

Sistema de semaforización inteligente en la ciudad de Bogotá para mejorar los tiempos de recorrido del sistema Transmilenio*

*Franz Joseph Rogelez***

*Ignacio Molina Hernández
y Andrés Mauricio Marín
Asesores****

Recibido: 22 de abril de 2013

Aceptado: 12 de junio de 2013

Cómo citar este artículo: Rogelez, F.J. (2013). Sistema de semaforización inteligente en la ciudad de Bogotá para mejorar los tiempos de recorrido del sistema Transmilenio. *Traza*, 4 (7), 10-29.

* Este artículo es la parte del proyecto final de carrera del autor, que desarrolla una propuesta para mejorar la movilidad a través del sistema de semaforización implementando inteligencia artificial en este.

** Ingeniero de sistemas de la Universidad Piloto de Colombia, Bogotá, Colombia. con especialización en Arquitectura de Software del SENA. Analista de desarrollo en IBM de Colombia. Correo electrónico: frogeliz86@upc.edu.co, franzjrcarvajal@gmail.com

*** Universidad Piloto de Colombia, Facultad Ingeniería de Sistemas, Bogotá, Colombia. Correo electrónico: frogeliz86@upc.edu.co, ignacio-hernandez@unipiloto.edu.co, andres-marin@unipiloto.edu.co.

Nota: a menos que se indique lo contrario, todas las fotos del artículo sin del autor.

Resumen

Este artículo describe el prototipo de un sistema de semaforización inteligente que controla los tiempos de iluminación verde, roja y amarilla de los semáforos, con el objetivo de minimizar la congestión vehicular de la ciudad. El sistema aprende cómo se comporta el tráfico y sabe cómo condicionarlo con base en los algoritmos genéticos. Para esto realiza un seguimiento continuo del flujo vehicular y con esta información crea las primeras generaciones de tiempos para los semáforos. Luego revisa cuáles tiempos se adaptaron mejor al sistema, es decir, aquellos que minimizaron el tráfico, y, finalmente, a través del cruce y la mutación que provee el método de aprendizaje de los algoritmos genéticos que simulan un proceso evolutivo, controla el tráfico de la ciudad y ayuda a minimizar los tiempos de recorrido del sistema de transporte masivo Transmilenio.

Palabras clave: movilidad, semaforización inteligente, algoritmos genéticos, inteligencia artificial.



Intelligent Traffic Light System in Bogotá for Improving Travel Times in the Transmilenio System

Abstract

This paper describes the prototype of an intelligent traffic light system that controls the green, red and yellow phases of traffic lights, with the purpose of minimizing traffic congestion in the city. The system learns traffic behavior and how to condition it based on genetic algorithms. For this purpose, the system continuously monitors traffic flow and, with this information, creates the first generations of traffic light timing. It then reviews which signal timings were best adapted to the system, that is, the ones that minimized traffic, and, finally, through crossover and mutation provided by the genetic algorithm learning method, simulating an evolutionary process, controls the city's traffic and helps to minimize travel times in the Transmilenio mass transportation system.

Keywords: Mobility, intelligent traffic light system, genetic algorithms, artificial intelligence.

Introducción

El sistema de transporte público masivo de la ciudad de Bogotá (Transmilenio) y el transporte público y privado en general sufren grandes atrasos en sus tiempos de recorrido debido a la movilidad de la ciudad que es traumática. Esto se debe a un atraso de más de cincuenta años en la malla vehicular que causa diversos problemas que afectan el tráfico. Uno de ellos es el funcionamiento del sistema de semaforización, el cual es dinámico porque no se adapta de manera satisfactoria a las necesidades de la población y no tiene en cuenta el alto volumen de vehículos.

Como una propuesta para disminuir los tiempos y mejorar la movilidad del sistema Transmilenio, se planteó implementar un sistema que implique el desarrollo de inteligencia artificial en el sistema de semaforización. De manera que se creó un algoritmo inteligente capaz de controlar la duración del tiempo de cada ciclo (rojo, amarillo y verde) de los semáforos en forma autónoma, controlando así el flujo vehicular y dando prioridad a las vías con mayor congestión. Dicho algoritmo aprende continuamente de los cambios efectuados en los ciclos de los semáforos respecto al tráfico y de esta manera busca obtener los tiempos óptimos para cada uno.

El prototipo creado cuenta con un simulador de un sistema de semaforización y un sistema de tráfico para poder experimentar y determinar en qué medida mejora la movilidad al implementar algoritmos genéticos en el sistema de semaforización. Este sistema inteligente captura la longitud de cola vehicular en cada semáforo, y da prioridad a las intersecciones con mayor afluencia vehicular. Luego selecciona los ciclos más óptimos adaptados al tráfico, con el fin de realizar nuevas generaciones a partir del proceso de aprendizaje de los algoritmos genéticos. Al momento de finalizar el análisis de ciclos, son realizadas actualizaciones en cada semáforo para volver a ejecutar el proceso, que se describe en detalle a continuación.

Proceso de control inteligente

Se describe a continuación el proceso que controla la duración del tiempo del ciclo (rojo, amarillo y verde) de los semáforos.

Capturar longitud de cola vehicular

Se captura la longitud de cola vehicular, es decir, la cantidad de vehículos que tiene un semáforo en cola. Esto se realiza por medio de un sensor ubicado en cada semáforo. Actualmente se han implementado en Bogotá dos tipos de sensores: uno de presión, que es ubicado debajo del asfalto, y uno de cámara, que es ubicado en la parte superior del semáforo (figura 1). Se propone utilizar en una futura implementación el sensor de cámara, no solo por ser el más económico, sino también porque tiene un rango de visión más amplio.

El prototipo retorna por cada semáforo la cantidad real de vehículos que se encuentran en la carretera, de manera parecida a como si contara con un sensor real. Luego de recorrer todas las intersecciones y de capturar la cantidad de vehículos por cada semáforo, se organizan de mayor a menor, siendo las intersecciones prioritarias las que mayor cantidad de vehículos tengan.

De esta manera, se crea la primera generación de tiempos de los ciclos verde y rojo, y el tiempo de ciclo amarillo se ha determinado como una constante con un valor de un segundo. Esta

primera generación creada se realiza a partir de la demanda de vehículos; se establecen unos límites para esta evitar que un punto del sistema sea sobrecargado.

Función de adaptación

Con los tiempos de los ciclos que se crearon en la primera generación se calcula el impacto en el sistema, es decir, se evalúa cuáles de los tiempos generados se adaptaron al sistema de tráfico de manera positiva, cuáles ayudaron a minimizar el tráfico y cuáles contaron con el menor tiempo del ciclo. Esto se hace comparando el flujo vehicular y los tiempos de ciclo, y el resultado es guardado para las siguientes generaciones de ese semáforo. Los tiempos que no se adaptaron al ciclo son desechados, siguiendo el esquema de evolución de Darwin (Holland, 1992), y así para cada semáforo (figura 2).

