

Tipo de artículo: Artículo original  
Temática: Matemática computacional  
Recibido: 7/05/2014 | Aceptado: 21/05/2014

## Una herramienta para simular y analizar el flujo de tráfico en un cruce regulado por semáforos con tiempos de ciclos variables

### *A tool to simulate and analyze traffic flow in an intersection controlled by traffic lights with variable cycle times*

Orlando Fonseca Guilarte<sup>1\*</sup>, Gabriel Santiago Pujol Fariña<sup>1</sup>, Sira Allende Alonso<sup>1</sup>, Gemayqzel Bouza Allende<sup>1</sup>

<sup>1</sup>Departamento de Matemática Aplicada, Facultad de Matemática y Computación, Universidad de La Habana. San Lázaro y L, Edificio Felipe Poey. La Habana, Cuba. Código Postal 10400. Correo-e: [gabrielpujol87@gmail.com](mailto:gabrielpujol87@gmail.com)

\*Autor para correspondencia: [ofonsek0702@hotmail.com](mailto:ofonsek0702@hotmail.com)

---

#### Resumen

El objetivo de este trabajo es presentar una herramienta que ayude a establecer una buena política de cambio en los tiempos de luces de los semáforos en las horas pico, basado en los resultados experimentales que se brindan. Se considera un cruce regulado por semáforos que puede estar compuesto por  $n$  calles y tener  $k$  fases. Se propone un simulador cuyos datos de entrada son las características propias de esta intersección y su salida la cantidad de autos que quedan, llegan y salen en cada fase. Se supone que las llegadas y salidas de los vehículos siguen una distribución de Poisson cuyo parámetro puede variar en el intervalo estudiado. La simulación está basada en eventos discretos y para la implementación del simulador se utilizó la plataforma .NET y el lenguaje de programación C#. La herramienta permite además obtener resultados estadísticos que ayudan al usuario en la toma de decisiones para identificar las fases problemáticas y tomar medidas para determinar la mejor opción, que será aquella que minimice la congestión del tráfico.

**Palabras clave:** control de semáforos, control de tráfico, optimización, simulación basada en eventos discretos.

### **Abstract**

*The goal of this paper is to present a tool that helps establishing a good change policy for the time of the traffic lights at peak hours, based on the given experimental results. The intersection is considered to be regulated by traffic lights, and be composed by  $n$  streets and have  $k$  different light phases. The proposed simulator's input consists of the characteristics of the intersection, and its output, the amount of cars that stay, arrive and exit it in each phase. It's assumed that the arrivals and departures of vehicles follow a Poisson distribution whose parameter can vary within the studied interval. The simulation is based on discrete events, and to implement the simulator the .NET framework and C# language were used. This tool also allows obtaining statistical results that help the user to identify problematic hours and hence, to take decisions to determine the best option to minimize the traffic congestion.*

**Keywords:** Optimization, simulation based on discrete events, traffic control, traffic lights control.

---

## **Introducción**

El control del tráfico en las zonas urbanas es un problema en constante evolución. De hecho, el aumento del tamaño de la población conduce a una mayor congestión ya que el volumen de tráfico crece proporcionalmente a la población. Las congestiones de tránsito provocan estrés, contaminación ambiental y pérdida de dinero debido a los largos tiempos de espera en las calles.

Por estas razones, las ciudades requieren sistemas de transporte más eficientes y un mejor control del tráfico. En particular, es importante contar con una buena política de control de semáforos en las intersecciones con el fin de resolver los problemas de congestión de tráfico y sobresaturación de vehículos. (Nagatani, 2006). De ahora en adelante, al mencionar intersecciones, se va a hacer alusión solamente a las controladas por semáforos.

Dada la importancia de la estimación de demoras y las complejidades presentes en las intersecciones semaforizadas, en distintas partes del mundo se han realizado numerosos esfuerzos con el fin de poder determinar la mejor y más realista forma de estimar y predecir las demoras en una intersección semaforizada (Larrain, 2012).

El problema de determinar las políticas óptimas de cambio de luces en los semáforos se ha modelado matemáticamente como un problema con restricciones de complementariedad (De Schutter, 1998; De Schutter, 2002) y como problemas de control discretos (Allende, 2009). Por sus soluciones prácticas, los métodos de lógica difusa (Valverde, 2010; HaiBo, 2010), redes neuronales (Kuei-Hsiang, 2008) y metaheurísticas (Sánchez, 2005) son los enfoques más utilizados.

