

Comparación de métodos heurísticos, quasi-Newton e híbridos para la solución de un problema de sincronización óptima de semáforos

Lema Fernández, Carmen Socorro [carmen.lemaf@udc.es]

Dpto. Economía Aplicada II

Universidade da Coruña

Allende Alonso, Sira [sira@matcom.uh.cu]

Bouza Allende, Gemayqzel [gema@matcom.uh.cu]

Universidad de La Habana (La Habana-Cuba)

RESUMEN

Las ciudades del mundo muestran serios problemas de tráfico en sus calles. Hay diferentes métodos para afrontar los problemas de congestión del tráfico en zonas urbanas, por ejemplo, a través de una política correcta de control de los semáforos se mejora la circulación en las intersecciones y se reducen los embotellamientos.

En este trabajo proponemos tres métodos para la solución de un problema de sincronización óptima de los ocho semáforos que regulan un cruce con seis fases en cada ciclo. Estos métodos son: un método heurístico basado en la metaheurística de recocido simulado, un método tipo quasi-Newton (método del gradiente discreto) y un método híbrido que combina ambas estrategias.

Usando datos teóricos y la programación en MATLAB de los diferentes algoritmos, se realiza una comparación de los resultados obtenidos con cada una de las estrategias mencionadas, que nos lleva a concluir cuál es el método con el que se obtiene una mayor satisfacción de los usuarios de la vía pública, en cuanto a tiempos de espera y largo de las colas.

ABSTRACT

The cities in the world show serious problems of traffic in the roads. There are different ways to face the problems of traffic congestion, for example, by means of a correct traffic lights control policy, the circulation in the intersection is better and traffic jam is avoided.

In this paper, we propose three methods for the solution of an optimal synchronization problem of eight traffic lights that regulate an intersection with six phases in each cycle. These methods are: a heuristic based on simulated annealing metaheuristic, a quasi-Newton method (discrete gradient method) and a hybrid method that combines both strategies.

Using theoretical data and the programming in MATLAB of the different algorithms, it is performed a comparison of the results obtained with each of the strategies mentioned, allowing us to conclude which is the method that more satisfies to the road users, in terms of waiting times and queues length.

Palabras clave:

Control óptimo de semáforos; recocido simulado; método quasi-Newton; método híbrido.

Área temática: Optimización

1. INTRODUCCIÓN

Cuando la intensidad del tráfico en una intersección es más grande que la que se puede admitir con una regulación de preferencia de paso y no se puede recurrir a un cruce a diferente nivel, la regulación mediante semáforos permite hacer frente a la situación con un buen nivel de seguridad, con el inconveniente de producir alguna demora a los vehículos que acceden a la intersección, de ahí que sea de vital importancia fijar la combinación de luces y su duración.

Los semáforos manejan los conflictos que aparecen en los cruces, decidiendo que carril tiene prioridad y por cuanto tiempo. Sin embargo, la duración de las luces debería estar definida para garantizar la eficiencia del sistema. Esta eficiencia se puede medir de diferentes formas. Dos formas habituales son: que los usuarios no deberían esperar demasiado tiempo para cruzar y que no existan muchos usuarios que estén esperando para cruzar. Así se reduce el consumo de combustible, la emisión de contaminantes al aire (CO_2), el ruido, los accidentes, el estrés, se evitan los obstáculos en las intersecciones más cercanas causados por colas largas, se mejora el movimiento de mercancías..., en general se puede decir que se mejora la calidad de vida de los ciudadanos, su salud y el crecimiento económico y desarrollo de nuestras ciudades.

En este trabajo exponemos un problema de control de semáforos en un cruce de dos carreteras con los dos sentidos de circulación y con dos carriles en cada sentido (con 8 semáforos en el cruce) en un entorno urbano. Los resultados están organizados así: en la próxima sección presentamos un modelo matemático para dicho problema en el cual los tiempos de las luces son variables, pero sólo dependen de los flujos de llegada, análogo al formulado en [Pedreira, L. et al. (2012)] como variante del de [de Schutter, B. and de Moor, B. (1998)] pero para seis fases en cada ciclo. En la siguiente exponemos nuestras propuestas de solución: una heurística basada en la metaheurística de recocido simulado, un algoritmo quasi-Newton para problemas de optimización no suaves y un algoritmo híbrido que combina ambas estrategias. A continuación mostramos algunos resultados numéricos obtenidos mediante la programación con MATLAB de dichos algoritmos, comparando las soluciones obtenidas para cada una de

las funciones objetivo con cada una de las tres propuestas y para finalizar enumeramos algunas conclusiones y líneas futuras de investigación.

2. DESCRIPCIÓN DEL PROBLEMA Y SU MODELACIÓN

En esta sección presentamos un modelo matemático para el problema de control de tráfico en un cruce. Se tiene una intersección a la que confluyen dos calles, ambas doble vía (con dos carriles en cada sentido de circulación) en las que se puede circular de frente, girar a derecha e izquierda (ver figuras 1, 2, 3, 4, 5 y 6). En cada carril, antes de rebasar el cruce hay un semáforo, por tanto es un cruce con ocho semáforos. Los semáforos T_1 , T_3 , T_5 y T_7 permiten seguir de frente o girar a la derecha, y los semáforos T_2 , T_4 , T_6 y T_8 sólo permiten girar a la izquierda. Se quiere hallar los períodos en que deben permanecer en verde o rojo las luces de los semáforos T_i , $i=1, \dots, 8$ para evitar la congestión de las vías, descrita por el número de vehículos en espera (longitud de la cola) en cada vía. Para ello se consideran distintos criterios: la suma de las longitudes medias de las colas de cada vía, longitud de la cola más larga, tiempo medio de espera o una combinación de ellas. Con este planteamiento, hemos de modelar y resolver un problema que en cada ciclo tiene seis fases:

- En la **primera fase** (ver figura 1) los semáforos T_1 y T_2 están en verde (los demás semáforos están en rojo), por tanto los vehículos que circulan por el carril L_1 rebasan el cruce, continuando de frente, o giran a la derecha y los que circulan por L_2 rebasan el cruce girando a la izquierda.
- En la **segunda fase** (ver figura 2) el semáforo T_1 continúa en verde (T_2 ha cambiado de verde a rojo) y T_3 cambia de rojo a verde, por tanto continúan rebasando el cruce siguiendo de frente o continúan girando a la derecha los vehículos que circulan por el carril L_1 y rebasan el cruce continuando de frente o giran a la derecha los que circulan por L_3 .
- En la **tercera fase** (ver figura 3) el semáforo T_3 continúa en verde (T_1 ha cambiado de verde a rojo) y T_4 cambia de rojo a verde, por tanto continúan rebasando el cruce siguiendo de frente o continúan girando a la derecha los

vehículos que circulan por el carril L_3 y los que circulan por L_4 rebasan el cruce girando a la izquierda.

