

COMPARACIÓN DE ESTRATEGIAS DE CONTROL DE FLUJO DE
TRÁFICO UTILIZANDO UN MARCO DE CO-SIMULACIÓN



GERMAN ANDRES INSUASTY DELGADO
GABRIEL CORAL CAICEDO



UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO

2019

**COMPARACIÓN DE ESTRATEGIAS DE CONTROL DE FLUJO DE
TRÁFICO UTILIZANDO UN MARCO DE CO-SIMULACIÓN**

**GERMAN ANDRES INSUASTY DELGADO
GABRIEL CORAL CAICEDO**

**Trabajo de grado presentado como requisito para optar al título de
Ingeniero Electrónico**

Asesor

PhD. ANDRES DARIO PANTOJA BUCHELI

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
SAN JUAN DE PASTO**

2019

RESPONSABILIDAD

“Las ideas y conclusiones aportadas en el siguiente trabajo son responsabilidad exclusiva de los autores”.

Artículo 1ro del Acuerdo No. 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de Aceptación:

Firma del Director del Trabajo de Grado

Firma del Jurado

Firma del Jurado

San Juan de Pasto, Marzo del 2019

AGRADECIMIENTOS

Agradecemos a la Universidad de Nariño, al Departamento de Ingeniería Electrónica y al grupo de investigación GIIEE, por el apoyo en los procesos académicos y financieros para llevar a cabo esta investigación. Un agradecimiento muy especial a nuestro asesor Andrés Darío Pantoja Bucheli, por su motivación, por la confianza depositada, el esmero y todo el tiempo dedicado a esta investigación, así mismo, se agradece al Ingeniero Pablo Ñañez por brindarnos asesoría y darnos a conocer el problema de investigación, lo cual dio origen a este trabajo. Además un agradecimiento a PTV GROUP por facilitarnos una herramienta óptima para el desarrollo de esta investigación, finalmente, se agradece a todas las personas que de una u otra manera contribuyeron para alcanzar nuestros objetivos y metas, muchas gracias.

German Andres Insuasty Delgado.

Gabriel Coral Caicedo.

Dedicado a Dios que me dio la fortaleza y me colmó de bendiciones permitiendo la culminación de este trabajo, a mis padres, Germán Insuasty y Amada Delgado, por su apoyo y confianza depositada en mí a lo largo de todo este tiempo y a mi hermano Santiago para que lo pueda tomar como un ejemplo de dedicación, esmero y orgullo.

German Andres Insuasty Delgado.

Dedicado a mi familia que siempre ha estado ahí para mí, sobre todo a mi madre Patricia Caicedo, la que ha luchado toda la vida por darnos un futuro mejor para mi hermano y para mí, a mi padre del cual he aprendido mucho a lo largo de mi vida, a mis abuelos que me apoyaron a lo largo de mi carrera y por último a mis amigos, que me ayudaron a afrontar varios problemas durante estos años y me apoyaron para seguir adelante.

Gabriel Coral Caicedo.

RESUMEN

Este trabajo propone una comparación de rendimiento de varios métodos de control para Intersecciones vehiculares semaforizadas en un entorno de co-simulación entre VISSIM y Matlab. Primero, se implementa los métodos en una intersección simple, y luego, se define los requisitos para extender las estrategias de control a varias intersecciones interconectadas. Los métodos que se evaluarán incluyen el tiempo fijo óptimo, por ejemplo la propuesta de F.V. Webster y algunas técnicas de adaptación basadas en estrategias bioinspiradas como por ejemplo Dinamicas de Replicador y Optimización por enjambre de partículas, además de incluir la comparación con una técnica de Control por Modelo Predictivo que ha mostrado resultados interesantes en trabajos anteriores. Aunque hay un interés creciente en la investigación sobre estrategias eficientes para el tráfico de vehículos urbanos, Implementar algoritmos de control en plataformas de simulación de tráfico podría ser una tarea difícil. Por otro lado, el software de control estándar como Matlab carece de la dinámica compleja emulada por VISSIM. La estructura de co-simulación propuesta facilita la implementación de algoritmos de optimización en Matlab, basada en las mediciones (por ejemplo, longitudes de cola promedio, tiempos de llegada, flujos de vehículos) obtenidas del entorno real emulado por el simulador de tráfico, luego, las respuestas óptimas son enviadas desde Matlab a VISSIM para controlar las fases del semáforo en cada intersección. De acuerdo con los resultados obtenidos, el algoritmo de optimización por enjambre de partículas (PSO) proporciona buenos resultados en comparación con la estrategia de tiempo fijo en entornos de intersecciones simples y múltiples. Sin embargo, necesita mediciones continuas para calcular la aptitud de cada individuo (fase verde en cada semáforo), buscando obtener el mejor resultado en un enjambre con información local y global. Del mismo modo, con el enfoque de la asignación óptima de recursos, la tecnica de Dinamicas de Replicador (RD) considera un tiempo como el recurso a ser dividido entre los semáforos para obtener el mejor bienestar común. Los resultados son similares a los obtenidos con PSO, pero el ajuste apropiado

de los parámetros de RD mejora la eficiencia de ciertos índices como la longitud media de la cola y consumo total de combustible. Luego, la comparación de resultados de las diferentes estrategias podría proporcionar ideas para nuevos métodos eficientes.

PALABRAS CLAVE: Control de tráfico urbano, Model Predictive Control, Particle Swarm Optimization, Replicator Dynamics.

ABSTRACT

This paper proposes a comparison of the performance of several control methods for vehicular traffic signals in a co-simulation environment between VISSIM and Matlab. First, the methods are implemented in a simple intersection, and then, the requirements are defined to extend the control strategies to several interconnected intersections. The methods to be evaluated include the optimal fixed time, for example the proposal of F.V. Webster and some adaptation techniques based on bioinspired strategies such as Replicator Dynamics and Particle Swarm Optimization, as well as including the comparison with a Control by Predictive Model technique that has shown interesting results in previous works. Although there is a growing interest in research on efficient strategies for urban vehicle traffic, implementing control algorithms on traffic simulation platforms could be a difficult task. On the other hand, standard control software such as Matlab lacks the complex dynamics emulated by VISSIM. The proposed co-simulation structure facilitates the implementation of optimization algorithms in Matlab, based on measurements (for example, average tail lengths, arrival times, vehicle flows) obtained from the real environment emulated by the traffic simulator, then, the optimal answers are sent from Matlab to VISSIM to control the traffic light phases at each intersection. According to the results obtained, the particle swarm optimization algorithm (PSO) provides good results compared to the fixed time strategy in simple and multiple intersection environments. However, it needs continuous measurements to calculate the aptitude of each individual (green phase in each traffic light), seeking to obtain the best result in a swarm with local and global information. In the same way, with the focus of the optimal allocation of resources, the Replicator Dynamics (RD) technique considers a time as the resource to be divided between the traffic lights to obtain the best common welfare. The results are similar to those obtained with PSO, but the appropriate adjustment of the RD parameters improves the efficiency of certain indices such as the average length of the tail and total fuel consumption. Then, comparing the results of the different

strategies could provide ideas for new efficient methods.

KEYWORDS: Urban traffic control, Model Predictive Control, Particle Swarm Optimization, Replicator Dynamics.

TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN	19
DEFINICIÓN DEL PROBLEMA	23
JUSTIFICACIÓN	24
OBJETIVOS	26
Objetivo General	26
Objetivos Específicos	26
MARCO TEÓRICO	27
Control por Tiempos Fijos	28
Particle Swarm Optimization (PSO)	29
PSO de Tiempo Ciclo Fijo	31
PSO de Tiempo Ciclo Variable	32
Replicator Dynamics (RD)	32
RD de Tiempo Ciclo Fijo	33
RD de Tiempo Ciclo Variable	34
Control Predictivo basado en Modelo (MPC)	35
METODOLOGÍA	40
 1 RESULTADOS Y ANÁLISIS	 44
1.1 Simulación en una intersección aislada	44
1.1.1 Comparación Entre PSO Convencional y PSO Reset	46
1.1.2 Determinación de Fitness para Replicator Dynamics	48
1.1.3 Comparación entre tiempos fijos (Webster), particle swarm optimization (PSO) y replicator dynamics (RD)	52
1.2 Simulación en dos intersecciones interconectadas	54
1.3 Simulación en ocho intersecciones interconectadas	56

2	CONCLUSIONES	62
3	TRABAJO FUTURO	63
	BIBLIOGRAFÍA	65
	ANEXOS	66

LISTA DE FIGURAS

	pág.
Figura 1 Criterios de actualización del mejor personal y mejor global para algoritmo de Particle Swarm Optimization.	38
Figura 2 Esquema general del controlador MPC.	39
Figura 3 Modelo de tráfico urbano Store-and-forward.	39
Figura 4 Comunicación Matlab-VISSIM por medio de una Interfaz de programación de aplicaciones, API.	41
Figura 5 Interfaz API para comunicación Matlab-Vissim actualizada para versiones 7.0 hasta 10.0 del simulador.	42
Figura 6 Diagrama de flujo que detalla la comunicación entre Matlab-VISSIM para el control de trafico.	43
Figura 7 Redes viales de una y dos intersecciones en VISSIM para la simulación de estrategias de control.	45
Figura 8 Longitud de cola acumulada en el tiempo entre el PSO Convencional y el PSO Reset.	47
Figura 9 Resultados comparativos en experimentos saturados y no saturados para fitness dependientes del flujo vehicular y longitud de cola.	50
Figura 10 Resultados comparativos para Replicator Dynamics con fitness de flujo, longitud de cola y flujo más longitud de cola, en un cambio de menor a mayor saturación en la red.	51
Figura 11 Longitud de cola acumulada en el tiempo para las diferentes metodologías de control en la red de una intersección aislada.	53
Figura 12 Longitud de cola acumulada en el tiempo para la red de dos intersecciones.	55
Figura 13 Red de trafico de ocho intersecciones interconectadas. Adaptada de [25].	57

Figura 14 Longitud de cola promedio en las 8 intersecciones para cada estrategia de control en condición 1 de simulación. 60

Figura 15 Red de tráfico con tres intersecciones interconectadas por un carril con dirección de flujo vehicular Sur-Norte. 64

Figura 16 Estado de los semáforos, verde y rojo, dispuestos en la vía Sur-Norte de la **Figura 15**. 64

LISTA DE TABLAS

pág.

Tabla 1	Resultados Simulación Comparación PSO Convencional y PSO Reset en una Intersección	47
Tabla 2	Resultados simulación en una intersección para algoritmos de F.V. Webster, PSO y RD	54
Tabla 3	Resultados Para la Condición 1. Valores finales promedio de los indicadores para las 8 intersecciones del sistema.	58
Tabla 4	Resultados Para la Condición 2. Valores finales promedio de los indicadores para las 8 intersecciones del sistema.	59
Tabla 5	Resultados Para la Condición 3. Valores finales promedio de los indicadores para las 8 intersecciones del sistema.	59

LISTA DE ANEXOS

	pág.
ANEXO A REPOSITORIO DE CÓDIGOS Y ARCHIVOS	67
ANEXO B ARTICULO CONGRESO IEEE	68

GLOSARIO

ALGORITMO: Conjunto ordenado de operaciones sistemáticas o reglas que permiten hallar la solución de un tipo de problemas mediante pasos sucesivos.

FASE: En este trabajo hace referencia al tiempo que dura cada luz de un semáforo.

FUNCIÓN DE FITNESS: Una función de fitness es un tipo particular de función objetivo que se utiliza para resumir, como una figura de mérito, qué tan cerca está una solución determinada de lograr los objetivos establecidos.

INTERSECCIÓN: En este trabajo hace referencia aquellos elementos de la infraestructura vial y de transporte donde se cruzan dos o más caminos.

MATLAB: Software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio.

VISSIM: Software de simulación de flujo de tráfico vehicular.

INTRODUCCIÓN

El transporte urbano en su conjunto es una actividad de gran magnitud y alto impacto en muchos aspectos cotidianos y en el desarrollo de las ciudades. Las consecuencias negativas de un alto nivel de tráfico involucran pérdidas significativas de tiempo, e incremento en el consumo de combustible que da como resultado una mayor producción de gases contaminantes como se muestra en (Perez, 2014). Como un agravante de la situación, el exceso de tráfico genera efectos perjudiciales en la salud relacionados con ambientes sonoros inaceptables y baja calidad del aire, con implicaciones directas e indirectas como el malestar general, trastornos de sueño, ansiedad, depresión e incluso enfermedades cardiovasculares después de un tiempo prolongado de exposición como se muestra en (Perez, 2012).