Un problema importante es validar las políticas propuestas, pero la comparación entre ellas es demasiado costosa para hacerse en la práctica. Para esto se propone un simulador que genera el número de autos que salen, entran y quedan en la intersección, cuya duración de fases puede variar de acuerdo a los criterios del usuario. Así, la elección de una política de control está justificada por la mejora en la movilidad de los vehículos basado en la reducción del tamaño de las colas.

## **Materiales y métodos**

En la implementación de la herramienta se utilizó el ambiente de desarrollo Visual Studio 2010, la plataforma .NET, el lenguaje de programación C#, y las bibliotecas gráficas Windows Forms y ZedGraph.

El lenguaje C# fue escogido por ser un lenguaje de alto nivel, de propósito general y orientado a objetos, que se fundamenta en una tecnología basada en componentes para el desarrollo de software. (Mayo, 2002)

El kit de desarrollo de software .NET Framework permite que los lenguajes compatibles con él, como lo es C#, puedan operar de manera intercambiable. Se puede escribir un componente en cualquier lenguaje compatible con .NET y ese mismo componente puede ser utilizado en cualquier otro lenguaje que también sea compatible, como lo son C++ y Visual Basic, entre otros. (Mayo, 2002)

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows basado en componentes para crear aplicaciones eficaces de alto rendimiento. Utiliza plataformas de desarrollo como la API de Windows, Windows Forms y Windows Presentation Foundation. Visual Studio incluye un editor de código que se apoya en herramientas como IntelliSense, así como la posibilidad de refactorización de código. (Microsoft, 2010)

ZedGraph es una biblioteca de clases con controles de usuario para formularios Windows Forms cuyo propósito es la creación de gráficos 2D sobre conjuntos de datos arbitrarios. Las clases proporcionan un alto grado de flexibilidad y, al mismo tiempo, el uso de las clases se mantiene simple, proporcionando valores por defecto para todos los atributos del gráfico. Las clases incluyen código para la elección de los rangos de escala apropiados y tamaños de paso en función de la gama de valores de los datos a representar gráficamente. (Champion, 2007).

La aplicación se diseñó siguiendo un grupo de patrones de diseño para obtener un marco robusto y ligero, que permitiera de forma coherente el desarrollo de la herramienta y reutilizara la mayor cantidad de código fuente. (Larman, 1999). Como esquema general de desarrollo se utilizó el enfoque ascendente (Bottom-Up), construyéndose una base sólida y

aplicándose después a casos específicos, y como método se utilizó el orientado a objeto, permitiendo que las abstracciones en el sistema sean claras, generales, manejables, preparadas para el cambio y reutilizables.

La herramienta desarrollada es una aplicación de escritorio compuesta por 4 módulos fundamentales. El método de diseño es modular porque se descompone el problema en un pequeño número de subproblemas menos complejos, interconectados mediante una estructura sencilla y lo suficientemente independientes. Además, cambios en la especificación sólo provoca cambios en un pequeño número de módulos, los cuales son totalmente comprensibles sin tener que examinar muchos de los otros. Cumple también con la Composición y la Protección modular. (Meyer, 2002)

## Resultados y discusión

### *Preliminares*

Esta sección contiene las definiciones y notaciones que se utilizarán a lo largo de este artículo.

**Intersección semaforizada:** lugar donde se interceptan dos o más calles, en el cual la circulación de vehículos está regulada permanente o mayoritariamente mediante sistemas de luces que establecen las prioridades de paso por la intersección.

**Fase:** combinación de luces para la cual una serie de semáforos de la intersección permanecen con la luz verde (color de la luz característica que permite el movimiento) y el resto con la luz roja. Se denota por  $\delta_k$  a la duración de la fase  $k$ .

**Estructura de fase:** movimientos permitidos a los carriles cuyos semáforos están en verde en una fase determinada.

**Duración de fase:** tiempo en que los semáforos de una fase permanecen con la luz verde.

**Ciclo:** combinación de diferentes fases de los semáforos de una intersección.

**Duración del ciclo:** suma de la duración de cada fase que compone el ciclo. Puede ser constante o variable. En el primer caso la fase  $k$  puede tener la misma duración para todos los ciclos y en el segundo caso la duración de las fases puede variar de un ciclo a otro.