- En la **cuarta fase** (ver figura 4) los semáforos T_5 y T_6 cambian de rojo a verde (T_3 y T_4 cambian de verde a rojo), por tanto rebasan el cruce siguiendo de frente, o giran a la derecha los vehículos que circulan por el carril L_5 y los que circulan por L_6 rebasan el cruce girando a la izquierda.
- En la **quinta fase** (ver figura 5) el semáforo T_5 continúa en verde (T_6 ha cambiado de verde a rojo) y T_7 cambia de rojo a verde, por tanto continúan rebasando el cruce siguiendo de frente o continúan girando a la derecha, los vehículos que circulan por el carril L_5 y rebasan el cruce siguiendo de frente o giran a la derecha los que circulan por L_7 .
- En la **sexta fase** (ver figura 6) el semáforo T_7 continúa en verde (T_5 ha cambiado de verde a rojo) y T_8 cambia de rojo a verde, por tanto continúan rebasando el cruce siguiendo de frente o continúan girando a la derecha los vehículos que circulan por el carril L_7 y los que circulan por L_8 rebasan el cruce girando a la izquierda.

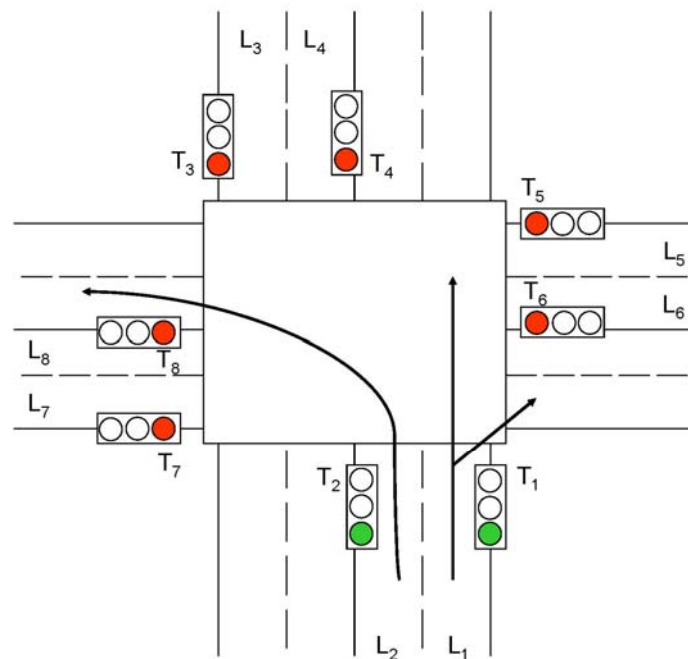


Figura 1

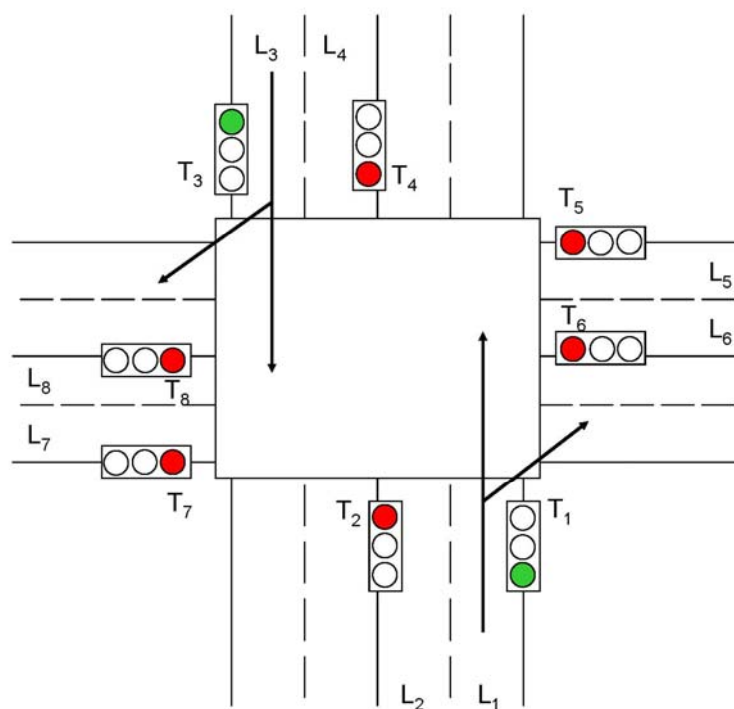


Figura 2

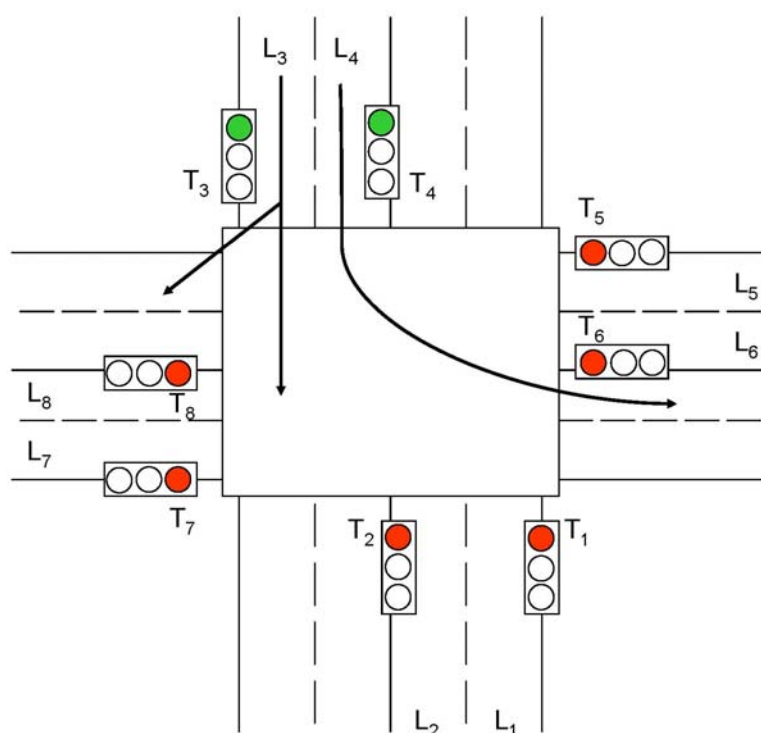


Figura 3

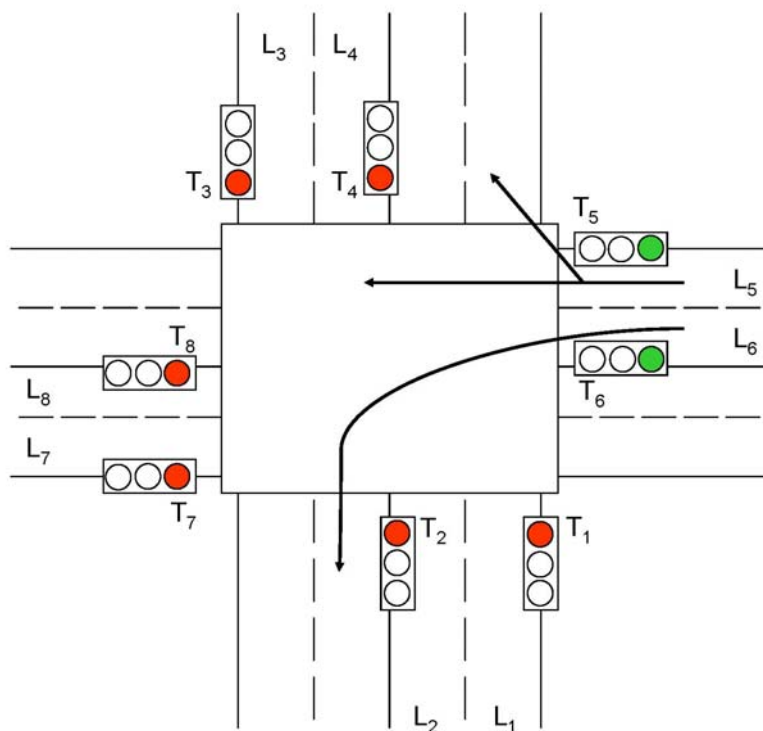


Figura 4

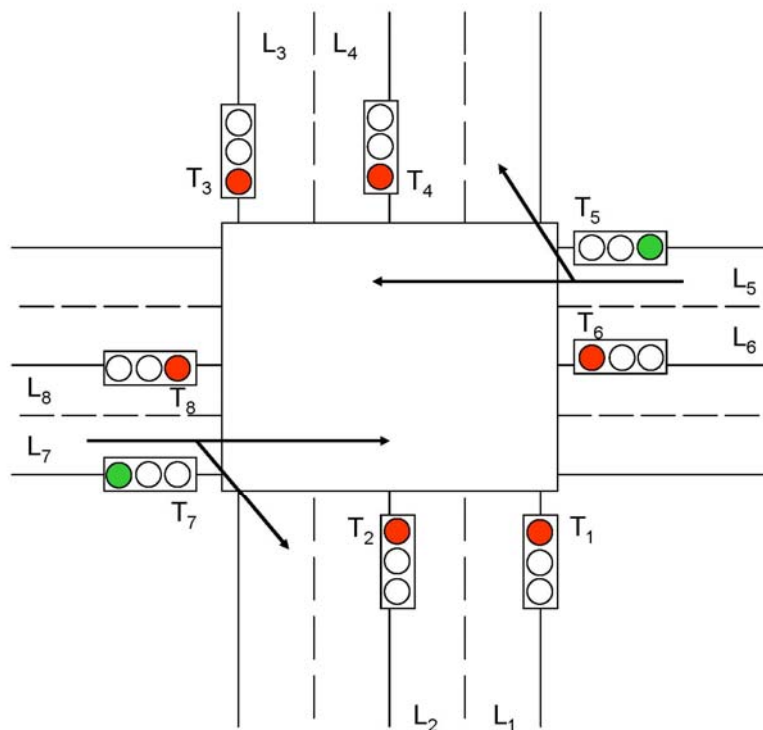


Figura 5

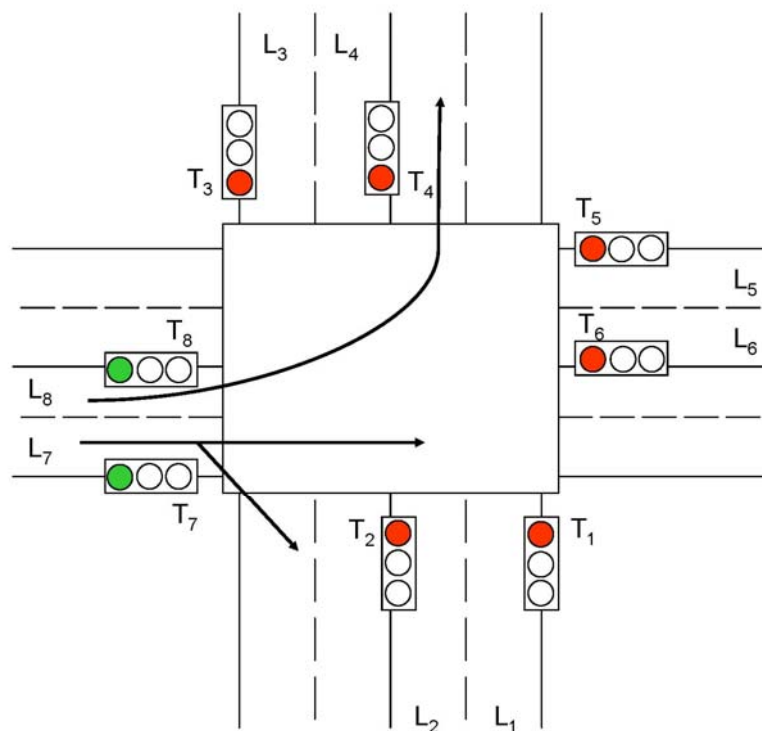


Figura 6

El ciclo comienza al principio de las luces verde de los semáforos T_1 y T_2 y acaba al final de sus luces rojas.