Aprendizaje de máquina

Los algoritmos genéticos proveen un método de aprendizaje que simula un proceso evolutivo que es sometido a cruces y a la mutación de los individuos —en este caso los tiempos de ciclo rojo y verde de los semáforos—, además de una selección de los tiempos que mejor se adaptan al sistema de tráfico. Se toman las dos mejores opciones adaptadas de un semáforo y son cruzadas, de manera que el tiempo resultante del cruce, a su vez, es mutado; luego es actualizado en el sistema de semaforización y posteriormente es pasado por la función de adaptación para determinar si se acomoda positivamente en el sistema (figura 3). Los métodos de selección y adaptación para cada tiempo de ciclo aplicado en un semáforo determinan si continúa en el algoritmo y es cruzado o si es desechado.

De esta forma se crean nuevos individuos (tiempos de ciclo) a partir de los operadores genéticos dentro de una población y por eso el cruce y la mutación son tan importantes en el algoritmo. El cruce toma los dos individuos que mejor se adaptaron y los cruza mediante el intercambio de material genético, en este caso los bits de la cantidad de tiempo de ciclo para cada semáforo. La mutación voltea bits aleatorios dentro de la población con una pequeña probabilidad, así se genera diversidad en la población, y el sistema aprende de los tiempos de ciclo que son mejores y busca constantemente un resultado óptimo.

Se utilizó el estándar de un punto de cruce en cadenas de longitud fija para realizar el cruce, el cual está dado de manera aleatoria en la cadena de bits. Por ejemplo, cada tiempo de ciclo está representado con una cadena de ocho bits que

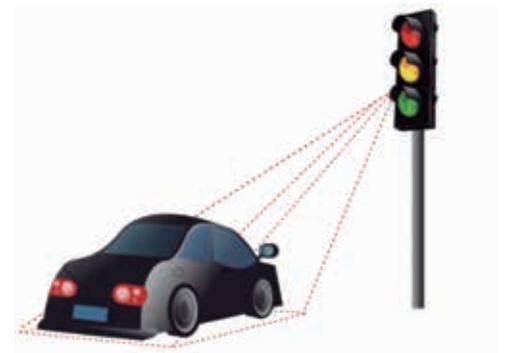


Figura 1. Sensor semafórico

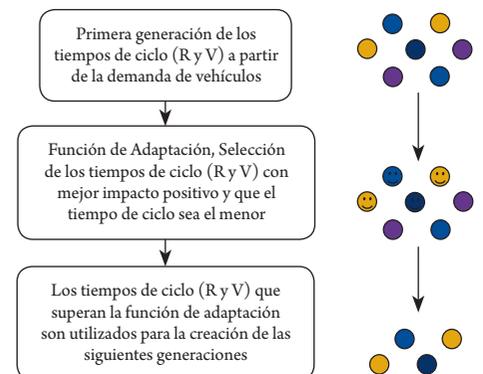


Figura 2. Generación de los tiempos y adaptación

hijos debajo de la línea. A la izquierda el punto de cruce se dio en toda la mitad de la cadena, y en la derecha el punto de cruce se dio en la tercera posición. En la mutación se puede ver que el cromosoma o gen aleatorio que muta es el número 5.

Ola verde

Si en determinada iteración el algoritmo evalúa que el sistema se encuentra en una carga de vehículos alta (figura 6), es decir, en un flujo vehicular alto de tráfico o congestión alta, es ejecutado un método denominado *ola verde*, que consiste en dar vía completa a los semáforos en verde mientras que se da espera a las vías que lo cruzan. Se determinan los tiempos de ciclo con niveles de desfase adecuados para que por lo menos la mitad de los vehículos que se encuentran en el primer semáforo puedan avanzar lo suficiente para reducir el tráfico. Las pruebas que se realizaron con el algoritmo en el simulador de tráfico determinaron que cuando el sistema se encuentra en un estado de alto tráfico y es utilizada la ola verde para minimizarlo, en promedio el 36 % de vehículos es liberado del sistema de tráfico. En la ola verde los tiempos de ciclo rojo y verde cuentan con un tiempo adicional debido al desfase, es decir, al tiempo de recorrido en que gasta un vehículo en ir de un semáforo a otro. Este tiempo está dado por la función $f(5, 10, 15, 20)$ (figura 5). El tiempo de desfase depende de la distancia entre los semáforos, aunque en el prototipo la distancia entre un semáforo y otro es constante en la troncal.

Diagrama del proceso de negocio del sistema inteligente

El proceso de negocio describe todo el sistema: la recolección de la información, la implementación de los algoritmos genéticos, la evaluación en el sistema, el recorrido de todos los semáforos y, de nuevo, la implementación de este proceso (figura 6).