Ante el crecimiento acelerado de la densidad vehicular, el desarrollo tecnológico en torno al control de tráfico ha permitido reducir la congestión como respuesta a las repercusiones económicas y sociales que conllevan los atascamientos en las redes viales. De esta manera, han surgido los denominados semáforos inteligentes (Santamaria, 2014), tecnología utilizada para mejorar el control del tráfico vehicular en las principales vías de cada ciudad. Ante los distintos retos presentes en el control vehicular, se han empleado diferentes metodologías que pueden ser clasificadas en dos grandes grupos. Las estrategias de control de lazo abierto son técnicas que implementan tiempos fijos en los semáforos a partir de datos estadísticos e históricos sobre el comportamiento del flujo vehicular. Por su parte, las estrategias de control en lazo cerrado son técnicas más sofisticadas que son sensibles a la demanda del tráfico y necesitan sensores para su implementación.

Dentro de la clasificación de estrategias de control en lazo abierto se han implementado métodos tradicionales como SIGSET (Allsop, 1971) y SIGCAP (Allsop, 1976), que son empleados en el control de intersecciones aisladas y buscan minimizar el tiempo de espera individual para los vehículos o maximizar la capacidad en la intersección, respectiva-

mente. El control de intersecciones viales con estrategias de tiempos fijos es una opción económica y simple de programar, por lo que se puede evolucionar fácilmente al diseño de controladores para intersecciones coordinadas, como las estrategias de MAXBAND y UTCS (Papageorgiou, 2003), que buscan maximizar la cantidad de vehículos que pasan por avenidas principales. De estas técnicas, MAXBAND es la única que considera la implementación de las denominadas “olas verdes”, que buscan que los autos viajen a través de una vía principal sin parar en ningún semáforo. También es de destacar la técnica TRANSYT (Robertson, 1969), clasificada también en lazo abierto, que utiliza un algoritmo de optimización por gradiente descendente sobre indicadores de desempeño como el número de paradas por cada vehículo en la intersección.

Dentro de la clasificación de control a lazo cerrado o sensible a la demanda de tráfico se encuentran técnicas que se ejecutan con base a mediciones en tiempo real de las entradas de la intersecciones. Por ejemplo, en (Pedraza, 2012) se aplica un modelo ANFIS (Adaptive Network Based Fuzzy Inference Systems) para optimizar el flujo vehicular en una intersección con 8 cruces semaforizados, controlando tiempos de sincronización y desfase. Como resultado, se muestra una reducción de la densidad vehicular en un 42.6% con respecto a la estrategia de tiempos fijos y una reducción del 61.7% en la cantidad promedio de vehículos en cola. De forma similar, en (Acuna, 2015) se emplean algoritmos evolutivos para resolver un problema de optimización multiobjetivo en un corredor vial, brindando como resultados una mejoría de hasta 15.3% en la velocidad del tráfico de buses y de 24.8% en la velocidad promedio de vehículos particulares. Además, se han realizado varias investigaciones que aplican técnicas de control avanzadas como MPC (Model Predictive Control) en (Jamshidnejad, 2018), esquemas óptimos (Dominguez, 2016), adaptativos (Penaranda, 2016), bioinspirados (Jovanovic, 2017) e inteligentes (Khooban, 2017), presentando todos resultados más eficientes en comparación a las técnicas de tiempos fijos con respecto al flujo vehicular, consumo de combustible y retrasos por vehículo, entre otros índices de rendimiento.

Debido a la magnitud y complejidad del control de tráfico urbano no es viable validar las estrategias de manera directa en una red real. Para esto se emplean plataformas

de simulación que pueden ser clasificadas de acuerdo al modelo de tráfico que utilizan, ya sean macroscópicos, microscópicos o mesoscópicos. Los modelos macroscópicos son usados en aplicaciones de gran escala, donde se tienen en cuenta variables relacionadas con las características generales de flujo como se presenta en (Perez, 2014). Los modelos microscópicos se emplean a escala más detallada para el análisis cercano del tráfico urbano, relacionando variables con respecto a comportamientos individuales de los vehículos. Por último, un modelo mesoscópico presenta una aproximación intermedia entre los microscópicos y macroscópicos, relacionando conceptos de ambos modelos.

Para realizar pruebas pertinentes de las técnicas de control es necesario emplear un programa que pueda simular la situación del tráfico de manera precisa, brinde la mayor cantidad de información posible y permita la fácil aplicación de diversos algoritmos de control. Actualmente existen varios simuladores de tráfico como TSIS-CORSIM (Owen, 2000), que analiza tráfico microscópico, empleando dos modelos para representar el entorno vehicular y brindando información del estado de las vías de manera precisa con un control de señales básico. TransModeler (Yang, 2011) es un software de simulación más completo ya que permite la simulación de los tres modelos de tráfico existentes, dando así la posibilidad de realizar simulaciones híbridas. Esta capacidad hace posible simular redes extensas con costos computacionales relativamente bajos, mientras permite diseñar sistemas de control actuados en relación a la demanda. SUMO (Krajzewicz, 2002) es uno de los pocos simuladores de tráfico microscópico de software libre que permite el modelado de sistemas intermodal que incluyen vehículos de carretera, transporte público y peatones. Tiene varias herramientas de soporte que manejan tareas como la búsqueda de rutas, visualización, importación de redes y cálculo de emisiones. Además, es posible generar un escenario cliente-servidor con Matlab, dando así la oportunidad de poder diseñar un control de señales mas complejo. Por su parte VISSIM (Park, 2003) es un software de simulación de tráfico microscópico que ofrece flexibilidad en varios aspectos como el concepto de enlaces y conectores, modelando geometrías a distintos niveles de complejidad. En esta plataforma se pueden definir atributos para los conductores

y vehículos, permitiendo parametrizaciones individuales. Además, ofrece la integración con otros sistemas vía puerto serial para controladores reales de señal, gestores de tráfico y esquemas de co-simulación.

Dadas las características de los simuladores, en este trabajo se usa VISSIM por ser una plataforma con gran cantidad de información disponible y que ofrece alta precisión por simulación. Para su uso se propone un ambiente de co-simulación con Matlab, donde el sistema vehicular simulado en VISSIM se constituye en la planta que suministra los datos necesarios, los esquemas de control (en matlab) encargados de procesar la información y enviar las señales de control (tiempos de luces de semáforos) nuevamente al sistema de tráfico para completar la realimentación. Como contribución, en este ambiente de co-simulación se proponen dos estrategias novedosas (Particle Swarm Optimization y Replicator Dynamics) para el control de varias intersecciones, en donde el recurso del tiempo de cada señal se optimiza en ciclos variables como respuesta a la interconexión de las vías y a los cambios aleatorios en los flujos vehiculares compartidos. La comparación entre controladores, incluyendo tiempos fijos, se realiza en distintos escenarios calculando varios índices de desempeño como tamaños de colas, consumo de combustible y generación de emisiones al ambiente.

El resto del trabajo se resume en las siguientes secciones: en la sección de estrategias de control de tráfico se trata sobre el fundamento teórico de cada técnica. Más adelante, en entorno de Simulación, se habla sobre los parámetros y configuración del simulador VISSIM en comunicación con Matlab, logrando el ambiente de co-simulación. Seguido a esto se encuentran los resultados y análisis de los resultados, mostrando el desempeño de cada técnica en los diferentes índices de rendimiento. Finalmente se muestran las conclusiones y las posibilidades de trabajo futuro.

DEFINICIÓN DEL PROBLEMA

¿Cuáles son las técnicas que presentan mayor efectividad en el control de un sistema de flujo de tránsito a través de un marco de co-simulación entre Matlab y un simulador?

JUSTIFICACIÓN

De manera continua existe un incremento en la circulación de vehículos que va acompañado de una creciente complejidad estructural de la red vehicular urbana, es por esto que las ciudades empiezan a mostrar algunos problemas de congestión de tráfico, provocando atascamientos, contaminación del aire, contaminación auditiva y aumento en accidentes de tránsito, entre otros. El atascamiento vehicular provoca diferentes aspectos negativos, como pérdidas de tiempo significativas, pérdidas económicas directas al aumentar el uso de combustible para el automotor, aumentando niveles de contaminación ambiental por la producción de gases contaminantes como óxido nitroso, monóxido de carbono, dióxido de carbono, compuestos orgánicos volátiles, además de micropartículas, tal y como se muestra en (Hernandez, 2018), sumado a esto, se pueden generar efectos perjudiciales para la salud, debido a que los residentes cercanos a focos de atascamiento vehicular vivirán en un ambiente sonoro normalmente calificado como inaceptable, con implicaciones directas e indirectas como el malestar general por la falta de un ambiente de tranquilidad, trastorno de sueño, ansiedad, depresión, náuseas, jaquecas, intolerancia, agresividad e incluso enfermedades cardiovasculares después de un tiempo prolongado de exposición. La tecnología asociada a los esquemas de control de tráfico trata de ayudar a mejorar los anteriores aspectos negativos, tanto para la población civil como para los entes gubernamentales, en un sistema de control de flujo de tránsito se puede influir en la velocidad de los diferentes automotores coincidiendo para las velocidades diseñadas de las carreteras y/o avenidas, además de lograr ambientes de conducción más amables para los conductores, pasajeros y peatones, creando una mejor conciencia ciudadana, fortaleciendo la autoridad y normas de tránsito, ayudando al crecimiento en la calidad de vida de la población en general. Para el desarrollo urbano moderno es muy importante lograr disminuir el tiempo recorrido de un vehículo de un punto a otro dentro de la ciudad en diferentes escenarios, para lograr esto, la mayoría de gobiernos opta por rediseñar las ciudades ampliando las calles y optimizando rutas, pero esto im-

plica un gran costo por parte de los gobiernos. El propósito de este proyecto es aportar a este mismo objetivo, pero a un costo mucho menor, con el diseño de estrategias de control de tráfico que puedan ser implementadas gracias al uso de un entorno simulado muy cercano a la realidad, teniendo en cuenta que el control de tráfico en redes urbanas se hace principalmente a través del manejo de los semáforos en intersecciones, y que en Colombia no existen sistemas de autopistas apropiadamente desarrollados, resulta relevante analizar las alternativas disponibles en la actualidad para llevar a cabo una gestión adecuada sobre las intersecciones semaforizadas, como se aborda en (Quijano, 2009) y en (Gutierrez, 2013). En consecuencia, surge la necesidad de un marco de co-simulación que permita una interacción entre el software especializado, encargado de simular condiciones reales de flujo vehicular, una plataforma de programación simple para la simulación y la prueba de diferentes técnicas y algoritmos de control. Esta configuración tipo Planta-Controlador facilita la implementación de estrategias avanzadas de control para lograr un desempeño óptimo del sistema, a la vez que sirve como una herramienta de desarrollo de nuevas investigaciones en este campo de los sistemas complejos.

OBJETIVOS

Objetivo General

Realizar un estudio comparativo de diferentes técnicas de control de tránsito en escenarios analizados en un entorno de co-simulación entre Matlab y una plataforma especializada de simulación de flujo vehicular.

Objetivos Específicos

- Analizar diferentes técnicas de control para optimización del tráfico observando los requerimientos del control y los parámetros de simulación de flujo.
- Implementar una interfaz de programación de aplicaciones (API por sus siglas en inglés) entre Matlab y un simulador especializado de tráfico para facilitar la prueba de las técnicas de control.
- Comparar diferentes técnicas de optimización aplicadas al control de flujo de tránsito, con base en distintos índices de desempeño obtenidos de escenarios co-simulados en la plataforma desarrollada.

MARCO TEÓRICO

Los esquemas de control de tráfico urbano usan semáforos coordinados en las intersecciones para detener y direccionar el flujo vehicular, definido como la cantidad de carros que pasan por un punto en un tiempo determinado. El flujo depende del número de vías que se juntan en un cruce definiendo las indicaciones de señal, que corresponden al encendido de una de las luces del semáforo o una combinación de varias luces al mismo tiempo (como las requeridas para avances y giros).

El tiempo que dura cada luz de un semáforo se conoce como fase, que es el tiempo asignado a las combinaciones de uno o más movimientos que reciben simultáneamente el derecho de paso durante uno o más intervalos. Una fase comienza con la pérdida del derecho de paso de los movimientos que entran en conflicto con aquellos vehículos que lo ganan. De acuerdo con las reglas internacionales, un movimiento pierde el derecho de paso en el momento que aparece la señal de indicación amarilla. Al tener varios semáforos en una intersección, se define un ciclo como el tiempo necesario para una secuencia completa de todas las fases en todos los semáforos.

Entre las medidas más utilizadas para evaluar el desempeño de sistemas de tráfico se tiene la longitud de cola en una vía, contada como el número de vehículos pendientes del paso. Esta medida es importante para determinar el porcentaje de saturación en la vía y la calidad del sistema de control. El valor del porcentaje de saturación en una intersección es una medida de la demanda actual en relación con su capacidad total. Este valor de la capacidad total es un parámetro que depende de la geometría de las calles y de criterios específicos como el grado de prioridad a cada vía. Con base en este tipo de medidas se pueden calcular otros indicadores como el consumo de combustible, los tiempos de espera, los retrasos y emisiones contaminantes, como se muestra en (Santamaria,2014).