Consideramos las siguientes hipótesis:

- La luz ámbar permanece encendida durante un espacio de tiempo fijo e igual a δ_{amb} .
- N : número de veces que los semáforos tienen la luz verde, es un entero dado.
- Las longitudes de las colas son variables continuas.
- La tasa media de llegadas y salidas de vehículos es constante (o varía muy despacio en el tiempo).
- Para cada carril, la tasa media de salidas durante la luz verde es mayor o igual que la tasa media de salidas durante la luz ámbar.

Las hipótesis tercera y cuarta merecen algunas consideraciones:

- El diseño de sucesiones temporales switching óptimas para semáforos es sólo útil si las tasas de llegada y de salida de vehículos en el cruce son altas. En ese caso, la aproximación de las longitudes de cola mediante variables continuas sólo introduce pequeños errores. Además, en la práctica, interesa más obtener rápidamente una buena aproximación de la sucesión temporal switching óptima para semáforos, que gastar una gran cantidad de tiempo en obtener la sucesión exacta switching óptima para semáforos.
- Suponer que la tasa media de llegadas y salidas es constante no es una restricción seria porque, en la práctica, el esquema switching óptimo para semáforos para los próximos K ciclos, se obtiene basándonos en la predicción de las tasas medias de llegada y salida usando los datos medidos durante los ciclos anteriores.

Describimos el problema *con las variables de control* δ_k , tiempo de duración de la luz verde en el k -ésimo cambio de luz (incluyendo el ámbar), es decir, duración de la k -ésima fase y *las variables de estado* $x_k = (x_{k1}, x_{k2}, x_{k3}, x_{k4}, x_{k5}, x_{k6}, x_{k7}, x_{k8})^t$, cantidad promedio de vehículos en los carriles $L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8$ en el momento del k -ésimo cambio de luz, es decir, al finalizar la k -ésima fase, $k = 1, \dots, 6N$.

Para cada carril L_j se definen las tasas medias de llegada y salida de los vehículos bajo las luces verde y ámbar y se denotan:

λ_j : tasa media de llegada de vehículos en el carril L_j (dada en vehículos por segundo).

μ_j : tasa media de salida en el carril L_j cuando el semáforo está en verde (dada en vehículos por segundo).

κ_j : tasa media de salida en el carril L_j cuando el semáforo está en ámbar (dada en vehículos por segundo).

Definimos (basándonos en [de Schutter, B. and de Moor, B. (1998)] pero considerando ocho carriles y seis fases en cada ciclo) los vectores $b_i, i=1, \dots, 18$ que representan el número de vehículos en cada carril teniendo en cuenta los movimientos asociados a cada fase:

$$\begin{aligned}
 b_1 &= \begin{bmatrix} \lambda_1 - \mu_1 \\ \lambda_2 - \mu_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{bmatrix}, & b_2 &= \begin{bmatrix} \lambda_1 - \mu_1 \\ \lambda_2 \\ \lambda_3 - \mu_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{bmatrix}, & b_3 &= \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 - \mu_3 \\ \lambda_4 - \mu_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 \\ \lambda_8 \end{bmatrix}, & b_4 &= \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 - \mu_5 \\ \lambda_6 - \mu_6 \\ \lambda_7 \\ \lambda_8 \end{bmatrix}, & b_5 &= \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 - \mu_5 \\ \lambda_6 \\ \lambda_7 - \mu_7 \\ \lambda_8 \end{bmatrix}, \\
 b_6 &= \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \\ \lambda_7 - \mu_7 \\ \lambda_8 - \mu_8 \end{bmatrix}, & b_7 &= \begin{bmatrix} 0 \\ (\mu_2 - \kappa_2)\delta_{amb} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & b_8 &= \begin{bmatrix} (\mu_1 - \kappa_1)\delta_{amb} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & b_9 &= \begin{bmatrix} 0 \\ 0 \\ (\mu_3 - \kappa_3)\delta_{amb} \\ (\mu_4 - \kappa_4)\delta_{amb} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \\
 b_{10} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ (\mu_6 - \kappa_6)\delta_{amb} \\ 0 \\ 0 \end{bmatrix}, & b_{11} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ (\mu_5 - \kappa_5)\delta_{amb} \\ 0 \\ 0 \\ 0 \end{bmatrix}, & b_{12} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ (\mu_7 - \kappa_7)\delta_{amb} \\ (\mu_8 - \kappa_8)\delta_{amb} \end{bmatrix},
 \end{aligned}$$

$$\begin{aligned}
 b_{13} &= \begin{bmatrix} 0 \\ \max((\lambda_2 - \kappa_2)\delta_{amb}, 0) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, b_{14} = \begin{bmatrix} \max((\lambda_1 - \kappa_1)\delta_{amb}, 0) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, b_{15} = \begin{bmatrix} 0 \\ 0 \\ \max((\lambda_3 - \kappa_3)\delta_{amb}, 0) \\ \max((\lambda_4 - \kappa_4)\delta_{amb}, 0) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 b_{16} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \max((\lambda_6 - \kappa_6)\delta_{amb}, 0) \\ 0 \\ 0 \end{bmatrix}, b_{17} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \max((\lambda_5 - \kappa_5)\delta_{amb}, 0) \\ 0 \\ 0 \\ 0 \end{bmatrix}, b_{18} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \max((\lambda_7 - \kappa_7)\delta_{amb}, 0) \\ \max((\lambda_8 - \kappa_8)\delta_{amb}, 0) \end{bmatrix}
 \end{aligned}$$

