva a cabo en una hoja de Microsoft Excel adjunta. Aquí podemos ver los resultados aplicados a nuestra red de prueba.

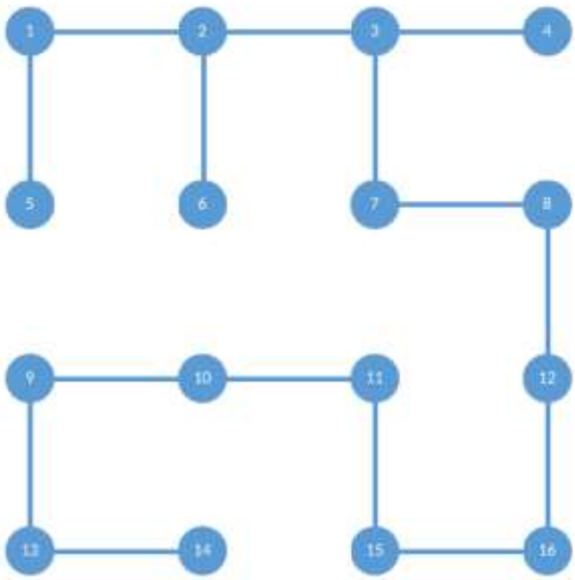


Imagen 5. Red de prueba 2

Tal y como podemos apreciar en la imagen de la Red de Prueba 2, se cumplen las dos condiciones indispensables ya que todos los nodos están conectados y no se aprecia ningún ciclo en el árbol obtenido.

Por constitución, la solución es un árbol. Si se elimina cualquiera de los arcos de este árbol, se produce un corte en la red. Por ejemplo, si en la red de la figura de la Red Prueba 2 se quita el arco (10-11) se produce la partición de los nodos en dos conjuntos. C1= {1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 15, 16} y C2= {9, 10, 13, 14}. Nos quedaría lo reflejado en la siguiente imagen (Onieva, Cortés, Muñuzuri, Guadix & Ibáñez, 2006).

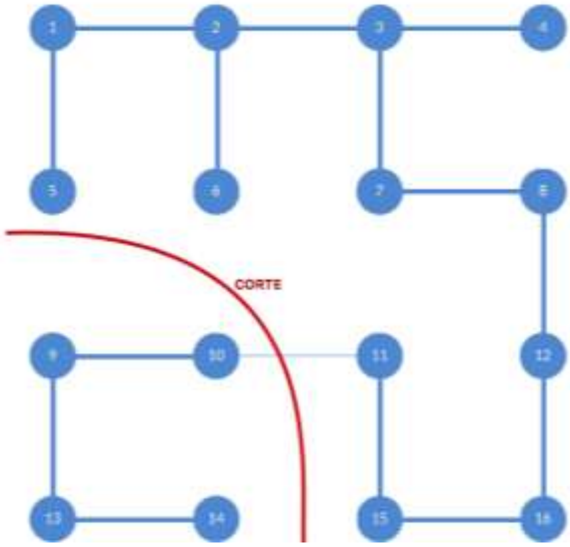


Imagen 6. Red de Prueba 3

Una propiedad importante del corte es que el arco (10-11) es de mayor peso que cualquiera de los restantes arcos que cruzan con el corte que no estén incluidos en el árbol de expansión. Esta propiedad la ha de tener la solución óptima y, si no fuera así, el arco de mayor distancia que cruza con el corte sería el que debería formar parte del árbol de expansión máximo. Esta propiedad es genérica y se recoge explícitamente (Onieva et al., 2006).

Propiedad 1. Los arcos del árbol son siempre los arcos de distancia máxima en cada uno de los cortes de la red que se produzcan al eliminar cada uno de ellos. Esta propiedad se refiere a los arcos que forman parte del árbol de distancia máxima. La otra característica de la solución se refiere a los arcos de la red que no forman parte del árbol de expansión máxima (Onieva et al., 2006).

Incorporando un arco cualquiera al árbol de distancia máxima se construye un ciclo. Por ejemplo añadiendo el arco (5-6) al árbol de expansión máxima de la red de la imagen Red Prueba 2 se obtiene el ciclo entre los nodos 1-2-5-6 que vemos en la imagen Red Prueba 4 (Onieva et al., 2006).

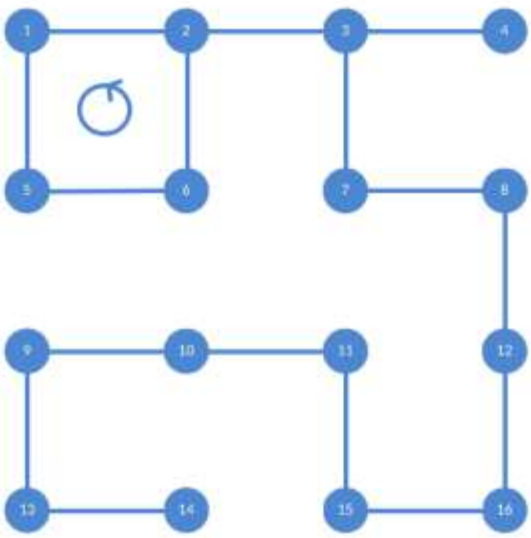


Imagen 7. Red de prueba 4

Dentro de ese ciclo se cumple que el peso del arco que se incorpora (5-6) es menor (o al menos igual) de la de cualquiera de los otros arcos (1-2), (2-6), (1-5) del árbol que están en ese ciclo. El argumento es sencillo. Si no fuera así, se quitaría del árbol el arco de menos peso del ciclo y se sustituiría por el arco nuevo externo que hemos incorporado. Se reconstruiría así otro árbol, pero de mayor expansión (Onieva et al., 2006).

Fijándonos en todos los posibles cortes que podemos realizarle a nuestra red de prueba, no encontramos ningún ejemplo que no satisfaga esta prioridad. Por ello, podemos decir que el modelo para el árbol de expansión máxima cumple con esta propiedad.

Propiedad 2. Los arcos (i, j) no incluidos en el árbol son siempre de peso menor o igual que la de los arcos de árbol que comunican los nodo i y j de dicho arco. Esta propiedad quiere expresar que, cualquier camino para llegar desde el nodo i al j, tendrá mayor suma del peso de sus arcos que el arco (i, j) no incluido en el arco. Sigamos con el ejemplo del arco 5-6 no incluido en el árbol. El peso del arco (5-6) es menor que la suma de los pesos de los arcos incluidos en el árbol que unen 5 y 6, que son (5-1), (1-2), (2-6). (Onieva et al., 2006).

De nuevo, el árbol de expansión máxima satisface esta propiedad. Junto con la Propiedad 1, esto valida finalmente el modelo utilizado y podemos decir que podemos utilizarlo para encontrar el árbol de expansión máxima de cualquier red. Sin embargo, en los anteriores apartados del trabajo comentábamos los problemas que aparecen en las redes grandes, así que propondremos distintas soluciones.

Para la red de prueba que hemos utilizado, si sumamos todos los pesos de los arcos que forman parte del árbol de expansión máxima, obtendremos una suma total de 1210. En esta red, no existe ninguna combinación de arcos que nos verifique la conexión de todos los nodos y la no aparición de ciclos y consiga una suma total mayor a la obtenida. Por todo ello, nos encontramos ante el árbol de expansión máxima de la red de prueba y podemos afirmar que el modelo es válido.

8 PRIMER MÉTODO. RANDOMIZED LOCAL SEARCH

La primera heurística que se ha utilizado es una variante obtenida del artículo "Randomized local search, evolutionary algorithms, and the mínimum spanning tree problem" de Frank Neumann e Ingo Wegener.

En este artículo se explica un método para encontrar el árbol de expansión mínimo, por ello, se han realizado ciertas modificaciones para que el resultado obtenido sea el árbol de expansión máximo.

8.1 Características del método

Si nos fijamos en un árbol de expansión, siempre vamos a encontrarnos con dos propiedades. La primera es que todos los nodos están interconectados entre sí por arcos y no queda ninguno desconectado. La segunda es que el número de arcos que van a estar activados siempre va a coincidir con el número total de nodos menos 1. Bajo estas condiciones, la función objetivo del problema será maximizar la suma total de pesos de los arcos elegidos. A esta función objetivo, debemos añadirle dos penalizaciones en el caso de que no se cumplan ciertas condiciones deseadas.

Debemos penalizar si hay menos de N-1 arcos y también si obtenemos más de N-1, siendo N el número de nodos. Si hay menos arcos de N-1, sabemos que podemos elegir otro arco más y así aumentar la suma total de los pesos de éstos y si hay más arcos de N-1, sabemos con seguridad que debe existir algún ciclo.

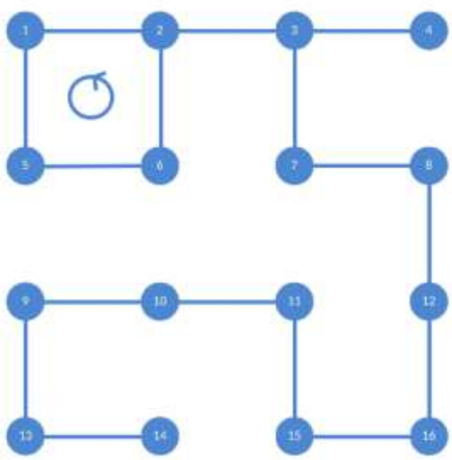


Imagen 8. Red Neumann 1

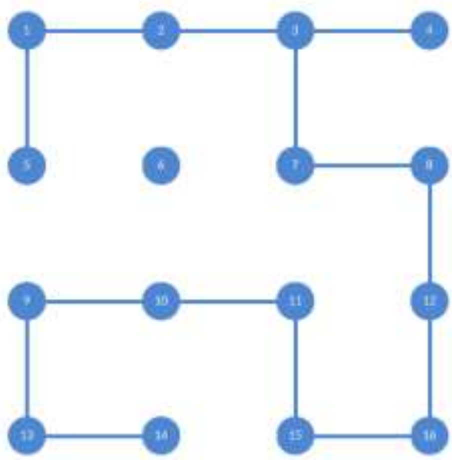


Imagen 9. Red Neumann 2

Analicemos las imágenes de Red Neumann 1 y 2. En la de la izquierda el grafo tiene 16 nodos activados, que coincide con el número de nodos. Un grafo con N o más arcos activados dará lugar a uno o más ciclos. En la imagen de la derecha, podemos observar que hay 14 arcos activados, número inferior a 15 y por lo tanto inferior a n-1. Un grafo con menos de n-1 arcos activados siempre dejará a uno o más nodos desconectados.

En definitiva, calcularemos el valor de la función fitness, la cual equivale a la función objetivo menos las dos penalizaciones antes citadas.

Ya tenemos la función fitness que vamos a evaluar. A continuación debemos elegir cómo obtener los datos que se evalúan en ella. Para ello introduciremos algunos conceptos, datos e indicaciones.

- En primer lugar, debemos enumerar cada arco del grafo.

- Vamos a trabajar con un vector S cuyo número de componentes coincide con el número de arcos de nuestro grafo. Los componentes de este vector son de carácter binario, por lo que, si por ejemplo S(4)

= 1, nos está indicando que el arco al que hemos designado el número 4 está activado. En caso contrario, no estaría activado. Los componentes de este vector S son los que vamos a ir variando, por lo que, podemos considerarlo como el vector "solución".

- C(S) es un valor el cual nos indica cuántos nodos estarían desconectados si nuestro árbol estuviera compuesto por los arcos cuyo componente en un vector S concreto es igual a 1. Nos interesa que C(S) sea cero, para que así evitar penalizar en la función fitness.

- E(S) nos indica cuántos componentes de un vector S son iguales a1, es decir cuántos arcos están activados. Nos interesa que E(S) siempre coincida con el número de nodos menos 1.

- W(i) indica el peso asociado a el arco i. Recordamos que debemos encontrar la combinación de arcos que nos ofrezca el mayor peso.

- $\sum_{i|Si=1} Wi$ es la suma total de todos los pesos de los arcos cuyo componente asociado en el vector S es igual a 1.

- Wub es una constante que vamos a utilizar como coeficiente de penalización. Debe ser lo suficientemente grande como para devaluar nuestra función fitness en el caso de que exista penalización pero prestando interés en que si es demasiado grande, el valor de la suma de los pesos de los arcos puede quedar insignificante al lado de éste y el método no funcionaría.