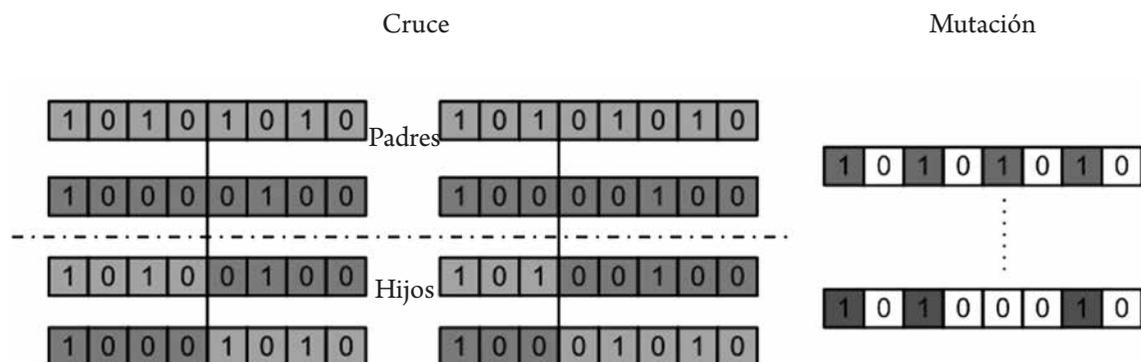


Figura 4. Cruce y mutación de algoritmos genéticos

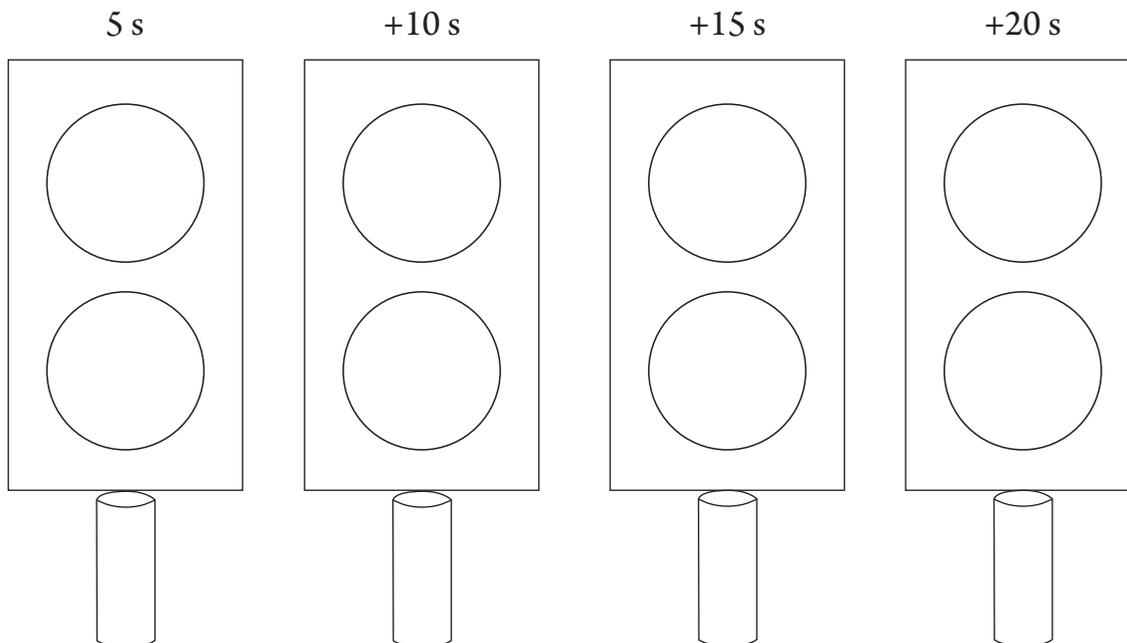


Figura 5. Desfase

En necesario tener en cuenta que este proceso se llevará a cabo para cada semáforo y depende del servidor desde donde sea ejecutado que el tiempo de respuesta sea mínimo, ya que la cantidad de semáforos en la vida real exige un equipo con buenas características físicas para poder soportar este tipo de operaciones que generan un alto consumo de *hardware*. Para solucionar esto es recomendable realizar el diseño y la implementación de forma paralela, ya que de esta manera el consumo de la máquina es mucho menor.

El proceso de negocio pasa por varias etapas: unas que van por la capa de presentación del simulador y luego por la capa de negocio, donde se encuentra el algoritmo inteligente. La mayoría de los procesos que se pueden ver ya fueron descritos anteriormente en el artículo, de manera que, resumiendo, el proceso comienza en la recolección de información de la cantidad de vehículos por cada semáforo; luego organiza estos tiempos de manera que los semáforos que tengan mayor congestión se encuentren primero; si ya se crearon las primeras generaciones de los tiempos por demanda entonces revisa la función de adaptación para cada tiempo y así determina el

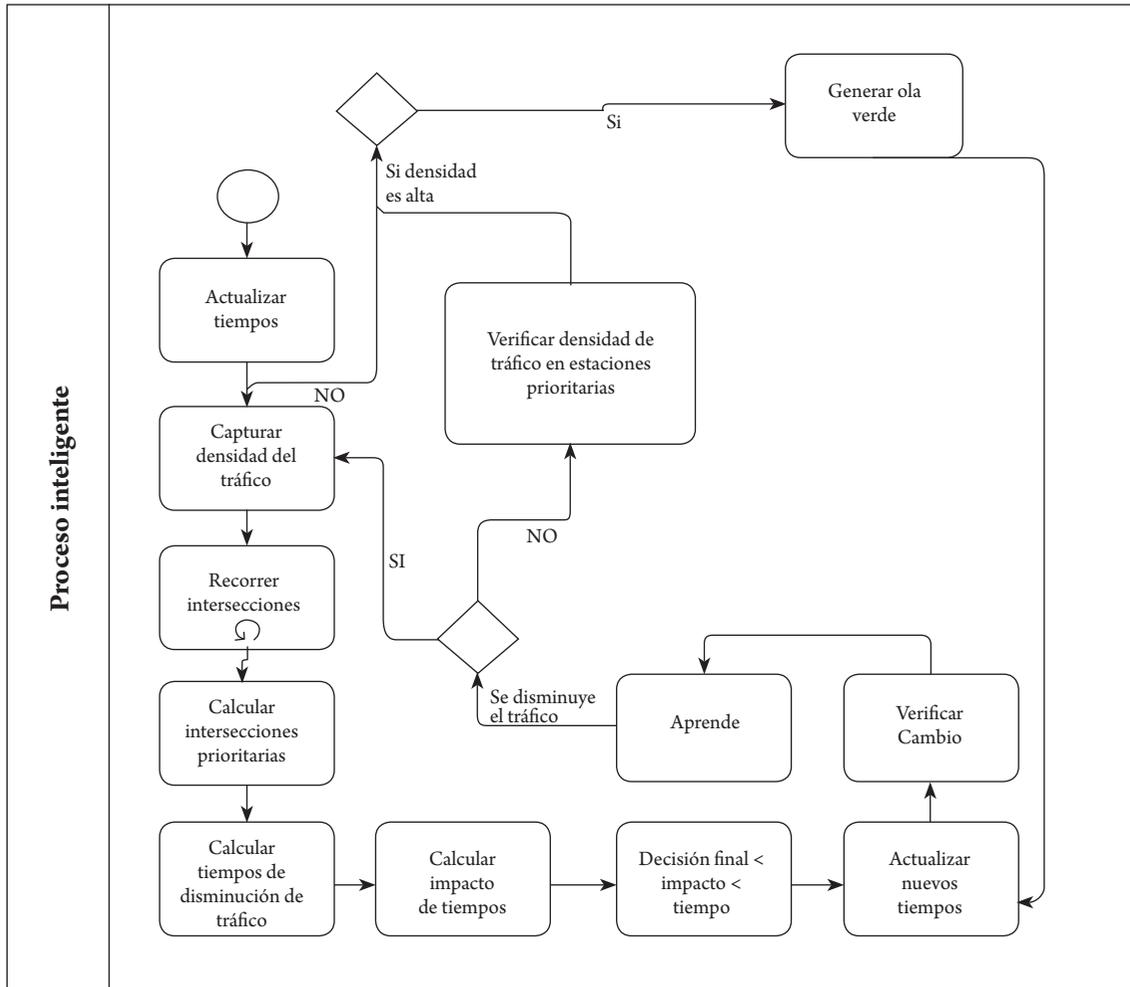


Figura 6. Proceso de negocio

impacto en el sistema; luego de cruzar y mutar los tiempos, y de aprender de aquellos que mejor se adaptaron, toma los resultantes y los actualiza en el simulador; y, por último, comienza a iterar de nuevo el proceso de manera que siempre está en busca del resultado óptimo. La mayoría de los algoritmos genéticos cuenta con una función de parada, es decir, donde el algoritmo determina que debe dejar de iterar. Para este problema de cambio constante no se puede determinar un estado de parada; por ello el proceso de negocio y el algoritmo genético no cuentan con esta condición.

