

Recibido <i>1/11/2009</i> Revisado <i>2/12/2009</i> Aceptado <i>21/12/2009</i>	Un método híbrido para resolver el problema de control óptimo de semáforos Lema Fernández, Carmen S. [colito@udc.es] Pedreira Andrade, Luis P. [lucky@udc.es] <i>Dpto. Economía Aplicada II. Universidade da Coruña</i> Bouza Allende, Gema [gema@matcom.uh.cu] Allende Alonso, Sira [sira@matcom.uh.cu] <i>Universidad de La Habana (La Habana-Cuba)</i>
----------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

RESUMEN

El objetivo de este trabajo es resolver un problema de control óptimo que consiste en asignar tiempos a luces de semáforos que regulan cruces de dos calles con los dos sentidos de circulación. El criterio de comparación a considerar se expresa en una función objetivo a minimizar que puede ser: longitud de las colas, cantidad de vehículos en el peor de los casos, tiempo de espera, una combinación de las anteriores, entre otras posibilidades. De esta forma se obtiene un modelo tipo problema de optimización con restricciones de equilibrio, en particular con restricciones de complementariedad lineal. Para su solución, proponemos un método híbrido que combina una heurística, tipo recocido simulado, para calcular una buena aproximación a la solución, que luego será mejorada con un algoritmo quasi-Newton para problemas de optimización no suaves. Además, presentaremos algunos ejemplos numéricos..

Palabras claves: Control óptimo de semáforos; quasi-Newton; recocido simulado

ABSTRACT

Let us consider an intersection of two two-way streets. We want to solve the problem of finding the switch times of the corresponding traffic lights minimizing a certain criterion, such as the length of the queues, the number of vehicles at the worst queue, waiting time, a combination, etc. The resulting model is an optimization problem with equilibrium constraints, namely a linear complementarity constraints problem. In this work we propose a hybrid solution strategy where an approximation to the solution is computed by means of a simulated annealing

algorithm and then it is improved by a quasi-Newton method for non-smooth optimization problems. We will also present some numerical examples.

Keywords:. optimal traffic light control; quasi-Newton; simulated annealing

1. INTRODUCCIÓN

Como el número de vehículos y las necesidades de transporte crecen, las ciudades del mundo muestran serios problemas de congestión de tráfico en sus carreteras. Los costes incluyen pérdida de tiempo (ya sea tiempo libre o de trabajo), aumento del consumo de combustible, contaminación, problemas de salud, stress y malestar, en general. Además, la congestión retarda el movimiento de mercancías y de servicios, de modo que, aumenta el precio de los productos y reduce la competitividad de los negocios. A partir de la década de los 80, los responsables de los departamentos de tráfico intuyeron que las soluciones tradicionales que se estaban aplicando no serían capaces de resolver esta problemática. Estas soluciones típicamente implicaban la construcción de nuevas infraestructuras o la ampliación de las ya existentes, pero en muchos casos no eran soluciones viables debido principalmente a los elevados costes, a la carencia de espacio disponible y al inevitable impacto medioambiental. Por tanto, los departamentos de tráfico de todo el mundo están muy interesados en la optimización de las infraestructuras existentes, con el objeto de mejorar al máximo sus prestaciones.

Uno de los problemas más relevantes en la gestión del tráfico es la optimización de los ciclos de los semáforos, es decir, la optimización de la secuencia de estados (rojo, verde, etc.) por la que pasan los semáforos y que se repite de manera cíclica, con el objeto de proporcionar una circulación suave y evitar largas colas en unas calles y circulación escasa en otras. Cada ciclo está asociado a una intersección, o sea, puede regir a varios semáforos. Cada fase del ciclo (o stage) se compone de los estados en los que están todos los semáforos de esa intersección durante un tiempo determinado.

Un modelo ELCP (de complementariedad lineal extendido) para el control de semáforos se considera en [de Schutter, B. and de Moor, B. (1998)]. Se comprueba que es NP-hard, de ahí que se resuelva un problema aproximado basado en la relajación del original. Desde el punto de vista algorítmico se han planteado soluciones mediante el uso de: redes neuronales para implementar la función de control de los semáforos [Spall, J.C. and Chin, D.C. (1994)], sistemas de control difuso para alargar o acortar el ciclo de los semáforos utilizando semáforos con sensores [You-Sik, H. et al. (1999)], redes de Petri para representar de manera modular las redes de tráfico urbano [di Febbraro, A. et al. (2002)] y algoritmos genéticos [Rouphail, N. et al. (2000)], Varia, H. R. and Dhingra, S. L. (2004), Sánchez, J. J. et al. (2008)].