**Tasa de llegada del carril  $l$ :** cantidad esperada de autos que llegan al carril  $l$  en cada segundo. Se denota por  $\lambda_l$ .

**Tasa de salida del carril  $l$ :** cantidad esperada de autos que salen del carril  $l$  en cada segundo. Se denota por  $\mu_l$  ( $\kappa_l$ ) si la luz verde (amarilla) está encendida para ese carril.

Los valores de  $\lambda_l$ ,  $\mu_l$ ,  $\kappa_l$  pueden variar durante el día. Durante determinados períodos como en el de tarde en la noche, los arribos no son frecuentes, por lo que  $\lambda_l$  es pequeño. En el caso de las horas picos, este parámetro es alto.

### *El Simulador*

El Simulador es una aplicación que contiene un modelo computacional. En su ejecución, los elementos de la entrada son procesados por un modelo matemático, cuya salida es mostrada de diversas formas al usuario.

La entrada del simulador contiene la estructura y el orden de las fases, así como el número de carriles que convergen en la intersección. El usuario también puede especificar el intervalo de tiempo deseado para estudiar el sistema, la duración de las fases  $\delta_k$ , el número de vehículos en cada carril (estado inicial) y las tasas de salida y de llegada, las cuales pueden ser definidas en función del tiempo como funciones constantes a trozos:  $\lambda_l(t)$ ,  $\mu_l(t)$ ,  $\kappa_l(t)$ . Para el análisis estadístico de los datos, el usuario debe especificar el número de simulaciones  $N$  que se ha de hacer y el estadígrafo que desee aplicar para ver su comportamiento en un gráfico.

La simulación está basada en eventos discretos dado que la ocurrencia asíncrona de dichos eventos fuerza la transición de estados (García, 2006). El estado del sistema viene dado por el número de vehículos en cada carril. Para el tiempo del simulador, no se estudiará los momentos en el que un vehículo entra o abandona el sistema. El proceso de nacimiento-muerte se estudia en cada fase al generar el número de automóviles a la entrada (o salida) del carril usando la distribución de Poisson correspondiente (Dodge, 2003), esto es  $\lambda_l \delta_k$  para el arribo al carril  $l$ ,  $\mu_l(\delta_k - \delta_{amb})$ , para las salidas en la luz verde y  $\kappa_l \delta_{amb}$ , para las salidas en la luz amarilla, donde  $\delta_k$  denota la duración de la fase  $k$ .

Teniendo en cuenta esto, se considera que el estado del sistema al final de cada fase está dado por el número de vehículos al principio de la fase más la diferencia entre los vehículos salientes y entrantes si es positivo.

El modelo matemático es un modelo probabilístico (la estocasticidad dentro del simulador viene dada por la generación de valores de variables aleatorias que distribuyen Poisson) y dinámico (dado por el reloj en el transcurso de los estados del sistema). Las variables de control son los tiempos en los cuales ocurren cambios de luces en los semáforos, pudiendo originar ciclos de longitud constante o variable según la estructura y duración de las fases que intervienen en estos ciclos. La medida de eficiencia empleada es la longitud de cola de cada carril, y como evaluación, se quiere determinar los ciclos críticos, que son los períodos en los que hay mayor congestión.

Como salida, el sistema genera un objeto que contiene para cada corrida, cada carril y cada fase, el número de autos que entraron, salieron y se quedaron en la intersección.

La herramienta se implementó de forma modular, con módulos que se integran para satisfacer todos los requerimientos y ayudar a la toma de decisiones de los usuarios. Contiene cuatro módulos independientes que garantizan que sea más flexible el trabajo con ellos y que se desarrollen otros módulos paralelamente. Estos son:

- La Configuración Inicial, es donde se especifican todas las variables de entrada del simulador. Este módulo se encarga de sincronizar en el tiempo todas las estructuras de datos que se tiene de forma tal que cada objeto tome el valor que le corresponda para cada intervalo de tiempo.
- El Modelo de Simulación, el cual constituye el núcleo del sistema y se encarga de manejar los eventos discretos, su reloj interno y la generación de las variables aleatorias.
- El Evaluador permite al usuario hacer validaciones experimentales, y entre sus implementaciones ofrece el cálculo de la media, la mediana, la moda, el máximo, la varianza y el intervalo de confianza para la media de todos los autos que quedan al final de cada fase de las N corridas realizadas.
- El Graficador, el cual ilustra mediante gráficos 2-D el conjunto de valores producidos para cada carril por el estadígrafo seleccionado en el Evaluador, mostrándose los datos en cada fase, ciclo y hora. Esto permite identificar la cantidad de ciclos problemáticos y el intervalo de tiempo en el que tiene lugar. Además se visualiza el total de autos que quedan en el cruce en general para cada fase (no se analizan los datos por cada carril sino que se ve el cruce como un todo) lo que posibilita saber si hay ciclos en los que todos los carriles tienen colas vacías o casi vacías.

### *Un ejemplo ilustrativo*

A continuación se presenta un ejemplo ilustrativo en el que la intersección tiene tres fases con el siguiente esquema, que no corresponde a una intersección real aunque podría serlo, ver Figura 1.

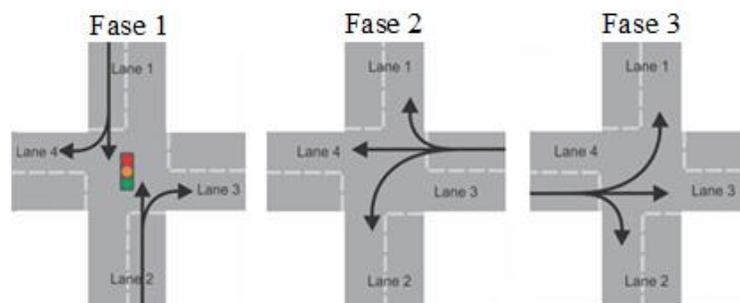


Figura 1. Ejemplo ilustrativo de una intersección con tres fases de luces de semáforo.

- Fase 1: los vehículos en los carriles L1 y L2 pueden seguir adelante o doblar a la derecha.
- Fase 2: los vehículos en el carril L3 pueden seguir adelante, doblar a la derecha o a la izquierda.
- Fase 3: los vehículos en el carril L4 pueden seguir adelante, doblar a la derecha o a la izquierda.

El número de autos que circulan por la intersección puede ser simulado en un periodo de 24 horas en el cual las tasas utilizadas varían en todo el período, siendo constantes por intervalos (ver Figura 2). Las tasas de salida son consideradas constantes, con valores para cada carril de [0,6; 0,5; 0,7; 0,7] respectivamente.

	00h–03h	03h–06h	06h–09h	09h–12h	12h–15h	15h–18h	18h–21h	21h–24h
Lane 1	0,01388	0,05555	0,00462	0,00231	0,00231	0,00231	0,00462	0,01388
Lane 2	0,00231	0,00231	0,00231	0,00231	0,00231	0,00925	0,05555	0,00462
Lane 3	0,00231	0,00231	0,00925	0,00231	0,00462	0,02777	0,22222	0,02777
Lane 4	0,00231	0,00231	0,22222	0,02777	0,00231	0,00925	0,00925	0,02777

Figura 2. Tasas de llegada a cada carril por intervalos de tiempo.

En un primer acercamiento al problema se fija un tiempo de fase constante para todo el intervalo de {30s, 30s, 30s}, se utiliza como medida la longitud media de cola por fase en cada carril, y se promedian 30 corridas de la simulación. Se observa que en los carriles 1 y 2 quedan 5 autos, y en los carriles 3 y 4, aproximadamente 30 (ver Figura 3).

En un segundo experimento, se intentó reducir el tiempo de la fase 1 para mejorar la situación promedio. Los tiempos probados fueron {10s, 30s, 30s}, como medida se mantuvo la media de la longitud de cola y la cantidad de corridas fue 30 (ver Figura 4). Se puede apreciar que los carriles 1 y 2 aumentaron su longitud de cola, pero en una cantidad casi insignificante, mientras que hay una gran mejoría en las colas de los carriles 3 y 4, que disminuyeron su longitud promedio a 12 autos.

Aunque en general los datos del cruce son aceptables, se desea disminuir aún más la longitud de las colas, principalmente en las horas de más congestión, para lo cual, continuar con un tiempo de fase constante para toda la simulación no es la mejor opción. Si se supone que el usuario necesita que en promedio no haya más de 8 autos en cola en un carril, entonces hay que hacer un cambio en la política de luces en las horas pico.