- N es el número total de nodos. A es el número total de arcos

Función fitness:

$$W(S) = \sum_{i|Si=1} Wi - C(S) * Wub - [E(S) - (N - 1)]$$

Para calcular E(S), sólo debemos de realizar la suma de todos los componentes de un vector S.

Para calcular C(S), debemos usar una matriz de incidencia, la cual nos dirá que nodo está inconexo. C(S) corresponde a la cantidad total de éstos.

Ya dijimos que se habían realizado algunas modificaciones en el algoritmo. Una de ellas ha sido evitar la penalización del número de arcos igual a N-1. Para ello debemos obligar a que los vectores S tengan siempre N-1 componentes iguales a 1.

Eliminamos la penalización y obtenemos la función fitness reducida.

Función fitness reducida:

$$W(S) = \sum_{i|Si=1} Wi - C(S) * Wub$$

8.2 Algoritmo 1 “Randomized Local Search” (RLS). Neumann, F., & Wegener, I. (2006).

1. Elegimos los componentes binarios de S aleatoriamente. Debemos asegurarnos que N-1 de ellos sean de valor 1.

2. Elegimos un número binario B aleatoriamente.
   2.1 Si B=0 elegimos un número K y T aleatoriamente, asegurándonos que S(K) sea distinto de S(T). Cambiamos los valores binarios de S(T) y S(K), si eran 1 serán ahora 0 y viceversa.
   2.2 Si B=1 elegimos cuatro números aleatoriamente K,T,R,L asegurándonos que el componente de S de dos de ellos valga 1 y el componente S de los otros dos valga 0. Cambiamos los valores binarios de S(T) ,S(K),S(R),S(L). Si eran 1 serán ahora 0 y viceversa.
   * Hacemos el cambio así para asegurarnos que en el vector S siempre hay un total de N-1 componentes que valen 1.

3. Tras los cambios, tenemos el vector S’

4. Si W(S’) es mayor que W(S), reemplazamos S con los valores de S’.

5. Volvemos al paso 2.

En esta aplicación, podemos fijar algún criterio de salida para parar el algoritmo cuando se llegue a un resultado concreto o se pueden fijar a priori un número de iteraciones, que será el número de veces que se repetirá el algoritmo. Se ha decidido utilizar la segunda opción ya que, a priori, no sabemos el resultado que podemos obtener.

En el Anexo 1 podemos encontrar el código del archivo Matlab (NEUMANN.m) donde se ejecuta este algoritmo para la red de prueba. En los primeros intentos obteníamos soluciones válidas pero tras varias iteraciones nos encontramos con un problema en el cual no habíamos reparado. A continuación se muestran los resultados obtenidos:

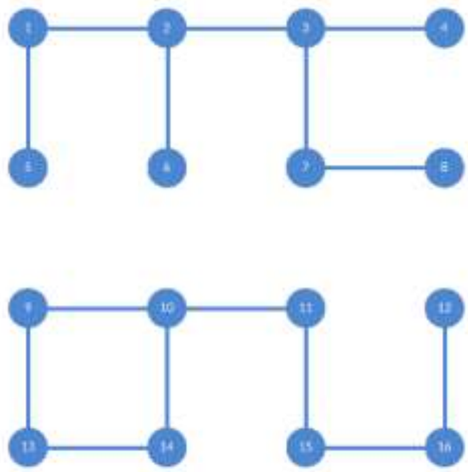


Imagen 10. Red Neumann 3

Como podemos comprobar, hemos obtenido una solución con N-1 arcos activados y todos los nodos conexos pero existe un ciclo. Esto se debe a que los árboles de expansión tienen estas dos características, pero existen más soluciones con esas características que no son árboles de expansión.

Podemos tener suerte, como en las primeras iteraciones, y encontrar soluciones válidas en redes pequeñas pero a manera que crezca la red tendremos más posibilidades de encontrar ciclos. Por lo tanto, este método no nos asegura encontrar el árbol de expansión máxima y su aplicación a la red de Sevilla va a resultar prácticamente inútil salvo una gran casualidad propia de la red.

8.3 Ventajas e Inconvenientes

Ventajas del método:

- Sencillo y fácil de programar
- Busca en la vecindad, de forma que facilita la búsqueda del óptimo
- Resulta muy rápido

Inconvenientes:

- No garantiza la no existencia de ciclos. Por lo tanto no nos soluciona el problema.

En la siguiente página podemos encontrar un diagrama de flujo que expresa el funcionamiento de este algoritmo.

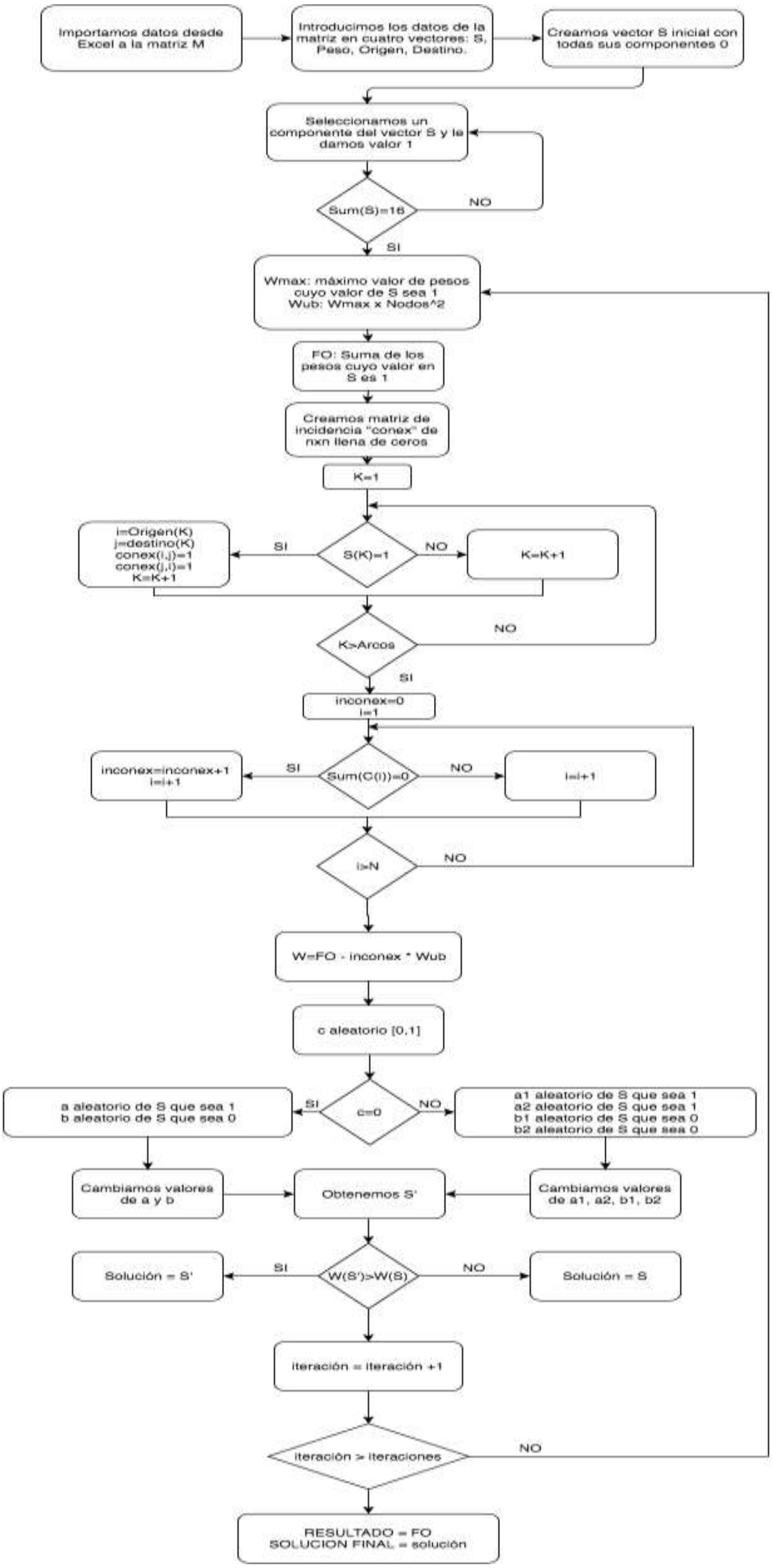


Imagen 11. Diagrama Neumann

9 SEGUNDO MÉTODO. SIMULACIÓN MONTECARLO

El método anterior concluía con el problema de no poder evitar la aparición de ciclos. Una solución habría sido intentar detectar los ciclos que se producen y penalizarlos en la función fitness. El problema de esta solución es que es complicado y laborioso elaborar un método para detectar ciclos, además de que siempre dependería del grafo que estuviéramos utilizando y sería realmente complicado crear un método genérico para obtener el árbol de expansión detectando y penalizando los ciclos.

Ante esta gran dificultad, se ha decidido tomar otro camino totalmente opuesto al realizado en el primer método. No vamos a implementar ningún algoritmo donde detectemos ciclos pero, en cambio, vamos a crear desde cero un algoritmo que nos proporcione un árbol, asegurándonos que no los tiene.

Una vez obtenido ese árbol, debemos evaluar la función objetivo, que se corresponderá con la suma total de pesos y compararlo con el valor de la función objetivo de otro árbol.

Con el método utilizado, cada árbol obtenido es completamente aleatorio y depende de las decisiones tomadas a partir de que se empieza a crear el árbol. Este hecho impide que podamos utilizar las soluciones vecinas como en la heurística anterior. El hecho de cambiar un arco por otro no nos asegura que no existan ciclos ya que cada arco activado, depende de la activación de todos sus arcos anteriores.

Por todo ello, se ha decidido utilizar el método de búsqueda Montecarlo.

9.1 Características del Método

> La simulación de Monte Carlo es una técnica cuantitativa que hace uso de la estadística y los ordenadores para imitar, mediante modelos matemáticos, el comportamiento aleatorio de sistemas reales no dinámicos (por lo general, cuando se trata de sistemas cuyo estado va cambiando con el paso del tiempo, se recurre bien a la simulación de eventos discretos o bien a la simulación de sistemas continuos).
> La clave de la simulación MC consiste en crear un modelo matemático del sistema, proceso o actividad que se quiere analizar, identificando aquellas variables (inputs del modelo) cuyo comportamiento aleatorio determina el comportamiento global del sistema. Una vez identificados dichos inputs o variables aleatorias, se lleva a cabo un experimento consistente en generar muestras aleatorias (valores concretos) para dichos inputs, y analizar el comportamiento del sistema ante los valores generados. Tras repetir n veces este experimento, dispondremos de n observaciones sobre el comportamiento del sistema, lo cual nos será de utilidad para entender el funcionamiento del mismo. Obviamente, nuestro análisis será tanto más preciso cuanto mayor sea el número n de experimentos que llevemos a cabo. (Faulín & Juan, 2005, p.3)

En definitiva, debemos crear X árboles que cumplan las condiciones que necesitamos y evaluarlos quedándonos con la solución que más nos satisfaga, siendo X el número de iteraciones. En nuestro caso, elegiremos la que mayor suma de pesos de los arcos elegidos nos proporcione.

Para asegurarnos de que nuestro árbol obtenido no tiene ciclos y conecta a todos los nodos, hemos seguido un proceso en el cual vamos seleccionando nodos y añadiéndolos a un conjunto, el cual llamaremos nodos de origen, pero evitando que los arcos unan nodos con otros del mismo de ese conjunto. Lo explicamos a continuación de una forma más concreta y gráfica.

9.2 Descripción del proceso

1. Elegimos un nodo inicial de manera aleatoria

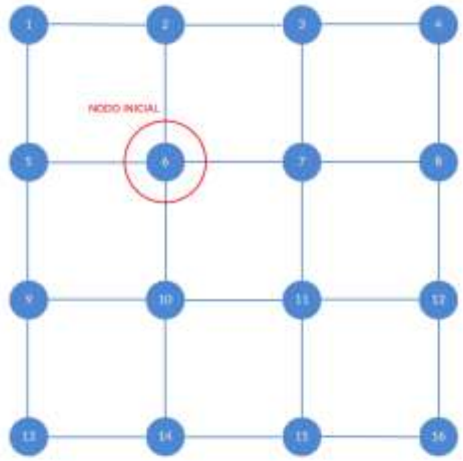


Imagen 12. Red Montecarlo 1

2. Lo introducimos en un conjunto o vector, donde se guardarán los nodos de origen. Lo llamaremos vector “O”

3. A continuación, debemos observar que nodos están conectados con alguno de los nodos que tenemos en el vector O, Hasta ahora sólo tenemos un nodo, por lo tanto debemos ver que nodos están conectados a él. Éstos serán los aspirantes para ser el nodo destino.