Los resultados están organizados así. En la próxima sección presentaremos un modelo matemático basado en el propuesto por Schutter-Moor (1998). En la tercera sección se muestra el problema de control óptimo y en la siguiente nuestra propuesta de solución a través de un método híbrido. Para finalizar exponemos algunos resultados numéricos obtenidos mediante la programación con MATLAB del algoritmo solución, así como las conclusiones y líneas futuras de investigación.

2. MODELO

En esta sección presentaremos el modelo matemático del problema de control de tráfico en un cruce simple. Se tiene una intersección a la que confluyen dos calles, ambas doble vía y en las cuales no es posible doblar hacia la izquierda. En cada esquina del cruce hay un semáforo. Se quiere hallar los períodos en que deben permanecer en verde los semáforos T_1 y T_3 , (rojo T_2 y T_4) y en cuales T_1 y T_3 muestran la luz roja (verde T_2 y T_4), de forma que no haya mucha congestión, ver Figura 1. Para ello se consideran distintos criterios: largo de la cola, tiempo de espera y largo de la peor cola o una combinación de ellas.

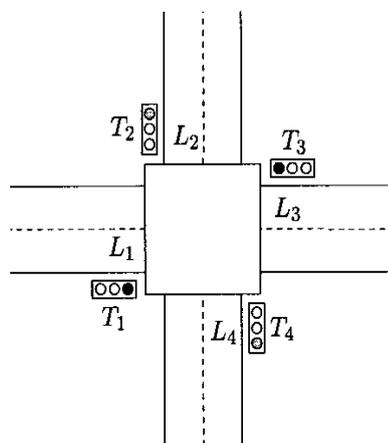


Figura 1: Cruce Simple

Usaremos los siguientes parámetros:

N : cantidad de veces que los semáforos tienen la luz verde.

δ_{amb} : tiempo de duración de la luz ámbar (constante que fluctúa entre los 3 y los 4 segundos).

Como simplificaciones del modelo trabajaremos con el comportamiento promedio por lo que asumimos que la cantidad de autos es continua. Sean:

λ_i : tasa media de llegada de vehículos en el carril L_i (dada en vehículos por segundo).

μ_i : tasa media de salida en el carril L_i cuando el semáforo está en verde.

κ_i : tasa media de salida en el carril L_i cuando el semáforo está en ámbar.

$$b_1 = \begin{bmatrix} \lambda_1 - \mu_1 \\ \lambda_2 \\ \lambda_3 - \mu_3 \\ \lambda_4 \end{bmatrix}, b_2 = \begin{bmatrix} \lambda_1 \\ \lambda_2 - \mu_2 \\ \lambda_3 \\ \lambda_4 - \mu_4 \end{bmatrix}, b_3 = \begin{bmatrix} (\mu_1 - \kappa_1)\delta_{amb} \\ 0 \\ (\mu_3 - \kappa_3)\delta_{amb} \\ 0 \end{bmatrix},$$

$$b_4 = \begin{bmatrix} 0 \\ (\mu_2 - \kappa_2)\delta_{amb} \\ 0 \\ (\mu_4 - \kappa_4)\delta_{amb} \end{bmatrix}, b_5 = \begin{bmatrix} \max((\lambda_1 - \kappa_1)\delta_{amb}, 0) \\ 0 \\ \max((\lambda_3 - \kappa_3)\delta_{amb}, 0) \\ 0 \end{bmatrix}, b_6 = \begin{bmatrix} 0 \\ \max((\lambda_2 - \kappa_2)\delta_{amb}, 0) \\ 0 \\ \max((\lambda_4 - \kappa_4)\delta_{amb}, 0) \end{bmatrix}$$

Definimos las siguientes variables:

δ_k : tiempo de duración de la luz verde en el k -ésimo cambio de luz (incluyendo el ámbar).