Simulador de tráfico

Como desarrollo final de este proyecto se realizó un simulador de tráfico para probar que el algoritmo inteligente que controlará el sistema de semaforización cumple con su cometido de minimizar los tiempos de recorrido del sistema Transmilenio. Así, se deja un marco de estudio en el que se puede pensar más adelante para implementar en la ciudad y con esto disfrutar de los resultados de esta investigación o de futuras investigaciones que pueden tomar como punto de partida la presente.

El simulador construido se hizo a partir del simulador de Martin Treiber (Germ, Kesting, Budden y Treiber, 2000), el cual utiliza un modelo inteligente de conducción (IDM, Intelligent-Driver-Model). Este modelo cuenta con un sistema inteligente que implementa autómatas celulares para cada vehículo que actúa como un ente independiente y, por lo tanto, decide su aceleración o desaceleración dependiendo de los factores del ambiente que lo rodean, que es muy parecido al ambiente real. Este modelo es explicado en detalle más adelante.

Alcance del simulador

El alcance actual está determinado por una vía principal en la ciudad de Bogotá y las calles que la cruzan. La vía principal es la troncal Caracas, entre calle 39, calle 45, calle 53, calle 57 y calle 63. Los semáforos controlados serán los dispuestos en estas intersecciones en ambos sentidos, teniendo en cuenta cómo operan en la vida real los semáforos de estas calles, es decir, en el sentido de sur-norte, norte-sur, oriente-occidente y occidente-oriente.

Los vehículos en el simulador están diferenciados de los vehículos pertenecientes al sistema Transmilenio. Los articulados son de tamaño largo y color rojo, y los vehículos particulares son de tamaño corto y color azul. Pese a que en la troncal en la ciudad de Bogotá también transitan vehículos particulares, al ser el objetivo de este proyecto el minimizar los tiempos de recorrido del sistema Transmilenio, el alcance determinado en el prototipo de los vehículos que transitan por la troncal serán únicamente los vehículos del sistema Transmilenio (los articulados), y solo vehículos particulares por las calles que lo cruzan.

Interfaz de usuario

Se muestra el simulador que detalla el mapa de las calles y los semáforos de las intersecciones. Se puede observar que cada semáforo tiene un *id* único que lo identifica y lo ubica en el mapa de semáforos (figura 7).

En la figura 8 de la interfaz del usuario se puede ver que las calles y la troncal están diferenciadas por sus nombres, que a su vez están conformados por el nombre de la calle y el sentido de orientación para que de esta manera sea fácil ubicarse en el plano real de la ciudad de Bogotá.

Modelo de tráfico longitudinal: el IDM

El IDM es un “auto siguiendo el modelo” (Treiber, 2011); es decir, el estado del tráfico en un momento dado se caracteriza por las posiciones, las velocidades y el índice de carril de los vehículos. La decisión de cualquier conductor para acelerar o frenar depende solo de su propia velocidad, de la del vehículo del frente y de la del inmediatamente atrás de él. Las decisiones de cambio de carril, sin embargo, dependen de todos los vehículos vecinos. En concreto, la

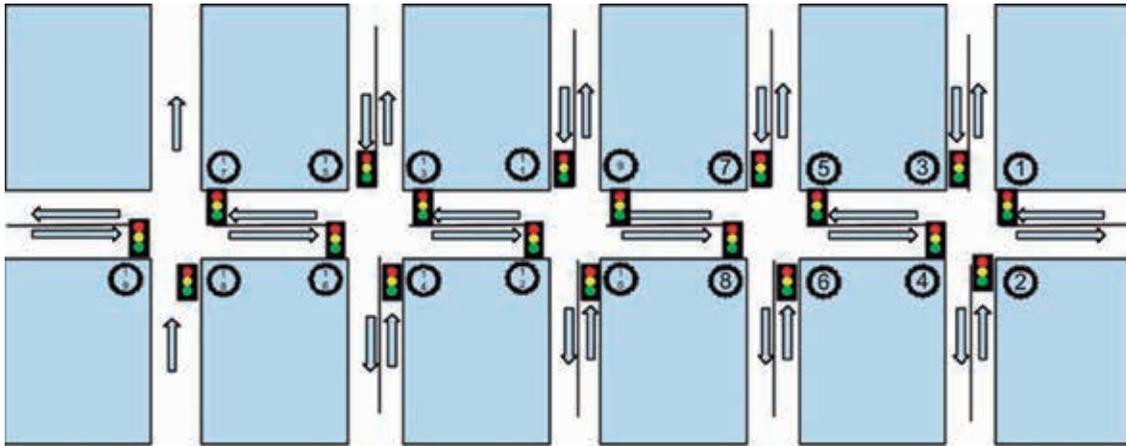


Figura 7. Esquema inicial de la interfaz del sistema inteligente de semaforización

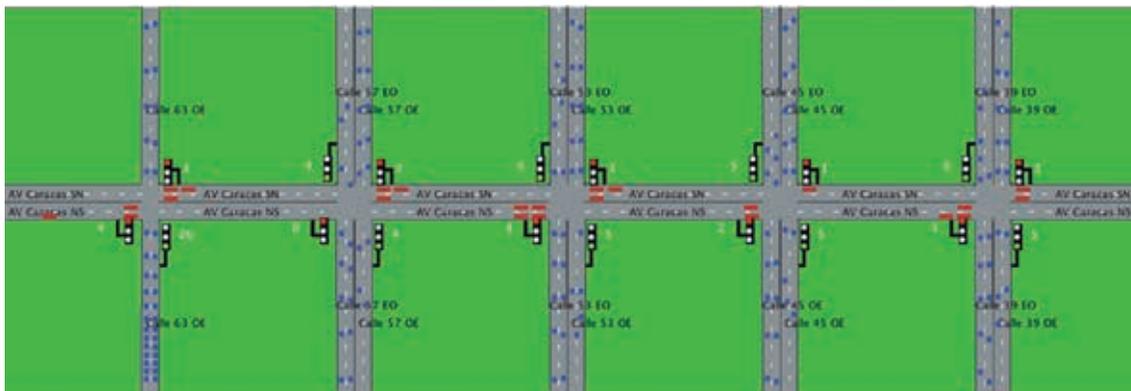


Figura 8. Interfaz del sistema inteligente de semaforización

aceleración dv/dt de un conductor determinado depende de su velocidad v , s en la distancia con el vehículo delantero, y de la velocidad v Delta diferencia (positiva al acercarse),

$$\frac{dv}{dt} = a \left[1 - \left(\frac{v}{v_0} \right) \delta - \left(\frac{s^*}{s} \right)^2 \right] \quad (1)$$

Donde

$$s^* = s_0 + \min \left[0, \left(vT + \frac{v\Delta v}{2\sqrt{ab}} \right) \right] \quad (2)$$

La aceleración se divide en una “deseada aceleración” a (3) en el camino libre, y reduce la velocidad de frenado inducida por el vehículo del frente.

$$\left[1 - \left(\frac{v}{v_0} \right) \text{delta} \right] \quad (3)$$

La aceleración en el camino libre disminuye a partir de la aceleración inicial a cero al acercarse a la “velocidad deseada” v_0 .