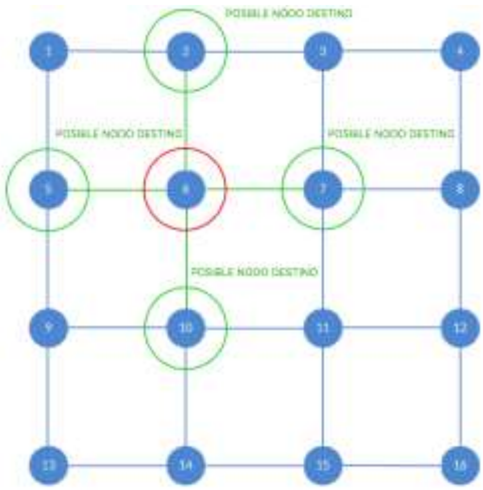


Imagen 13. Red Montecarlo 2

4. Aleatoriamente, elegimos un nodo destino. Este nodo destino no puede formar parte del vector de nodos de origen. El arco que los une, ya forma parte del árbol de expansión. El nodo destino ahora pasa a formar parte del vector origen y se convierte en un nodo origen.

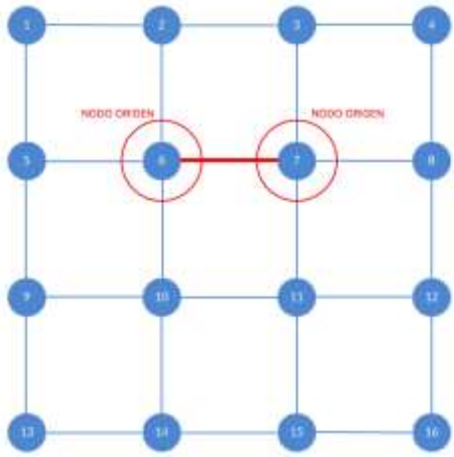


Imagen 14. Red Montecarlo 3

5. Repetimos el procedimiento de elección de nuevo nodo destino

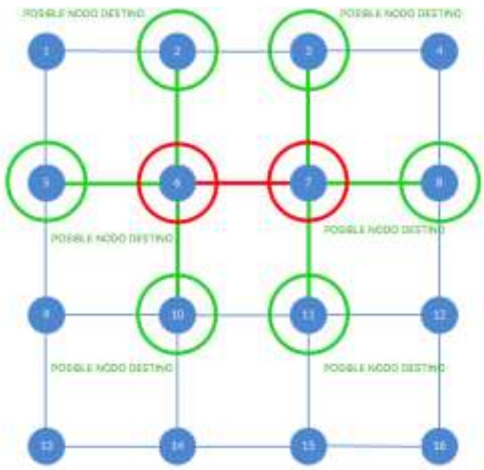


Imagen 15. Red Montecarlo 4

6. Hemos de repetir el procedimiento hasta que todos los nodos estén en el vector de origen, eso significará que todos los nodos están conectados.

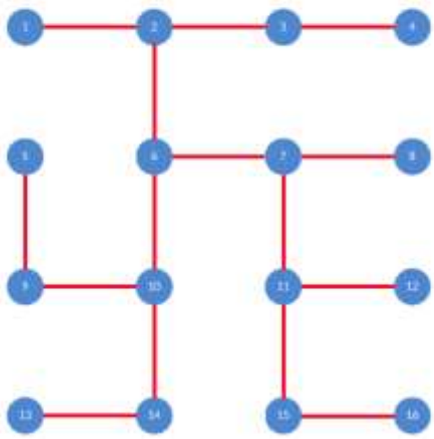


Imagen 16. Red Montecarlo 5

En la anterior imagen podemos observar el resultado final donde comprobamos que hemos obtenido un árbol de expansión con los N nodos conectados y N-1 arcos activados. Probablemente no sea el de expansión máxima debido a que hemos elegido los nodos y arcos de una forma totalmente aleatoria. Aquí es donde entra en juego la simulación Monte Carlo, pues debemos ir creando un determinado número árboles aleatorios hasta quedarnos con el de expansión máxima. Mientras mayor sea el número de árboles evaluados, más probabilidades habrá de encontrar el máximo y mientras mayor sea el número de nodos y arcos, más complicado será. Por tanto, a mayor número de nodos y arcos, más cantidad de árboles debemos evaluar.

En al Anexo 2 podemos encontrar el código del archivo Matlab (MONTECARLO.m) donde se ejecuta este algoritmo para la red de prueba. Empezando por un máximo igual a 0, cada vez que el algoritmo encontraba un valor de la función objetivo mayor que el máximo, actualiza éste máximo y almacena el valor encontrado. Esto significa que aunque no lleguemos al árbol de expansión máxima, podemos quedarnos con el que se acerque más, al menos en el valor de la función objetivo ya que la forma del árbol de expansión obtenido no tiene obligación de parecerse al árbol de expansión máximo. Cuantificando un poco el proceso, la primera vez que puse en marcha el algoritmo, necesité 11507 iteraciones para encontrar el árbol de expansión máxima, sin embargo, la segunda necesité 5893 iteraciones. Esto describe la pura naturaleza aleatoria del método ya que, cuantas más iteraciones más probabilidades de encontrar el máximo tenemos pero nada nos impide encontrarlo en la primera iteración o ni siquiera encontrarlo. Cada solución no tiene nada que ver con la anterior o la siguiente, esto se debe al no uso de las soluciones vecinas ya que el método nos lo impide. Para la red de Sevilla vamos a necesitar muchas más iteraciones y mucho más tiempo debido a que tiene muchos más nodos y arcos.

Analicemos entonces los resultados obtenidos. Mirando la imagen podemos comprobar que hemos obtenido el mismo árbol que obteníamos cuando solucionábamos el problema con el software de optimización Solver. Esto indica que hemos conseguido encontrar el árbol de expansión máxima. No encontramos los problemas que teníamos en el primer algoritmo ya que hemos forzado que cada árbol tuviese N nodos conectados, N-1 arcos activados y hemos evitado a toda costa los ciclos.

Podemos considerar éste método bastante menos eficiente respecto al tiempo de búsqueda que el primero, sin embargo es mucho más eficaz dado que las soluciones que aporta siempre serán válidas.

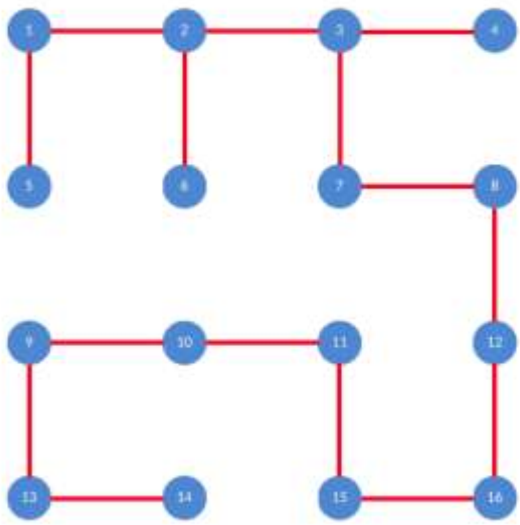


Imagen 17. Red Montecarlo 6

9.3 Ventajas e Inconvenientes

Ventajas del método:

- Garantiza la no existencia de ciclos. Por lo tanto nos ofrecerá siempre un árbol de expansión, sea o no el máximo.

Inconvenientes:

- A pesar de un código no muy largo, bastante más complicado de programar.
- No busca en la vecindad, lo que dificulta la búsqueda del óptimo.
- Resulta muy lento.

En la siguiente página podemos encontrar un diagrama de flujo que expresa el funcionamiento de este algoritmo.

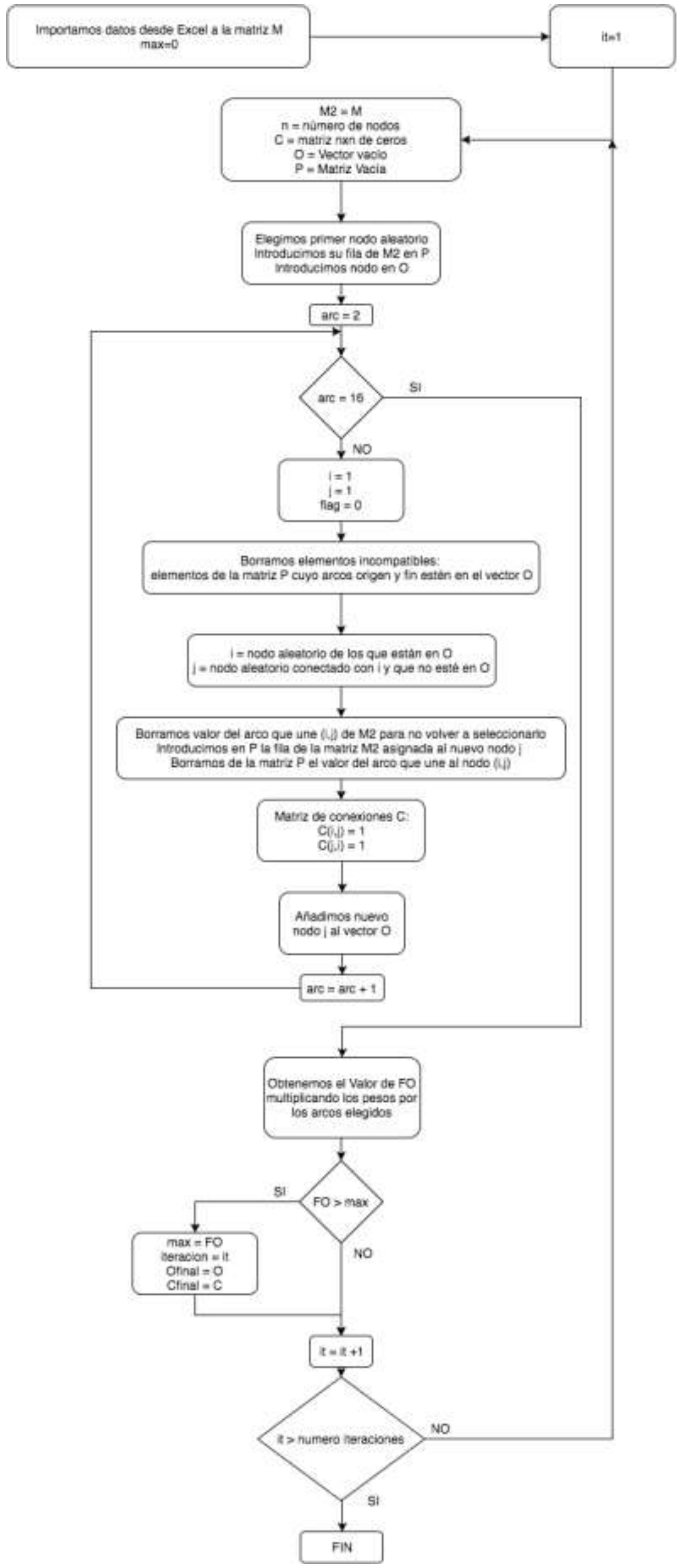


Imagen 18. Diagrama Montecarlo

10 RED DE SEVILLA

10.1 Introducción a la Red de Sevilla.

Una vez validado el modelo y elegido la heurística que vamos a utilizar para la búsqueda del árbol de expansión máxima, llegamos al principal propósito de este proyecto, calcular el árbol de expansión máxima para la red de Sevilla obteniendo así una propuesta semafórica.

