

Planificación de Ciclos en Semáforos con PSO: Casos de Estudio sobre Málaga y Sevilla

José M. García-Nieto, Enrique Alba, y Ana C. Olivera

Resumen— Actualmente, el constante crecimiento de las áreas metropolitanas en nuestras ciudades conlleva un incremento de semáforos que controlan el flujo del tráfico vehicular. La existencia de cientos e incluso miles de semáforos, cuya programación debe ser conjuntamente sincronizada, supone una labor altamente compleja para los ingenieros civiles en los centros de control de tráfico. Dicha programación, efectuada sobre los ciclos y periodos de colores en los semáforos, debe ser además lo más eficiente posible para evitar la congestión del tráfico rodado, con todos los problemas de: seguridad, ahorro energético, ahorro en tiempo de trabajo, etc., que lleva asociados. Por tanto, el empleo de técnicas automáticas inteligentes para la programación optimizada de semáforos se convierte en una tarea indispensable hoy en día. En este trabajo, proponemos el uso de un algoritmo de optimización PSO (Particle Swarm Optimization), el cual, utilizando el simulador de tráfico SUMO para su entrenamiento, es capaz de obtener programas de ciclos de semáforos optimizados para áreas urbanas extensas. Como casos de estudio, nos hemos centrado en las ciudades de Málaga y Sevilla. Los programas de ciclos resultantes tras una serie de experimentos obtienen reducciones significativas en términos de congestión de tráfico y en tiempo medio de viaje.

Palabras clave— Programación de ciclos, Particle Swarm Optimization, Simulador de Tráfico SUMO.

I. INTRODUCTION

Hoy en día, el volumen de tráfico vehicular existente en las grandes ciudades provoca severos problemas relacionados con la polución y ruido ambiental, la congestión, la seguridad, así como la disponibilidad de plazas de estacionamiento. Debido a que las mejoras de las infraestructuras físicas son costosas y no siempre abordables, los investigadores en este área están de acuerdo en que una planificación correcta y eficiente de los semáforos puede ayudar a reducir estos problemas mediante la mejora de la circulación de vehículos en las ciudades [1] [2]. Además, el número de semáforos operativos está creciendo y su planificación conjunta cada vez más compleja debido a la gran cantidad de combinaciones de periodos de colores que se deben manejar. Por tanto, la utilización de sistemas automáticos inteligentes para la programación óptima de ciclos de semáforos es una tarea indispensable en la gestión del tráfico rodado.

Recientemente, las iniciativas más prometedoras en esta línea están enfocadas en el uso de simuladores [1] [2] [3], ya que éstos proveen una inmediata y continua fuente de información sobre el flujo del tráfico tomando modelos realistas de circulación. En

los últimos años, la mayoría de los esfuerzos están encaminados hacia la programación eficiente de ciclos de luces en semáforos [4] [5].

En este sentido, el uso de técnicas inteligentes automáticas, y en concreto, de algoritmos metaheurísticos [6] [7], ha ido ganando en importancia ya que se ha demostrado su adecuación en la planificación de semáforos [2] [8] [9] [10]. Sin embargo, el uso de dichos sistemas inteligentes se limita por lo general a la optimización de instancias académicas pequeñas, con uno o dos cruces y con 8 semáforos como mucho, o que están diseñados para áreas urbanas muy específicas, con escasos vehículos y semáforos. Además, los resultados reportados en estos trabajos no se compararon con otras técnicas ni con planificaciones generadas por personal experto.

Todas estas razones nos motivaron para proponer una estrategia de optimización mediante la cual, una metaheurística puede encontrar programas de ciclos de colores optimizados para semáforos en áreas urbanas heterogéneas y extensas. En particular, hemos utilizado en este trabajo un algoritmo PSO (Particle Swarm Optimization) [11] como optimizador de programas de ciclos de semáforos. PSO es un algoritmo de fácil configuración que por lo general desarrolla una rápida convergencia a soluciones de calidad [11], lo que lo hace adecuado para la resolución problemas de simulación con alto coste computacional. Para la evaluación de los programas de ciclos generados (codificados como vectores solución), hemos utilizado el simulador de tráfico microscópico SUMO (Simulator of Urban Mobility) [12]. Este simulador trabaja a modo de caja negra que tiene como entrada el programa de ciclos para una determinada instancia de área urbana y devuelve los valores de adecuación de dicho programa para la instancia evaluada. Para este trabajo hemos utilizado dos instancias localizadas en los centros urbanos de Málaga y Sevilla. Los resultados y comparaciones con otras técnicas: Random Search (RANDOM) y Sumo Cycle Program Generator (SCPG) [12]; ponen de relieve las mejoras significativas obtenidas por nuestra propuesta en términos de flujo de tráfico y de tiempo medio de viaje.

Este trabajo se organiza de la siguiente manera. En la Sección II se presenta el problema de la programación óptima de ciclos de luces en semáforos. La Sección III describe nuestra estrategia de optimización. Las Secciones IV y V presentan los experimentos llevados a cabo y los subsecuentes análisis, respectivamente. Las conclusiones y el trabajo futuro se detallan en la Sección VI.

II. EL PROBLEMA: OPTIMIZACIÓN DE PROGRAMAS DE CICLOS EN SEMÁFOROS

Un escenario de tráfico urbano se compone básicamente de: intersecciones, semáforos, vías, direcciones y vehículos circulando por sus rutas previamente establecidas. Los semáforos se sitúan en las intersecciones y controlan el flujo del tráfico mediante sus programas de estados de colores y periodos de duración. Todos los semáforos en una misma intersección están gobernados por un programa común, ya que deben estar necesariamente sincronizados por motivos obvios de seguridad. Además, para todos los semáforos en una misma intersección, la combinación de colores durante un periodo de ciclo debe siempre ser válida, siguiendo reglas de tráfico específicas de dichas intersecciones con el fin de evitar colisiones y accidentes en general. En nuestro modelo trabajamos exclusivamente con combinaciones válidas de estados de colores para cada intersección, las cuales permanecen constantes durante el proceso de optimización. Así se evita la aparición de combinaciones de estado no válidas y se restringe la optimización únicamente al paso de estados factibles.