El término de frenado se basa en una comparación entre la “distancia deseada dinámica” (2) y la real diferencia con el vehículo precedente. Si la brecha real es aproximadamente igual a la (2), entonces la desaceleración rompe tan esencialmente que compensa la parte de aceleración en vacío, así que la aceleración resultante es casi cero. Esto significa que (2), corresponde a la brecha al seguir a otros vehículos en constante flujo de tráfico. Además, (2) aumenta dinámicamente cuando se aproxima a los vehículos más rápidos y disminuye cuando el vehículo delantero es más lento. Como consecuencia de ello la desaceleración del impulso se incrementa con la disminución de la distancia al vehículo delantero (se quiere mantener una cierta “distancia de seguridad”), se incrementa la velocidad propia (que aumenta la distancia de seguridad), y aumenta la diferencia de velocidad en el vehículo delantero (cuando se aproxima el vehículo frente a una tasa muy alta puede ocurrir una situación peligrosa) (Treiber, 2011).

Resultados del experimento

Para comenzar este experimento se tomó un conjunto de datos reales extraídos de las estadísticas del tráfico de la ciudad de Bogotá (Hernández, 2012; DANE, 2012a; DANE, 2012b), los cuales se convirtieron en los datos de entrada para crear el tráfico simulado. Con esto fue posible disponer de unos resultados más cercanos a la realidad que fueron utilizados para los dos escenarios de prueba y comparación.

El tráfico que se creó a partir de los datos encontrados en las estadísticas de la ciudad es el conjunto de valores iniciales que determina la generación de números pseudoaleatorios que corresponden al flujo vehicular generado para el prototipo. Esta generación de números pseudoaleatorios es creada utilizando un algoritmo establecido por el estadista Dagoberto Saboyá con la información recolectada, que es utilizada como semilla para generar el nuevo tráfico.

En el primer escenario los tiempos de los ciclos rojo y verde de los semáforos son los mismos que tienen los semáforos en la ciudad de Bogotá, y que fueron recogidos a través de una

investigación que se realizó para saber el funcionamiento del sistema de semaforización real de la ciudad. Esta investigación se encuentra en detalle en el documento completo de este proyecto (Rogez, 2012, p. 15). En el segundo escenario los tiempos de los ciclos rojo y verde son los tiempos que el algoritmo genético asigna a cada semáforo. Ambos escenarios de prueba son comparados para visualizar en qué medida la implementación de algoritmos genéticos en el sistema de semaforización mejora la movilidad respecto al funcionamiento del sistema actual.

Información base

La información que fue recolectada sobre el tránsito de articulados del sistema Transmilenio se describe a continuación (tabla 1). Y en la tabla 2 se agrega la velocidad media de estos vehículos.

Tabla 1. Promedio de articulados por día y por mes en la ciudad

Troncal	Promedio mensual de vehículos vinculados			Promedio diario de vehículos en servicio		
	2010	2011	2012	2010	2011	2012
	1176	1269	1291	1114	1240	1252

Flota troncal disponible vinculada al 30 de abril de 2012: 1291 buses.

Fuente: DANE.

La cantidad de vehículos articulados con la que cuenta Transmilenio y los vehículos articulados que utiliza en promedio para sus actividades son representados en las figuras 9 y 10, respectivamente.

Resultados del algoritmo creado GPAN

En veintiún días de observación en el tiempo del simulador se realizó un promedio de la cantidad de vehículos generados por el algoritmo creado Generados de Números Seudoaleatorios (GPAN). Estos datos generados son solo una parte, ya que el alcance del proyecto toma únicamente una fracción de la troncal del sistema Transmilenio, por ello los datos generados son un porcentaje menor a los datos recolectados.

Resultados de los escenarios de pruebas

La tabla 2 describe la comparación entre las velocidades máximas promedio alcanzadas por los dos escenarios de pruebas.

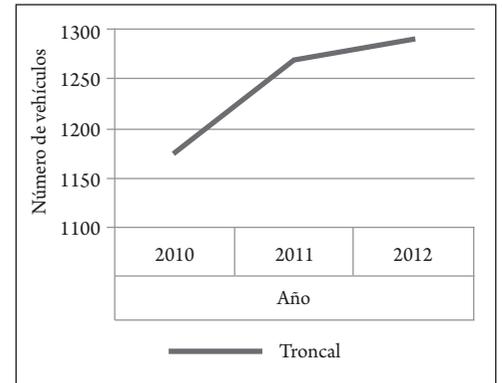


Figura 9. Promedio mensual de vehículos vinculados, 2010 a 2012

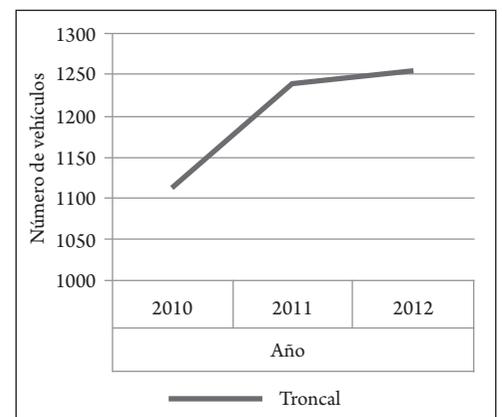


Figura 10. Promedio mensual de vehículos en servicio, 2010 a 2012

Franz Joseph Rogelez

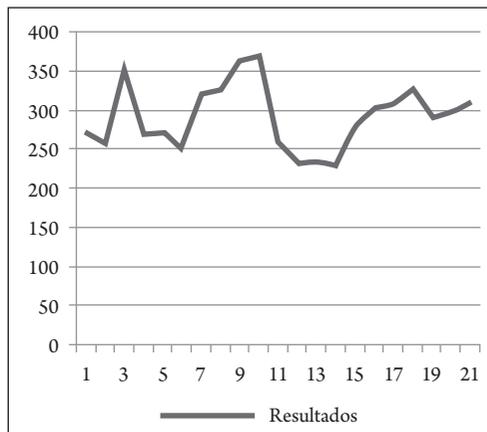


Figura 11. Flujo vehicular creado (actuado) para las calles 39 a 63
Fuente: Rogelez (2012, p. 99).

Tabla 2. Velocidad promedio: datos simulados y datos reales

Velocidad	V	Medida
Velocidad promedio flota troncal real	26,08	km/h
Velocidad máxima promedio flota troncal simulada	63,09	km/h
Velocidad promedio flota troncal de la simulación con AG	36,03	km/h
Velocidad máxima promedio flota troncal simulación con AG	68,06	km/h

Velocidad promedio de la flota troncal en abril de 2012: 26,08 km/h.

Fuente: Rogelez (2012, p. 15).

Como se puede ver en la figura 12, hubo un aumento de la velocidad vehicular en el experimento respecto a la velocidad real.