Cuando decidimos elegir la segunda opción de cálculo para el árbol de expansión máxima que corresponde con el método de búsqueda Montecarlo aplicado a un algoritmo del que obteníamos un árbol de expansión, hablábamos de que teníamos una gran pérdida de eficiencia ya que no podemos utilizar las soluciones vecinas en cada árbol sino que debemos crear cada uno aleatoriamente desde el primer nodo. Debemos tener en cuenta también, que el tiempo que tarda el algoritmo generador de árboles de expansión en que se basa el método aumentará exponencialmente con cada nodo y arco nuevo añadido a la red. Esto significa que encontraremos una diferencia de tiempos abismal entre un árbol de 16 nodos como nuestra red de prueba y otro, por ejemplo, de 40 nodos. Pues bien, la red de Sevilla podría constar sin problemas de más de 40 nodos, incluso trazando una red reducida de ésta. Finalmente, también podemos percatarnos de que, mientras más arcos tenga la red, más posibilidades de árboles de expansión distintos existen y por lo tanto, más cantidad de árboles hay que probar para llegar al de expansión máxima.

En resumen, aplicando el método a la red de Sevilla, el algoritmo tardará mucho más en crear un árbol y debemos realizar muchas más iteraciones debido a que existen un mayor número de árboles diferentes. El tiempo de espera para encontrar la solución máxima va a incrementarse mucho, pudiéndose tratar incluso de días con un equipo de gama media salvo excepciones.

A continuación vamos a detallar el tipo de red que vamos a crear para Sevilla ya que no entrarán todas las calles y semáforos en su totalidad.

Nos vamos a centrar en las principales avenidas y calles dado que nos interesa mover el mayor flujo de vehículos por nuestra Onda Verde. Las calles pequeñas suelen tener poco flujo pero si en cualquier caso las incluyéramos en nuestra red de nodos y arcos, estos arcos tendrían un peso muy pequeño comparado con el peso de los arcos de grandes avenidas y es bastante probable que participaran en nuestro árbol de expansión máxima.

En el caso concreto de Sevilla, hay una gran zona que vamos a obviar, dado que está compuesta principalmente por calles pequeñas y semáforos cuya utilidad es proporcionar paso a peatones salvo alguna excepción. Se trata de la zona Centro, la cual delimita al norte con Calle Resolana y Ronda Capuchinos, al oeste con Calle Torneo y Paseo Cristóbal Colón que transcurren paralelas al Río Guadalquivir, al este con Calle Recaredo y Avda. Menéndez Pelayo y por el sur el Costurero de la Reina.

La zona de más densidad de circulación de Sevilla corresponde con la circunvalación de la SE-30. Antes la de construcción de esta circunvalación-autovía, la ronda que estaba en su lugar estaba plagada de semáforos, los cuales desparecieron tras la construcción de ésta. La Ronda Urbana Norte, que comprende entre el Puente del Alamillo y el cruce de la Avda. Kansas City con la SE-30 es un tramo de vía que conecta dos extremos de SE-30. Esta ronda sí posee varios semáforos, los cuales sí van a formar parte de nuestra red a diferencia del tramo de autovía correspondiente a la SE-30.

Los últimos tramos que no vamos a incluir en nuestra red de Sevilla corresponden a la Ronda Supernorte (SE-20). Esta ronda se extiende desde las inmediaciones del Estadio de La Cartuja, donde se encuentra con la SE-20 convirtiéndose la Avda. Carlos III, hasta la Autovía de Andalucía en las inmediaciones del Aeropuerto de Sevilla. Se creó en la Expo'92 como un enlace entra la Isla de La Cartuja y el Aeropuerto y contaba con varios semáforos. En la actualidad queda sólo un semáforo y corresponde con la entrada a una vía auxiliar sin mucha importancia por lo que no tiene mucho sentido incluir esta ronda por un semáforo donde tenemos prácticamente asegurado que la Onda Verde dará prioridad a los vehículos que circulan por la SE-20.

10.2 Construcción

Para construir la red de Sevilla, se ha partido desde unos datos iniciales. Estos datos vienen reflejados en unas tablas de Excel donde se puede diferenciar una cantidad determinada de nodos y los respectivos arcos que los conectan. Inicialmente, partimos con una red de 1221 nodos y 4310 arcos. Al final del método de búsqueda elegido, concluíamos que el tiempo de cálculo aumentaba exponencialmente a medida que añadíamos arcos y nodos a la red. Comentábamos que había una gran diferencia entre el tiempo para calcular el árbol de expansión máxima de una red de 16 nodos que, por ejemplo, una red de 40. Si con estas cantidades esta diferencia era tan grande, la diferencia entre el número de nodos y arcos de la red de prueba y el número de nodos y arcos de la red inicial de Sevilla es tan grande comparada con la otra que ni siquiera se nos ocurre plantearnos la resolución de esta red.

En el apartado anterior hacíamos algunos apuntes de zonas de Sevilla que íbamos a obviar dada su baja importancia o lo poco que nos condiciona los resultados. A continuación vamos a transformar la red inicial de 1221 nodos y 4310 arcos en una red simplificada la cual nos permitirá al menos encarar el problema a revolver y obtener una solución factible. Vamos a observar también la evolución que sufre la red con cada transformación, hasta llegar a la red simplificada.

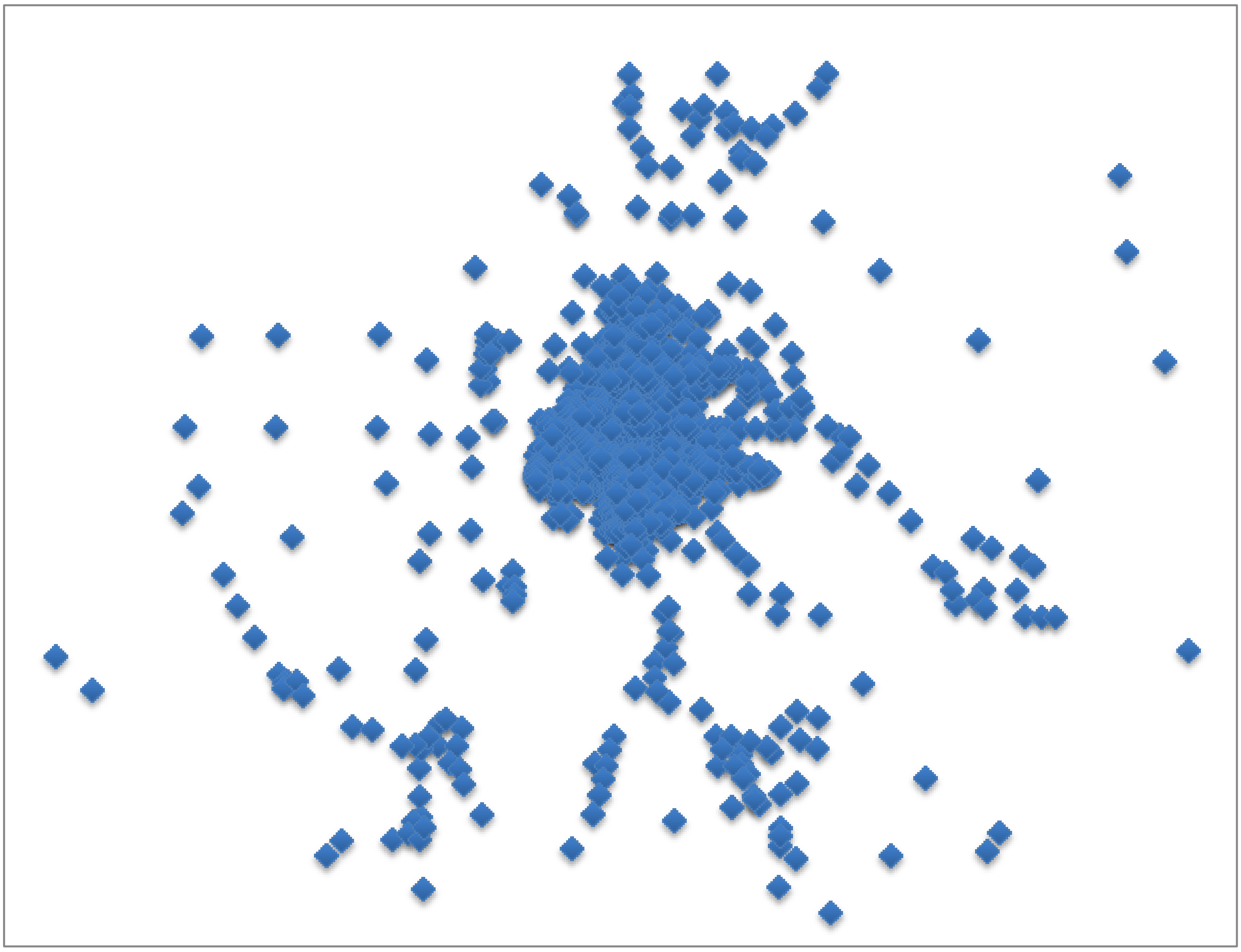


Imagen 19. Red Sevilla Inicial

La anterior imagen nos muestra la red inicial. Como podemos observar, es casi imposible diferenciar las vías de circulación de la ciudad. Difícilmente nos percataríamos de que se trata de la ciudad de Sevilla sin saberlo a priori.

Un dato que sabemos de partida, es que los nodos con referencia del 1 al 183 son nodos ficticios, por lo tanto no intervendrán en nuestra red, al igual que cualquier arco que conecte a alguno de éstos. Los 183 nodos eliminados no son apenas nada comparado con los 1221 que tenemos, por ello la red permanecería

prácticamente igual a la imagen original.
A continuación vamos a realizar la primera criba importante en nuestra red. Observando las tablas, en concreto la de los arcos, nos percatamos de que el peso de los nodos oscila entre 0 y 4771. Esto quiere decir que hay una serie de arcos cuyo peso es desproporcionadamente bajo comparado con los arcos de más peso. Traduciéndolo a nuestro problema, estos arcos van a tener muy poca prioridad a la hora de ser elegidos para formar parte del árbol de expansión máxima. La primera criba consiste en eliminar los nodos cuyos arcos no superen el valor de 1000 en el peso. Nos quedamos pues, con los nodos que tengan al menos un arco con valor superior a 1000. Veamos cómo quedaría la red.