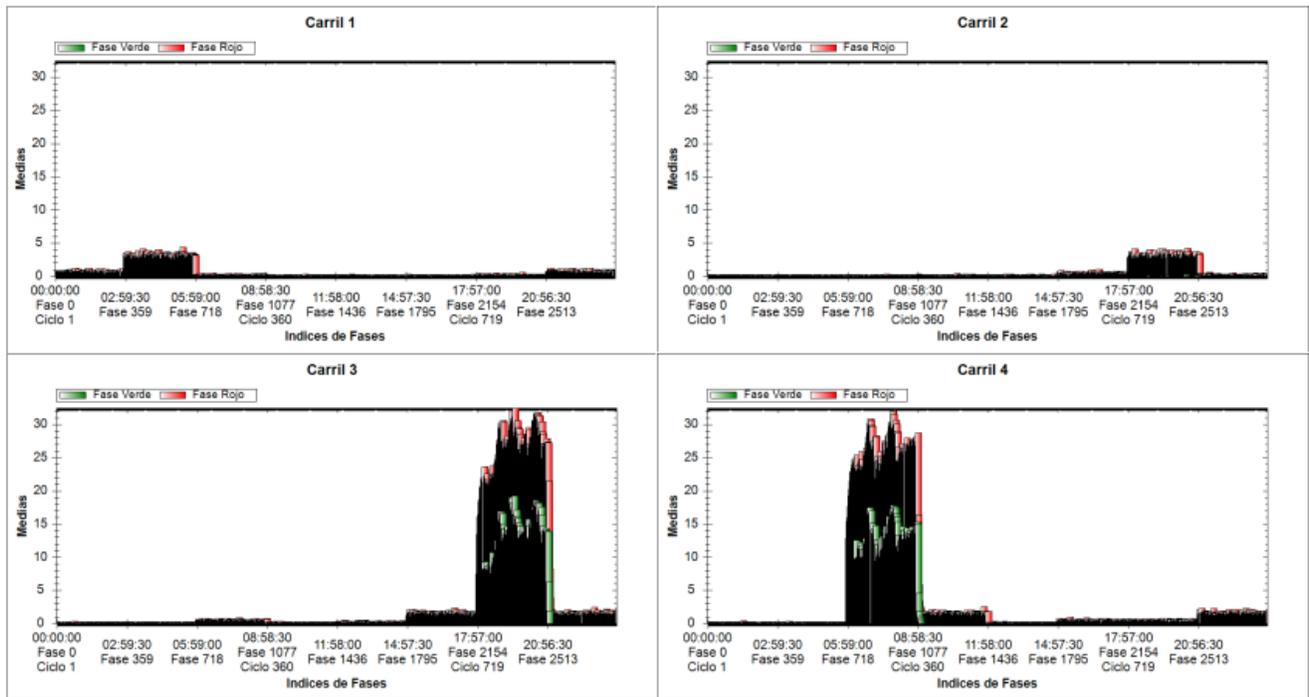


Figura 3. Comportamiento promedio del cruce para la primera configuración de ciclo.