La tabla 3 muestra la comparación de la longitud de cola vehicular capturada por cada semáforo en los dos escenarios de prueba. Esta información que se muestra a continuación es el promedio del conjunto total de los datos recolectados en el experimento.

Tabla 3. Longitud de cola vehicular promedio simulada. Respuesta de los semáforos con los tiempos de los ciclos reales y los ciclos que genera el algoritmo genético

ID semáforo	Longitud cola simulación	Longitud cola algoritmos genéticos
1	10	6
2	15	16
3	31	31
4	15	10
5	13	12
6	15	10
7	13	12
8	10	6
9	9	5
10	6	3
11	5	3
12	10	6
13	15	16
14	13	15
15	10	8
16	40	32
17	13	10
18	12	8
19	6	6

Fuente: Rogelez (2012, p. 103).

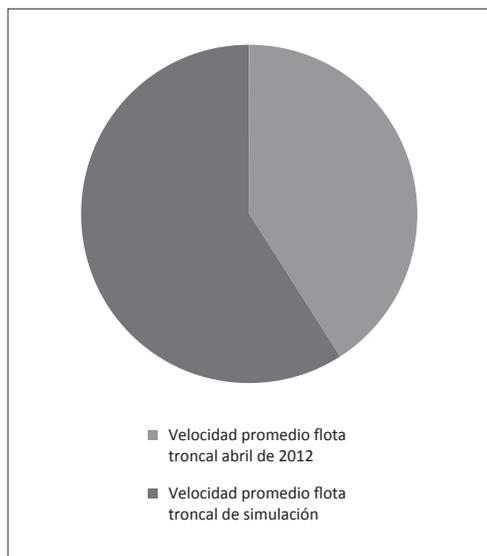


Figura 12. Comparación de la velocidad promedio real y la simulada
Fuente: Rogelez (2012, p. 104).

Se puede ver la mejora de la implementación de los algoritmos genéticos frente al actual funcionamiento del sistema de semaforización de la ciudad de Bogotá así: un 17 % de disminución de la longitud de cola vehicular aproximadamente, una mejora en la velocidad promedio de un 38,1 % y en la velocidad máxima de un 7,8 % aproximadamente (figura 12).

Funcionamiento del sistema de semaforización de la ciudad actual en los cambios de tiempos de ciclo de los semáforos

El sistema de semaforización vial de la ciudad de Bogotá cuenta actualmente con 1183 intersecciones que están manejadas por 973 equipos de control marca Siemens, de los cuales el 49,3 % corresponde a tecnología antigua y el 50,7 %, a tecnología de punta. Dependiendo de la intersección el semáforo tiene un aumento o una disminución en el ciclo rojo, amarillo o verde. En general el sistema actual cambia los ciclos de la siguiente manera: los ciclos varían los tiempos del semáforo en rojo y verde en dos periodos diarios (mañana-tarde) y en tres periodos semanales (lunes-jueves; viernes; sábado-domingo). De esta

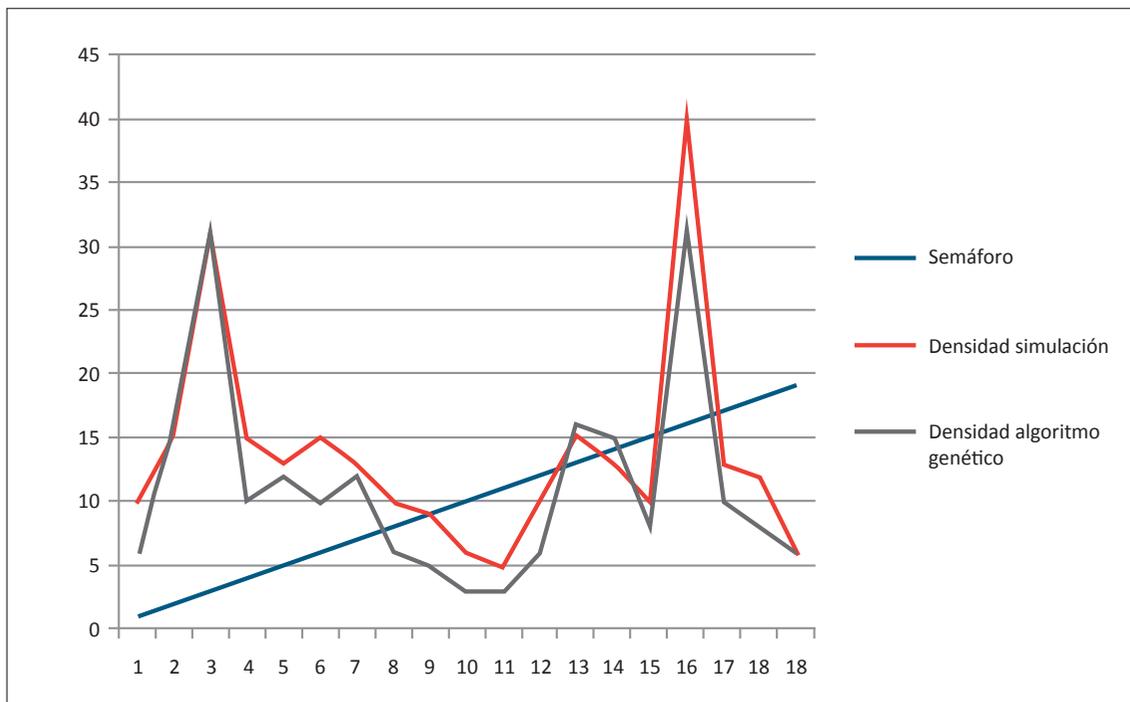


Figura 13. Comparación del promedio de la longitud de cola vehicular de los ciclos de los semáforos reales y los ciclos de los semáforos generados por el algoritmo genético

Fuente: Rogeiz (2012, p. 104).

manera todo el año se realizan cambios cada tres, seis o doce meses —aunque dependen eventualmente de estudios del tráfico— en las programaciones de los tiempos de rojo y verde.

Problemas de implementación

Al tratar de solucionar los problemas del tráfico de una ciudad, en general se habla de implementar un modelo de optimización con un conjunto de parámetros y variables, como el flujo vehicular, la distancia entre semáforos, los tiempos de duración de los ciclos de los semáforos, y de un objetivo específico, como la cantidad de paradas de los vehículos, la cantidad de gasolina gastada, el tiempo total de espera en el semáforo. En este caso, y por el alcance dado inicialmente en el proyecto, se intenta mejorar el tiempo de recorrido de los articulados del sistema Transmilenio y minimizar el tiempo de espera en los semáforos. El problema en sí es bastante denso, ya que interfieren complejas variables en el sistema y unas son dependientes de las otras. Por ejemplo, si solo se da tiempo de ciclo rojo y verde a los semáforos por demanda de vehículos, significa que a mayor demanda en un semáforo más tiempo tendría en el ciclo verde y eso causaría congestión en las vías contiguas. Con esto podría ocurrir un colapso, ya que se sobrecarga un punto en el sistema de tráfico y en el de semaforización. Sin embargo, la implementación del algoritmo genético determina entonces cuáles tiempos de ciclo rojo y verde son los más indicados para ser asignados a cada semáforo, y así se elimina esta situación.