Entonces:

$$\begin{aligned}
 x_{6k+1} &= \max(x_{6k} + b_1\delta_{6k+1} + b_7, b_{13}) \text{ para } k=0, 1, \dots, N-1, \\
 x_{6k+2} &= \max(x_{6k+1} + b_2\delta_{6k+2} + b_8, b_{14}) \text{ para } k=0, 1, \dots, N-1, \\
 x_{6k+3} &= \max(x_{6k+2} + b_3\delta_{6k+3} + b_9, b_{15}) \text{ para } k=0, 1, \dots, N-1, \\
 x_{6k+4} &= \max(x_{6k+3} + b_4\delta_{6k+4} + b_{10}, b_{16}) \text{ para } k=0, 1, \dots, N-1, \\
 x_{6k+5} &= \max(x_{6k+4} + b_5\delta_{6k+5} + b_{11}, b_{17}) \text{ para } k=0, 1, \dots, N-1, \\
 x_{6k+6} &= \max(x_{6k+5} + b_6\delta_{6k+6} + b_{12}, b_{18}) \text{ para } k=0, 1, \dots, N-1,
 \end{aligned}$$

ya que para calcular, por ejemplo, la cantidad promedio de vehículos en cada carril al final de la primera fase de cada ciclo (x_{6k+1}), hemos de usar los vectores b_1 , b_7 y b_{13} , pues en el carril L_1 durante esa fase, la luz del semáforo T_1 está en verde y por tanto sólo hay llegadas y salidas con luz verde, mientras que en el carril L_2 hay llegadas y salidas con luz verde y con luz ámbar y en los demás carriles sólo hay llegadas, pues los respectivos semáforos están en rojo.

Las características de la zona de tráfico pueden imponer otras regulaciones, tales como establecer cotas a las duraciones mínimas y máximas para los tiempos verde y rojo, así:

$\delta_{min.verde,i}$: cota mínima de duración de la luz en la fase i ; $i=1, \dots, 6$, en cada ciclo.

$\delta_{max.verde,i}$: cota máxima de duración de la luz en la fase i ; $i=1, \dots, 6$, en cada ciclo.

El problema de regulación de los semáforos puede representarse a través del siguiente modelo (basándonos en [Pedreira, L. et al (2012)]) pero considerando seis fases en cada ciclo):

$$\text{Minimizar } J \quad (1)$$

sujeto a:

$$\delta_{min.verde,1} \leq \delta_{6k+1} - \delta_{amb} \leq \delta_{max.verde,1} \quad \text{para } k=0,1,\dots,N-1, \quad (2)$$

$$\delta_{min.verde,2} \leq \delta_{6k+2} - \delta_{amb} \leq \delta_{max.verde,2} \quad \text{para } k=0,1,\dots,N-1, \quad (3)$$

$$\delta_{min.verde,3} \leq \delta_{6k+3} - \delta_{amb} \leq \delta_{max.verde,3} \quad \text{para } k=0,1,\dots,N-1, \quad (4)$$

$$\delta_{min.verde,4} \leq \delta_{6k+4} - \delta_{amb} \leq \delta_{max.verde,4} \quad \text{para } k=0,1,\dots,N-1, \quad (5)$$

$$\delta_{min.verde,5} \leq \delta_{6k+5} - \delta_{amb} \leq \delta_{max.verde,5} \quad \text{para } k=0,1,\dots,N-1, \quad (6)$$

$$\delta_{min.verde,6} \leq \delta_{6k+6} - \delta_{amb} \leq \delta_{max.verde,6} \quad \text{para } k=0,1,\dots,N-1, \quad (7)$$

$$x_{6k+1} = \max(x_{6k} + b_1 \delta_{6k+1} + b_7, b_{13}) \quad \text{para } k=0,1,\dots,N-1, \quad (8)$$

$$x_{6k+2} = \max(x_{6k+1} + b_2 \delta_{6k+2} + b_8, b_{14}) \quad \text{para } k=0,1,\dots,N-1, \quad (9)$$

$$x_{6k+3} = \max(x_{6k+2} + b_3 \delta_{6k+3} + b_9, b_{15}) \quad \text{para } k=0,1,\dots,N-1, \quad (10)$$

$$x_{6k+4} = \max(x_{6k+3} + b_4 \delta_{6k+4} + b_{10}, b_{16}) \quad \text{para } k=0,1,\dots,N-1, \quad (11)$$

$$x_{6k+5} = \max(x_{6k+4} + b_5 \delta_{6k+5} + b_{11}, b_{17}) \quad \text{para } k=0,1,\dots,N-1, \quad (12)$$

$$x_{6k+6} = \max(x_{6k+5} + b_6 \delta_{6k+6} + b_{12}, b_{18}) \quad \text{para } k=0,1,\dots,N-1. \quad (13)$$

Si los carriles difieren en importancia al evaluar el nivel de congestión de los mismos, se representa el nivel de importancia del carril, fijando diferentes pesos o ponderaciones w_j , $j=1,\dots,8$. La función J a minimizar puede ser:

- **Suma (ponderada) de las longitudes medias de las colas en todos los carriles**

$$J_1 = \sum_{j=1}^8 w_j \frac{\sum_{i=1}^{6N} x_{ij} \delta_i}{\sum_{i=1}^{6N} \delta_i} \quad (14)$$

- **Longitud (ponderada) media de las colas en el carril con mayores colas**

$$J_2 = \max_j w_j \frac{\sum_{i=1}^{6N} x_{ij} \delta_i}{\sum_{i=1}^{6N} \delta_i} \quad (15)$$

- **Longitud (ponderada) de la cola más larga**

$$J_3 = \max_{i,j} w_j x_{ij} \quad (16)$$

- **Suma (ponderada) de los tiempos medios de espera considerando las colas de todos los carriles**

$$J_4 = \sum_{j=1}^8 \frac{w_j}{\lambda_j} \frac{\sum_{i=1}^{6N} x_{ij} \delta_i}{\sum_{i=1}^{6N} \delta_i} \quad (17)$$

- **Tiempo (ponderado) medio de espera en el carril con mayor tiempo de espera**

$$J_5 = \max_j \frac{w_j}{\lambda_j} \frac{\sum_{i=1}^{6N} x_{ij} \delta_i}{\sum_{i=1}^{6N} \delta_i} \quad (18)$$

Se puede analizar una sexta opción que consiste en considerar una combinación de los criterios anteriores. Esto nos lleva a un modelo multi-objetivo que podemos resolver tomando una combinación positiva de las distintas funciones objetivo.

$$J_6 = \sum_{i=1}^5 \alpha_i J_i \quad (19)$$

donde los valores α_i corresponden a los pesos que se les dan a los distintos usuarios.

Como consecuencia de lo anteriormente expuesto, conocido δ (vector de los tiempos de cada luz), la cantidad de autos en cada carril queda determinado mediante

las ecuaciones (8) – (9) – (10) – (11) – (12) – (13). Denotamos por $x(\delta)$ la matriz de 6N filas y 8 columnas donde $x_{ij}(\delta)$ indica la cantidad de autos en el carril j en el momento del cambio de luz i.