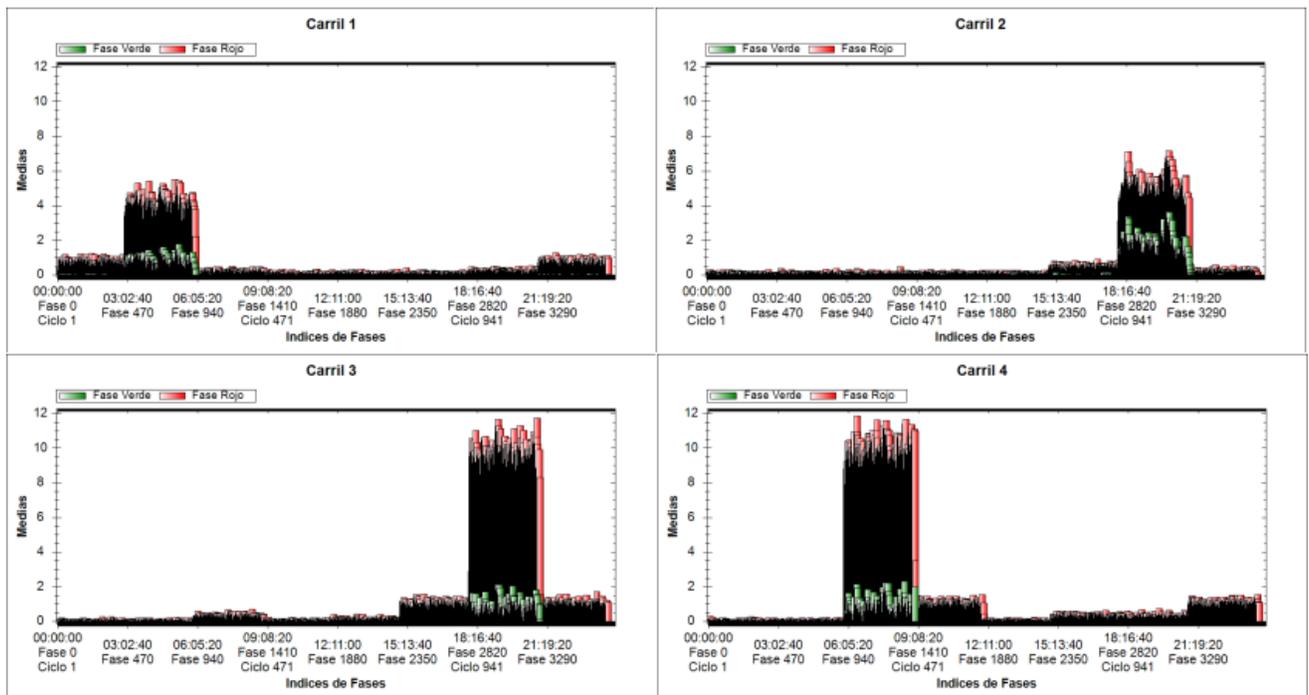


Figura 4. Comportamiento promedio del cruce para la segunda configuración de ciclo.

Para el tercer experimento, se ajustó la duración de los ciclos para que cambiaran en el tiempo como respuesta a los períodos de mayor congestión. Se experimentó utilizando {10s, 30s, 30s} de 0h a 24h, exceptuando: {10s, 10s, 35s} de 6h a 9h, y {10s, 10s, 35s} de 18h a 21h. Como medida se mantuvo la longitud media de cola por fase en cada carril, y la cantidad de corridas también se mantuvo en 30 (ver Figura 5).

Como se observa se ha mejorado notablemente el flujo de tráfico en la intersección al tener variabilidad la duración de los ciclos. Los carriles 1 y 2 permanecen estables con los nuevos cambios y el 3 y el 4 han disminuido su valor medio de 12 autos a menos de 6.