En este contexto, nuestro principal objetivo consiste en encontrar **programas de ciclos optimizados** (OPCS) para todos los semáforos situados en una determinada área urbana. Nos referimos a los programas de ciclos como el periodo de tiempo en el que un conjunto de semáforos (en un cruce) permanecen con sus estados de luces/colores. Al mismo tiempo, estos programas deben también coordinarse con semáforos en intersecciones adyacentes, mejorando así el flujo de vehículos que circulan conforme a la regulación vial establecida.

A. Codificación

Para nuestra estrategia hemos codificado el OPCS mediante un vector de números enteros (positivos) siguiendo la estructura de SUMO para la programación de ciclos, a través de la cual, cada elemento del vector (variable) representa una duración de fase de los semáforos implicados en una determinada intersección (véase la Fig. 1). A pesar de su simplicidad, esta representación permite tener en cuenta la interdependencia entre variables, no sólo para duraciones de fase en una misma intersección, sino que también para semáforos en intersecciones adyacentes.

B. Función de Fitness

Para evaluar cada programa de ciclos generado por nuestro algoritmo se ha formulado la siguiente función de fitness, mediante la cual se considera la información obtenida de los eventos sucedidos durante la simulación previa:

$$fitness = \frac{TV + TE + (ND * TS)}{V^2 + P} \quad (1)$$

El objetivo principal (Ecuación 1) consiste en maximizar el número de vehículos que alcanzan sus des-

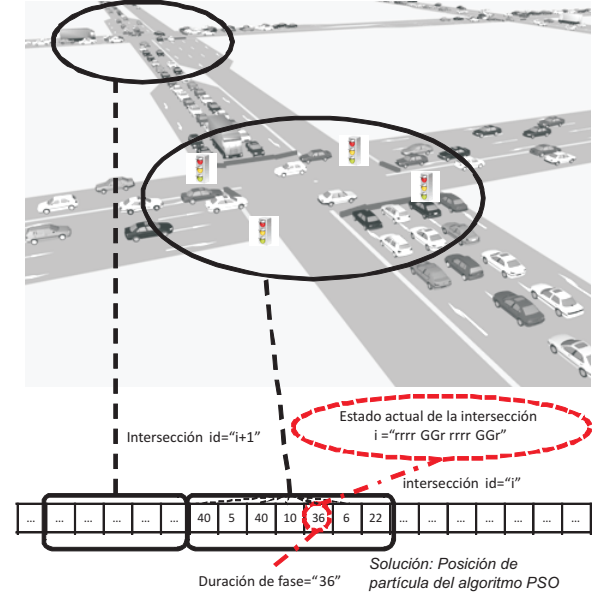


Fig. 1. Programa de ciclos en una intersección

tinios (V), así como minimizar el tiempo medio de viaje de todos los vehículos (TV) durante el tiempo de simulación (TS). El número de vehículos que llegan a sus destinos se eleva al cuadrado (V^2) para hacerlo prioritario sobre los demás términos y factores. Obviamente, el número de vehículos que no alcanzan sus destinos durante la simulación (ND) debe ser minimizado. El tiempo medio de viaje se refiere a la agregación de los tiempos de viaje de todos los vehículos que alcanzan sus destinos durante el tiempo de simulación. Por el contrario, los vehículos que no completan sus viajes serán considerados con tiempo de viaje igual al tiempo de simulación, lo cual implica una penalización adicional. Otro importante término considerado en esta función es el estado en el que los semáforos están en un preciso momento, ya que éste influencia el tiempo que cada vehículo debe parar y esperar (TE), con el consecuente retardo en su propio tiempo de viaje.

Finalmente, una proporción bien balanceada de colores en las duraciones de fase de los estados debería promocionar aquellos estados con más semáforos en verde situados en vías o calles con un gran número de vehículos circulando, y los semáforos en rojo situados en calles con un bajo volumen de tráfico. La proporción de colores en cada fase (pf) de todas las intersecciones int se formula mediante la Ecuación 2.

$$P = \sum_{k=0}^{int} \sum_{j=0}^{pf} d_{k,j} * \left(\frac{v_{k,j}}{r_{k,j}} \right), \quad (2)$$

Donde $v_{k,j}$ es el número de semáforos en verde y $r_{k,j}$ es el número de semáforos en rojo en el estado j (con duración de fase $d_{k,j}$) y en la intersección k . El mínimo valor de $r_{k,j}$ es 1 para evitar la división por cero.

III. ESTRATEGIA DE OPTIMIZACIÓN

Nuestra estrategia de optimización se compone por dos partes principales: la optimización con PSO y la simulación con SUMO.