$x_k = (x_{k,1}, x_{k,2}, x_{k,3}, x_{k,4})^t$ cantidad promedio de autos en los carriles L_1, L_2, L_3, L_4 en el momento del k -ésimo cambio de luz. Entonces:

$$x_{2k+1} = \max(x_{2k} + b_1\delta_{2k+1} + b_3, b_5) \quad \text{para } k = 0, 1, \dots, N-1,$$

$$x_{2k+2} = \max(x_{2k+1} + b_2\delta_{2k+2} + b_4, b_6) \quad \text{para } k = 0, 1, \dots, N-1,$$

3. PROBLEMA DE CONTROL ÓPTIMO

Podemos imponer algunas condiciones extras, tales como las duraciones mínimas y máximas para los tiempos verde y rojo, así:

$\delta_{min.verde,i}$: cota mínima de duración de la luz verde en semáforo $i=1,2$.

$\delta_{max.verde,i}$: cota máxima de duración de la luz verde en semáforo $i=1,2$.

El problema a resolver es el siguiente:

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

sujeto a:

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

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

$$x_{2k+1} = \max(x_{2k} + b_1 \delta_{2k+1} + b_3, b_5) \quad \text{para } k=0,1,\dots,N-1, \quad (4)$$

$$x_{2k+2} = \max(x_{2k+1} + b_2 \delta_{2k+2} + b_4, b_6) \quad \text{para } k=0,1,\dots,N-1, \quad (5)$$

donde J es el criterio a optimizar, el cual puede ser:

- Longitud de la cola sobre todas las colas

$$J_1 = \sum_{j=1}^4 w_j \sum_{i=1}^{2N} x_{ij} \delta_i \quad (6)$$

- Longitud de la cola sobre la peor cola

$$J_2 = \max_j w_j \sum_{i=1}^{2N} x_{ij} \delta_i \quad (7)$$

- Longitud de cola en el peor caso

$$J_3 = \max_{i,j} w_j x_{ij} \quad (8)$$

- Tiempo de espera sobre todas las colas

$$J_4 = \sum_{j=1}^4 \frac{w_j}{\lambda_j} \sum_{i=1}^{2N} x_{ij} \delta_i \quad (9)$$

- Tiempo de espera sobre la peor cola

$$J_5 = \max_j \frac{w_j}{\lambda_j} \sum_{i=1}^{2N} x_{ij} \delta_i \quad (10)$$

donde (w_1, \dots, w_4) representan la importancia que se le otorga a cada uno de los cuatro carriles.

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.

$$J_6 = \sum_{i=1}^5 \alpha_i J_i \quad (11)$$

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

Sea el parámetro:

$$\Delta_{ij} = \begin{cases} 1 & \text{si } i \equiv j(2) \\ 0 & \text{si no} \end{cases}$$

Una forma más compacta de escribir el problema, sin tener que diferenciar entre índices pares (correspondientes al cambio de luz rojo-verde del semáforo 1-3) y los impares (donde cambia la luz del semáforo 2-4) es la siguiente:

$$\text{Minimizar } J \quad (12)$$

sujeto a:

$$\Delta_{k1} \delta_{\min.\text{verde},1} + \Delta_{k2} \delta_{\min.\text{verde},2} \leq \delta_k - \delta_{\text{amb}} \leq \Delta_{k1} \delta_{\max.\text{verde},1} + \Delta_{k2} \delta_{\max.\text{verde},2} \quad \text{para } k=1, \dots, 2N \quad (13)$$

$$x_{k,j} = \max[x_{k-1,j} + (\lambda_j - \Delta_{kj} \mu_j) \delta_k + \Delta_{kj} (\mu_j - \kappa_j) \delta_{\text{amb}}, \max(\Delta_{kj} (\lambda_j - \kappa_j) \delta_{\text{amb}}, 0)] \quad \text{para } k=1, \dots, 2N; j=1, 2, 3, 4 \quad (14)$$

Este modelo es un problema de optimización con restricciones de complementariedad y de grandes dimensiones. Se puede asumir que las variables δ son enteras, aunque no es una hipótesis que se tome en cuenta en [de Schutter, B. and de Moor, B. (1998)], pues hace más complejo el problema. En cuanto a su solución, el tamaño del problema en un caso real (10N variables, 4N restricciones suaves y 8N no suaves) hace que sea muy costoso aplicar un algoritmo iterativo [de Schutter, B. (2002)]. Es por eso que para su solución proponemos un método híbrido que combina una heurística, tipo recocido simulado, con un algoritmo quasi-Newton para problemas de optimización no suaves.