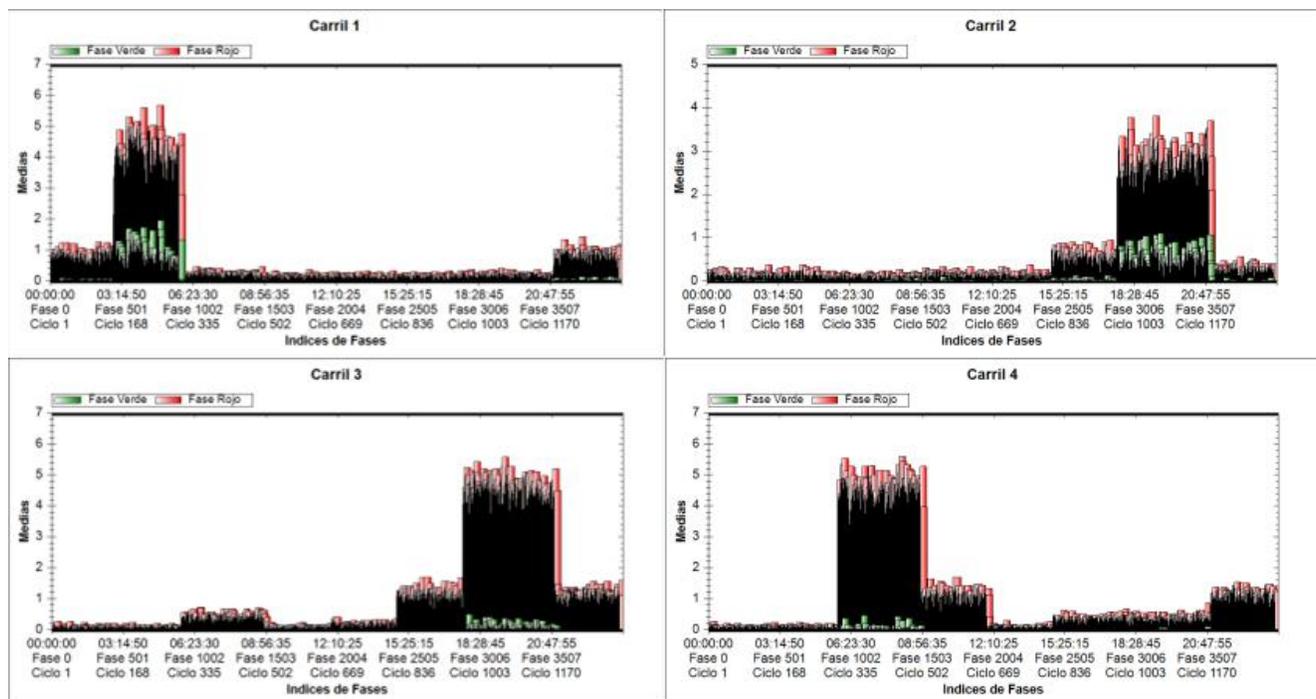


Figura 5. Comportamiento promedio del cruce para la tercera configuración de ciclo.

Para mostrar la variación del comportamiento promedio de todo el cruce durante los tres experimentos realizados, se hizo una comparación general que tomó en cuenta el promedio de las colas de todos los carriles. Hecho esto, se comprobó la importancia de tener una política de cambio de luces que se adapte a los requerimientos. La Figura 6 muestra esto:

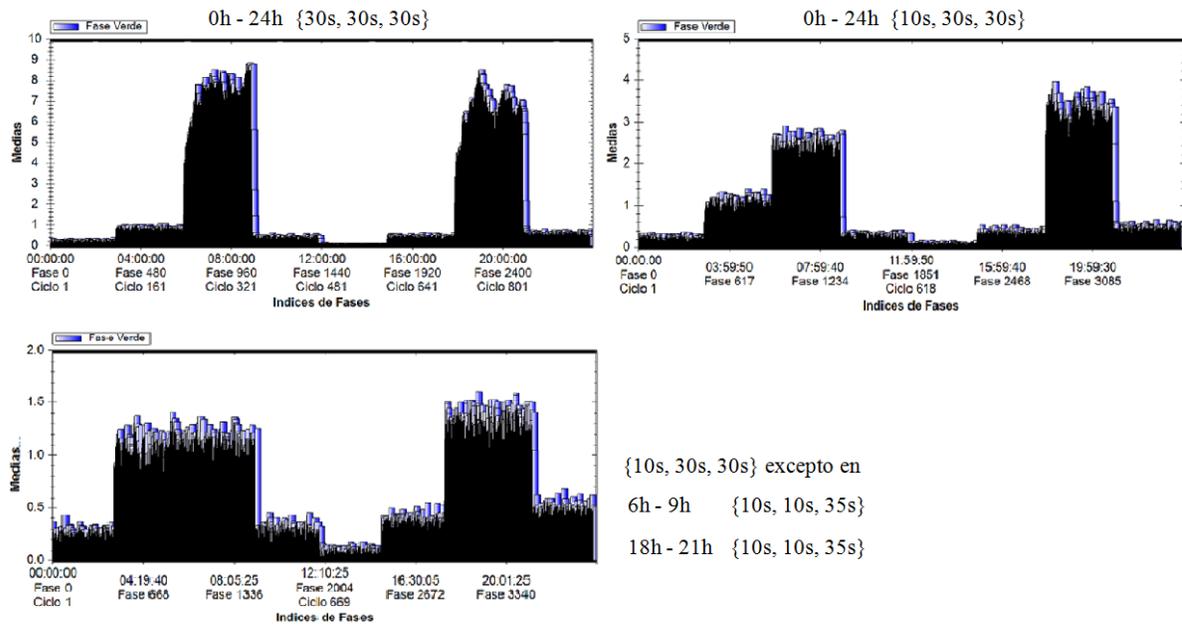


Figura 6. Comportamiento promedio del cruce en cada prueba tomando en cuenta el promedio de los resultados de todos los carriles.