El algoritmo de PSO [11] es una metaheurística poblacional inspirada en el comportamiento social de las bandadas de pájaros y las agrupaciones de individuos en general. Fue inicialmente diseñado para la resolución problemas de optimización continua. En PSO, cada solución potencial al problema se codifica mediante la *posición* de una partícula y a la población de partículas se le llama *cúmulo* o *enjambre* (*swarm*). Para el desarrollo de nuestro PSO hemos seguido las especificaciones del estándar de 2007 [13]. En este algoritmo, cada posición de partícula x^i se actualiza cada iteración g mediante la Ecuación 3.

$$x_{g+1}^i = x_g^i + v_{g+1}^i \quad (3)$$

donde el término v_{g+1}^i es la velocidad de la partícula, que viene dada por la Ecuación 4.

$$v_{g+1}^i = w \cdot v_g^i + U[0, \varphi_1] \cdot (p_g^i - x_g^i) + U[0, \varphi_2] \cdot (b_g^n - x_g^i) \quad (4)$$

En esta fórmula, p_g^i es la mejor solución personal que la partícula i ha encontrado durante su proceso, b_g^n es la mejor partícula en un vecindario de n partículas (conocida como *el mejor social*) aleatoriamente seleccionado (uniforme) del cúmulo y w es el factor de inercia de la partícula (que controla el balance entre exploración-explotación). Por último, φ_1 y φ_2 son los coeficientes de aceleración, los cuales controlan el efecto relativo de los mejores personal y social de la partícula, mientras que $U[0, \varphi_k]$ es un valor aleatorio uniforme en el intervalo $[0, \varphi_k]$, $k \in 1, 2$. Éste último se genera de nuevo para cada componente en el vector de velocidad y para cada iteración.

Debido a que el OPCS se codifica mediante vectores de números naturales (representando ciclos de duración de fase), hemos utilizado el método de cuantificación (*quantisation*) provisto en la especificación del estándar PSO 2007 [13]. El método de cuantificación se aplica a cada nueva partícula (generada con la Ecuación 3) y transforma las variables continuas a discretas. Básicamente consiste en un cuantificador uniforme de Mid-Thread como se especifica en la Ecuación 5. El paso de cuantificación está inicializado como $\Delta = 1$.

$$Q(x) = \Delta \cdot \lfloor x/\Delta + 0,5 \rfloor \quad (5)$$

El pseudocódigo de Algoritmo 1 describe el Estándar PSO 2007 para OPCS. Este algoritmo comienza con la inicialización de las partículas del cúmulo (Línea 1). Los elementos correspondientes de cada posición de partícula (solución) son inicializados con valores aleatorios representando las duraciones de fase. Estos valores son generados en el intervalo de tiempos $[5, 60] \in \mathbb{Z}^+$, que constituye el rango de periodos temporales posibles (en segundos) en

Algoritmo 1 Estándar PSO 2007 para OCP

```

1: inicializaCúmulo()
2: while  $g < \text{maxIteraciones}$  do
3:   for partícula  $x_g^i$  do
4:      $b_g^n = \text{mejorVecino}(x_g^i, n)$ 
5:      $v_{g+1}^i = \text{actualizaVel}(w, v_g^i, x_g^i, \varphi_1, p_g, \varphi_2, b_g^n)$  //Eq. 4
6:      $x_{g+1}^i = Q(\text{actualizaPos}(x_g^i, v_{g+1}^i))$  //Eqs. 3 y 5
7:     evalúa( $x_{g+1}^i$ ) //SUMO Simulación y Eq. 1
8:      $p_{g+1}^i = \text{actualiza}(p_g^i)$ 
9:   end for
10: end while

```

los que un semáforo puede mantener un determinado color (únicamente verde o rojo, ya que el amarillo es constante). Para un número máximo de iteraciones, cada partícula se mueve a través del espacio de búsqueda mediante la actualización de su velocidad y posición (Líneas 4, 5, y 6), se evalúa (Línea 7) y su mejor posición personal se actualiza también (Línea 8). Finalmente, el algoritmo devuelve como resultado la mejor partícula encontrada durante todo el proceso de optimización.

La simulación se utiliza para asignar un valor cuantitativo de adecuación (fitness) a las soluciones codificadas en las partículas, para un determinado escenario urbano. Esta simulación se lleva a cabo instanciando el simulador microscópico de tráfico SUMO. Así, cuando el algoritmo PSO genera una nueva solución con un nuevo programa de ciclos, se inicia SUMO para simular la instancia con calles, direcciones, obstáculos, semáforos, vehículos, velocidades, rutas, etc., respecto al nuevo programa de ciclos. Tras la simulación, SUMO devuelve los valores necesarios para computar la función de fitness (Ecuación 1). Cada evaluación de solución requiere una única simulación pues las rutas de vehículos en SUMO se generan de manera determinista. Cada nuevo programa de ciclos se carga estáticamente en cada proceso de simulación. En este sentido, lo demandado actualmente en ingeniería civil para la planificación de semáforos son programas de ciclos constantes para áreas específicas y para franjas horarias preestablecidas (hora punta, horario nocturno, festivos, etc.), lo cual nos llevó a abordar el problema con este enfoque.

IV. EXPERIMENTOS

Puesto que estamos interesados en desarrollar un optimizador capaz de tratar con áreas urbanas lo más realistas posibles, hemos utilizado en este trabajo dos escenarios generados a partir de información real en mapas digitales. Tales escenarios cubren áreas similares de aproximadamente $0,75 \text{ km}^2$ y están físicamente localizados en las ciudades de Málaga y Sevilla. La información utilizada para ello consiste en: reglas de tráfico regulado, situación de las señales y otros elementos, edificios, direcciones, calles, intersecciones, etc. Además, se ha determinado el número de vehículos circulando, así como sus velocidades a partir de las especificaciones de la De-