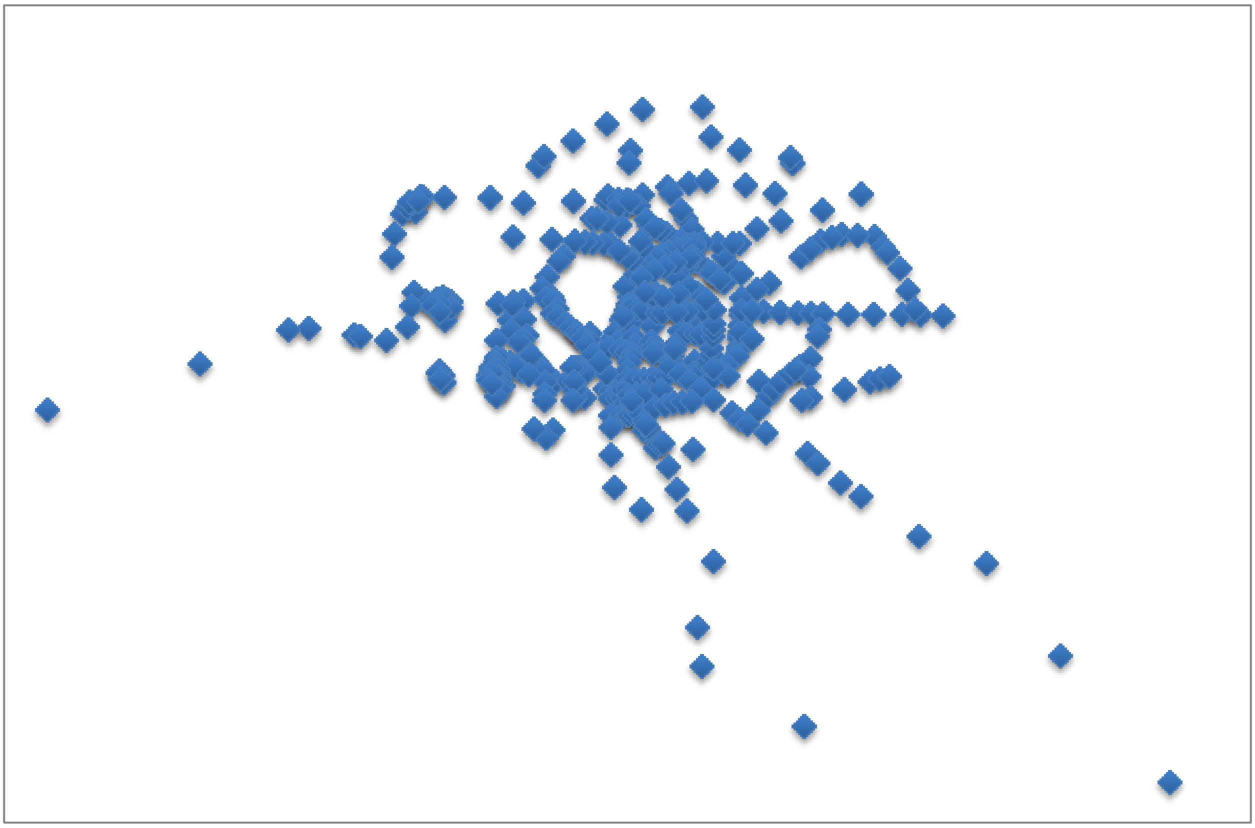


Imagen 20. Red Sevilla Filtrada

Ahora ya sí podemos diferenciar algunas áreas de Sevilla. Por ejemplo, podemos percatarnos de que la zona que corresponde al centro se ha quedado vacía tal y como se esperaba, debido al poco flujo de vehículos y a la poca importancia de sus semáforos como explicábamos en la introducción de la red de Sevilla. También podemos observar algunas vías radiales que parecen salir desde el centro hacia los extremos. Éstas se corresponden a las diferentes vías de acceso a Sevilla, tales como la Autovía de Andalucía (A-4) al norte, la A-92 dirección Granada por el este, la A-376 hacia Montequinto al suroeste, la entrada de la A-4 que enlaza con la Autopista AP-4 por el sur y la A-49 hacia el Aljarafe.

Esta reducción ya sí es bastante considerable ya que pasamos a tener 411 nodos 712 arcos. Es una reducción importante pero, aun así, sigue siendo inviable calcular el árbol de expansión máxima con nuestro algoritmo.

El hecho de quitar los arcos y nodos a ciegas puede ocasionarnos ciertos problemas a la hora de comparar nuestra red con el mapa original de Sevilla. Una vía con poco flujo puede ser clave para conectar dos vías importantes, por lo tanto, el arco correspondiente a esa vía deberá ir incluido.

Para continuar con la criba de nodos y arcos, no es recomendable seguir un criterio de pesos del arco, ya que podemos considerar que todos los arcos con peso mayor de 1000 van a tener un factor importante en nuestra red. La solución que nos queda es eliminar nodos manualmente. Es un trabajo largo y bastante laborioso, pero a la hora de calcular el árbol de expansión máxima, nuestro algoritmo nos lo agradecerá.

Vamos a establecer una serie de criterios y métodos de eliminación de nodos y sus correspondientes arcos.

1. No incluimos SE-30 y SE-20 (Ronda Supernorte). De las medidas que vamos a tomar, ésta es probablemente la que menos reduzca el tamaño de nuestra red, pero simplificaría bastante ya que se tratan de dos vías importantes que ya no formarían parte de nuestro problema. Además de estas dos vías, tenemos que tener en cuenta el tramo que une el Puente del Alamillo con la Avda. Carlos III en La Isla de La Cartuja ya que en éste no encontramos semáforos pero sí los encontramos al entrar en la avenida citada.

2. Los arcos y nodos a los cuales llamaremos "radiales" tampoco van a intervenir en nuestra red. Estos nodos se encuentran a lo largo de vías de salida o entrada de Sevilla, y en cuanto superan el límite de la ciudad, estos dejan de estar conectados con más nodos que el que le precede y/o sucede. Veamos un ejemplo gráfico.

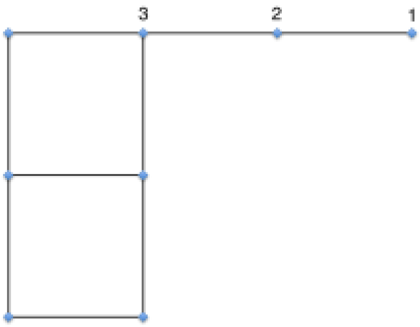


Imagen 21. *Red Radiales 1*

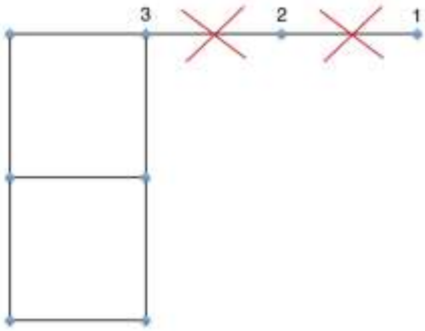


Imagen 22. *Red Radiales 2*

Como vemos en la imagen de la izquierda, tenemos dos nodos aislados que salen desde el "núcleo" de la red hacia fuera y sólo dependen del anterior y/o siguiente nodo con el que está conectado. Llamaremos "nodos radiales" a los nodos 1 y 2. Recordamos que para obtener nuestro árbol de expansión máxima todos los nodos deben están interconectados y el nodo que está más al extremo, nodo 1, sólo posee un arco con el que puede estar conectado, es decir, tiene una única posibilidad de conexión, por lo tanto ese arco que conecta nodo 1 con nodo 2 obligatoriamente va a formar parte del árbol de expansión. Como sabemos esta obligación, podemos eliminar este nodo, ya que ralentizaría el proceso y al fin y al cabo el resultado obtenido va a ser el mismo. Lo mismo sucede con el nodo 2. Debido a que el nodo 2 está vinculado a la fuerza con el 1 y éstos a su vez deben estar conectados con el árbol entero, el nodo dos debe de tener otra conexión, la cual lo una al árbol. La única opción que le queda al nodo 2 es el arco que lo une con el nodo 3, el cual también va a formar parte del árbol de expansión máximo obligatoriamente. Al quitar estos nodos debemos tener en cuenta que no los estamos eliminando totalmente. Es decir, no los eliminamos de la red de prueba porque no son importantes sino que los obviamos porque consideramos que su incorporación al árbol es obligatoria en cualquiera de los casos.

3. Además de los nodos radiales, existe otro tipo de disposición en el grafo la cual designa una serie de nodos a los cual llamaremos "nodos intermedios". Un nodo intermedio es un nodo que sólo está conectado con el nodo que le precede y con el que le sucede, formando una cadena que irá desde un nodo a otro, no intermedios. Los nodos que enlazan la cadena formada por los arcos de los nodos intermedios pasarán a llamarse nodos iniciales y finales. Éstos sí podrán poseer arcos con los distintos nodos del grafo. Veamos un ejemplo.

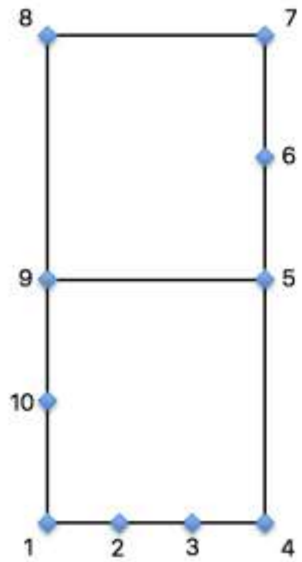


Imagen 23. Red Intermedio 1

En la imagen anterior podemos observar una cadena de 10 nodos conectados mediante 11 arcos. Si observamos el nodo 6, podemos ver que sólo está conectado con el nodo 5 y con el 7, por lo tanto, podemos considerarlo nodo intermedio. Lo mismo ocurre con los nodos 10, 2 y 3. Consideramos entonces los arcos 1, 4, 5, 7, 8, 9 como nodos iniciales o finales. El procedimiento de edición del grafo consiste en reemplazar los arcos que unen a los nodos intermedios con un arco único que una al nodo inicial y final. La siguiente ilustración nos lo muestra.

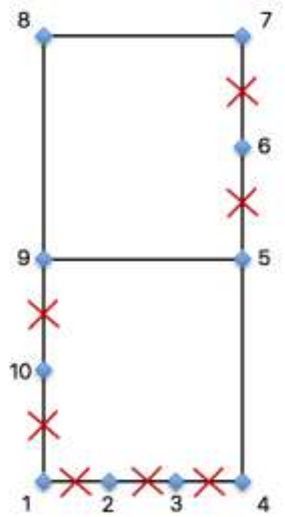


Imagen 24. Red Intermedio 2

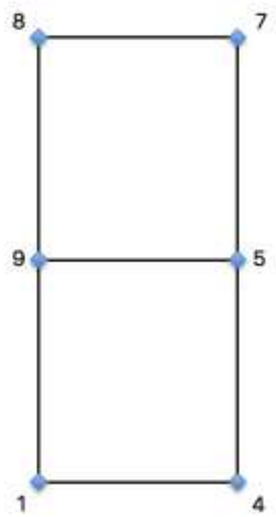


Imagen 25. Red Intermedio 3

Siendo el nodo 1 nodo inicial y el 4 final, o viceversa, hemos sustituido la cadena que formaba los arcos 1-2, 2-3, 3-4 por un único arco 1-4.

Siendo el nodo 1 nodo inicial y el 9 final, o viceversa, hemos sustituido la cadena que formaba los arcos 1-10 y 10-9 por un único arco 1-9.

Siendo el nodo 7 nodo inicial y el 5 final, o viceversa, hemos sustituido la cadena que formaba los arcos 7-6 y 6-5 por un único arco 7-5.

Finalmente tenemos una red de 6 nodos conectados 7 arcos.

Si nos fijamos bien en la red, los nodos 7, 8, 1 y 4 también podrían haber sido elegidos como nodos intermedios. La elección de estos nodos es personal y no necesariamente debemos elegir todos los nodos intermedios disponibles, sino que depende mucho de la red real de la que estemos tratando. Aplicando este criterio de eliminación de nodos, podemos imaginar una avenida con semáforos a lo largo de ella, pero que no todos ellos tienen la misma importancia o se encuentran en un cruce con otras vías. Los semáforos menos importantes serían los nodos intermedios.

Al reemplazar los arcos que unen a los nodos intermedios por el arco que va desde el nodo inicio al nodo fin, debemos asignarle un peso a éste último. El criterio de elección de peso para el arco ha sido elegir el mayor valor de peso de los arcos que unen a los nodos intermedios, asegurando la dirección a la que se obtiene ese valor ya que en los datos que se nos ha facilitado no tendrá el mismo valor el flujo 1-2 que el 2-1.

Tengamos en cuenta que si en el algoritmo el arco 1-4, por ejemplo, no es elegido para formar parte del árbol de expansión máxima, no quiere decir que ninguno de los arcos intermedios no formen parte de él. El hecho de que el arco 1-4 no entre en la solución indica que el semáforo 1 no estará sincronizado con el 4, pero la Onda Verde podría llegar, por ejemplo, del 4 al 3 y del 3 al 2 nunca llegando al 4. Adaptándolo al lenguaje común, podríamos circular por la vía 1-4 encontrándonos todos los semáforos en verde de los nodos intermedios pero al llegar al nodo 4 lo encontraríamos en rojo y ahí acabaría la Onda Verde.

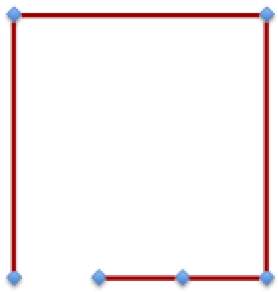


Imagen 26. Red Intermedio 4

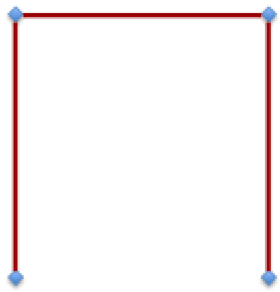


Imagen 27. Red Intermedio 5

Según nuestro criterio de edición de la red real de Sevilla, las imágenes superiores son equivalentes. La imagen de la izquierda es la que obtendríamos si no eliminásemos los nodos intermedios y la derecha corresponde al resultado eliminando los nodos intermedios.