## Conclusiones

La simulación concebida ayuda a visualizar las características de las intersecciones estudiadas. Con esta herramienta se puede observar las consecuencias de la variación de los parámetros tales como las tasas de entrada y salida y la duración de las fases. El simulador desarrollado permite analizar la fluidez del tráfico y comprobar la eficacia de las nuevas políticas de control de una forma barata y rápida, siendo el manejo de la herramienta muy fácil para el usuario. De hecho, sin experimentar con cada una de las políticas consideradas en la intersección, se puede estimar el comportamiento del tráfico con el fin de evitar las congestiones. Por supuesto, es necesario tener en cuenta que cuando se comparan estos resultados con situaciones de la vida real, los errores de medición pueden aparecer, produciéndose resultados poco confiables. Sin embargo, con datos más precisos, las mejoras son posibles. Las variaciones en la duración de los ciclos pueden ser obtenidas a través de otros modelos y metodologías, por ejemplo metaheurísticas, y ser validadas con esta herramienta, siendo posible además complementar el análisis de las horas pico empleando técnicas de minería de datos, gracias a la independencia modular del sistema.

## Referencias

- ALLENDE, S.; LEMA, C. S.; PEDREIRA, L. P.; BOUZA, G. Un método híbrido para resolver el problema de control óptimo de semáforos. *Rect@*, 10(1):285–298, 2009.
- CHAMPION, J. A flexible charting library for .NET. [en línea]. Code project, for those who code, 2007. [Consultado el 19 de marzo de 2014]. Disponible en <http://www.codeproject.com/Articles/5431/A-flexible-charting-library-for-NET>
- DE SCHUTTER, B.; DE MOOR, B. Optimal traffic light control for a single intersection. *European Journal of Control*, 4(3):260–273, 1998.
- DE SCHUTTER, B. Optimizing acyclic traffic signal switching sequences through an extended linear complementarity problem formulation. *European Journal of Operational Research*, 139(2):400–415, 2002.
- DODGE, Y. (ed): *The Oxford Dictionary of Statistical Terms*, Oxford University Press, 2003.
- GARCÍA, L; PÉREZ, L; MARTÍ, L. Temas de simulación. Editorial Félix Varela. 2006. p 17-27.
- HAIBO, M.; JIANNING, Y.; LINZHONG, L. Traffic signals control of urban traffic. In *7th International Conference on Fuzzy Systems and Knowledge Discovery*, 2010. p 763–767.
- KUEI-HSIANG, CH; REN-HAO, L.; MENG-HUI, W. An intelligent traffic light control based on extension neural network. In *Proceedings of the 12th International Conference on Knowledge-based Intelligent Information and Engineering Systems*, 2008. p 17-24.
- LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Prentice Hall Hispanoamérica, 1999. p 272-279.
- LARRAIN, A. I. Consideraciones para la estimación de demoras en intersecciones semaforizadas. Tesis de Maestría en Ciencias de la Ingeniería. Escuela de Ingeniería. Pontificia Universidad Católica de Chile, 2012. p 6.
- MAYO, J. *C# al descubierto*. Pearson Educación, Madrid, 2002. p 9.
- MEYER, B. *Object-Oriented Software Construction*. Prentice Hall PTR, 2002. p 40-61,823.
- MICROSOFT. Recursos para Visual Studio. [en línea]. MSDN, 2010. [Consultado el 19 de marzo de 2014]. Disponible en <http://msdn.microsoft.com/es-es/vstudio/cc136611>
- NAGATANI, T. Dispersion and scaling of fluctuating vehicles through a sequence of traffic lights. *Physica A*, 361(2):619–629, 2006.

- SÁNCHEZ, J.; GALÁN, M.; RUBIO, E. Bit level versus gene level crossover in a traffic modeling environment. In International Conference on Computational Intelligence for Modelling control and Automation (CIMCA 2005), 1:1190–1195, 2005.
- SÁNCHEZ, J.; GALÁN, M.; RUBIO, E. Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. In Proceedings of The Congress on Evolutionary Computation 2004 (CEC2004), 2:1668–1674, 2004.
- VALVERDE, L.; SANTOS, M.; LOPEZ, V. Fuzzy Decision System for Safety on Roads, Intelligent Decision Making Systems, Proceedings of the 4th International ISKE Conference, p: 326-331, 2010.