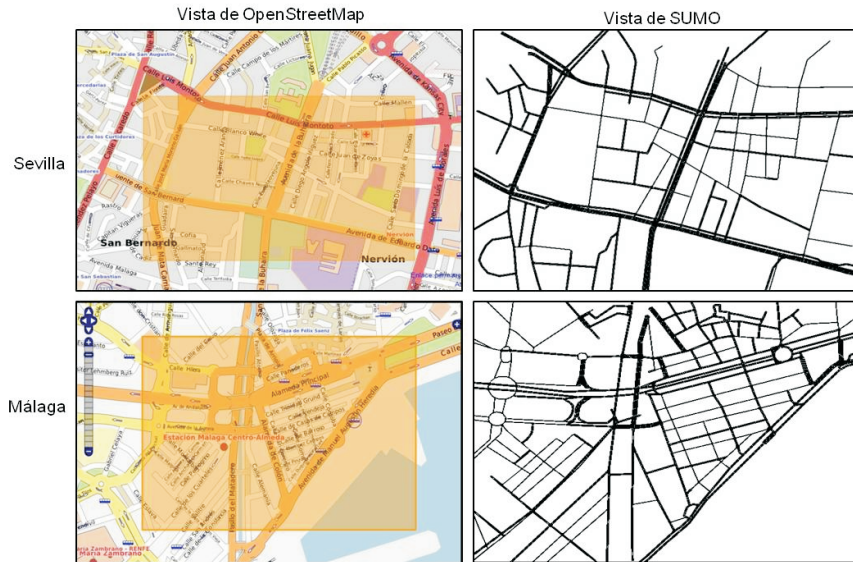


Fig. 2. Instancias de Málaga y Sevilla exportadas a formato de SUMO

legación de Movilidad del Ayuntamiento de Málaga (<http://movilidad.malaga.eu/>). Esta información se obtuvo en esta delegación a partir de puntos sensorizados en ciertas calles de la ciudad, calculando así una medida de la densidad de tráfico en varios intervalos temporales. Para el caso de Sevilla, se consultó la Delegación de Movilidad de este ayuntamiento (<http://www.trajano.com/>).

La ilustración de Fig. 2 muestra las áreas seleccionadas para las dos ciudades tratadas con sus correspondientes capturas de OpenStreetMap y SUMO, describiendo el proceso de generación de los escenarios urbanos. Dichas áreas urbanas contienen las siguientes características:

1. *Sevilla*. Situada en el popular distrito de Nervión en el centro de dicha ciudad (Fig. 2, arriba). Está compuesta por intersecciones entre calles incluyendo cada una de 4 a 17 semáforos. El área completa, de organización regular, tiene todos sus cruces conectando entre tres y cuatro calles. Las principales avenidas que cruzan este escenario son: Menéndez Pelayo, Eduardo Dato, San Francisco Javier, Montoto, Galván y Buharúa.
2. *Málaga*. Situada en la zona entre el centro de la ciudad y el puerto. Con un trazado irregular (Fig. 2, abajo). Está compuesta por calles con diferentes anchuras y longitudes, además de varias rotondas. Contiene intersecciones con un número de semáforos entre 4 y 16. Las principales avenidas encontradas en esta instancia son: Alameda Principal, Andalucía, Manuel Agustín Heredia, Colón y Aurora.

En concreto hemos elegido estos dos escenarios debido a que constituyen diferentes áreas metropolitanas con estructuras heterogéneas. El número de intersecciones estudiadas es de 70, con 500 vehículos circulando en cada una de las instancias. Es necesario clarificar que, a pesar de tener los dos escenarios el mismo número de intersecciones (70), el número

de semáforos no es el mismo ya que dichas intersecciones son muy diferentes entre sí, sumando 368 semáforos en Sevilla y 312 en el caso de Málaga.

En las simulaciones, cada uno de los vehículos recorre su propia ruta desde el origen hasta su destino circulando con una velocidad máxima de 50 km/h (velocidad típica en las áreas urbanas). Las rutas fueron generadas a priori aleatoriamente. El tiempo de simulación es de 500 segundos (iteraciones de microsimulación) para cada instancia. Este tiempo de duración fue determinado como el tiempo máximo que tarda un coche en completar su ruta, incluso si éste debiera parar en todos los semáforos que encontrara, es decir, si encontrara todos los semáforos iniciando fase en rojo. Cuando un vehículo abandona el recorrido del escenario, ya no volverá a aparecer de nuevo durante la simulación.

Para la ejecución del algoritmo hemos utilizado la implementación en C++ de PSO que provee la biblioteca MALLBA [6]. La fase de simulación se realiza mediante el simulador SUMO en su versión 0.12.0 para Linux. Los experimentos se realizaron en las computadoras de los laboratorios del departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga. La mayoría de estas máquinas están equipadas con procesadores dual core, con 1GB de RAM y sistema operativo Linux Devian. Todas estas máquinas operan bajo la plataforma middleware Condor [14], que actúa como un planificador de tareas distribuido (cada tarea se encarga de una ejecución independiente de PSO).

Para cada escenario se realizaron 30 ejecuciones independientes de nuestro PSO. El tamaño del cúmulo es de 100 partículas, realizando 300 iteraciones de optimización (resultando de esta manera unas 30.000 evaluaciones, e.g., simulaciones de SUMO) por ejecución e instancia. Como ya comentamos en la sección anterior, el tamaño de la partícula

TABLA I
PARÁMETROS DE SIMULACIÓN Y DE PSO

Fase	Parámetro	Valor
Simulación	Tiempo de Simulación (pasos)	500 s
	Area de Simulación	0,75 km ²
	Numero de Vehículos	500
	Velocidad de Vehículo	0-50 km/h
	N. de Intersecciones	70
PSO	Max. N. de Evaluaciones	30.000
	Tamaño del Cúmulo	100
	Tamaño de Partícula (N. Semáforos)	368
		312
	Coeficiente Local (φ_1)	2,0
	Coeficiente Social (φ_2)	2,0
	Máximo de Inercia (w_{max})	0,5
	Mínimo de Inercia (w_{min})	0,1

depende directamente del número de semáforos de la instancia. El resto de parámetros están resumidos en la Tabla I, siendo éstos el resultado de ejecuciones preliminares de nuestra estrategia de optimización con el escenario de Málaga. Respecto a los parámetros específicos de PSO, se utilizaron aquellos recomendados en el estudio sobre la convergencia de este algoritmo en [15].