4. En ciertos casos, nos encontramos con cierta disposición de nodos la cual es una combinación de los apartados 2 y 4. Es decir, encontramos arcos radiales unidos a nodos intermedios. Haciendo uso de los métodos explicados en sus respectivos apartados, podemos obviar algunos nodos y así reducir aún más la red. Veamos un ejemplo.

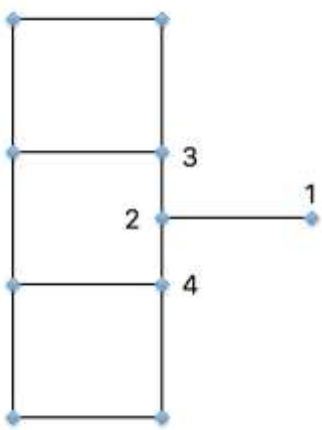


Imagen 28. Red Intermedio y Radial 1

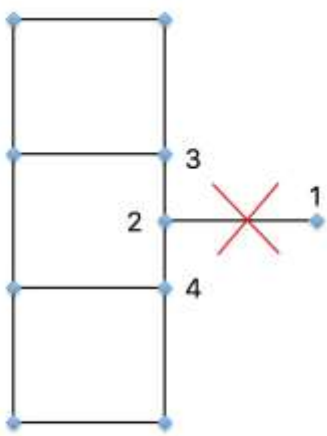


Imagen 29. Red Intermedio y Radial 2

En primer lugar, encontramos un nodo radial, el nodo 1. Atendiendo al apartado 2, el arco que une el nodo 1 con el nodo 2 puede ser obviado de nuestra red e incluido directamente en la solución final ya que es el único que conecta al nodo 1 y todos los nodos deben estar conectados. Nos quedaría de este modo.

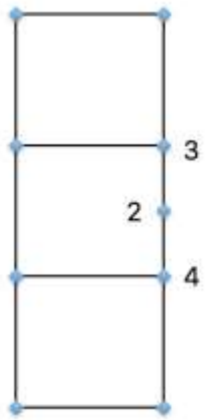
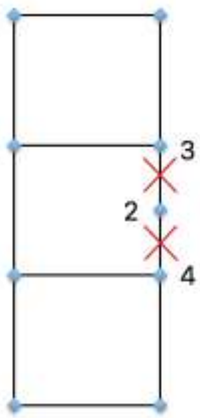


Imagen 30. Red Intermedio y Radial 3

El hecho de obviar el nodo 1 radial y quitarlo de nuestra red nos convierte el nodo dos en un nodo intermedio que antes no lo era. Podemos visualizar que en nodo 2 está conectado sólo por dos arcos a los nodos 3 y 4. Si no hubiéramos eliminado el arco 2-1 del grafo, el nodo 2 no sería un nodo intermedio.



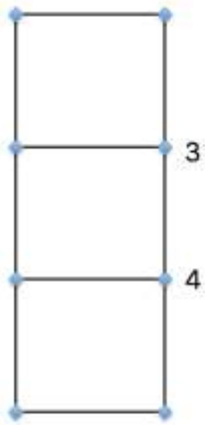


Imagen 31. Red Intermedio y Radial 4 *Imagen 32. Red Intermedio y Radial 5*

Procedemos a realizar el reemplazo de arcos sustituyendo los arcos 2-3 y 2-4 por el 3-4
Recordemos que a la hora de elegir el peso asignado al arco 3-4 vamos a tomar el mayor valor del peso contando todos los arcos intermedios, que son 2-3, 3-2, 2-4, 3-4.

Este último criterio de eliminación posee cierto problema. En el caso de que el arco que ha sustituido a los arcos intermedios no sea elegido por el algoritmo para formar parte del árbol de expansión máxima, el nodo radial se verá afectado ya que estará conectado con un nodo intermedio pero éstos estarán aislados del árbol. Veamos una ilustración.

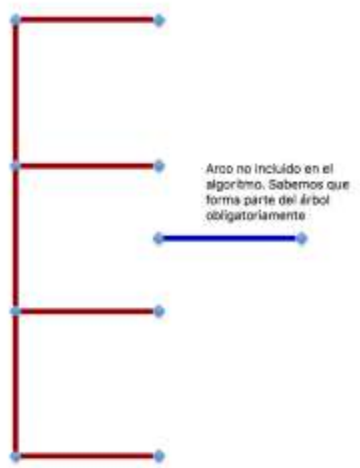


Imagen 33. Red Intermedio y Radial 6

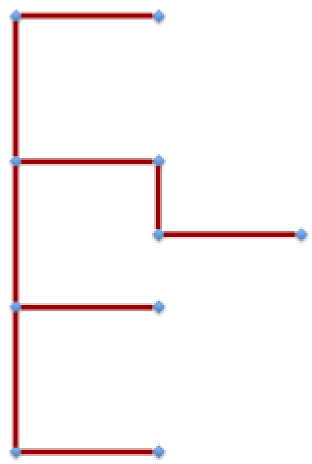


Imagen 34. Red Intermedio y Radial 7

En la figura de la izquierda apreciamos como el arco radial queda desvinculado del árbol. Tengamos en cuenta que ese arco no está siendo usado en el algoritmo, ya que si estuviera siendo utilizado, la solución no sería válida debido a que no cumplimos que el árbol esté formado por N-1 arcos, siendo N el número de nodos. Al no estar el arco radial dentro del algoritmo, sólo estamos utilizando los 8 nodos de la izquierda, sin contar con el radial ni el intermedio y la solución es viable.

Ante esta incongruencia, debemos saber que si el arco que ha sustituido a los arcos intermedios no entra en la solución de árbol, debemos añadir arcos intermedios hasta llegar a conectar al nodo que conecta con el arco radial. Ante la duda de que arco o arcos radiales elegir, elegiremos la combinación que nos proporcione mayor peso de los arcos. En la imagen de la derecha podemos observar el resultado.

La medida que más ha tenido que ver con el resultado final ha sido la eliminación de los nodos intermedios, junto a la primera criba de nodos que sólo estuvieran conectados por arcos menores de 1000. A continuación podemos observar el resultado.

Imagen 35. Red Sevilla Simplificada

Tomando estas medidas, hemos realizado una significativa reducción de arcos, y por lo tanto de nodos. La cantidad final de nodos es 65 y la de arcos de 109. Podemos ver los arcos finales en el Anexo 4. Esto supone una reducción considerable comparando con los 1221 nodos y 4310 arcos con los que contábamos en la red inicial. Esta cantidad de nodos hace posible que nos planteemos resolver el problema e incluso nos da ciertas esperanzas de la posibilidad de recalar en el árbol de expansión máxima con el algoritmo. Aun así, el tiempo que va a tardar en llegar a tal resultado va a ser considerablemente mayor dado que nuestra Red de Sevilla Simplificada tiene nada más y nada menos que la cantidad de nodos de la Red de Prueba multiplicado por 4, siendo 4.5 el factor de multiplicación en el caso de los arcos.

Aunque el tiempo de búsqueda sea notablemente mayor, no tiene sentido seguir reduciendo la cantidad de nodos y arcos ya que la red podría terminar perdiendo la semejanza a la red viaria principal de Sevilla, además de que si reducimos lo suficiente la red, podría ser posible resolverla con el modelo para el árbol de expansión máxima aplicado a un software de optimización y los esfuerzos para obtener el algoritmo que resuelva el árbol de expansión máxima para una red grande habrían sido en vano.

10.3 Resolución de la Red de Sevilla

Una vez tenemos el método de resolución y la red donde queremos aplicarlo, ya sólo queda poner en marcha el algoritmo y esperar mientras buscamos el óptimo.

En al Anexo 3 podemos encontrar el código del archivo Matlab (MONTECARLOSEVILLA.m) donde se ejecuta este algoritmo para la Red de Sevilla Simplificada. El único cambio que tiene efecto en éste es la matriz que contiene tantos los arcos como el peso de cada uno y el número de nodos y arcos. Como hemos comentado en apartados anteriores, esta red supone un gran incremento en cuanto a número de nodos y arcos y, además de que tardará más en crear un árbol de expansión y evaluarlo, tenemos que tener en cuenta que si tenemos más nodos y arcos también tendremos más posibilidades de árboles y por ello el número de iteraciones será mayor, al igual que el tiempo empleado en llevar a cabo éstas.

Si hablábamos que para la Red de Prueba bastaban unas horas para llegar al óptimo, incluso menos, para la Red de Sevilla se han llevado a cabo 4 búsquedas concluyendo en todas con la misma solución. Cada búsqueda ha sido aproximadamente de 15 horas, siendo aproximadamente 60 el tiempo total de búsqueda. El hecho de que en las 4 ocasiones hayamos encontrado la misma solución, nos indica que podemos estar en el óptimo y por ellos no se han realizado más búsquedas pero recordemos que se trata de una heurística, lo cual no nos garantiza alcanzar el óptimo y podría tratarse de mera casualidad.

Tabla 2. Búsquedas Árbol Expansión Máxima

Búsqueda	Resultado	Número de iteración
1	111894	85414
2	111894	31873
3	111894	82610
4	111894	104778

Si nos fijamos en la tercera columna, que nos indica en la iteración en la que se encontró el mayor valor alcanzado, el cual coincide en los cuatro casos, podemos percatarnos de la variabilidad de este factor ya que puede encontrarse en la iteración 31873 como en otra tres veces mayor que la primera. Esta variabilidad es fruto del carácter totalmente aleatorio del que hablábamos cuando explicábamos el Método Montecarlo.

Hemos obtenido una suma total de pesos de arcos que formar el árbol de expansión de 11894. Si ordenáramos los pesos de los arcos y eligiéramos los 64 mayores, que es el número de arcos que formarán el árbol de expansión obtendríamos el valor de 119007. Este valor nos indica el valor máximo al que podemos llegar, sin tener en cuenta obtener un árbol de expansión o no. Comparando ambos resultados comprobamos que nuestro algoritmo se ha acercado mucho a este valor y por lo tanto hay muchas probabilidades de que hayamos obtenido el árbol de expansión máxima.

A continuación se pueden visualizar los resultados aplicados a la Red Simplificada de Sevilla, y comprobar que tenemos un árbol de expansión, sin ciclos y todos los nodos conectados.

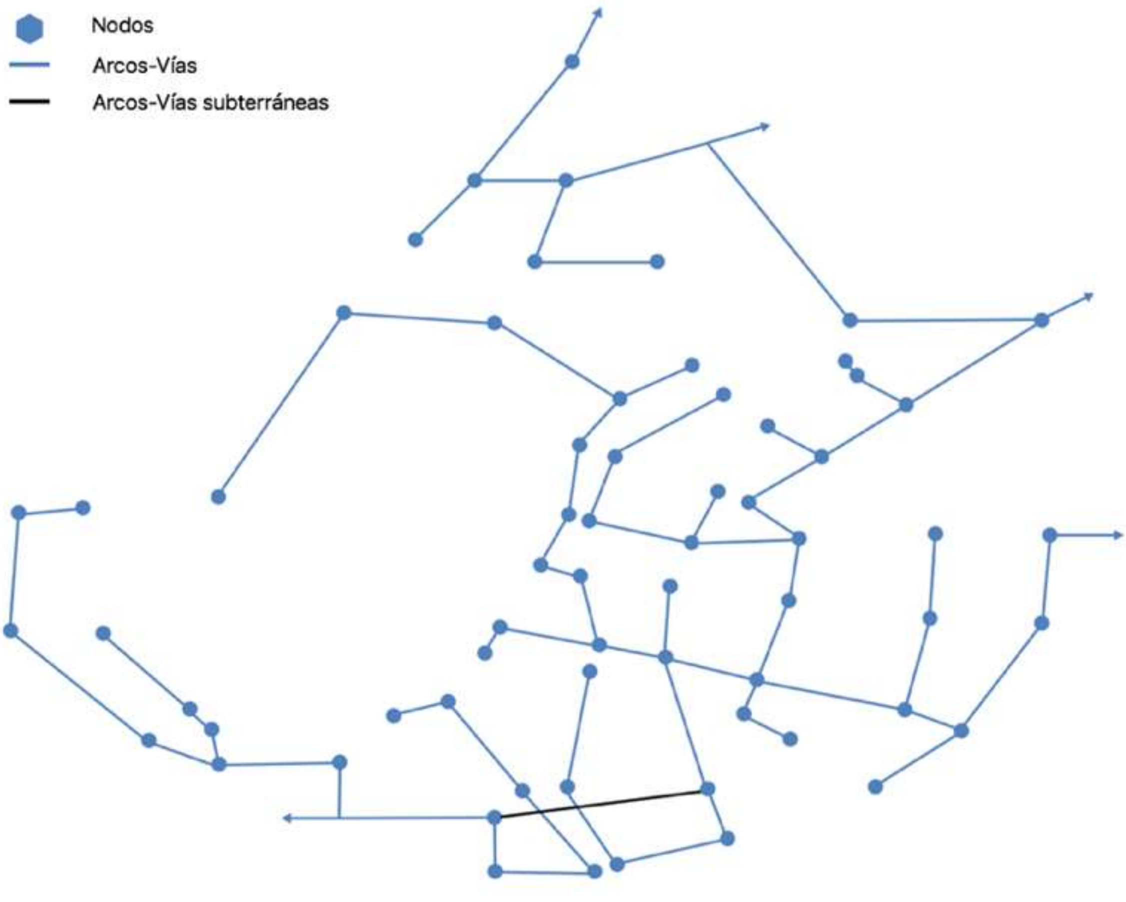


Imagen 36. Red Sevilla Resuelta

Cuando hablábamos de la reducción de nodos y arcos, concretamente cuando tratábamos con los arcos intermedios, decíamos que el peso que elegiríamos sería el mayor entre todos los arcos de esos arcos intermedios y siempre recordando el sentido de éste. Pues una vez que tenemos elegidos los arcos, podemos representar sobre la red el sentido de las vías.