Una clasificación general de las estrategias de control de tráfico define controladores de tiempo fijo y de tiempo variable. Los primeros hacen referencia a un control donde el ci-

clo, la duración y la secuencia de intervalos son invariables, son definidos con anticipación y su calibración depende de datos estadísticos que se deben evaluar para calibraciones posteriores. Los controladores de tiempo variable son más complejos y permiten mejorar el rendimiento de la red por medio de mediciones del número de vehículos en cada intersección. Normalmente están implementados en tres pasos básicos, iniciando con la recolección de la información (cantidades de vehículos, longitudes de cola y saturaciones, entre otras medidas) para calcular indicadores que se deben optimizar. Luego, se ejecuta un algoritmo para el ajuste de los tiempos en los semáforos (variables de control), y finalmente, ajustar los parámetros y mejorar el rendimiento en el siguiente ciclo, de manera similar a (Peñaranda, 2016).

A continuación se presentan las estrategias implementadas en este trabajo para el análisis de diferentes casos de estudio y su comparación.

Control por Tiempos Fijos

Las estrategias de control basadas en tiempo fijo establecen las fases de cada semáforo con base en datos y mediciones estadísticas previos a la implementación tal y como se muestra en (Papageorgiou, 2003), (Robles 2009). F. V. Webster (Webster, 1958), basándose en observaciones de campo y simulación de condiciones de tránsito, determinó una longitud de tiempo de ciclo óptimo dada por

$$C = \frac{1.5L + 5}{1 - \sum_{i=1}^{\varphi} Y_i}, \quad (1)$$

donde L es el tiempo en segundos perdido por ciclo, φ es el número de fases y Y_i es el máximo valor de la relación entre el flujo promedio y el flujo de saturación para el acceso i .

Cabe notar que el tiempo de pérdida en el ciclo está dado por la duración de las fases en amarillo y en rojo. El cálculo del tiempo en amarillo se define por

$$T_y = t + \frac{v}{2a}, \quad (2)$$

donde t representa el tiempo de percepción o reacción del conductor, v es la velocidad media del flujo vehicular y a es la tasa de desaceleración. La estimación del tiempo en

rojo T_r se calcula con

$$T_r = \frac{W + L_v}{v}, \quad (3)$$

donde W es el ancho de la intersección y L_v es la longitud de los vehículos. Luego, es posible calcular el tiempo de pérdida en cada ciclo con

$$L = T_r + T_y. \quad (4)$$

Para determinar la demora mínima en la intersección, el tiempo verde efectivo total del ciclo dado por $C - L$ se debe distribuir de manera proporcional a los valores de relación de saturación Y_i

$$G_i = \frac{Y_i(C - L)}{\sum_{j=1}^{\varphi} Y_j}, \quad (5)$$

expresión en la que G_i es el tiempo en verde efectivo para la fase i de la intersección. De esta manera se establecen los tiempos óptimos para cada fase, que son programados de manera permanente en el sistema de control de tránsito.

Particle Swarm Optimization (PSO)

Esta técnica se basa principalmente en modelos de conductas sociales, tales como el comportamiento de una bandada de aves o un banco de peces, como se muestra en (Kennedy, 1997). Este algoritmo permite optimizar la solución de un problema a partir de una población de individuos que representan soluciones candidatas. Los individuos se pueden movilizar por todo el espacio de búsqueda según criterios y restricciones, para encontrar una mejor solución, tal y como se demuestra en (Qian, 2007), (Eberhart, 2001).

El movimiento del i -ésimo individuo tiene como criterio de actualización la mejor posición personal que haya tenido hasta el momento ($pBest_i$) y la mejor posición global que haya encontrado el grupo ($gBest$). Estos dos parámetros junto con un factor aleatorio influyen en la velocidad y la posición para moverse en el espacio de búsqueda en el siguiente instante. En cada iteración del algoritmo se hace una actualización de velocidad y posición (Kennedy, 1997) para cada individuo definidas respectivamente por

$$V_i(k + 1) = wV_i(k) + c_1r_1(pBest_i - X_i(k)) + c_2r_2(gBest - X_i(k)) \quad (6)$$

y

$$X_i(k+1) = X_i(k) + V_i(k+1). \quad (7)$$

En estas ecuaciones, $X_i(k)$ es la posición del individuo i en el instante k , w es el peso de inercia; c_1 y c_2 son coeficientes que determinan el peso relativo de la influencia del mejor personal $pBest_i$ y el mejor global $gBest$, respectivamente. r_1 y r_2 son valores aleatorios en el intervalo $[0, 1]$ y los valores de c_1 y c_2 son constantes asignadas mediante pruebas de simulación escogiendo los valores que entregan los mejores resultados. Cabe destacar que el valor de c_2 es considerablemente menor a c_1 , ya que el problema a trabajar requiere que se le de prioridad al valor del mejor personal $pBest_i$, por encima del valor del mejor global $gBest$ en cuanto a la asignación de los tiempos en verde de cada calle. La posición de cada individuo es evaluada por medio de una función de bienestar o fitness. Si la nueva posición del individuo es mejor que la anterior, $pBest_i$ es reemplazada por X_i y si es descubierta una mejor posición global, $gBest$ es actualizado. Para emplear el algoritmo en control de tráfico, se asume que existen φ fases de movimiento, reuniendo todos los semáforos. Este número de fases corresponden al número de individuos pertenecientes al enjambre para PSO, donde la posición de cada individuo X_i , representa el tiempo en verde de la fase a la que pertenece. En concordancia con el problema, la función objetivo o *fitness* que se quiere minimizar para cada individuo se define por

$$f_i = \frac{Qlen_i}{Cap_i}, \quad (8)$$

representa la relación entre la longitud de cola actual $Qlen_i$ y la capacidad máxima de la vía Cap_i , del semáforo i . Claramente la minimización de estas funciones objetivo significa una mejora en la red y es consecuencia directa de la implementación de los tiempos de cada fase X_i .

En la ejecución del algoritmo, al inicio de cada ciclo de tiempo C se evalúan las condiciones de longitud de cola para cada vía y se determina el valor de la fitness (8) para calcular la velocidad y posición mediante las expresiones (6) y (7). Esto se logra con

el cálculo de $pBest_i$ y $gBest$, que son actualizados al presentarse una disminución en la fitness del iésimo individuo, reemplazando su antiguo valor con el X_i responsable del cambio. Así, el algoritmo en cada intervalo de tiempo evalúa las condiciones y actualiza el control tratando de converger a un valor de longitud de cola mínimo.

Sin embargo, existen inconvenientes para esta configuración de PSO, puesto que presenta un problema al tratar de encontrar el mínimo de este sistema, cuyas características (longitudes y flujos) cambian constantemente en el tiempo. Esto es debido a los criterios de actualización de los valores de $pBest$ y $gBest$, ya que estos solo se actualizan si el valor de fitness evaluado en el instante k , es menor al valor que ya tenía asignado en $k-1$. De esta manera si el mínimo del sistema cambia por un valor mayor al que tenía con anterioridad, el algoritmo podría cambiar. El flujo de tráfico es un sistema dinámico, por lo que se propone modificar el algoritmo PSO clásico restableciendo los valores de $pBest_i$ y $gBest$, una mejor adaptación de PSO en el contexto de trabajo.

En la **Figura 1** se presentan los diagramas de flujo de los criterios que se añadieron al algoritmo de PSO para resetear los valores de $pBest_i$ y $gBest$. En cuanto al valor asignado a $pBest_i$, se evalúa si el valor de la fitness no ha mejorado durante 2 ciclos de tráfico C . Si no se logra un mínimo mejor que el $pBest_i$ actual, este es reemplazado por el valor actual de la fitness de la calle i .

El valor $gBest$ es actualizado por el valor mínimo de $pBest$ cuando $gBest$ tenga un valor de fitness menor a 0.1, equivalente a un nivel de saturación del 10%. Con esto se evita que el algoritmo falle al afirmar que una de las calles esta prácticamente vacía cuando en verdad puede tener una cola mayor. Estos criterios son aplicados antes de evaluar la velocidad y la posición de cada individuo para asegurar el hecho de que se tienen los valores correctos de $pBest$ y $gBest$ para la situación actual de cada intersección.

En la sección de resultados se hace una comparación entre el PSO convencional y el PSO modificado, denominado PSO Reset en las siguientes secciones, mostrando así las capacidades de cada uno frente al problema planteado.

PSO de Tiempo Ciclo Fijo

En un esquema de tiempo fijo, el tiempo del ciclo se mantiene constante en el sistema (Gong, 2012), es decir que $\sum_{i=1}^{\varphi} X_i = C$. Sin embargo, en el algoritmo PSO cada X_i es el resultado de un proceso evolutivo, por lo que en general la restricción no se cumple. Para satisfacer esta restricción, al final de cada iteración se ajustan los valores de los individuos calculando la diferencia entre el ciclo C y $\sum_{i=1}^{\varphi} X_i \neq C$, y recortando o aumentando los valores de X_i de manera proporcional para satisfacer el tiempo fijo requerido.

PSO de Tiempo Ciclo Variable

Para permitir una longitud de ciclo variable y mayor flexibilidad en la optimización de los tiempos en verde, se realiza un cambio al algoritmo descrito anteriormente. Aunque se mantienen todos los cálculos para estimar posición y velocidad en el espacio de búsqueda, se modifica la restricción de $\sum_{i=1}^{\varphi} x_i = C$. De esta manera, se propone que la sumatoria de tiempos para las fases puede variar dentro de un rango dado por $C - \frac{a}{2} < \sum_{i=1}^{\varphi} x_i < C + \frac{a}{2}$, donde a es una constante positiva de sintonización de acuerdo con la red que se esté analizando. Si existe un sobrepaso de los límites, se recorta o adiciona tiempo a cada individuo de forma proporcional. En consecuencia, se pasa de tener un solo intervalo de tiempo C a tener varios intervalos de tiempos C_i , permitiendo así que el algoritmo determine el tiempo de ciclo óptimo para la situación actual del sistema.

Replicator Dynamics (RD)

El modelo de RD describe cómo ciertas características de una población pueden llegar a evolucionar mediante la interacción mutua de los individuos y la evaluación de su estado de bienestar, tal y como se demuestra en (Pantoja, 2019). En analogía, el tráfico urbano también puede ser visto desde un enfoque de dinámicas poblacionales, puesto que cada vehículo tiene un comportamiento individual y se organizan en conjunto persiguiendo ciertos objetivos tales como tomar la vía que lo lleve a su destino en el menor

tiempo posible o el menor gasto de combustible recorriendo la menor distancia. En estos escenarios, la población se caracteriza por presentar comportamientos replicables entre individuos en busca de objetivos comunes (Gutierrez, 2013).

Para describir la estrategia de RD, se asume que se tiene un conjunto de n hábitats, en donde hay una cantidad determinada de individuos de una especie. En este caso, $p_i(t) \geq 0$ es la cantidad de individuos que optan por el hábitat i , mientras que $p = [p_1, \dots, p_n]^\top$ es el estado de la población y P_{tot} es la cantidad total de la población en el ambiente. Los cambios en cada una de las poblaciones define el modelo de RD, que está dado por (Taylor, 1978) así,

$$\frac{dp_i}{dt} = p_i(f_i(p) - \bar{f}(p)), \forall i = 1, \dots, n, \quad (9)$$

donde $f_i(p)$ representa la función de bienestar percibida en el i -ésimo hábitat y $\bar{f}(p)$ es el promedio ponderado de las funciones de bienestar de la población,

$$\bar{f}(p) = \frac{1}{P_{tot}} \sum_{i=1}^n p_i f_i(p). \quad (10)$$

Una consecuencia interesante de la definición en (10), es que el total de la población se mantiene constante, es decir que $\sum_{i=1}^n p_i = P_{tot}$ para todo el tiempo (Hofbauer, 1998). Además, observando la ecuación (9), el estado estable se logra cuando $p_i^*(f_i(p^*) - \bar{f}(p^*)) = 0$, donde $p^* = [p_1^* \ p_2^* \ \dots \ p_n^*]^\top$ es un punto de equilibrio. Si $p_i^* > 0 \ \forall \ i$, el punto de equilibrio satisface la condición $f_i(p^*) = \bar{f}^*$, donde $\bar{f}^*$ es el promedio de la función de bienestar en equilibrio. Luego, en estado estacionario todos los hábitats tienen la misma función de bienestar.

Realizando analogías apropiadas con los esquemas poblacionales, se ha mostrado que la elección apropiada de funciones de bienestar conlleva a encontrar un punto de equilibrio estable que soluciona diversos problemas de optimización y control en ingeniería, como se muestra en (Quijano, 2017).