4. PROPUESTA DE SOLUCIÓN

El método de recocido simulado es una heurística en que se resuelve un problema de optimización $\min J(x) \quad x \in M$, 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.

Su pseudo-código se presenta a continuación.

- 0) Fijar $\varepsilon > 0$, valor pequeño, $t^0 > \Delta t > 0$ (temperatura inicial y variación de temperatura respectivamente). Escoger J_i , $i=1, \dots, 6$, y construir J , 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. $J_{mejor} = J(x(\delta_0), \delta_0)$, temperatura $t = t^0$, $i=0$.
- 2) Mientras $t > T$, $q=1$.
- 3) Mientras $q < Q$, tomar δ un punto vecino de δ_i , y hallar $x(\delta)$. Si $J(x(\delta), \delta) < J_{mejor}$, $\delta_{i+1} = \delta$ e ir a 6.
- 4) Generar r , número aleatorio de acuerdo a la ley uniforme en $(0,1)$.
- 5) Si $r < \exp\left(\frac{J_{mejor} - J(x(\delta), \delta)}{t}\right)$, $\delta_{i+1} = \delta$, si no, $\delta_{i+1} = \delta_i$ y construir la solución $x(\delta_{i+1})$ asociada.
- 6) $q = q + 1$, ir a 3.
- 7) $t = t - \Delta t$ e ir a 2.

Se dice que δ es vecino de δ^* si difieren en solo una componente y $\|\delta - \delta^*\| = 1$. En aras de mantener factibilidad se tiene en cuenta que el nuevo punto cumpla las restricciones en (13). Claramente las vecindades así definidas, son un subconjunto discreto del conjunto de soluciones factibles, pero es una forma rápida de explorar soluciones factibles.

Teniendo en cuenta las restricciones (14) 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 [Beliakov, G. and Bagirov, A. (2006)]. 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 ∇J por el método de gradiente discreto [Beliakov, G. and Bagirov, A. (2006)] 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 $\sigma_s = \operatorname{argmax}\{\sigma \geq 0, J(\delta_{k,s} + \sigma d/|d|) - J(\delta_{k,s}) \leq -c_2 \cdot \sigma |d|\}$.
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 , $|g_i| = 1$ 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 $w_k = \operatorname{argmin}\{|w| / w \in D\}$.
4. Si $|w_k| < \alpha$ fin.
5. Hallar $g_{k+1} = w_k/|w_k|$. Si $J(\delta + \gamma g_{k+1}) - J(\delta) \leq c\gamma|w_k|$ 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.

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.

Este método arrojó en la mayoría de los casos mejores resultados que la heurística, pero a un costo computacional superior. Usando las ventajas de ambos algoritmos, proponemos buscar una buena aproximación a la solución mediante el método de recocido simulado. Luego se mejora el comportamiento en una vecindad del punto mediante el algoritmo tipo quasi-Newton para funciones no suaves ya expuesto.

Presentamos ahora un ejemplo para ilustrar el comportamiento del algoritmo, así como los resultados obtenidos mediante la programación en MATLAB del método propuesto.

5. EJEMPLO

Datos: 5 ciclos, tiempo de ámbar 3 sg.

Tabla 1: Parámetros

	Carril 1	Carril 2	Carril 3	Carril 4
Tasa-llegada	0.3	0.3	0.2	0.25
Tasa-sal. ver.	0.7	0.7	1	0.5
Tasa-sal.ámba	0.25	0.25	0.5	0.5

En cuanto a las cotas del tiempo de las luces, se tomaron:

$$\delta_{min,verde,1} = \delta_{min,verde,2} = 5 \text{ sg. y } \delta_{max,verde,1} = \delta_{max,verde,2} = 30 \text{ sg. Pesos } w=[1 \ 1 \ 1 \ 1]$$

Para la función objetivo J_1 , obtuvimos los siguientes resultados numéricos:

Tabla 2: Duración de la luz verde en cada ciclo

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5
Semáforos 1-3	5	6,73	8,42	8,62	10,29
Semáforos 2-4	6,73	6,3	9,07	9,01	6,36