11 COMPARACIÓN CON LA PROPUESTA ACTUAL

En este último apartado vamos a comparar los resultados obtenidos con un plano de Coordinación Semafórica de Sevilla que ha facilitado el tutor. La llamaremos "Propuesta 1" para agilizar las explicaciones.

A continuación se muestra la imagen del plano. Como podemos visualizar, la imagen se encuentra girada 90º. En esta propuesta de coordinación que nos ofrece la imagen podemos comprobar que no se han tenido en cuenta la formación de ciclos, sino que se limita a nombrar las vías por las cuales debería implementarse una Onda Verde, que se corresponde obviamente con las vías de mayor flujo de vehículos, e indicar el sentido de circulación. Notemos que al igual que en nuestra Red Simplificada de Sevilla, se han eliminado ciertas vías que, a pesar de ser muy transitadas, no poseen semáforos o no nos aportan nada importante a la hora de tomar decisiones para las Ondas Verdes. Podemos observar la no aparición de la Ronda Supernorte y la SE-30. Probablemente se trate de una propuesta de coordinación antigua ya que incluye por ejemplo la Avda. Constitución, la cual en la actualidad es peatonal y no se permite la circulación de vehículos salvo el tranvía.

Imagen 37. Red Propuesta 1

A continuación adjuntamos nuestra Red Simplificada de Sevilla con la diferencia de que en esta ocasión se han añadido el nombre de las vías más importantes para facilitar la comparación. A diferencia de los realizadores del plano de coordinación anterior, hemos decidido no voltear la imagen 90º, pero, como se puede comprobar en todas las imágenes de la Red de Sevilla, la relación de ejes no coincide con la realidad ya que para facilitar la modificación de ésta se ha realizado una reducción en el eje Y, quedando la red un poco "achatada".

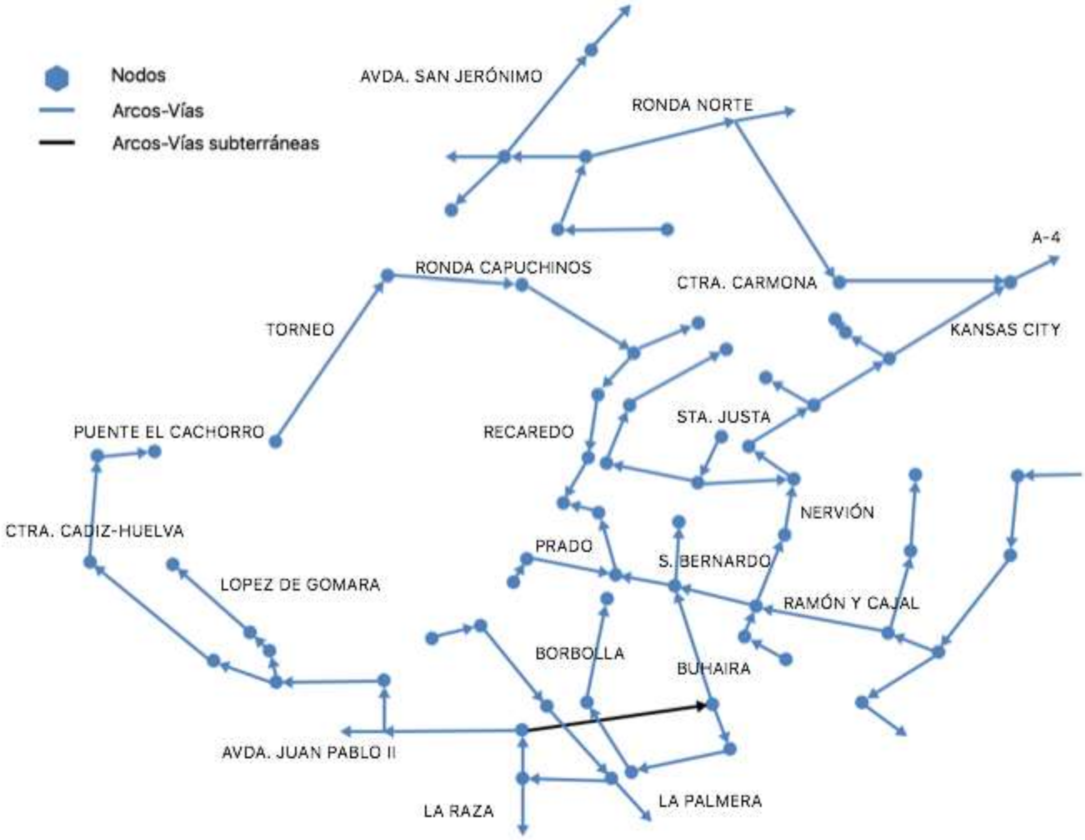


Imagen 38. Red Sevilla con sentido de la marcha

Como antes indicábamos, en la propuesta de coordinación con la cual nos estamos comparando, no tenemos un árbol de expansión, por lo tanto la comparación se va a realizar teniendo en cuenta dos factores.

En primer lugar, podemos observar que hay vías de nuestra propuesta que no aparecen en la otra y viceversa. Vamos a analizar estas vías, además del sentido de circulación de las vías que aparezcan en ambas propuestas simultáneamente.

Si empezamos por el norte geográfico, podemos comprobar que en la Propuesta 1 no se ha tenido en cuenta tanto la Avda. de San Jerónimo ni el tramo de Ronda Norte que va desde el Puente del Alamillo hacia la Glorieta Berrocal, llamada Rotonda del Carrefour Macarena comúnmente. Sin embargo sí se ha tenido en cuenta el tramo desde esta glorieta hacia el corte de la Avda. Alcalde Manuel del Valle con Carretera Carmona, coincidiendo el sentido de la circulación con nuestra propuesta. En cambio, notamos la aparición de Avda. Miraflores en la Propuesta 1, a diferencia de la nuestra, que fue eliminada en la primera criba de los arcos con peso menor de 1000.

El sentido del tramo de carretera Carmona de nuestra propuesta que está sincronizado coincide con la Propuesta 1.

Continuando por el este, encontramos la primera gran diferencia. La Avda. Kansas City, una de las más importantes está coordinada en la Propuesta 1 para dar prioridad a los vehículos entrantes en Sevilla, a diferencia que en nuestra propuesta. Notamos la ausencia de la Onda Verde en la Avda. Andalucía, hecho que nos sorprende. La prioridad la tiene en este caso la Avda. Cruz del Campo y San Francisco Javier, esta última en Nervión. La Avda. Cruz del Campo coincide en el sentido de la marcha, a diferencia que la Avda. San Francisco Javier, que es puesta entre las propuestas.

La Ronda del Tamarguillo, coincidiría totalmente en ambas propuestas además de ser la zona en la que más claro se ve la coincidencia de la Onda Verde

En la parte central del mapa, podemos observar como la Avda. Ramón y Cajal está coordinada en el mismo sentido en ambas propuestas, sin embargo, en el Prado San Sebastián se produce una diferencia. Esta diferencia es causada a la antigüedad de la Propuesta 1, ya que en ese tramo está impedida la circulación en el sentido que propone la Propuesta 1. Respecto a la Avda. Eduardo Dato, nuestra propuesta divide el sentido de la marcha según el tramo del que se trate, al contrario que la Propuesta 1. Esto significa que habrá tramos que el sentido coincida y los habrá que no coincida.

La Buhaira y la Avda. Borbolla toman importancia en la parte sur del mapa, estando ambas sincronizadas al revés en las propuestas. No se ha tenido en cuenta la Avda. de la Raza en la Propuesta 1 posiblemente por la antigüedad del plano pero podemos encontrar la Avda. de La Palmera en las dos propuestas. Eso sí, coordinadas al contrario en cada una.

Al suroeste encontramos una coincidencia en la coordinación en el Paseo de las Delicias, ahora llamado Avda. Juan Pablo II.

Finalmente, por el este impacta la no aparición de República Argentina en nuestra propuesta y la aparición del tramo de la Carretera Cádiz-Huelva que rodea a Triana. El tramo que une a ésta con la Avda. Juan Pablo II sí aparece y además está coordinada del mismo modo en las dos propuestas. Por el contrario, el sentido de coordinación de Ronda Triana no coincide.

Observando nuestra propuesta podemos percatarnos del detalle de que, en las rondas más externas de la ciudad, el sentido de circulación coincide con el sentido del movimiento de las agujas del reloj, subiendo hacia el norte por la izquierda y bajando por la derecha del mapa.

Por las vías internas a estas rondas, podemos observar como el sentido de las vías va hacia el sur por Avda. de Torneo, conectando con La Avda. de la Palmera, siendo las vías que transitan la Zona de Nervión, Gran Plaza, San Bernardo las que el sentido señala al norte.

12 CONCLUSIONES

Tras obtener el árbol de expansión máxima, con él realizar la propuesta para la sincronización semafórica en Sevilla y compararlo con la Propuesta 1 que corresponde con la propuesta actual, hemos obtenido algunas conclusiones respecto a los resultados.

El método que hemos utilizado, evitando los ciclos, proporciona aspectos tanto buenos como malos. El hecho de dividir la red viaria de Sevilla en nodos y arcos en vez de una vía completa concluye en que podemos encontrar varios tramos de una vía que forman parte de la Onda Verde y otros tramos que no. Sin embargo, representa sobre el grafo cómo actuaría la Onda Verde en la realidad, ya que la no aparición de ciclos asegura el flujo continuo de vehículos en cada cruce sin retroalimentarlo. Nuestra propuesta es viable, la otra sería imposible llevarla a cabo ya que si ponemos en marcha todas esas Ondas Verdes, se entorpecerían entre ellas, causando paradas y por tanto, la no ejecución de éstas.

Nuestra propuesta, incluso si resultados obtenidos no tuvieran nada que ver con la realidad, es una propuesta analítica que hemos implementado con unas restricciones. Ante los datos de densidad de tráfico de las vías de Sevilla, sería muy fácil elegir a simple vista las vías por donde pasaría la Onda Verde, obviamente por las vías con más flujos. Lo verdaderamente complicado es implementar una única Onda Verde que sincronice todas estas vías eligiendo las adecuadas para que el movimiento de vehículos seas máximo. Esa es la propuesta de este trabajo.

Podemos ver claramente un cambio real en la estructura viaria de la ciudad. La propuesta con la cual nos hemos comparado es antigua, y eso se puede observar en los resultados. La única coincidencia totalmente clara entre ambas se da en la Ronda del Tamarguillo, dado que el cambio del tráfico en ésta ha sido insignificante a lo largo de los años. En Kansas City encontramos otra gran coincidencia, aunque el sentido de la marcha esté invertido.

Las nuevas vías e infraestructuras, como el tramo subterráneo en la Calle Cardenal Bueno Monreal abren nuevas opciones y facilidades para el tráfico y hace que los flujos de las vías puedan cambiar, como ha pasado en este caso.