Con el fin de poder establecer comparaciones, hemos implementado además un algoritmo de Random Search, también en el ámbito de la biblioteca MALLBA. De esta forma, sometiendo a este algoritmo al mismo procedimiento experimental que a PSO esperamos obtener algunos indicios sobre el poder de nuestra propuesta, es decir, cómo de inteligente es respecto a un algoritmo que no utiliza información heurística en su funcionamiento. El pseudocódigo del algoritmo de Random Search (RANDOM) se muestra en Algoritmo 2. El número máximo de evaluaciones a realizar por RANDOM fue también iniciado a 30.000.

Algoritmo 2 Pseudocódigo de RANDOM

```

1: inicializaSolución(x)
2:  $i \leftarrow 0$ 
3: while  $i < \text{maxEvaluaciones}$  do
4:   genera( $x_i$ ) //nueva solución
5:   if  $f(x) \geq f(x_i)$  then
6:      $x \leftarrow x_i$ 
7:   end if
8:    $i \leftarrow i + 1$ 
9: end while

```

Por otra parte, SUMO provee de un algoritmo determinista para la generación de programas de ciclos (SCPG). Por tanto, podemos así comparar los programas de ciclos obtenidos por nuestro PSO contra aquellos generados por SUMO. Este último algoritmo consiste básicamente en asignar a las duraciones de fase de los semáforos nuevos valores en el rango [6,31], de acuerdo a tres factores diferentes:

1. La proporción de estados en verde de las fases,
2. el número de vías de entrada en las intersecciones,
3. el tiempo de frenado de los vehículos cuando se acercan a los semáforos.

La información completa sobre el funcionamiento del algoritmo SCPG puede ser encontrada en [12].

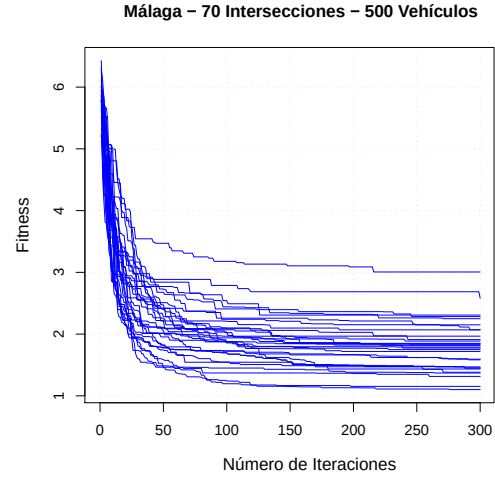


Fig. 3. Trazas de progreso de PSO para Málaga

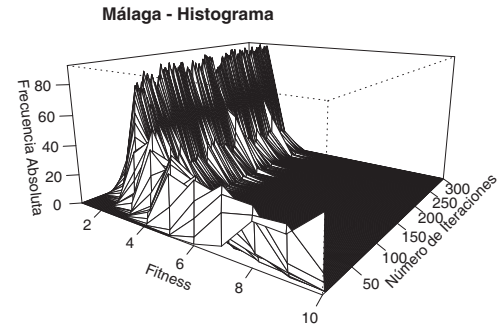


Fig. 4. Histograma de valores de fitness del cúmulo completo

A. Resultados y Comparaciones

En primer lugar, en esta sección analizamos el comportamiento interno de nuestro PSO en la programación óptima de ciclos de semáforos.

Gráficamente, en Fig. 3 podemos observar las trazas de progreso de las 30 ejecuciones independientes de PSO en la planificación de la instancia de Málaga. Podemos ver en esta figura cómo para todas las ejecuciones nuestro algoritmo converge tras las primeras 200 iteraciones (20.000 evaluaciones), utilizando el resto del tiempo en pequeñas mejoras de refinamiento de soluciones. Todos las soluciones finales resultaron con cierta dispersión en cuanto a la calidad del fitness y en cuanto a su composición. Casi todos los valores de fitness resultantes están en el rango entre 1 y 3. En términos de convergencia y robustez, éstas son características deseables ya que así podemos ofrecer a los expertos en el área un conjunto variado de programas de ciclos desde un estado inicial de la optimización.

Más en profundidad, respecto a cada ejecución independiente, la figura Fig. 4 muestra un ejemplo representativo de traza donde podemos observar la frecuencia absoluta de la distribución de fitness del cúmulo completo. En concreto, se refiere a una ejecución de nuestro PSO en la resolución del escenario de Málaga. En esta gráfica, las partículas iniciales