Otro ejemplo es el que tiene que ver con el tiempo del ciclo y el desfase, que son variables dependientes, lo que hace que al modificar una se tenga que modificar la otra. Por ejemplo, el sistema tiene que tener en cuenta que el tiempo rojo en un semáforo debe contar como el verde en el semáforo contiguo, es decir, que además de los vehículos en cola en el propio semáforo debe estar atento de los semáforos contiguos en la intersección. Eso en el sistema real es dado por una caja de control que no permite que en una intersección todos los semáforos estén en el ciclo verde o en el ciclo rojo al mismo tiempo.

Por otro lado, en la simulación se estiman variables óptimas. Por ejemplo, el modelo inteligente de conducción elimina los accidentes ya que respeta la distancia mínima de seguridad entre un vehículo y otro, y mantiene una velocidad mesurada al alcanzar un vehículo lento para evitar acciones de choque. En la vida real esta situación no se da todas las veces; la mayoría de los accidentes de tráfico ocurre por estas circunstancias.

Trabajo futuro

Este trabajo puede ser evaluado para ser implementado en la ciudad de Bogotá. Según los experimentos realizados la implementación mejoraría la movilidad en un porcentaje óptimo, aunque estos datos pueden cambiar a favor al aumentar las horas de prueba. Si se desea implementar, se tienen que tener en cuenta aspectos mencionados en el punto “problemas de implementación”.

En la implementación real pueden ser creadas futuras investigaciones de cómo el sensor captura la información y de cómo hacerla aun más valiosa capturando mayor cantidad de detalles. Por el alcance del proyecto el sensor solo devuelve la cantidad de vehículos, pero sin duda un sensor

real podrá capturar la velocidad de los vehículos y utilizar esta información para alimentar el algoritmo genético.

Con la información captada se podrían realizar unas proyecciones del flujo vehicular del tráfico. El modelo quedaría completo si se tomaran los procesos estocásticos para realizar esta tarea; pero solo el hecho de comprender el tema tomaría entre uno o dos semestres más, así que es bueno tener en cuenta este ítem como la continuación de un proyecto, o como una opción de trabajo para la maestría y, por qué no, para el doctorado.

También puede ser mejorado el algoritmo genético. Por ejemplo, en la investigación se pudo observar que el uso de un operador de cruce uniforme puede ser más eficiente en otros contextos que el cruce en un punto. Si se probara de esta manera, podría determinar en qué medida mejora el hecho de cambiar el punto de cruce.

Aunque en esta investigación se utilizan algoritmos genéticos por ser un método de optimización y por ser un método evolutivo que se adapta particularmente a la idea de mejoramiento continuo al sistema de semaforización, no quiere decir que sea el único método inteligente a utilizar para la optimización de un sistema de semaforización. Por ello esta investigación puede ser tomada como punto de partida para implementar otros métodos inteligentes.

Conclusiones

Este prototipo presenta una alternativa para la mejora de la movilidad, aunque para ello se requiere de una cantidad de información real importante que está en manos de Transmilenio y del Sistema Integrado de Transporte Público (SITP), para que los datos simulados se aproximen más a la realidad.

Es importante destacar que gracias al simulador de tráfico *movsim* (Germ, Kesting, Budden y Treiber, 2000) fue posible realizar el experimento, el cual se utilizó para probar el algoritmo genético con un tráfico simulado. En el marco científico el *software* libre contribuye a la expansión del conocimiento, y esto ayudó para poder comprobar las teorías y los objetivos planteados en el inicio de este proyecto.

Los resultados del algoritmo genético, aunque en un principio no parecieron prometedores, demostraron que el experimento mejoraría continuamente el tráfico. Se presentó un 17 % de disminución de la longitud de cola vehicular, hubo una mejora de 38,1 % en la velocidad promedio y de 7,8 % en la velocidad máxima promedio, aproximadamente. Esto llevaría a mejorar la movilidad tanto para los articulados del sistema Transmilenio como para los automóviles particulares. También es importante destacar que estos porcentajes podrían ser mucho mayores por la cantidad de iteraciones del algoritmo genético; mientras más veces sea iterado el proceso de negocio, el resultado será mucho más cercano a un resultado óptimo.

El equipo utilizado en las pruebas del experimento tuvo problemas de rendimiento, por ello es necesario contar con un buen equipo para realizar mejores pruebas que exigen más recursos de *hardware*. Además, las pruebas realizadas requieren de mucho más tiempo para tomar buenos resultados respecto a algoritmo genético.

Aunque la cantidad de vehículos del sistema Transmilenio en las horas pico parece baja, es importante destacar que se hay que tener en cuenta ciertos factores para cumplir la demanda. Por ejemplo, se debe mantener una distancia de seguridad entre articulados para evitar acciones de choque, por eso la cantidad de buses no puede superar un tope, aun en las horas pico.

Los conocimientos adquiridos durante la carrera proporcionan la manera adecuada de abordar los problemas a los que es enfrentado un ingeniero de sistemas. Las etapas de análisis y diseño son mucho más importantes que la implementación, ya que proporcionan las bases adecuadas para realizar un buen proyecto de *software*, y al momento de llegar a la implementación son mucho menos las preguntas que tiene que hacer el desarrollador sobre cómo abordar el problema.

Las buenas prácticas de desarrollo facilitan la implementación del código y lo hacen mucho más rápido; de igual manera que al momento de llegar al mantenimiento del *software* se hace más sencillo de abordar. La implementación estadística en un proyecto de este alcance facilita la muestra de resultados y da la confianza de que estos resultados estén cercanos a la realidad.

Epílogo

Ignacio Hernández Molina
Asesor de Investigación del Proyecto
Profesor Universidad Piloto de Colombia

En general las universidades tienen dentro de su razón de ser un compromiso con la investigación. En términos kantianos es un imperativo categórico el tener que fomentar los procesos investigativos, bien sea con la conformación de semilleros de investigación, con el desarrollo de equipos profesionales de investigación, conformados por sus docentes y aglutinados alrededor de los centros de investigación, o a través de lo que algunos desafortunadamente siguen considerando anacrónicas “tesis de grado” o proyectos de grado, bajo la modalidad de seminarios de investigación, o talleres de investigación; en realidad el nombre es lo de menos, acá lo importante es la teleología que se persigue en la universidad, como centro productor de conocimientos.

Desde esta última perspectiva, la de generar —como dice Gaston Bachelard— “un espíritu científico” en el pregrado, en donde surgen inquietudes bastante interesantes y complejas por parte de los estudiantes, para contribuir a la transformación de la realidad social —de por sí ya bastante dinámica—, producto, entre otras cosas, del permanente avance en el desarrollo de las tecnologías y, para el caso que nos ocupa, de tecnologías de la información y la comunicación.