RD de Tiempo Ciclo Fijo

Asumiendo que el número total de hábitats n representa el número de fases φ de la red de semáforos, entonces la cantidad de individuos en cada hábitat p_i representaría

el tiempo en verde para la fase i . Así, la condición de población constante está dada por $\sum_{i=1}^{\varphi} p_i = P_{tot} = C$, donde C es el tiempo ciclo para las fases en la intersección de tráfico. Luego, la restricción de tiempo fijo es respetada por los RD todo el tiempo. Ahora, una función de bienestar apropiada a la analogía está dada por

$$f_i(p) = \frac{w_1 q_i + w_2 Q_i}{w_1 S_{q_i} + w_2 S_{Q_i}}, \quad (11)$$

en donde para el intervalo de tiempo C , q_i es el flujo vehicular, Q_i es la longitud de cola, y S_{q_i} y S_{Q_i} son los valores de saturación de flujo y cola, respectivamente, para la fase i . Además, w_1 y w_2 son pesos ajustables para determinar la influencia de los flujos o colas, teniendo en cuenta que $w_1 + w_2 = 1$.

La función de fitness es determinada por las variables de longitud de cola y flujo vehicular con una relación directamente proporcional, ya que ante una mayor fitness, es necesario un mayor tiempo en verde. La combinación de estos parámetros permite aprovechar las ventajas de cada uno. Por ejemplo, al estimar el flujo vehicular se tiene en cuenta la cantidad de vehículos que ingresan a la red para el próximo ciclo, mientras que la longitud de cola solo tiene en consideración la situación actual de la red. Además, la longitud de cola tiene cambios muy lentos en comparación a la medición de flujo vehicular cuando existen condiciones de extrema saturación, es decir, donde hay un atascamiento masivo y por tanto poca movilidad de los vehículos. En estos casos, la tasa de vehículos entrantes a la red sería muy baja y el cálculo de la fitness solamente con el flujo estimaría que la red se encuentra poco saturada. Esta indicación es errónea y puede solventarse adicionando el control por longitud de cola, que es capaz de observar que existe un gran atascamiento y darle prioridad a la calle en cuestión. De esta manera, estos dos parámetros en conjunto pueden adaptarse mejor ante diversos estados de la red de tránsito. Finalmente, los parámetros de peso w_1 y w_2 son ajustados para determinar la prioridad de las variables, que en este caso son escogidos tal que $w_1 > w_2$, ofreciendo mejor desempeño en condiciones de tráfico normal.

RD de Tiempo Ciclo Variable

Para ajustar el algoritmo de Replicator Dynamics a un ciclo variable, se debe modificar

la restricción de $\sum_{i=1}^{\varphi} p_i = C$ a una menos restrictiva que permita que $\sum_{i=1}^{\varphi} p_i < C$. En este caso, el concepto de tiempo a repartir en cada fase puede ser tomado como un recurso con un límite superior. Si se agrega un hábitat más al ambiente en la ecuación (9), se tendrían $\varphi + 1$ ecuaciones, sumando una fase al sistema de control de tránsito y logrando cumplir $\sum_{i=1}^{\varphi+1} p_i = C$. La fase adicional es un “agente ficticio” (Pantoja, 2019) que no está asignado a ningún semáforo, pero que hace las veces de una variable que almacena el tiempo que no se utiliza en las fases reales. La función de bienestar para la fase ficticia se define por

$$f_{\varphi+1}(p) = \frac{w_1 \sum_{i=1}^{\varphi} q_i + w_2 \sum_{i=1}^{\varphi} Q_i}{w_1 \sum_{i=1}^{\varphi} S_{q_i} + w_2 \sum_{i=1}^{\varphi} S_{Q_i}}, \quad (12)$$

que tiene en cuenta las funciones de bienestar de todas las fases reales, monitoreando el estado completo del sistema. La función de bienestar promedio se mantiene en estructura, pero debe considerar la nueva fase en la forma

$$\bar{f}(p) = \frac{1}{P_{tot}} \sum_{i=1}^{\varphi+1} p_i f_i(p). \quad (13)$$

Cabe resaltar que con la adición de la nueva fase, en el punto de equilibrio todas las funciones de bienestar siguen siendo iguales (incluyendo a la ficticia), manteniendo las mismas propiedades importantes de los RD, como la estabilidad y optimalidad de su punto de equilibrio (Pantoja, 2019). De esta manera el control con un límite superior para el tiempo ciclo de las intersecciones puede adaptarse mejor a las condiciones dinámicas de un flujo de tráfico real, sin cambiar la estructura del controlador.

Control Predictivo basado en Modelo (MPC)

Un controlador MPC hace uso de un modelo para predecir la respuesta del sistema, para que con una estimación de valores futuros de la planta, se minimice una determinada función objetivo. El método utilizado en este trabajo es tomado de (Gutierrez, 2013), en donde se parte de una referencia que corresponde al estado final deseado y el modelo realiza una predicción de los estados del sistema. Estas predicciones se comparan con el punto de referencia y la diferencia entre estos dos parámetros será recibida por un

optimizador que provee las señales de control. El método descrito se puede observar en la **Figura 2**.

Existen varias alternativas al plantear un modelo de tráfico urbano, sin embargo, es destacable el trabajo realizado en (Tettamanti, 2008) que emplea un modelo Store-and-Forward. Específicamente, las dos partes básicas del modelo son la intersección y la calle de conexión a la intersección, donde cada intersección semaforizada tiene sus propios parámetros asociados a la calle de entrada y de salida. La **Figura 3** muestra la relación de dos intersecciones interconectadas.

Las dinámicas de la calle de conexión z están descritas por

$$x_z(k+1) = x_z(k) + T(q_z(k) - h_z(k) + r_z(k) - s_z(k)), \quad (14)$$

donde $x_z(k)$ mide el número de vehículos en la calle z en el instante kT , de lo cual también se puede estimar la longitud de la cola en ese instante, considerando T como el periodo de control y $k = 1, 2, 3, \dots$ como el índice de tiempo discreto. Las variables $q_z(k)$ y $h_z(k)$ son flujos de entrada y salida respectivamente, mientras que $r_z(k)$ y $s_z(k)$ son la demanda y el flujo de salida durante el periodo $[kT, (k+1)T]$.

Aunque la ecuación (14) representa el movimiento de vehículos para la calle z , cuando se desea definir el modelo de una red de tráfico cada calle de cada conexión debe ser modelada con su propia ecuación. Además, las ecuaciones deben estar relacionadas por medio de los flujos que comparten, agrupando las cantidades de vehículos de las calles en un vector de estados $x(k)$. El modelo se puede expresar en una representación de espacio de estados lineal de la forma

$$x(k+1) = Ax(k) + Bu(k) + d(k), \quad (15)$$

donde A es la matriz identidad (el estado futuro depende del estado presente sumados los flujos netos) y el producto $Bu(k)$ se deriva de $T[q_z(k) - h_z(k)]$ en la ecuación (14), representando la diferencia del flujo de entrada y de salida de la red durante el intervalo de control T . La entrada de control u es el vector de los tiempos en verde de todas las fases. Para obtener la matriz B se estima el parámetro S (flujo de saturación), o se

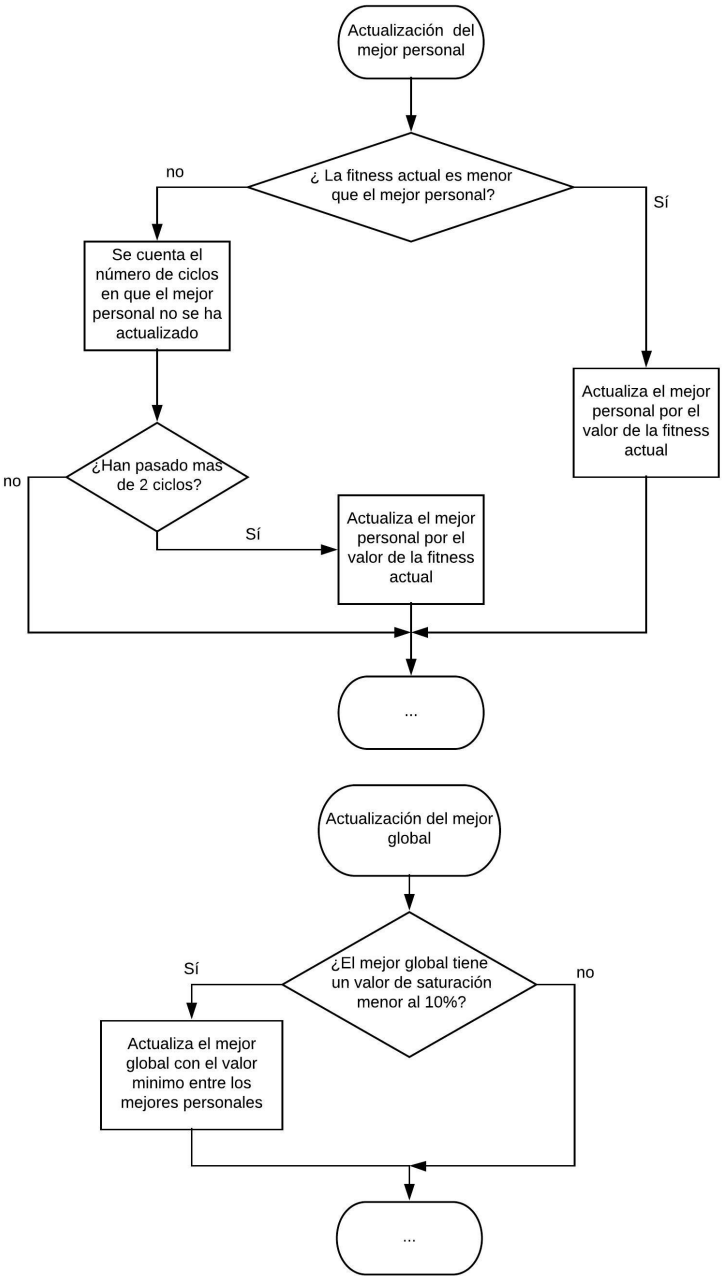
puede usar un valor estándar como $S = 0.5$ veh/s (Tettamanti, 2008). El término $d(k)$ representa una perturbación aditiva al sistema que está compuesta por $T[r_z(k) - s_z(k)]$, donde $r_z(k)$ y $s_z(k)$ son consideradas medidas de perturbación y reflejan la diferencia en la demanda de flujos de salida de una vía durante el intervalo de control. Adicionalmente, en la perturbación se incluye $p_z(k)$, que es un parámetro relacionado a los límites de la red de tráfico.

Una vez obtenido el modelo, existen varias opciones para el indicador a optimizar. La función más utilizada es una ecuación cuadrática de costo de la forma

$$J(k) = \frac{1}{2} \{x(k)^\top Q x(k) + u(k)^\top R u(k)\}, \quad (16)$$

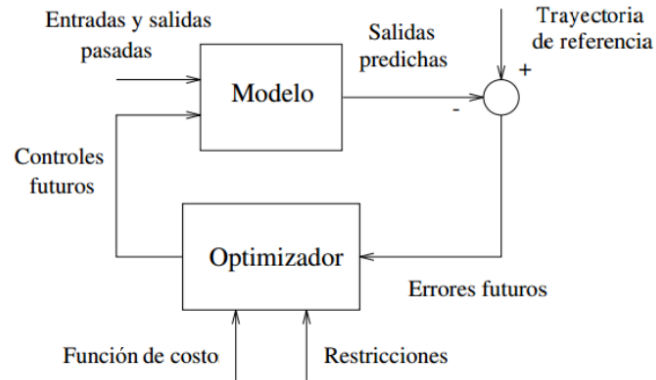
donde x es el vector de estado, representando el número de vehículos que se encuentran en cada vía de la intersección a controlar, u es el vector de las entradas de control (tiempos en verde), y Q y R son matrices constantes para ponderar la relevancia de estados y entradas en la función de costo. La minimización de esta función permite reducir el número de vehículos con los menores tiempos en verde posibles. Para esto, se debe encontrar el vector $u(k)$, teniendo en cuenta que los estados sean coincidentes con los predichos por el modelo de la ecuación (15). Este vector óptimo representa los mejores tiempos en verde posibles para obtener el mínimo de $J(k)$.

Figura 1. Criterios de actualización del mejor personal y mejor global para algoritmo de Particle Swarm Optimization.



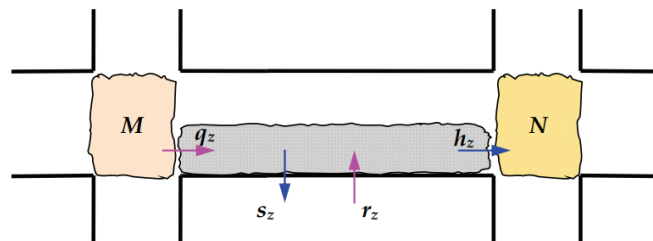
Fuente: Producto de este trabajo, realizado con software Lucidchar.