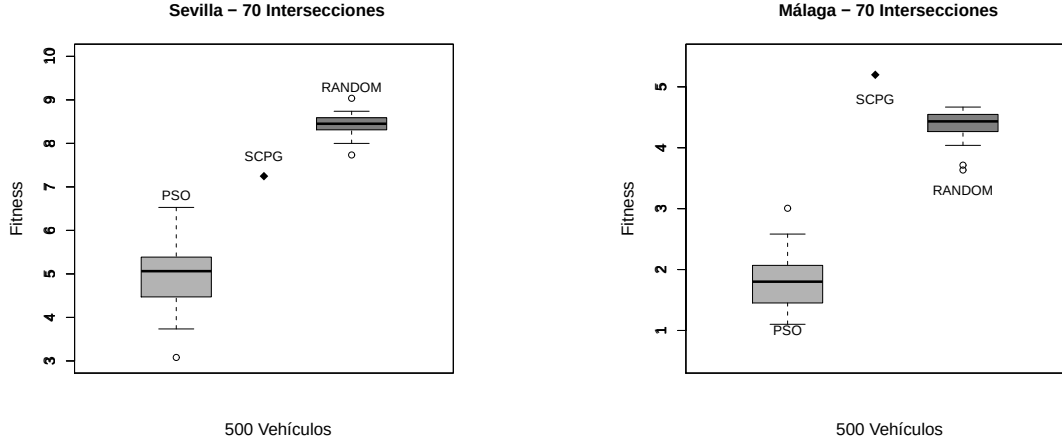


Fig. 5. Boxplots de las distribuciones resultantes para Sevilla y Málaga. Los resultados de SCPG se representan con ♦

TABLA II
VALORES DE FITNESS OBTENIDOS POR PSO, RANDOM Y
SCPG PARA LAS INSTANCIAS DE SEVILLA Y MÁLAGA

Instance	Value	PSO	RANDOM	SCPG
Sevilla	Máximo	6.53E+00	9.04E+00	7.24E+00
	Mediana	5.06E+00	8.45E+00	7.24E+00
	Mínimo	3.08E+00	7.73E+00	7.24E+00
	Media	4.95E+00	8.43E+00	7.24E+00
	Desv. Tip.	6.68E-01	2.58E-01	0.00E+00
Málaga	Máximo	3.01E+00	4.67E+00	5.19E+00
	Mediana	1.80E+00	4.43E+00	5.19E+00
	Mínimo	1.10E+00	3.63E+00	5.19E+00
	Media	1.81E+00	4.37E+00	5.19E+00
	Desv. Tip.	4.22E-01	2.55E-01	0.00E+00

presentan valores de fitness diversos y altos ($\simeq 8$), para ir convergiendo durante la segunda mitad del proceso de optimización hacia soluciones con valores de fitness ($\leq 1,5$) similares y bajos.

En cuanto a los resultados generales, la Tabla II contiene: el máximo, la mediana, la media, el mínimo y la desviación estándar de los valores de fitness obtenidos (en las 30 ejecuciones independientes) por la estrategia propuesta para las dos instancias: Málaga y Sevilla. Junto a éstos se presentan también en la misma tabla los resultados referentes a los algoritmos RANDOM y SCPG, para así poder establecer comparaciones fácilmente. De este modo, podemos comprobar en esta tabla que PSO obtuvo los mejores resultados (en negrita), ya que incluso los máximos valores obtenidos por nuestra propuesta son más bajos (estamos minimizando) que los valores medios mostrados por los otros dos algoritmos.

Con la intención de dotar a estas comparaciones de significado estadístico hemos aplicado diferentes *t-tests* [16] a los valores numéricos y las distribuciones de los resultados. Hemos utilizado estos tests paramétricos ya que las distribuciones resultantes mostraron condiciones de normalidad e igualdad de varianzas adecuadas. El nivel de confianza utilizado fue del 95 % ($p\text{-value}=0,05$), lo cual nos permite asegurar que dichas distribuciones son estadísticamente diferentes si los test resultan con un $p\text{-value}<0,05$.

Efectivamente, el *t-test para muestras independientes* aplicado a las distribuciones de PSO y RANDOM (Tabla II) resultó con $p\text{-values} \ll 0,05$, para las dos instancias: Málaga y Sevilla. De manera similar, el *t-test para una muestra* aplicado a la media de la distribución de PSO contra el valor de SCPG también resultó con $p\text{-values} \ll 0,05$. Por tanto, podemos destacar que PSO obtuvo resultados estadísticamente mejores que los otros dos algoritmos comparados: RANDOM (con búsqueda totalmente estocástica) y SCPG (determinista); lo que nos lleva a confirmar que nuestra estrategia de optimización es inteligente y adecuada en comparación con algoritmos de información ávida y con conocimiento de los expertos en el área.

Desde un punto de vista gráfico, Fig. 5 muestra los boxplots de las distribuciones resultantes de PSO y RANDOM. Los resultados de SCPG están representados mediante un punto (♦) ya que ésta técnica siempre devuelve un mismo valor determinista para cada instancia. Como era de esperar, las distribuciones de PSO muestran mejores cuartiles inferiores, medianas y cuartiles superiores que RANDOM y SCPG. Para la instancia de Sevilla, el algoritmo RANDOM obtuvo peores resultados que SCPG. Posiblemente, el alto número de semáforos en el caso de Sevilla hace que SCPG obtenga programas de ciclos más eficientes que los obtenidos por RANDOM, lo cual muestra el mal comportamiento de este último algoritmo para la instancia más compleja (Sevilla). El uso de información heurística adicional en los algoritmos resulta imprescindible a la hora de resolver este problema.

V. ANÁLISIS DE LOS PROGRAMAS DE CICLOS RESULTANTES

Esta sección está dedicada al análisis de programas de ciclos ejemplos resultantes de nuestro PSO, así como los posibles beneficios que pueden ofrecer a los usuarios en este campo. De esta forma mostramos el impacto real que se deriva de la utilización

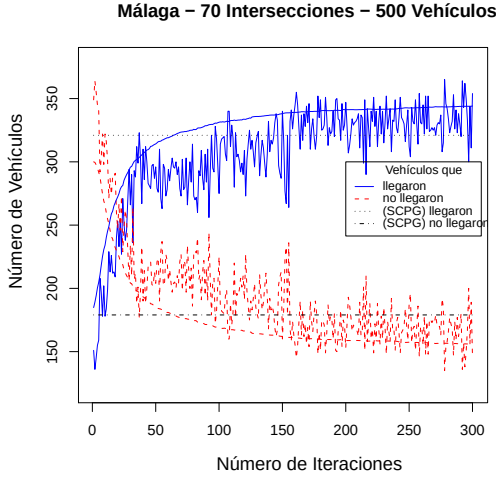


Fig. 6. Número de vehículos que llegaron (no llegaron)

de nuestra estrategia de optimización en la computación de programas de ciclos de semáforos.