Bajo estos criterios, de familiarizar al estudiante con el proceso de investigación, primero incentivando la imaginación, luego enseñándoles a abstraer los problemas de la realidad, y luego poniéndolos en contexto, surgió la idea en el Taller de Investigación del Programa de Ingeniería de Sistemas de la Universidad Piloto de encontrar una solución al problema de la semaforización en las intersecciones que desembocan en la troncal de Transmilenio, entre las calles 39 a 63. Esto con el fin de priorizar el paso de los articulados, y que los autos particulares

no quedaran invadiendo las intersecciones en el cambio semafórico de rojo, amarillo y verde. Fue así como surgió el proyecto de grado “Prototipo de un sistema de semaforización inteligente en la ciudad de Bogotá para mejorar los tiempos de recorrido del sistema Transmilenio”, el cual obtuvo mención de grado meritoria en la sustentación de su proyecto.

Bajo este incentivo el trabajo fue presentado al concurso latinoamericano de proyectos de cómputo “Proyecto Multimedia”, edición 2013, versión Colombia, realizado en la Universidad Minuto de Dios de Bogotá. Además fue acreditado para representar a Latinoamérica en la muestra científica juvenil, en Trujillo, Perú del 30 de octubre al 1 de noviembre de 2013. Finalmente, la Sociedad Latinoamericana de Ciencia y Tecnología Aplicada le otorgó reconocimiento a Franz Joseph Rogelez Carvajal por haber obtenido medalla de oro y primer puesto en dicho concurso.

Esto demuestra que es más mito que realidad el hecho de que las tesis se guarden en los anaqueles de las universidades. Por el contrario, una salida que nos queda para mover el motor del desarrollo es la investigación científica; incentivar en los estudiantes la importancia de la investigación y las satisfacciones que genera como valor agregado los puede convertir en las generaciones de reemplazo de la comunidad científica nacional.

Agradecimientos

El autor agradece a Klaus Banse, a Sebastián Thrun, a Peter Norvig, a Ralph Germ, a Arne Kesting, a Martin Budden, a Martin Treiber, a Dagoberto Saboyá, a Erik Schwartz, a Ignacio Hernández, a Gilberto Pedraza, a Fredy Pérez, a Giovanni Fajardo, a Ibo Cerra, a Fidel Barbosa y a Andrés Marín por sus contribuciones al proyecto.

Referencias

- Azarang, R. y García, E. (1996). *Simulación y análisis de modelos estocásticos* (1.ª ed.). México: McGraw Hill.
- Baek, T., Hoffmeister, F., y Schwefel, H. A. (1991). *Survey of evolution strategies. Proceedings of the Fourth International Conference on Genetic Algorithms*. La Jolla, CA: Morgan Kaufmann.
- Banse, K. (2011). *Protocolos Its para sistemas de semaforización*. Recuperado de <http://www.sit-colombia.com/download/presentaciones/201109%2029-%20co%20pso%20avante%20protocolos.pdf>
- DANE (2012a). *Estadística movimiento Transmilenio*. Recuperado de: http://www.dane.gov.co/files/investigaciones/boletines/transporte/anexos_trans_IITrim11.xls 2011
- DANE (2012b). *Distribución Transmilenio*. Recuperado de http://www.dane.gov.co/files/investigaciones/boletines/transporte/bolet_trans_IITrim11.pdf 2011

- DiCyt (2011). *Información sobre el tráfico y el sistema de semaforización de la ciudad de Bogotá*. Recuperado de <http://www.dicyt.com/noticias/bogota-estudia-la-implantacion-de-sistemas-inteligentes-de-regulacion-del-trafico>
- García, R. (2009). *Utilización de algoritmos genéticos en sistemas inteligentes autónomos*. Recuperado de <http://iidia.com.ar/rgm/tesistas/steinhilber-tesis-de-grado-en-sistemas-de-informacion.pdf>
- Germ, R., Kesting, A., Budden, M. y Treiber, M. (2011). *Información sobre el simulador*. Recuperado de <http://www.h2063376.stratoserver.net/movsim/#about>
- Guerra Hernández, A. (2004). *Aprendizaje automático: algoritmos genéticos*. Xalapa: Universidad Veracruzana.
- Hernández, I. (2002). *La investigación científica. Un camino a la imaginación*. Bogotá: Universidad Piloto de Colombia.
- Hernández, I. (2012). *La formulación de proyectos. Ciencias e ingenierías*. Bogotá: Universidad Piloto de Colombia.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. Michigan: University of Michigan Press.
- IBM (2010). *Congestión en el tráfico. Encuestas*. Recuperado de http://www.ibm.com/smarterplanet/co/es/traffic_congestion/ideas/index.html. 2010
- IBM (2011a). *The Globalization of Traffic Congestion*. Recuperado de <http://goo.gl/mMLNB>
- IBM (2011b). *Tráfico inteligente*. Recuperado de http://www.ibm.com/smarterplanet/us/en/traffic_congestion/ideas/
- IBM (2011c). *Sensores de flujo vehicular*. Recuperado de <http://www.ibm.com/podcasts/howit-works-/040207/index.shtml>
- Jong, K. A., Spears, W. M. y Gordon, D. F. (1993). *Using Genetic Algorithms for Concept Learning*. Boston: Springer.
- Laurence, A. (1995). *Adapting Operator Probabilities in Genetic Algorithms*. San Francisco: Morgan Kaufmann Publishers In.
- Lba, E. (1999). *Análisis y diseño de algoritmos genéticos paralelos distribuidos*. Málaga: Universidad de Málaga.
- Palmisano, S. J. (mayo de 2010). *Intelligent Transportation Society of America. Annual Meeting & Conference, Houston, Texas*. Recuperado de en: http://www.ibm.com/smarterplanet/us/en/transportation_systems/article/palmisano_itsa_speech.html
- Rogelez, F. (2012). *Prototipo de un sistema de semaforización inteligente en la ciudad de Bogotá para mejorar los tiempos de recorrido del sistema Transmilenio*. Bogotá: Universidad Piloto de Colombia.

- Spears, W. y Anand, V. A. (1991). *Study of Crossover Operators in Genetic Programming*. Cambridge, Washington: Springer.
- Thrun, S. y Norvig, P. (2011). *Artificial Intelligence, Stanford Engineering*. Recuperado de <https://www.ai-class.com/>
- Transmilenio (2012). *Estadísticas generales Transmilenio*. Recuperado de http://www.transmilenio.gov.co/WebSite/Contenido.aspx?ID=TransmilenioSA_TransmilenioEnCifras_EstadisticasGenerales
- Treiber, M., Hennecke A. y Helbing D. (2000). *Congested Traffic States in Empirical Observations and Microscopic Simulations*. Stuttgart: University of Stuttgart.
- Treiber, M. (13 de marzo de 2011). *IDM Información*. Recuperado de <http://www.vwi.tu-dresden.de/~treiber/MicroApple-t/IDM.html>