Figura 2. Esquema general del controlador MPC.



Fuente: Microscopic simulation model calibration and validation: case study of vissim simulation model for a coordinated actuated signal system pp 185.

Figura 3. Modelo de tráfico urbano Store-and-forward.



Fuente: Development of road traffic control by using integrated vissim-matlab simulation environment pp 43.

METODOLOGÍA

Para este trabajo se emplea un ambiente de co-simulación entre VISSIM y Matlab, a manera similar a como se utiliza en (Tettamanti, 2012), (Dabiri, 2016), (Tettamanti, 2015), siendo VISSIM la planta o sistema a controlar y Matlab el encargado en procesar la información y enviar las señales de control nuevamente al sistema.

La interfaz VISSIM-COM define un modelo jerárquico en donde las funciones y parámetros del simulador, que originalmente se muestran de manera gráfica, pueden ser manipuladas por comandos de programación en diferentes lenguajes capaces de manejar objetos COM, tales como C++, Visual Basic, Java, Python, Matlab, entre otros. Anteriormente, el uso de la interfaz no estaba ampliamente difundido y la simulación y control se debía hacer por medio de una *Interfaz de programación de aplicaciones, API*. La **Figura 4** muestra la interdependencia de las dos plataformas con una API, lo que eleva un costo computacional y puede llegar a ser confuso a la hora de implementar simulaciones coordinadas. Este proceso se detalla en (Tettamanti, 2012). En este trabajo se actualiza una API desarrollada en (Gutierrez, 2013), en la cual se puede configurar distintos parámetros para la simulación tales como el tiempo ciclo, tiempo inicial en verde para cada controlador, asignación de sensores y tiempo de simulación, entre otros. La interfaz actualizada de esta API puede ser vista en la **Figura 5**.

Hoy en día, por medio de la programación COM, el usuario tiene la capacidad de manipular los atributos de los objetos internos de la simulación de una manera dinámica y directa, sin necesidad de usar una API. La programación por puerto COM es explorada también para hacer la simulación entre Matlab y Vissim, debido a que Matlab brinda una mayor facilidad al momento de diseñar y depurar algoritmos de control, además de ser un entorno de programación con herramientas y funciones de control y optimización. En la **Figura 6** se observa un esquema general de cómo opera el ambiente de co-simulación planteado. Primero, se establecen unos parámetros iniciales para el control del tráfico (tiempo de simulación, red de tráfico a utilizar, flujos vehiculares de entrada y semilla

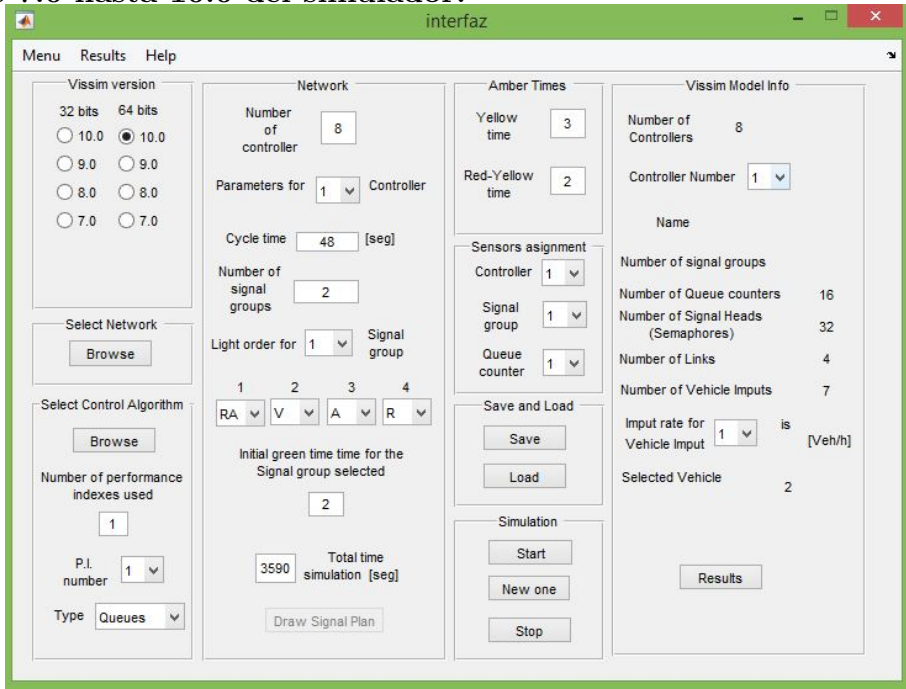
de aleatoriedad, entre otros) que definen el escenario de trabajo. Estos parámetros se cargan en Matlab y posteriormente se envían a VISSIM, iniciando la simulación. Dentro del entorno de Matlab se espera hasta que se cumpla el tiempo para el muestreo de datos y se solicita la información relevante al simulador de tráfico (e.g., longitud de cola, flujos, retrasos, consumos). Con esta información se ejecuta el algoritmo de control para el cálculo de tiempos en verde tratando de mejorar el estado de la red de tráfico, tiempos que se comunican e implementan en VISSIM repitiéndose hasta que se cumpla el tiempo de simulación.

Figura 4. Comunicación Matlab-VISSIM por medio de una Interfaz de programación de aplicaciones, API.



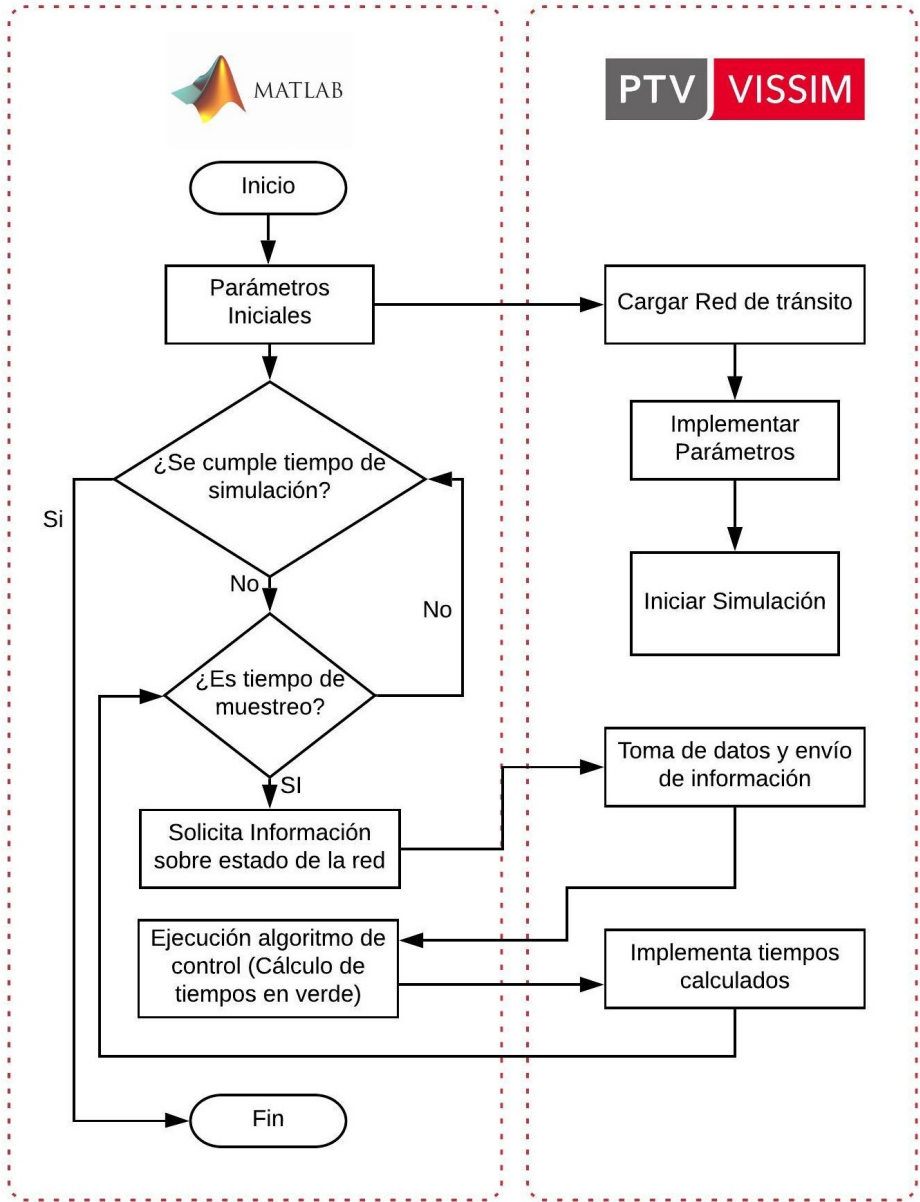
Fuente: Producto de este trabajo.

Figura 5. Interfaz API para comunicación Matlab-Vissim actualizada para versiones 7.0 hasta 10.0 del simulador.



Fuente: Producto de este trabajo realizado en software Matlab.

Figura 6. Diagrama de flujo que detalla la comunicación entre Matlab-VISSIM para el control de trafico.



Fuente: Producto de este trabajo, realizado con software lucidchar.

1. RESULTADOS Y ANÁLISIS

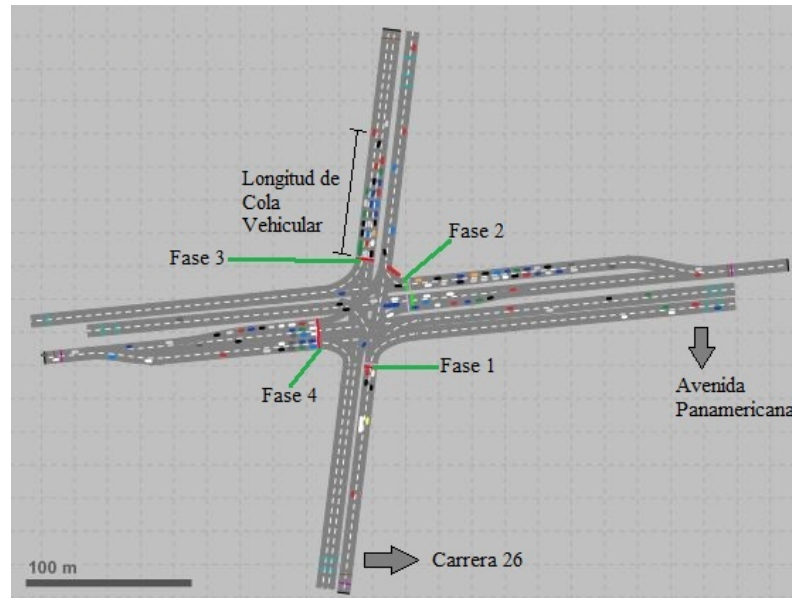
A continuación se muestran los resultados obtenidos y el análisis correspondiente para distintos escenarios de redes viales. En primera instancia se hace la implementación de algoritmos en una intersección aislada, ajustando y calibrando parámetros para cada técnica de control y tratando de obtener los mejores comportamientos posibles para diferentes datos de saturación. Posteriormente se implementan dos intersecciones interconectadas, para analizar el comportamiento de los algoritmos en casos donde el flujo de salida de una intersección se convierte en el flujo de entrada de la otra. Finalmente, para observar el desempeño del control en una red compleja, se aplican las técnicas a una red de ocho intersecciones interconectadas y se hace la comparación con los diferentes indicadores de desempeño.

1.1 Simulación en una intersección aislada

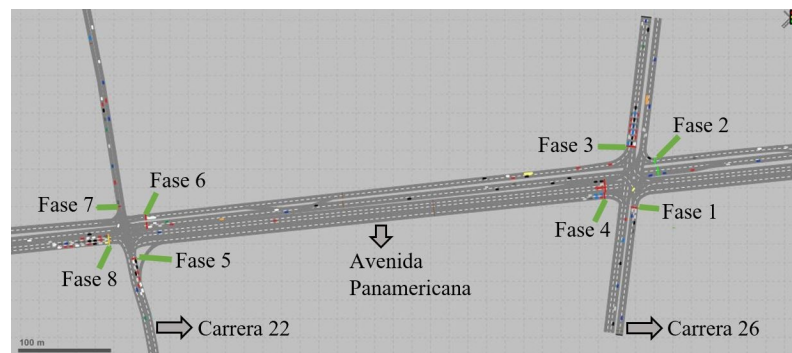
Las simulaciones son evaluadas sobre una intersección inspirada en la red vial de la ciudad de Pasto que presenta variaciones frecuentes en el flujo vehicular durante el transcurso del día. Una imagen de la intersección en el simulador VISSIM se muestra en la **Figura 7(a)**.