Tabla 3: Promedio de autos en el momento del cambio de luz

	Carril 1	Carril 2	Carril 3	Carril 4
Ciclo1-1 ^{er} cambio	0,15	1,5	0	1,25
Ciclo1-2 ^o cambio	2,17	0,16	1,35	0
Ciclo2-1 ^{er} cambio	0,83	2,18	0	1,68
Ciclo2-2 ^o cambio	2,72	1	1,26	0,11
Ciclo3-1 ^{er} cambio	0,7	3,53	0	2,21
Ciclo3-2 ^o cambio	3,42	1,26	1,81	0
Ciclo4-1 ^{er} cambio	1,32	3,84	0	2,15
Ciclo4-2 ^o cambio	4,03	1,58	1,8	0
Ciclo5-1 ^{er} cambio	1,26	4,67	0	2,57
Ciclo5-2 ^o cambio	3,17	3,48	1,27	0,98

En este ejemplo hemos obtenido buenos resultados ya que las colas son cortas (la peor cola tiene 4,67 autos). Además observamos que se evitó la congestión con períodos largos sin cambio de luz cuando la cantidad de tráfico es grande. El valor de la función objetivo es 489,94. Si hubiésemos utilizado sólo la heurística, el valor de la función objetivo sería 1920. Se observa la mejoría del resultado utilizando el método híbrido, aunque el coste computacional es mayor, ya que sólo la heurística usaría 1,61 sg., sin embargo el híbrido necesita 7,69 sg.

Probamos con los otros criterios y obtuvimos resultados similares.

Para el segundo criterio la solución fue:

Tabla 4: Duración de la luz verde en cada ciclo

	Ciclo1	Ciclo2	Ciclo3	Ciclo4	Ciclo5
Semáforos 1-3	5	7,91	8,64	9,97	5,62
Semáforos 2-4	6,27	7,21	8,77	5,58	5,06

$J_2 = 207,67$, mayor cola 5,42 autos.

Para el tercer criterio obtuvimos:

Tabla 5: Duración de la luz verde en cada ciclo

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5
Semáforos 1-3	5,03	6,92	8,35	13,69	8,64
Semáforos 2-4	5	8,91	13,03	8,41	5

Y la peor cola tuvo 4,83 autos. Si hubiésemos usado sólo la heurística, la peor cola tendría 7,15 autos. Por tanto, se observa la mejora obtenida usando el método híbrido, frente al uso único de la heurística. En cuanto al tiempo computacional, el ordenador emplea 2,59 sg. en la heurística y 9,69 sg. en el método completo.

Usando el cuarto criterio los períodos son:

Tabla 6: Duración de la luz verde en cada ciclo

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5
Semáforos 1-3	8,15	8,88	12,24	8,64	8,43
Semáforos 2-4	7,74	10,47	7,49	6,8	5

El valor de la función objetivo es 2274, de ahí que el resultado no fuera muy bueno pues el tiempo de espera total fue alto, aunque es mucho mejor que si usásemos sólo la heurística, ya que en ese caso el valor de J_4 es 6877. La mayor cola tuvo 6,3 autos.

Los resultados con el quinto criterio son:

Tabla 7: Duración de la luz verde en cada ciclo

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5
Semáforos 1-3	6,05	7,23	6,52	7,06	8,94
Semáforos 2-4	6,21	5,95	5,77	17,22	5

Los resultados fueron bastante buenos, la peor cola tuvo 6,42 autos y el valor de la función objetivo fue 793,28.

Culminamos con el problema multi-objetivo, modelado mediante la minimización de la función J_6 ver (11). Utilizamos el vector de pesos $\alpha = (1 \ 1 \ 1 \ 1 \ 1)$ para no hacer diferencias entre los distintos criterios. La solución fue:

Tabla 8: Duración de la luz verde en cada ciclo

	Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4	Ciclo 5
Semáforos 1-3	5,09	14,96	8,99	7,91	6,85
Semáforos 2-4	6,62	6,04	5,48	5,36	5