En cuanto al algoritmo, tomamos una decisión al elegir la segunda opción en base a la fiabilidad desechando la eficiencia. Con la red de nodos y arcos de Sevilla sin simplificar tomaría muchísimo tiempo obtener el árbol de expansión máxima, sin embargo, finalmente encontraría una solución en base a las condiciones que le hemos impuesto.

Al elegir una red de nodos y arcos no dirigidos, simplificábamos mucho el problema, sobre todo la búsqueda heurística. La resolución de una red de arcos dirigidos podría proporcionar una propuesta algo más firme y en concordancia con la realidad.

En definitiva, ésta es una propuesta simple que nos proporciona una primera visión de cómo podría ser la implementación de una Onda Verde en Sevilla. Un estudio mucho más minucioso siguiendo esta vía podría proporcionar la propuesta óptima que requiere una ciudad con una densidad de tráfico tan alta como Sevilla.

REFERENCIAS Y ENLACES

[1] Al-Mudhaffar, A. (2006). *Impacts of Traffic Signal Control Strategies*. (Doctoral Thesis). Royal Institute of Technology, Stockolm , Sweden.

[2] Bonneson, J., Sunkari, S., Pratt, M., & Songchitruksa, P. (2001). *Traffic signal operations handbook, Second Edition*. EEUU. Texas Transportation Institute.

[3] Capaldo, F., & Biggiero, L. (2012). Some surveys in order to static traffic light coordination. *International Conference on Traffic and Transport Engineering*. Belgrade

[4] Cardoso, E. & Moreno, V. (2012). Análisis y diseño de software para la sincronización de intersecciones semafóricas. *RIELAC Revista de Ingeniería Electrónica, Automática y Comunicaciones*, *33, 16-24.*

[5] Diaz, Mª.C., Diaz, J.F., Ferreiro, J.I., Pérez, Mª.T., Serrano, M.G., Tomás, R., Sentana, I., & Sentana, E. (2002, junio). Métodos Geométricos de Coordinación de Intersecciones Reguladas por Semáforos. *XIV Congreso Internacional de Ingeniería Gráfica.* Ingegraf, Santander. Descargado de https://rua.ua.es/dspace/bitstream/10045/21683/1/210.pdf

[6] Faulín, J., & Ángel, J.A. (2005). *Simulación de Montecarlo con Excel*. Descargado de http://www.uoc.edu/in3/emath/docs/Simulacion_MC.pdf

[7] García-Nieto, J., Alba, E., & Olivera, A. (2012). *Planificación de Ciclos en Semáforos con PSO: Casos de Estudio sobre Málaga y Sevilla.*

[8] Neumann, F. & Wegener, I. (2006). Randomized local search, evolutionary algorithms, and the minimunn spanning tree problem. *Theoretical Computer Science, 378, 32-40*.

[9] Onieva, L., Cortés, P., Muñuzuri, J., Guadix, & J., Ibáñez, J.N. (2006). *Métodos Cuantitativos y Organización de la Producción.* Madrid: Síntesis.

[10] Temario General de la ESTT – OEP. Parte 3: Gestión Técnica del Tráfico. (2014). Descargado de http://www.dgt.es/Galerias/la-dgt/empleo-publico/oposiciones/doc/2014/TEMA-3.26.doc

[11] Warberg, A., Larsen, J., & Jorgensen, R. M.(2008). Green Wave Traffic Optimization. *Informatics and Mathematical Modelling.* Technical University of Denmark

ANEXOS

ANEXO 1. Algoritmo Randomized Local Search

```
M=xlsread('DATOSPRUEBA');
w=0; %%valor inicial fitness%%

%%introducimos los datos de la matriz en P (pesos), D (destino)
y O (objetivo)%%
n=16;
k=1;
for i=1:n
    for j=1:n
        if j >= i
            if M(i,j)>0
                P(k)=M(i,j);
                O(k)=j;
                D(k)=i;
                k=k+1;
            end
        end
    end
end

%%creamos vector inicial s que nos cumpla e(s)=16
arc=length(P); %%arc: número de arcos%%
S=zeros(1,arc); %%S: vector solución%%
r=0;
while r < n-1
    ale=randi(arc,1,1);
    S(ale)=1;
    r=sum(S);
end

%%obtenemos wmax y wub%%
wmax=0;
for i=1:arc
    if (P(i)*S(i))>wmax
        wmax = P(i)*S(i);
    end
end

wub = n*n*wmax;

%%obtenemos sumatorio wi%%
FO=0;
for k=1:arc
    FO=FO+(S(k)*P(k));
end

%%matriz "conex"%%
conex=zeros(n);
for k=1:arc
```

```
    if S(k)==1
        i=D(k);
        j=O(k);
        conex(i,j)=1;
        conex(j,i)=1;
    end
end

%%observamos columnas vacías, nos proporciona los nodos
inconexos%%
inconex=0;
for i=1:n
    if sum(conex(i,:))==0;
        inconex=inconex+1;
    end
end

%%función fitness%%
 w=FO-(inconex*wub);
 solucion=FO;
 conexionesreales=conex;
 vectorsolucion=S;


 %%iteraciones%%
 for it=1:20000;
 FO=0;
 SOL=S;
 c=randi([0,1],1,1);
 if c==0
     a=randi(arc,1,1);
     b=randi(arc,1,1);
     while SOL(a)==0
         a=randi(arc,1,1);
     end
     while SOL(b)==1
         b=randi(arc,1,1);
     end
     SOL(a)=0;
     SOL(b)=1;
 else
     a1=randi(arc,1,1);
     a2=randi(arc,1,1);
     b1=randi(arc,1,1);
     b2=randi(arc,1,1);
     while SOL(a1)==0 && SOL(a2)==0
         a1=randi(arc,1,1);
         a2=randi(arc,1,1);
     end
     while SOL(b1)==1 && SOL(b2)==1
         b1=randi(arc,1,1);
         b2=randi(arc,1,1);
     end
```

```
      SOL(a1)=0;
      SOL(a2)=0;
      SOL(b1)=1;
      SOL(b1)=1;
 end

 %%obtenemos sumatorio wi%%
FO=0;
for k=1:arc
    FO=FO+(SOL(k)*P(k));
end

%%matriz "conex"%%
conex=zeros(n);
for k=1:arc
    if SOL(k)==1
        i=D(k);
        j=O(k);
        conex(i,j)=1;
        conex(j,i)=1;
    end
end

%%observamos columnas vacías, nos proporciona los nodos
inconexos%%
inconex=0;
for i=1:n
    if sum(conex(i,:))==0;
        inconex=inconex+1;
    end
end

%%función fitness%%
 w1=FO-(inconex*wub);

%%comprobamos si la fitness del nuevo es mejor o peor del
anterior%%
if w1>=w
    w=w1;
    solucion=FO;
    iteracion=it;
    S=SOL;
    conexionesreales=conex;
    vectorsolucion=SOL;
end
 end
```

ANEXO 2. Método Montecarlo.

```
clear
clc

max=0;
M=xlsread('DATOSPRUEBA');

M2=M;
n=16;
C=zeros(16);

%%elegimos primer nodo al azar e introducimos en V%%
O(1)=randi(n,1,1);


%%introducimos en matriz P%%
P(1,:)=M2(O(1),:);
for arc=2:16
 i=1;
 j=1;
 flag=0;

%%borramos elementos incompatibles%%
for origen=1:length(O)
    for destino=1:length(O)
        P(origen,O(destino))=0;
    end
end


[fil,col]=size(P);

%%elegimos siguiente(s) nodo(s) que cumplan que no vayan a los
que están en
%%V y que exista el arco (en la matriz sea distinto de 0)
while flag==0 && sum(sum(P))>0
    iprim=randi(fil,1,1);
    i=O(iprim);
    jprim=randi(n,1,1);
    if P(iprim,jprim)>0
        flag=1;
        j=jprim;
    end
end

if flag==1
M2(i,j)=0;
M2(j,i)=0;
P((fil+1),:)=M2(j,:);
P(iprim,j)=0;
P((fil+1),iprim)=0;
```

```
C(i,j)=1;
C(j,i)=1;

%%añadimos a la lista de O y D%%
O(fil+1)=j;
end
end

%%vemos valores de FO%%
FO=0;
for isuma=1:16
    for jsuma=1:16
        if jsuma>=isuma
           FO=FO+(M(isuma,jsuma)*C(isuma,jsuma));
        end
    end
end
```

ANEXO 3. Método Montecarlo aplicado a Red Simplificada de Sevilla

```
clear
clc

max=0;
M=xlsread('datosSevilla');

for it=1:4000000
M2=M;
n=65;
C=zeros(65);
O=[];
D=[];
P=[];

%%elegimos primer nodo al azar e introducimos en O y D%%
O(1)=randi(n,1,1);
D(1)=O(1);

%%introducimos en matriz P%%
P(1,:)=M2(O(1),:);

for arc=2:65
 i=1;
 j=1;
 flag=0;

%%borramos elementos incompatibles%%
for origen=1:length(O)
    for destino=1:length(D)
        P(origen,D(destino))=0;
    end
end

[fil,col]=size(P);

%%elegimos siguiente(s) nodo(s) que cumplan que no vayan a los
que están en
%%V y que exista el arco (en la matriz sea distinto de 0)%%
while flag==0 && sum(sum(P))>0

    iprim=randi(fil,1,1);
    i=O(iprim);

    jprim=randi(n,1,1);
    if P(iprim,jprim)>0
        flag=1;
        j=jprim;
    end

end
```

```
M2(i,j)=0;
M2(j,i)=0;
P((fil+1),:)=M2(j,:);
P(iprim,j)=0;
P((fil+1),iprim)=0;
C(i,j)=1;
C(j,i)=1;

%%añadimos a la lista de O y D%%

O(fil+1)=j;
D(arc)=j;
end

%%obtenemos Suma total de pesos%%
FO=0;
for isuma=1:65
    for jsuma=1:65
        if jsuma>=isuma
           FO=FO+(M(isuma,jsuma)*C(isuma,jsuma));
        end
    end
end

if FO>max
    max=FO
    iteracion=it
    Ofinal=O;
    Cfinal=C;
end
end
```

ANEXO 4. Tabla de arcos Red Simplificada de Sevilla

NODO	NODO	FLUJO
3	5	1802
3	1	2690
3	4	363
2	1	2322
6	4	1307
2	4	1494
7	6	1120
5	8	1802
7	8	1011
6	5	452
9	2	2292
7	11	1443
11	18	2391
8	12	2105
11	12	1167
18	9	1934
12	13	2526
13	16	1307
16	18	1246
17	16	1288
65	13	1522
17	15	1448
14	65	1127
15	14	1385
16	14	1413
22	21	2326

NODO	NODO	FLUJO
20	22	1386
13	24	1575
19	21	1173
18	19	1215
17	20	2161
19	20	1501
27	26	1397
24	23	1496
25	24	1496
23	22	1638
26	25	1525
62	10	1532
9	10	2317
24	28	1646
63	62	1768
61	63	1390
28	27	1049
63	60	2868
28	29	1711
59	62	1438
29	23	1760
60	59	1058
31	26	2194
10	58	2136
27	30	1243
29	30	1001
58	48	2127
31	30	1352

NODO	NODO	FLUJO
60	57	1110
37	21	1624
22	36	1635
57	55	1290
23	35	1391
55	60	2341
29	34	1514
59	58	1058
48	47	1719
30	33	1805
48	64	1351
31	32	2082
64	55	1249
36	39	1556
55	56	1657
36	37	1178
35	36	1178
47	46	1190
33	34	1093
34	35	1100
46	45	1520
32	33	1093
45	44	552
41	44	1717
38	37	2036
39	38	1223
41	34	1710
46	49	1388

NODO	NODO	FLUJO
54	49	1179
40	39	1811
55	56	2236
54	53	2758
40	35	1319
53	52	2406
40	41	1533
43	52	1344
43	32	1767
45	40	1631
40	44	1017
49	50	1035
47	38	1347
53	50	1268
51	50	1131
39	46	1203
43	42	1451
42	51	1131
33	42	1769
42	41	1826
50	45	1359
44	51	2500
51	55	2500
52	56	2500