Los parámetros de la red se fijan con un tiempo de simulación de 22500 segundos, equivalentes a 6.25 horas, en las cuales existen cinco cambios de flujo vehicular cada 4500 segundos. En primera instancia todas las vías tienen un flujo de entrada de 1000 veh/h, tratando de simular un contexto cotidiano de saturación de la red. Seguido a esto, los flujos cambian a 1500 veh/h y 1000 veh/h para las vías dispuestas de manera vertical y horizontal, respectivamente, obteniendo el escenario donde las vías verticales presentan condiciones de una mayor saturación. Al finalizar esta etapa se intercambian los valores, donde se simula un mayor flujo en la vía horizontal. Luego se reemplazan los valores de flujo de entrada vehicular por 1000 veh/h para la Carrera 26 y 500 veh/h para la vía Panamericana, otorgando un escenario donde dos de las cuatro fases presentan nivel

Figura 7. Redes viales de una y dos intersecciones en VISSIM para la simulación de estrategias de control.



(a) Red vial de una intersección con 4 fases inspirada en la Avenida Panamericana con Carrera 26, Pasto, Colombia.



(b) Red vial de dos intersecciones con 8 fases, inspirada en las vías de Avenida Panamericana entre Carrera 22 y 26, Pasto, Colombia.

Fuente: Producto de este trabajo, realizado en software VISSIM.

bajo de saturación. Al final, estos últimos valores se permutan cumpliendo los cinco de flujo vehicular. La evaluación de la simulación se hace con una tasa de muestreo $T = 115$ s.

Los resultados presentados en esta sección tienen las siguientes convenciones:

Qlen: Longitud promedio de la cola vehicular [metros].

QlenMax: Longitud máxima de cola vehicular [metros].

VehsAll: Cantidad de vehículos que han pasado por la intersección en el intervalo de tiempo.

VehsDelayAll: Tiempo promedio de demora de todos los vehículos. La demora de un vehículo se obtiene al restar el tiempo de viaje teórico (ideal) del tiempo de viaje real. El tiempo de viaje teórico es el tiempo de viaje que podría lograrse si no hubiera otros vehículos, sin controles de señal u otras razones para detenerse.

StopDelayAll: Retraso promedio en segundos por vehículo.

StopAll: Promedio de paradas por vehículo en el intervalo de tiempo.

EmissionCO: Cantidad de monóxido de carbono [gramos].

EmissionNOX: Cantidad de óxido de nitrógeno [gramos].

EmissionVOC: Emisiones de compuestos orgánicos volátiles [gramos].

FuelConsumption: Gasto de combustible [galones estadounidenses].

Las pruebas en esta intersección se realizan en primera instancia analizando los dos modelos planteados de PSO para observar cuál se adecúa más al flujo de transito. Luego se realizan las mismas pruebas al algoritmo RD para poder identificar la mejor función de fitness. Una vez definidos estos parámetros se comparan los resultados de los algoritmos propuestos (tiempos fijos de Webster, PSO y RD).

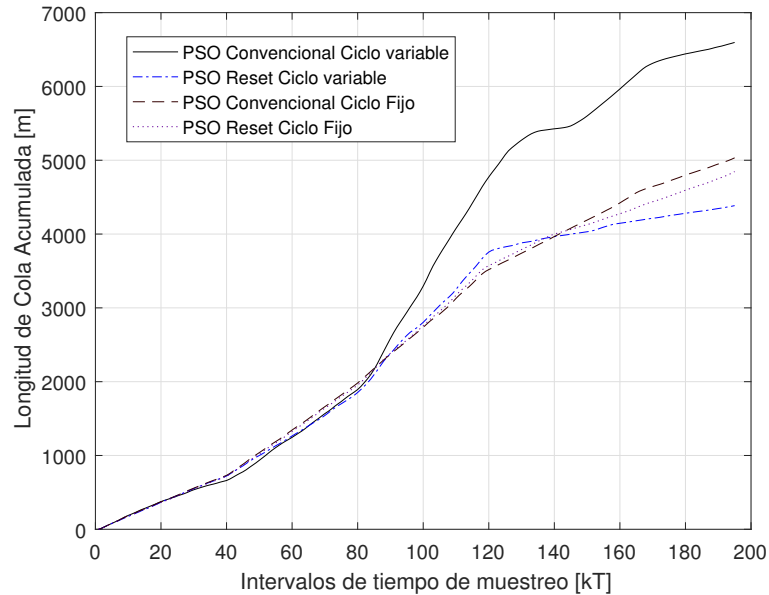
1.1.1 Comparación Entre PSO Convencional y PSO Reset En la **Tabla 1** y en la **Figura 8** se presentan los resultados obtenidos en los modelos del algoritmo PSO funcionando con los criterios planteados y evaluados según las restricciones en el tiempo de ciclo. Como convención se tiene que CFC es la técnica de ciclo fijo convencional, CFR es ciclo fijo reset, CVC es ciclo fijo variable y CVR ciclo variable reset.

Tabla 1. Resultados Simulación Comparación PSO Convencional y PSO Reset en una Intersección

	Qlen	QLenMax	VehsAll	VehsDelayAll	StopDelayAll
PSO CFC	5033.1	23461.4	24759.5	9127.1	7597.9
PSO CFR	4846.65	22020.4	24754.1	8749.2	7236.4
PSO CVC	6596.2	26333.9	24772.5	9869.3	7671.8
PSO CVR	4385.4	20284.6	24768.5	7095.7	5403.9
	StopAll	EmissionCO	EmissionNOX	EmissionVOC	FuelConsumption
PSO CFC	169.1	33581.5	6533.7	7782.8	480.4
PSO CFR	165.1	32830.9	6387.7	7608.9	469.7
PSO CVC	259.6	40376.3	7855.8	9357.6	577.6
PSO CVR	187.1	31964.8	6219.2	7408.2	457.3

Fuente: Producto de este trabajo.

Figura 8. Longitud de cola acumulada en el tiempo entre el PSO Convencional y el PSO Reset.



Fuente: Producto de este trabajo. Realizado en Matlab.

La **Figura 8** muestra la integral de la longitud de cola promedio en metros que se obtiene en los instantes kT . Esta variable es escogida porque normalmente es asociada con el atascamiento en el tráfico urbano y es un buen índice para evaluar el desempeño del control en la red. Se observa que en ciclo fijo y en ciclo variable, PSO Reset obtiene mejores resultados que PSO convencional. Con ciclo fijo, PSO Reset logra reducir la longitud de cola en un 4% con respecto al PSO convencional, pero con ciclo variable se llega a reducir en un 34% con respecto al algoritmo convencional.

La razón para que los resultados sean tan diferentes al cambiar la restricción en el tiempo de ciclo es que PSO convencional es propenso a errores, y más aún con un tiempo de ciclo variable. Como se explicó anteriormente, PSO convencional funciona para sistemas invariantes en el tiempo, por lo que en este caso, este método no entrega un valor mínimo confiable con respecto a la saturación de las vías. En el caso de tiempo de ciclo fijo, los resultados son cercanos entre PSO convencional y PSO reset, ya que el algoritmo en ambos casos tiene un tiempo determinado para repartir entre los semáforos de cada intersección. Adaptando los tiempos para que la suma sea el tiempo C , establecido como parámetro al inicio de la simulación. En el caso en donde el algoritmo no tiene restricción en el tiempo de ciclo, PSO asigna los tiempos a cada semáforo sin que estos tengan que ser modificados para cumplir la condición. Sin embargo, para el PSO convencional se llega un punto en que el error es mayor porque este algoritmo alcanza mínimos que no cambian a medida que evoluciona la simulación. En cambio, el PSO Reset tiene los mejores resultados al reiniciar periódicamente sus mejores personales y globales, reaccionando apropiadamente a los cambios de saturación de las vías.

En concordancia con las longitudes de cola, PSO Reset de ciclo variable presenta también mejores resultados para los indicadores de cantidades de vehículos, paradas y emisiones mostrados en la **Tabla 1**.

1.1.2 Determinación de Fitness para Replicator Dynamics la función de estado de bienestar de la población o fitness, se evalúa en cada iteración del algoritmo como indicador que sirve de referente a la nueva asignación de recursos (tiempos en

verde). Por esto se deben seleccionar apropiadamente los parámetros que interactúan en esta función, que es parte fundamental de la ecuación (9). Como primera medida se deduce que la función de bienestar debe estar relacionada de manera directa al flujo vehicular entrante a la red por la congestión o nivel de atascamiento que este flujo produce. Sin embargo, un flujo determinado en una vía estrecha producirá una longitud de cola mayor que en una vía ancha. Luego, dependiendo de la geometría de la vía, se pueden tener diferentes condiciones de saturación de flujo S_{qi} y de cola S_{Qi} .

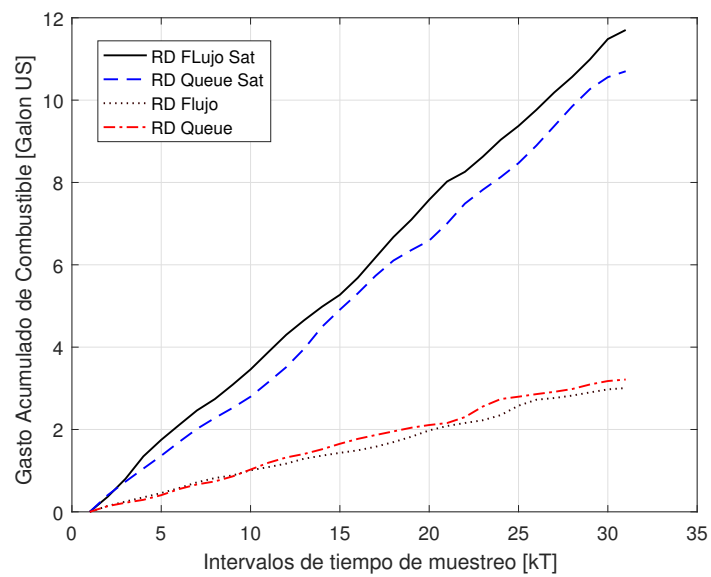
Para encaminar la influencia del flujo (qi) y de la longitud de cola (Qi), se examina el desempeño de los RD, ecuación (9), con las siguientes funciones de fitness

$$f_i(p) = \frac{q_i}{S_{qi}}, \forall i = 1, \dots, n, \quad (17)$$

$$f_i(p) = \frac{Q_i}{S_{Qi}}, \forall i = 1, \dots, n. \quad (18)$$

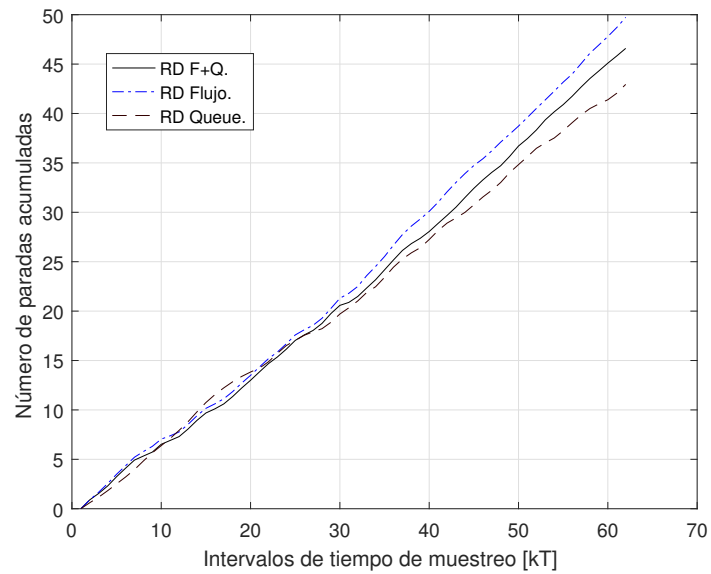
Estas fitness son puestas a prueba en dos contextos diferentes de condiciones de tráfico sobre la misma red vial mostrada en la **Figura 7(a)**. Como primer experimento, todas las fases tienen una entrada de flujo vehicular de 1000 veh/h, simulando una condición de red no saturada. En segunda instancia, dos de las cuatro fases incrementan su flujo vehicular de entrada en un factor de 10, simulando una condición desigual de saturación. Los resultados sobre el número de paradas promedio acumulado, se muestran en la **Figura 9**, mientras que los demás índices de rendimiento demuestran comportamientos similares. De estos dos escenarios se observa que en situación de saturación de red, tiene un mejor comportamiento el control de RD con una fitness que involucre la longitud de la cola. A este escenario se lo denominará *RD Queue Sat*, pues logra el mismo objetivo de control en la intersección, con un número menor de paradas al de la fitness con flujo, escenario denominado como *RD Flujo Sat*. Por su parte en la situación no saturada de la red, el control con una fitness de flujo, (*RD Flujo*), tiene un menor gasto de combustible. Teniendo en cuenta estos resultados en distintas situaciones, se opta por implementar una fitness combinada como la que se muestra en la ecuación (11). Los resultados

Figura 9. Resultados comparativos en experimentos saturados y no saturados para fitness dependientes del flujo vehicular y longitud de cola.