Sean los parámetros:

$$\Delta_{ij} = \begin{cases} i \equiv 1(6) & j = 1,2 \\ i \equiv 2(6) & j = 1,3 \\ i \equiv 3(6) & j = 3,4 \\ i \equiv 4(6) & j = 5,6 \\ i \equiv 5(6) & j = 5,7 \\ i \equiv 0(6) & j = 7,8 \\ 0 & \text{si no} \end{cases} \quad \tilde{\Delta}_{ij} = \begin{cases} i \equiv 1(6) & j = 2 \\ i \equiv 2(6) & j = 1 \\ i \equiv 3(6) & j = 3,4 \\ i \equiv 4(6) & j = 6 \\ i \equiv 5(6) & j = 5 \\ i \equiv 0(6) & j = 7,8 \\ 0 & \text{si no} \end{cases}$$

(donde, por ejemplo, por $i \equiv 1(6)$ estamos indicando que i es congruente con 1 módulo 6, o sea, existe un entero n tal que $i = 1 + 6n$)

Una forma más compacta de escribir el problema, sin tener que diferenciar entre las seis fases es la siguiente:

$$\text{Minimizar } J \quad (20)$$

sujeto a:

$$\Delta_{k1}\delta_{\min.verde1} + \Delta_{k2}\delta_{\min.verde2} + \Delta_{k3}\delta_{\min.verde3} + \Delta_{k4}\delta_{\min.verde4} + \Delta_{k5}\delta_{\min.verde5} + \Delta_{k6}\delta_{\min.verde6} \leq \delta_k - \delta_{amb} \leq \Delta_{k1}\delta_{\max.verde1} + \Delta_{k2}\delta_{\max.verde2} + \Delta_{k3}\delta_{\max.verde3} + \Delta_{k4}\delta_{\max.verde4} + \Delta_{k5}\delta_{\max.verde5} + \Delta_{k6}\delta_{\max.verde6}$$

para $k=1, \dots, 6N$ (21)

$$x_{kj} = \max[x_{k-1j} + (\lambda_j - \Delta_{kj}\mu_j)\delta_k + \tilde{\Delta}_{kj}(\mu_j - \kappa_j)\delta_{amb}, \max(\tilde{\Delta}_{kj}(\lambda_j - \kappa_j)\delta_{amb}, 0)] \text{ para } k=1, \dots, 6N$$

$j=1, \dots, 8$ (22)

Como habíamos mencionado anteriormente, ya introducimos una aproximación extra al considerar las longitudes de cola continuas. Además, en la práctica, hay también alguna incertidumbre y variación en el tiempo de las tasas de llegada y de salida, lo que hace recomendable obtener una buena aproximación de la solución óptima que sea calculable en un tiempo corto y fácilmente ajustable a los cambios en los valores de los parámetros. De hecho [de Schutter, B. and de Moor, B. (1998)] se trata de un problema de optimización con restricciones de complementariedad lineal (problema NP-duro),

que involucra 54N variables, 12N restricciones suaves y 48N no suaves, lo que haría muy costoso la aplicación de un algoritmo iterativo [de Schutter, B. (2002)].

3. PROPUESTA DE SOLUCIÓN

Se proponen tres métodos diferentes de solución:

- Un método heurístico basado en la metaheurística de recocido simulado.
- Un método tipo quasi-Newton, el método del gradiente discreto.
- Un método híbrido que combina ambas estrategias.

3.1. Método heurístico basado en la metaheurística de recocido simulado

El método de recocido simulado es una heurística en que se resuelve un problema de optimización $\min F(x) \ x \in M$, simulando el proceso de enfriamiento en el que a medida que baja la temperatura el movimiento de los átomos es menos probable [S. Kirkpatrick, S. et al. (1983)]. Dado un candidato a solución Ω el algoritmo busca en una vecindad del mismo, un punto con mejor evaluación de la función objetivo. De no existir, se toma un punto con peor evaluación como nueva solución con una cierta probabilidad. Esta probabilidad que se calcula a partir de la ley de Boltzman, es proporcional a la temperatura del sistema e inversamente proporcional a la variación de los valores de la función objetivo al pasar al nuevo punto. Cada cierto número de iteraciones se disminuye la temperatura del sistema. Los distintos criterios de parada son: encontrar un punto con un valor adecuado de la función objetivo y alcanzar una temperatura suficientemente baja en el sistema.

El pseudo-código de este algoritmo para el problema de sincronización de semáforos es:

- 0) Fijar $t_0 > 0$ (temperatura inicial), $0 < \Delta t < 1$ (variación de la temperatura). Escoger J_i , $i=1, \dots, 6$, y construir F , función objetivo del problema. Q cantidad de pasos con igual temperatura. T menor temperatura a alcanzar por el sistema.
- 1) Construir un vector δ_0 y la solución $x(\delta_0)$ asociada. $F_{mejor} = F(x(\delta_0), \delta_0)$, $\delta_{mejor} = \delta_0$ temperatura $t = t_0$, $i = 0$.
- 2) Mientras $t > T$,
 - 2.1) $q = 1$

2.2) Mientras $q < Q$

2.2.1) Tomar δ un punto vecino de δ_i y hallar $x(\delta)$. Si $F(x(\delta), \delta) < F_{mejor}$,

$\delta_{i+1} = \delta$, $\delta_{mejor} = \delta$ e ir a **2.2.4)**

2.2.2) Generar r , número aleatorio de acuerdo a la ley uniforme en $(0,1)$.

2.2.3) Si $r < \exp\left(\frac{F_{mejor} - F(x(\delta), \delta)}{t}\right)$, $\delta_{i+1} = \delta$, si no, $\delta_{i+1} = \delta_i$ $\delta_{mejor} = \delta_{i+1}$ y

construir la solución $x(\delta_{i+1})$ asociada.

2.2.4) $q = q + 1$ e ir a **2.2)**

2.3) $t = t \cdot \Delta t$ e ir a **2).**

3) Mejor solución δ_{mejor} con valor de la función objetivo $F(x(\delta_{mejor}), \delta_{mejor})$. Fin

Se dice que δ es vecino de δ^* si difieren en solo una componente y $\|\delta - \delta^*\| = 1$, es decir existe i tal que $\delta_i = \delta_i^* \pm 1$ y $\delta_j = \delta_j^*$ para todo $j \neq i$. En aras de mantener factibilidad se tiene en cuenta que el nuevo punto cumpla las restricciones de acotación en (21). Claramente las vecindades así definidas, son un subconjunto discreto del conjunto de soluciones factibles, pero es una forma rápida de explorar soluciones factibles.