Con esta idea en mente, para cada iteración de PSO y cada partícula del cúmulo, hemos tomado toda la información obtenida de cada simulación (evaluación de una solución) realizada. Esta información consiste en el número de vehículos que alcanzan su destino durante el tiempo de simulación, la duración media de sus viajes y el momento en el que abandonaron el escenario. De esta forma podemos distinguir la mejora progresiva en el flujo del tráfico con respecto a las soluciones iniciales.

Un ejemplo representativo puede observarse en el proceso de optimización de la instancia de Málaga. En primer lugar, en la figura Fig. 6 se muestra la traza (ejecución de la mediana) respecto al número de vehículos que alcanzaron sus puntos de destino (curva continua superior) versus el número de vehículos que no alcanzaron sus destinos (curva discontinua inferior), para cada paso de iteración de PSO. Las curvas solapadas muestran el número medio de vehículos (en 30 ejecuciones independientes) que llegaron (y no llegaron) a sus destinos. Además, los resultados de SCPG están mostrados para la misma instancia mediante líneas rectas discontinuas.

A partir de Fig. 6 se puede observar fácilmente cómo la cantidad de vehículos que llegaron (no llegaron) a sus destinos crece (decrece) hasta que el algoritmo alcanza su condición de parada a las 300 iteraciones. De hecho, en los pasos iniciales del proceso de optimización, el número de vehículos que llegaron a sus destinos era menor que aquellos resultantes del programa de ciclos generado por SCPG. Sin embargo, en los pasos intermedios y finales de PSO, las soluciones resultantes muestran una mayor calidad en términos de flujo de tráfico, ya que 349 vehículos de los iniciales 500 (69.8 %) finalizaron satisfactoriamente sus trayectos. Más aún, en las soluciones finales de PSO, una media de 340 vehículos completaron sus viajes (media de 30 ejecuciones). Esto último contrasta con los 314 vehículos que completaron sus

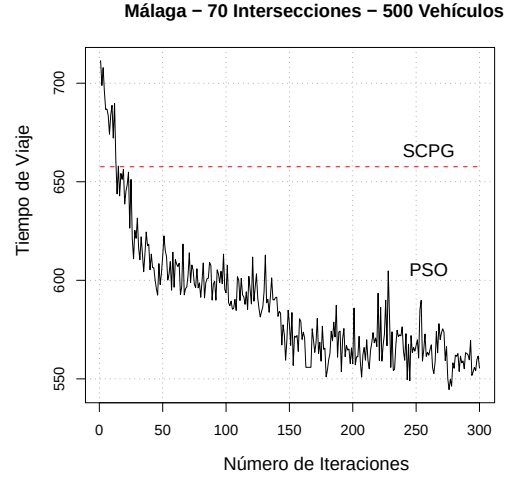


Fig. 7. Tiempo medio de viaje

trayectos con el programa de ciclos de SCPG, resultado una mejora del 10.13 % de PSO sobre SCPG.

Desde un punto de vista diferente, la gráfica presentada en Fig. 7 nos muestra la traza del tiempo medio de viaje empleado por los vehículos con los programas de ciclos generados por PSO durante todas las evaluaciones de una ejecución. En este caso, el tiempo de viaje es cada vez más corto a medida que el algoritmo alcanza su condición de parada. Es necesario señalar que, en el cálculo del tiempo de viaje, los vehículos que no llegaron a sus destinos durante el tiempo de simulación tomaron un tiempo de 500 segundos, es decir, el tiempo de simulación completo. Por este motivo, las soluciones de SCPG mostraron un tiempo medio acumulado de 660 segundos mientras que las de PSO obtuvieron tiempos de 557 segundos, lo cual representa una mejora del 15.7 % de los programas de PSO con respecto a los de SCPG.

Finalmente, con la intención de comprender mejor las implicaciones de utilizar (o no utilizar) un programa de ciclos optimizado, en Fig. 8 se muestran las trazas de simulación del flujo de tráfico resultante tras aplicar programas generados por SCPG (izquierda) y PSO (derecha). Estas imágenes corresponden a capturas del último paso de simulación (segundo 500) para dos áreas seleccionadas de simulaciones de Málaga que incluyen: la avenida Andalucía, la avenida Aurora y la calle Guadalmedina. Como podemos observar, la densidad en el tráfico generado por el programa de ciclos de SCPG es claramente mayor que con el programa generado por PSO (mediana de la 30 ejecuciones independientes), incluso mostrando el primero algunas intersecciones con atascos severos. En el caso del programa generado por PSO, no se registró ningún atasco en sus intersecciones al final de la simulación.