Fuente: Producto de este trabajo. Realizado en Matlab.

Figura 10. Resultados comparativos para Replicator Dynamics con fitness de flujo, longitud de cola y flujo más longitud de cola, en un cambio de menor a mayor saturación en la red.



Fuente: Producto de este trabajo. Realizado en Matlab.

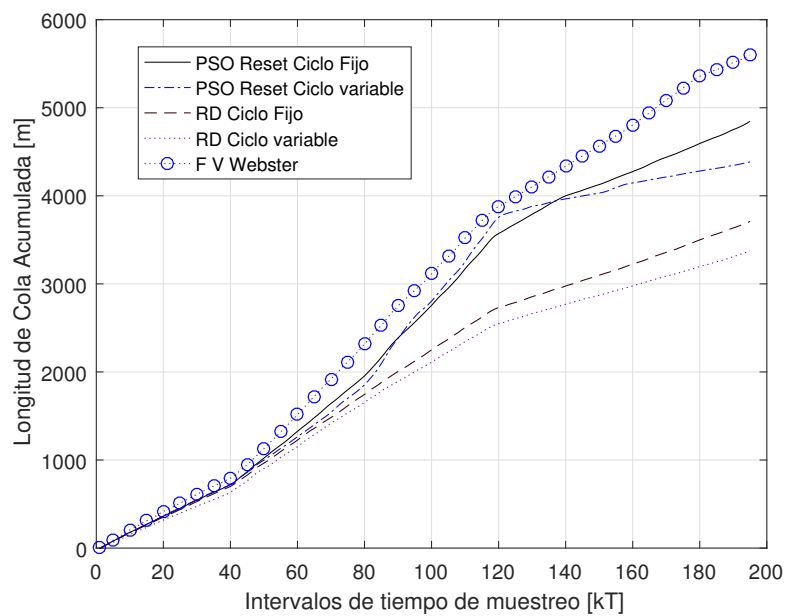
comparativos se pueden ver en la **Figura 10**, en donde el comportamiento combinado de una fitness con flujo y longitud de cola, aprovecha las características de cada uno en las diversas situaciones que se pueden presentar. Para analizar la influencia, se plantea un caso donde los primeros 30 intervalos de simulación se tiene un escenario no saturado y luego la saturación se aumenta en un factor de 10 para 2 de las 4 fases. La función combinada de fitness tiene un comportamiento equilibrado en el número de paradas, gasto de combustible y demás índices de desempeño, durante todo el tiempo, llegando a un punto medio de rendimiento, pero aprovechando sus características y ventajas para los diferentes escenarios de tráfico.

1.1.3 Comparación entre tiempos fijos (Webster), particle swarm optimization (PSO) y replicator dynamics (RD) Para hacer la comparación entre algoritmos se toma como punto de referencia los resultados dados por el algoritmo F. V. Webster, pues corresponde al esquema de tiempos fijos que generalmente se implementa en las redes viales reales. Los resultados se presentan en la **Tabla 2** y **Figura 11**.

Con la comparación realizada entre algoritmos PSO, en esta sección se usa solamente PSO Reset tanto en ciclo fijo (PSOR CF) como en ciclo variable (PSOR CV). Con los indicadores obtenidos en la intersección aislada, es notorio que el mejor desempeño se logra con el algoritmo de Replicator Dynamics con ciclo variable, puesto que reduce en un 40% la longitud de cola, que implica también una disminución en el gasto de combustible y emisiones de CO de un 21%.

El algoritmo de PSO Reset en tiempo ciclo fijo y variable, muestra un desempeño mejor que Webster, pues reducen respectivamente la longitud de cola en un 13% y 22%, logrando también una disminución en el gasto de combustible de 9% y 11%. Los resultados de los demás índices de desempeño tienen un comportamiento similar, siendo RD de ciclo variable el algoritmo con mejores indicadores. Los detalles pueden verse en la **Tabla 2**, donde se destacan en cursiva los resultados más favorables.

Figura 11. Longitud de cola acumulada en el tiempo para las diferentes metodologías de control en la red de una intersección aislada.



Fuente: Producto de este trabajo. Realizado en Matlab.

Tabla 2. Resultados simulación en una intersección para algoritmos de F.V. Webster, PSO y RD

	Qlen	QLenMax	VehsAll	VehsDelayAll	StopDelayAll
Webster	5596.9	24901.6	24761.1	10133.2	8536.5
PSOR CF	4846.6	22020.4	24754.1	8749.2	7236.4
PSOR CV	4385.4	20284.6	24772.5	7095.7	5403.9
RD CF	3709.6	18331.6	<i>24765.5</i>	7337.4	5901.5
RD CV	<i>3373.6</i>	<i>16897.5</i>	<i>24765.5</i>	<i>6671.5</i>	<i>5205.6</i>
	StopAll	EmissionCO	EmissionNOX	EmissionVOC	FuelConsumption
Webster	178.8	36022.5	7008.7	8348.6	515.3
PSOR CF	165.1	32830.9	6387.7	7608.9	469.7
PSOR CV	187.1	31964.9	6219.2	7408.2	457.3
RD CF	<i>154.3</i>	29490.4	5737.8	6834.7	421.9
RD CV	157.3	<i>28620.5</i>	<i>5568.5</i>	<i>6633.1</i>	<i>409.4</i>

Fuente: Producto de este trabajo.

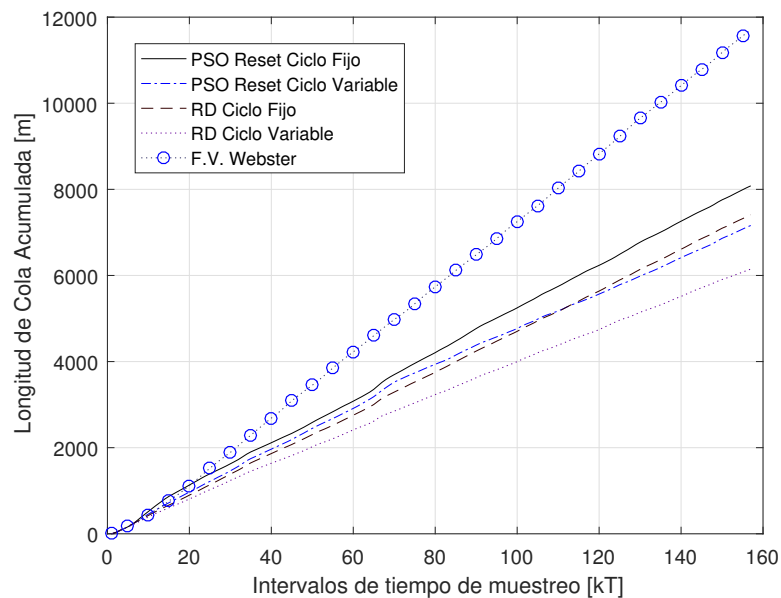
1.2 Simulación en dos intersecciones interconectadas

El análisis de las conexiones de varias intersecciones viales es importante ya que representa la mayoría de casos reales, permitiendo observar la interdependencia de los cruces. Para una buena operación, la influencia de la capacidad de las vías no es tan relevante si no se tiene un buen esquema de control en las intersecciones.

Para simular estos esquemas se lleva a cabo una ampliación de la intersección aislada de la **Figura 7(a)**, obteniendo un esquema de dos intersecciones interconectadas con 8 fases, como se presenta en la **Figura 7(b)**. Es de destacar que las intersecciones comparten flujo vehicular, por lo que el desempeño del sistema depende también de una coordinación implícita entre los controladores de cada una. La simulación de esta red tiene un tiempo de 18000 segundos, equivalentes a 5 horas, con un tiempo de muestreo de $T = 115$ s, que corresponde al valor de ciclo óptimo calculado por la técnica de F. V. Webster. Las vías mostradas en la **Figura 7** tienen una distribución geométrica aproximadamente igual.

Los resultados se muestran en la **Figura 12**, donde se observa la acumulación en el tiempo de la longitud de cola promedio por ciclo. Los datos muestran que los algoritmos adaptativos tienen un buen comportamiento frente a la técnica de tiempos fijos, pues se logra reducir el promedio de las longitudes de cola en 31% y 39% para PSO con tiempo de ciclo fijo y variable respectivamente, y aún mejor para Replicator Dynamics que reduce en 36% y 43%. Este resultado también implica una disminución en el gasto de combustible y tiempo de espera para los vehículos.

Figura 12. Longitud de cola acumulada en el tiempo para la red de dos intersecciones.



Fuente: Producto de este trabajo. Realizado en Matlab

El control con tiempos de ciclo fijo es restrictivo pues no permite a los sistemas reaccionar o predecir efectivamente la saturación de las calles. En cambio, con un control con tiempos de ciclo variable, es posible notar en algunos instantes de simulación un desfase en los valores de tiempo en verde de las vías interconectadas, dando lugar así a una coordinación de fases, o también llamado una ola verde, como en (Corman, 2009). Esta coordinación está relacionada con la distancia de separación entre intersecciones y

la velocidad promedio de los vehículos. Así, cuando un vehículo abandona la primera intersección y llega a la próxima, es probable que la encuentre habilitada para pasar, mejorando los índices de rendimiento. Sí un vehículo logra entrar en una ola verde, tendrá un recorrido con una mínima demora y posiblemente sin paradas, disminuyendo gasto de combustible y emisiones de gases, entre otros aspectos. Es interesante que los algoritmos ejercen un control independiente en cada intersección, sin compartir información entre los sistemas. Sin embargo, es posible inferir una coordinación de fases, con una mejoría de 3% en los índices de longitud de cola con Replicator Dynamics para la red de dos intersecciones en comparación con la intersección aislada.

1.3 Simulación en ocho intersecciones interconectadas

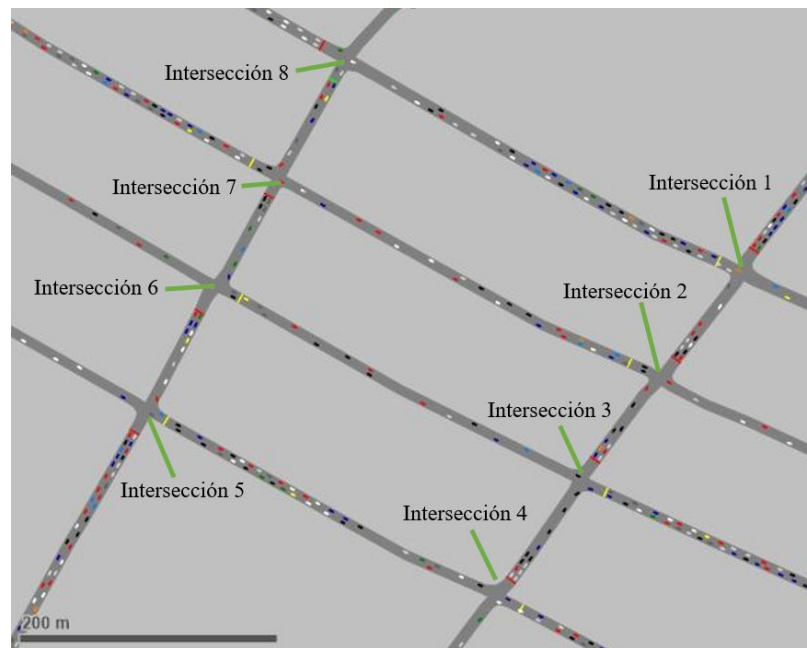
Con el fin de analizar una red similar al de un área de tráfico de una ciudad, se hace la extensión de los algoritmos a una red de ocho intersecciones interconectadas, ver **Figura 13**, cuenta con 16 fases (2 por cada cruce)(Tettamanti, 2008) como la presentada en (Gutierrez, 2013), en donde se aplican varios controles, incluyendo MPC.

Las principales características de esta red son que las vías que conectan cada intersección portan dos carriles y el área que estas vías forman en las ocho intersecciones es aproximadamente igual.

Los algoritmos propuestos en este trabajo son técnicas no inspiradas en modelo, que muestran un buen comportamiento adaptativo al cambio de flujo vehicular tanto en una intersección aislada como en un sistema interconectado. Sin embargo, es interesante hacer la comparación de desempeño con algoritmos exitosos como el controlador predictivo basado en modelo (MPC) implementado en (Gutierrez, 2013) y (Tettamanti, 2008) descrito anteriormente, que tiene como objeto maximizar el flujo por cada intersección. Para mostrar la reacción de los controladores ante diferentes características de saturación de la red, se han propuesto los siguientes casos:

Condición 1: En este experimento se busca simular la condición no saturada de la red, donde la tasa de entrada de vehículos por carril en cada intersección es de 1500 Veh/h. El flujo de tránsito ingresa a las intersecciones de la red respetando el sentido de cada

Figura 13. Red de trafico de ocho intersecciones interconectadas. Adaptada de [25].



Fuente: Control de tráfico urbano, anexos.

vía.