Tabla 9: Promedio de autos en el momento del cambio de luz

	Carril 1	Carril 2	Carril 3	Carril 4
Ciclo1-1 ^{er} cambio	0,15	1,52	0	1,27
Ciclo1-2 ^o cambio	2,14	0,23	1,32	0
Ciclo2-1 ^{er} cambio	0,15	4,72	0	3,74
Ciclo2-2 ^o cambio	1,96	3,65	1,21	2,23
Ciclo3-1 ^{er} cambio	0,15	6,35	0	4,48
Ciclo3-2 ^o cambio	1,79	5,5	1,10	3,11
Ciclo4-1 ^{er} cambio	0,15	7,88	0	5,08
Ciclo4-2 ^o cambio	1,76	7,09	1,07	3,75
Ciclo5-1 ^{er} cambio	0,37	9,14	0	5,46
Ciclo5-2 ^o cambio	1,87	8,49	1	4,21

En este caso los resultados fueron bastante malos, ya que tuvimos colas con 8 y 9 vehículos y un menor número de ceros que con el criterio uno. A pesar de ello el valor de la función objetivo fue 5155, mucho mejor que el valor que se obtiene con la

aplicación de la heurística que es 8207. En cuanto al tiempo computacional, fue mayor que en los otros criterios, 3,42 sg. en la heurística y 9,9 sg. en el método completo.

6. CONCLUSIONES Y LÍNEAS FUTURAS DE INVESTIGACIÓN

- Siguiendo el modelo propuesto por Schutter-Moor, hemos estudiado el problema de control óptimo de semáforos para un cruce de dos carreteras con los dos sentidos de circulación sin posibilidad de girar a la izquierda. Su programación en MATLAB se realizó sin necesidad de distinguir entre índices pares e impares.
- Hemos propuesto un método híbrido de solución donde se combina la filosofía de recocido simulado y un método quasi-Newton adecuado. Obtuvimos resultados bastante buenos (las colas son cortas), con un tiempo computacional no demasiado grande, ya que mediante la heurística fuimos capaces de explorar la región de soluciones “escapando” de extremos locales, y mediante el algoritmo determinístico se mejoró considerablemente la solución obtenida a través de la heurística.
- En el futuro realizaremos más experimentos numéricos en los que se incluyan diferencias entre los distintos carriles, un mayor número de ciclos y en el caso del problema multiobjetivo, vectores de peso que ponderen más un objetivo que otro.
- Modelaremos también otros tipos de intersecciones en los que se permita girar a la izquierda, y usando datos obtenidos a través del departamento de tráfico de la ciudad de A Coruña, derivar actuaciones que permitan una mejora del tráfico rodado en ciertos cruces con embotellamientos frecuentes en dicha ciudad.

7. REFERENCIAS BIBLIOGRÁFICAS

- BELIAKOV, G. and BAGIROV, A. (2006) “Non-smooth optimization methods for computation of the conditional value-at-risk and portfolio optimization”, *Optimization*, 55, 5-6, pp. 459-479.
- DE SCHUTTER, B. and 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.

- DI FEBBRARO, A. et al. (2002). “On applying Petri nets to determine optimal offsets for coordinated traffic light timings”. The IEEE 5th International Conference on Intelligent Transportation Systems, pp. 773-778.
- KIRKPATRICK, S.; GELATT, J.R. and VECCHI, M.P. (1983) “Optimization by simulated annealing”. *Science*, 220, pp. 671-680.
- ROUPHAIL, N. et al. (2000). “Direct Signal Timing Optimization: Strategy Development and Results”. In XI Pan American Conference in Traffic and Transportation Engineering.
- SÁNCHEZ, J. J.; GALÁN, M. J. and RUBIO, E. (2008). “Applying a Traffic Lights Evolutionary Optimization Technique to a Real Case: Las Ramblas, Area in Santa Cruz de Tenerife” *IEEE Transactions on Evolutionary Computation*, 12, 1, pp. 25-40.
- SPALL, J. C. and CHIN, D. C. (1994). “A model-free approach to optimal signal light timing for system-wide traffic control”. 33rd IEEE Conference on Decision and control, 1994, pp. 1868-1875.
- VARIA, H. R. and DHINGRA, S. L. (2004). “Dynamic Optimal Traffic Assignment and Signal Time Optimization using Genetic Algorithms”. *Computer-Aided Civil and Infrastructure Engineering*, 19, pp. 260-273.
- YOU-SIK, H. et al. (1999). “New Electrosensitive Traffic Light using Fuzzy Neural Network”. *IEEE Transactions on Fuzzy Systems*, VII, 6, pp. 759- 767.