VI. CONCLUSIONES

En este trabajo presentamos una estrategia de optimización basada en PSO capaz de encontrar pro-

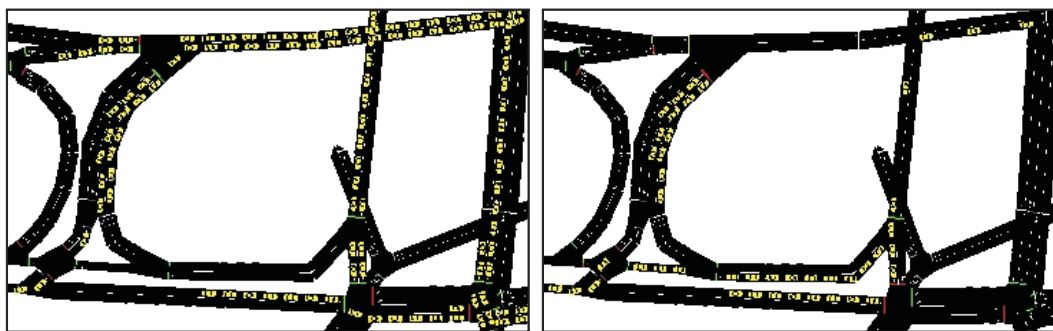


Fig. 8. Congestión de tráfico generado por SCPG (izquierda) y PSO (derecha) en Málaga

gramas de ciclos eficientes para semáforos de manera automática. Como casos de estudio hemos utilizado dos instancias realistas localizadas en los centros metropolitanos de Málaga y Sevilla. Los experimentos y análisis realizados con el sistema propuesto se presentan desde dos enfoques diferentes: el comportamiento de la técnica de optimización utilizada y la calidad de las soluciones generadas en el dominio de la planificación de semáforos. Las principales conclusiones que podemos extraer son los siguientes:

1. Nuestra estrategia muestra un comportamiento satisfactorio en escenarios de tráfico extensos y de diseño realista. Para las dos instancias estudiadas, nuestro PSO obtiene resultados estadísticamente mejores que los otros dos algoritmos comparados: el generador de programas de ciclos de SUMO (SCPG) y el de Random Search (RANDOM).
2. Las soluciones finales obtenidas por PSO mejoran tanto en el número de vehículos que llegan a sus destinos como en la duración de viaje, para las dos instancias. En particular, para el caso de Málaga, la mejora obtenida ronda el 10.13% en el número de viajes completados y el 15.7% en la duración de dichos viajes, respecto a SCPG.
3. El análisis en cuanto a la congestión del tráfico muestra que los programas de ciclos generados por nuestro PSO mejoran en general el flujo de vehículos en la red vial. Todo esto supone una mejora real en el tráfico para los ciudadanos.

Como trabajo futuro, con el propósito final de asistir a los expertos en la toma de decisiones, pretendemos abordar el problema de la planificación de ciclos de semáforos con versiones más modernas PSO y otras metaheurísticas. También estamos interesados en abordar el problema desde un punto de vista multiobjetivo y con nuevas instancias de redes viales.

AGRADECIMIENTOS

Este trabajo está financiado por la Junta de Andalucía (CICE), mediante el proyecto P07-TIC-03044 (DIRICOM <http://diricom.lcc.uma.es>) y el Ministerio Español de Ciencia e Innovación (MICINN) y FEDER con los proyectos TIN2008-06491-C04-01 (M* <http://mstar.lcc.uma.es>) y TIN2011-28194 (RoadMe <http://roadme.lcc.uma.es>). José García-Nieto disfruta de una beca con código BES-2009-018767 de MICINN. Ana Carolina Olivera disfruta de un contrato postdoctoral del Consejo Nacional de Investigaciones Científicas y Técnicas de Argentina (CONICET, www.conicet.gov.ar).

REFERENCIAS

- [1] J. McCrear and S. Moutari, "A hybrid macroscopic-based model for traffic flow in road networks," *Euro. Jour. of Oper. Res.*, vol. 207, no. 2, pp. 676 – 684, 2010.
- [2] J. Sánchez, M. Galán, and E. Rubio, "Applying a traffic lights evolutionary optimization technique to a real case: "Las Ramblas" area in Santa Cruz de Tenerife," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 25 –40, feb. 2008.
- [3] C. Karakuzu and O. Demirci, "Fuzzy logic based smart traffic light simulator design and hardware implementation," *App. Soft Comp.*, vol. 10, no. 1, pp. 66 – 73, 2010.
- [4] E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg, "Optimizing traffic lights in a cellular automaton model for city traffic," *Phys. Rev. E*, vol. 64, no. 5, pp. 56–132, 2001.
- [5] T. Nagatani, "Effect of speed fluctuations on a green-light path in a 2d traffic network controlled by signals," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 19, pp. 4105 – 4115, 2010.
- [6] E. Alba, G. Luque, J. García-Nieto, G. Ordonez, and G. Leguizamón, "Mallba: a software library to design efficient optimisation algorithms," *Int. J. of Innovative Computing and Applications 2007 (IJICA)*, vol. 1, no. 1, pp. 74–85, 2007.
- [7] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.
- [8] J. García-Nieto, E. Alba, and A. C. Olivera, "Swarm intelligence approach for the traffic light scheduling: Application to real urban areas," *Engineering Applications Of Artificial Intelligence*, vol. -, pp. In Press, 2011.
- [9] L. Peng, M. Wang, J. Du, and G. Luo, "Isolation niches particle swarm optimization applied to traffic lights controlling," in *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, dec. 2009, pp. 3318 –3322.
- [10] F. Teklu, A. Sumalee, and D. Watling, "A genetic algorithm approach for optimizing traffic control signals considering routing," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, pp. 31–43, 2007.
- [11] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, 2001.
- [12] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package SUMO," *RoboCup 2006 Infrastructure Simulation Competition*, 2006.
- [13] M. Clerc et al., "Standard PSO 2011," Tech. Rep. [online] <http://www.particleswarm.info/>, Particle Swarm Central, January 2011.
- [14] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: the condor experience.," *Concurrency - Practice and Experience*, vol. 17, no. 2-4, pp. 323–356, 2005.
- [15] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [16] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, CHAPMAN & HALL/CRC STATISTICS, D. J. Sheskin, UK, 2003.