3.2. Método del gradiente discreto

Teniendo en cuenta las restricciones (22) podemos reducir el modelo implícitamente y calcular los valores de las funciones involucradas y sus derivadas de forma iterativa. Así resolvemos el problema con un algoritmo determinístico tipo quasi-Newton para funciones no suaves [Bagirov, A.M., Karasözen, B. y Sezer, M. (2008)]. La estrategia es:

1. Escoger un vector inicial δ_0 y la solución $x(\delta_0)$ asociada y $k=0$. Fijar sucesiones $\alpha_k, \beta_k, \gamma_k, \zeta_k \rightarrow 0^+, k \rightarrow \infty$ y números $c_1 > c_2 > 0$.
2. Fijar $s=0$, $\delta_{k,s} = \delta_k$.
3. Buscar una dirección de descenso d , estimando ∇F por el método de gradiente discreto [Bagirov, A.M., Karasözen, B. y Sezer, M. (2008)], con $(\alpha, \beta, \gamma, \zeta, c) = (\alpha_k, \beta_k, \gamma_k, \zeta_k, c_1)$.
4. Si $|d| \leq \alpha_k$, $\delta_{k+1} = \delta_{k,s}$, $k = k + 1$ ir a 2
5. Sea $\sigma_s = \operatorname{argmax}\{\sigma \geq 0, F(\delta_{k,s} + \sigma d/|d|) - F(\delta_{k,s}) \leq -c_2 \cdot \sigma/|d|\}$.

$$6. \delta_{k,s+1} = \delta_{k,s} + \sigma_s d/|d|.$$

7. $s=s+1$. Ir a 3.

La dirección de descenso se calcula de la siguiente manera

1. Fijar g , $|g|=1$ y fijar e , vértice del cubo unitario en $\mathbb{R}^n$.

2. Calcular el gradiente discreto $v_1 = \Gamma^i(\delta, g_1, e, \beta, \gamma, \zeta)$. $D = \{v_1\}$, $|g_i| > \sqrt{\frac{1}{2n}}$ $k=1$.

3. Hallar $w_k = \operatorname{argmin}\{|w| \mid w \in D\}$

4. Si $|w_k| < \alpha$ fin

5. Hallar $g_{k+1} = w_k/|w_k|$. Si $F(\delta + \gamma g_{k+1}) - F(\delta) \leq c\gamma|w_k|$ fin.

6. Calcular el gradiente discreto $v_{k+1} = \Gamma^i(\delta, g_{k+1}, e, \beta, \gamma, \zeta)$, $|g_i| > \sqrt{\frac{1}{2n}}$ $D = D \cup \{v_k\}$ e

ir a 3.

Es decir: se fija una dirección inicial, se calcula el gradiente discreto con respecto a dicha dirección inicial, se halla la distancia entre el cono convexo de todos los gradientes discretos calculados y el origen. Si la distancia es menor que la tolerancia, se acepta el punto como punto estacionario aproximado, en otro caso, se calcula una nueva dirección de búsqueda y se comprueba si esta dirección es una dirección de descenso. Si lo es se para, y ya tenemos calculada la dirección de descenso, en otro caso, se calcula otro gradiente discreto en esta nueva dirección y se actualiza el cono convexo.

El gradiente discreto $\Gamma^i(\delta, g_1, e, \beta, \gamma, \zeta)$ se calcula componente a componente mediante un esquema de diferencias divididas entre puntos consecutivos que difieren en $\zeta^j \beta$ unidades en la componente j , a lo que se agrega una combinación de las componentes $j \neq i$ para la i -ésima componente.

3.3. Método híbrido

Se comienza buscando una buena aproximación a la solución mediante el método de recocido simulado y luego se mejora el comportamiento en una vecindad del punto mediante el algoritmo quasi-Newton para funciones no suaves.

4. RESULTADOS COMPUTACIONALES

Los algoritmos y métodos de resolución propuestos han sido programados en MATLAB y se han implementado en un ordenador con procesador Intel Core i7, 950 que trabaja a 307 GHz. Todos ellos se corren partiendo de los datos que a continuación se especifican, obteniéndose los resultados para las distintas funciones objetivo. El programa permite variar los parámetros del algoritmo (temperatura inicial y final, número de repeticiones para una temperatura fija y estrategia de disminución de la temperatura), a fin de posibilitar el ajuste de sus valores en distintas situaciones de tráfico (por ejemplo, en cuanto a densidad de tráfico o número de ciclos).

Datos: 5 ciclos ($N=5$), tiempo de ámbar 3s ($\delta_{amb}=3s$)

Se usaron los parámetros que se indican en la siguiente tabla:

Carriles	1	2	3	4	5	6	7	8
Tasa llegada (λ)	0.1	0.15	0.2	0.15	0.2	0.1	0.15	0.2
Tasa salida verde (μ)	0.7	0.7	2	1.5	0.7	0.7	1	1.5
Tasa salida ámbar (κ)	0.25	0.25	0.45	0.2	0.25	0.25	0.45	0.2

En cuanto a las cotas del tiempo de las luces, se tomaron: $\delta_{min.verde,i} = 5s$, $\forall i=1, \dots, 6$; $\delta_{max.verde,i} = 30s$, $\forall i=1, \dots, 6$

Solución inicial: δ_0 aleatorio

Todos los carriles son iguales, es decir $w = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$

Todos los usuarios tienen el mismo peso, es decir $\alpha_i = 1 \ \forall i=1, \dots, 5$

Temperatura inicial $t_0 = 100000000$

$\Delta t = 0.001$

Número de iteraciones con igual temperatura $Q = 20$

Criterio de parada $T < 0.0001$

Los resultados obtenidos en dos corridas diferentes para las distintas funciones objetivo y los tres algoritmos propuestos aparecen en las tablas siguientes:

Función objetivo → Algoritmo y T.C. ↓	J ₁	J ₂	J ₃	J ₄	J ₅	J ₆
Heurístico	55.23 autos	14.3 autos	37.9 autos	340.1 s	128.6s	569.8
Tiempo computacional	0.0156 s	0.0156 s	0.0156 s	0.0156s	0.0156s	0.0156s
Quasi-Newton	45.03 autos	11.5 autos	25.94 autos	263.3 s	55.25s	494
Tiempo computacional	0.5781 s	0.4375 s	1.07 s	3.95 s	5.4 s	7.96 s
Híbrido	40.77 autos	10.1 autos	24.67 autos	196.11 s	44.92s	368.69
Tiempo computacional	0.5313 s	0.2813 s	0.6406 s	3.93 s	6.56 s	8.7 s

Función objetivo → Algoritmo y T.C. ↓	J ₁	J ₂	J ₃	J ₄	J ₅	J ₆
Heurístico	57.67 autos	15.3 autos	31.35 autos	310.3 s	108.2s	699.8
Tiempo computacional	0.0156 s	0.0156 s	0.0469 s	0.0313s	0.0156s	0.0156s
Quasi-Newton	53.47 autos	12.4 autos	24.64 autos	220.9 s	44.92s	548.16
Tiempo computacional	0.4844 s	0.4063 s	0.8906 s	3.31 s	5.34 s	8.75 s
Híbrido	47.72 autos	11.7 autos	24.59 autos	197.74 s	40.94s	488.18
Tiempo computacional	0.375 s	0.3125 s	1.4 s	3.5 s	6.75 s	8.84 s

A la vista de los resultados que se recogen en las tablas anteriores, podemos realizar las siguientes consideraciones:

- El algoritmo que mejor se comporta (en cuanto a la obtención de los valores mínimos de las diferentes funciones objetivo) es el algoritmo híbrido consistente en buscar una aproximación a la solución por medio de recocido simulado, y luego mejorarla a través del método del gradiente discreto.
- En cuanto a los métodos que se usan en solitario, va mejor el método quasi-Newton. Sin embargo el tiempo computacional es bastante superior que el empleado por la heurística, ya que en los primeros pasos este algoritmo es muy lento, pues le es complicado encontrar un entorno en el que pueda localizarse la solución óptima (problema que se soluciona con el método híbrido).
- En cuanto a los resultados poco adecuados obtenidos por la heurística, quizá sería necesario replantearse las condiciones que se usan para su construcción: política de enfriamiento, número de pasos con igual temperatura, criterio de

parada, etc., para que los resultados de ésta sean más ajustados a los del método híbrido, lo que además permitiría que éste pudiese mejorar al mejorar la heurística.

5. CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

- En este trabajo hemos estudiado un problema de control óptimo de semáforos para un cruce de dos calles con los dos sentidos de circulación, con dos carriles en cada sentido, y con un semáforo que regula la circulación en cada carril, por tanto un cruce regulado por ocho semáforos en el que en cada ciclo existen seis fases.
- Hemos presentado un modelo para dicho problema en el cual los tiempos de las luces son variables, pero sólo dependen de los flujos de llegada.
- Hemos propuesto tres estrategias de solución: una heurística basada en la metaheurística de recocido simulado, un método quasi-Newton para problemas de optimización no suaves (método del gradiente discreto) y un método híbrido que combina ambas estrategias.
- Se programaron en MATLAB las tres estrategias propuestas y, usando datos teóricos, se implementaron en un ordenador con procesador Intel Core i7. Se compararon los resultados obtenidos con los tres algoritmos, obteniendo que la mejor respuesta (en cuanto a largo de las colas y tiempo medio de espera) se alcanza con el híbrido, ya que mediante la heurística fuimos capaces de explorar la región de soluciones “escapando” de extremos locales, y mediante el algoritmo exacto se mejoró considerablemente la solución obtenida a través de la heurística.
- En cuanto a los resultados obtenidos con la heurística, se ha constatado que si se usa, por ejemplo, $\Delta t = 0,8$ y se hacen 30 iteraciones con la misma temperatura, aunque el tiempo computacional se incrementa apreciablemente aproximándose al que necesita el método quasi-Newton, los valores que se obtienen para las diferentes funciones objetivo mejoran bastante y se aproximan a los valores que se obtienen con dicho método, por lo que, como continuación a esta

investigación, trataremos de implementar los tres algoritmos propuestos usando estos nuevos criterios para la construcción de la heurística, probando con diferentes soluciones iniciales generadas y cambiando el criterio de parada, lo que, suponemos, nos permitirá obtener incluso mejores resultados para el método híbrido que los expuestos en este trabajo, aunque el tiempo computacional sea superior, lo cual en este contexto no es excesivamente relevante.

- En el futuro aplicaremos el algoritmo híbrido a cruces que cumplen las características expuestas en este trabajo, como el existente en la ciudad de A Coruña en el cruce de la avenida de Arteixo con la Ronda de Outeiro.
- También nos proponemos, en dichas intersecciones, la experimentación con datos reales de flujo de vehículos, incluyendo en nuestro estudio diferencias entre los distintos carriles y un mayor número de ciclos. El enfoque multiobjetivo por ponderación de los objetivos se podría incluir en este trabajo y dejar para el futuro otros enfoques por metas o usando heurísticas diseñadas para problemas multiobjetivo como Pareto Recocido Simulado.

6. REFERENCIAS BIBLIOGRÁFICAS

- BAGIROV, A.M., KARASÖZEN, B. y SEZER, M. (2008) “Discrete gradient method: Derivative-free method for non-smooth optimization”, *Journal of Optimization Theory and Applications*, 137, 2, pp. 317-334.
- DE SCHUTTER, B. y DE MOOR, B. (1998) “Optimal traffic light control for a single intersection”. *European Journal of Control*, 4, 3, pp. 260-276.
- DE SCHUTTER, B. (2002) “Optimizing acyclic traffic signal switching sequences through an extended linear complementarity problem formulation”, *European Journal of Operational Research*, 139, 2, pp. 400-415.
- KIRKPATRICK, S.; GELATT, J.R. y VECCHI, M.P. (1983) “Optimization by simulated annealing”. *Science*, 220, pp. 671-680.

- PEDREIRA, L. et al. (2012) “Regulación óptima de las fases de los semáforos en un cruce e impacto medioambiental”. Anales de ASEPUMA nº 20: 103, pp. 103.1-103.22.