Condición 2: Se simula la condición en la que las calles de la red se encuentren saturadas, con una tasa de entrada de vehículos a cada una de las calles de 2500 Veh/h, mientras que las carreras presentan una situación de flujo normal de 1500 Veh/h.

Condición 3: Presenta saturación en las carreras con una tasa de entrada de 2500 Veh/h, mientras que las calles presentan un flujo normal de 1500 Veh/h.

Los resultados del MPC muestran que el algoritmo de optimización mejora alrededor de un 20% con respecto a una estrategia de tiempos fijos. Sin embargo, al compararlo con los controles basados en PSO y Replicator Dynamics, estos últimos obtienen resultados promedio en las intersecciones considerablemente mejores, como se muestra en la Tablas: **Tabla 3**, **Tabla 4** y **Tabla 5**. Haciendo detalle en los resultados de la **Tabla 3**, se encuentran mejoras en el tiempo de retraso en parada, pasando de un 7% de MPC, a un 20% de PSO y 50% en Replicator Dynamics. Cabe notar que la técnica de MPC obtiene un flujo vehicular 3% mejor que los otros algoritmos, resultado coherente puesto que en el modelo matemático busca maximizar esta variable (Tettamanti, 2008).

Tabla 3. Resultados Para la Condición 1. Valores finales promedio de los indicadores para las 8 intersecciones del sistema.

	Qlen	QlenMax	VehsAll	VehsDelayAll	StopDelayAll
Webster	10398.1	36061.5	21888.5	13677.2	4284.7
MPC	10181.9	35308.5	22482.5	14276.9	3968.3
PSO	7788.7	31865.1	21711.1	11154.5	3432.2
RD	5390.3	24681.2	21886.5	9794.9	2132.4
	StopsAll	EmissionCO	EmissionNOX	EmissionVOC	FuelConsumption
Webster	909.1	23027.5	4480.3	5336.8	329.4
MPC	984.3	25050.4	4873.9	5805.6	358.4
PSO	643.7	18066.4	3515.1	4187.1	258.4
RD	602.3	16871.5	3282.6	3910.1	241.3

Fuente: Producto de este trabajo.

En la **Figura 14** se observa el comportamiento de la longitud de cola promedio en

Tabla 4. Resultados Para la Condición 2. Valores finales promedio de los indicadores para las 8 intersecciones del sistema.

	Qlen	QlenMax	VehsAll	VehsDelayAll	StopDelayAll
Webster	15353.3	50862.5	22300.5	16574.8	5250.4
MPC	14647.4	49664.1	22917.5	16427.8	4627.7
PSO	16549.2	55763.1	23583.5	14691.7	5954.8
RD	9239.4	37936.2	24393.1	14037.2	3041.5
	StopsAll	EmissionCO	EmissionNOX	EmissionVOC	FuelConsumption
Webster	1549.7	34985.4	6806.9	8108.2	500.5
MPC	1606.6	36699.6	7140.4	8505.5	525.1
PSO	965.9	27203.3	5292.8	6304.6	389.2
RD	1192.5	31252.6	6080.6	7243.1	447.1

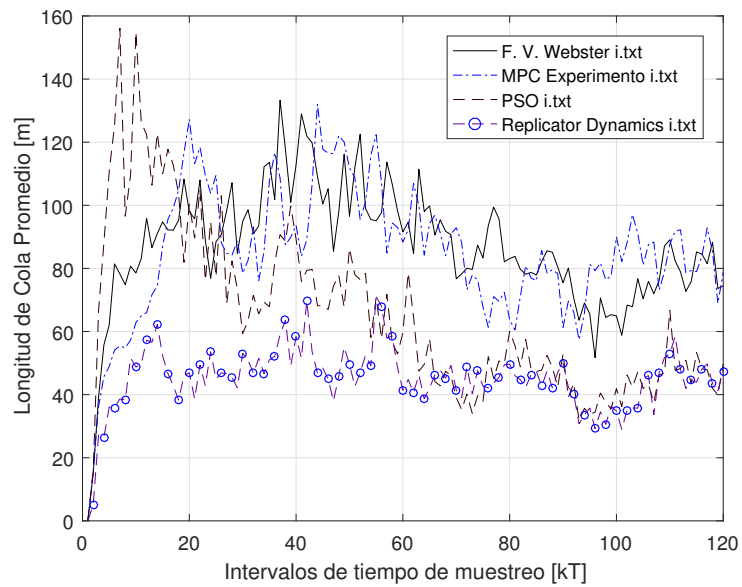
Fuente: Producto de este trabajo.

Tabla 5. Resultados Para la Condición 3. Valores finales promedio de los indicadores para las 8 intersecciones del sistema.

	Qlen	QlenMax	VehsAll	VehsDelayAll	StopDelayAll
Webster	26749.1	81726.9	24235.5	23815.8	7452.5
MPC	25244.6	77182.2	24317.1	24559.1	7003.3
PSO	28058.8	93038.3	25840.5	19270.8	8938.3
RD	16605.9	61194.1	25930.1	20285.4	4314.7
	StopsAll	EmissionCO	EmissionNOX	EmissionVOC	FuelConsumption
Webster	2903.2	63091.1	12275.2	14621.9	902.6
MPC	3241.6	69111.4	13446.6	16017.2	988.7
PSO	1371.5	38496.5	7490.1	8921.9	550.7
RD	2579.1	59916.3	11657.5	13886.2	857.2

Fuente: Producto de este trabajo.

Figura 14. Longitud de cola promedio en las 8 intersecciones para cada estrategia de control en condición 1 de simulación.



Fuente: Producto de este trabajo. Realizado en Matlab.

las intersecciones a lo largo del tiempo, variable normalmente relacionada con el atascamiento en las vías. Es interesante notar que el algoritmo de PSO en los primeros instantes de funcionamiento presenta un manejo poco eficiente del tráfico, ya que es un algoritmo que mejora su desempeño en el espacio de búsqueda en cada iteración. Así, en los primeros instantes de simulación está alejado de los tiempos óptimos para el control de flujo, pero a medida que avanza la simulación existe una mejoría en el manejo del tránsito, llegando a valores muy similares de la técnica de Replicator Dynamics.

Los resultados para las distintas condiciones de saturación de las carreras y las calles (Tablas: **Tabla 3**, **Tabla 4** y **Tabla 5**), muestran que en los escenarios el valor de los índices de desempeño varía de manera directa con el valor del flujo de entrada vehicular. Entre más saturada se encuentre una intersección vial, los algoritmos de PSO y Replicator Dynamics reducen su tiempo ciclo, pues necesitan hacer una repartición de tiempos óptimos de manera más frecuente para atender el flujo vehicular entrante. MPC y F. V. Webster tienen un tiempo de ciclo fijo, por lo que las condiciones de saturación influyen en la distribución del tiempo efectivo en verde, dando prioridad a la fase con mayor demanda. Uno de los índices mas destacables de PSO esta en la reducción del gasto de combustible, pues en comparación a la técnica de F. V. Webster se disminuyó en aproximadamente 22%, 23% y 39% para cada condición de saturación, respectivamente.

En general, haciendo un promedio de los índices de evaluación, la técnica de Replicator Dynamics de ciclo variable logra tener el mejor rendimiento, pues logra mejorar el desempeño con respecto a la técnica de tiempos fijos de en un 30%, 20% y 16% para cada condición, respectivamente, mientras que PSO tiene unos porcentajes de mejoría de 19%, 11% y 20%. La técnica de MPC en promedio tiene ligeramente un mejor desempeño que la tecnica de tiempos fijos.

2. CONCLUSIONES

El tráfico es un sistema complejo con características especiales, en donde las técnicas de control a emplearse deben ser estudiadas a detalle y tener posibilidad de adaptación. De esta manera, las técnicas de PSO y Replicator Dynamics resultan adecuadas para controlar estos sistemas de manera óptima y en tiempo real, mejorando los resultados de técnicas ya implementadas en redes viales físicas y mostrando ser métodos versátiles y útiles en la práctica.

El carácter aleatorio del algoritmo de PSO puede presentar dificultad al inicio del algoritmo. Sin embargo, si se mantiene dentro de un rango adecuado brinda una mejoría en búsqueda de unos tiempos óptimos para el control de flujo. Con respecto a la técnica de Replicator Dynamics, es de gran importancia el criterio para escoger una función de bienestar, ya que agregar variables a esta función puede mejorar el control al obtener más información sobre el estado actual y futuro de la red.

Es de destacar que el aporte realizado en este trabajo sobre controles con tiempo ciclo variable se constituye en una herramienta que mejora los indicadores de manejo del tránsito. Esto se corrobora con las simulaciones realizadas, en donde es posible notar que el tiempo ciclo de cada fase tiene relación inversa con la saturación de la intersección.

3. TRABAJO FUTURO

Como trabajo futuro se propone comprobar la sincronización autónoma, percibida visualmente en sistemas de control aislados con Replicator Dynamics que comparten flujo vehicular. Se recomienda hacer un estudio relacionando los datos de los tiempos en verde de los semáforos con los valores de flujo y longitud de cola vehicular en cada momento de simulación.

Para analizar la posible sincronización de fases se implementó una red de tres intersecciones interconectadas como se muestra en la **Figura 15**, donde se establecieron flujos vehiculares determinísticos, para reducir las características aleatorias en el experimento, los flujos se fijaron en 1000 Veh/h para la vía vertical y 500 Veh/h para las vías horizontales. La simulación se llevó a cabo en un escenario simplificado para la transición de fases, donde solo se consideran el derecho y restricción del paso (señales en verde y rojo respectivamente). Esta red cuenta con 6 fases (2 fases en cada intersección). Haciendo un análisis de tiempos para los semáforos dispuestos en la vía vertical, enumerados en la dirección de flujo vehicular de sur a norte en la **Figura 15**, se pudo observar un cierto desfase en la asignación de tiempos verdes (**Figura 16**). Esta gráfica de fases muestra el estado de los semáforos en los momentos avanzados de una simulación de 3600 s, donde se nota que en momentos donde el semáforo 1 emite una señal de luz verde, después de un cierto tiempo el semáforo 2 también lo hace en coordinación con el semáforo 3. Esta situación permitiría predecir que este último semáforo está notoriamente desfasado. Sin embargo, dado que en esta parte final de la red también recibe flujo vehicular de las vías horizontales, supondría una luz verde en tiempos aproximadamente iguales al semáforo 2, generando movimiento en su carril, haciendo que los vehículos entrantes hagan el menor número de paradas en esta ultima intersección. El experimento se hace con el algoritmo de Replicator Dynamics, en su versión de tiempo ciclo variable, donde las 3 intersecciones muestran un tiempo ciclo promedio aproximadamente de 25 s. Aunque los estados de los semáforos mostrados en la **Figura 16** muestran una cierta tendencia a

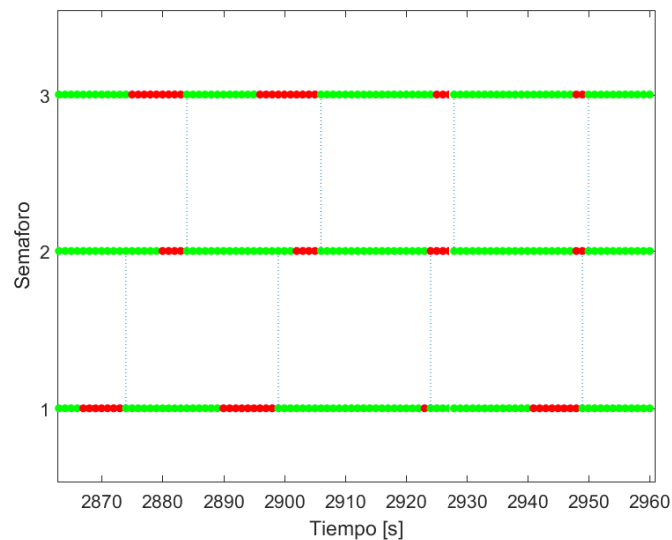
la sincronización, no se puede demostrar una secuencia o patrón específico que corrobore la formación de olas verdes.

Figura 15. Red de tráfico con tres intersecciones interconectadas por un carril con dirección de flujo vehicular Sur-Norte.



Fuente: Producto de este trabajo. Realizado en VISSIM.

Figura 16. Estado de los semáforos, verde y rojo, dispuestos en la vía Sur-Norte de la Figura 15.



Fuente: Producto de este trabajo. Realizado en Matlab.

BIBLIOGRAFÍA

ACUNA, Álvaro, ARRECHE, Efraín y NESMACHNOW, Sergio. Planificación de tráfico y transporte colectivo en el corredor garzón utilizando un algoritmo evolutivo, Montevideo, 2015. Trabajo de investigación (Ingeniero electrónico). Universidad de la Republica. Facultad de ingeniería.

ALLSOP, Richard E. Sigset: a computer program for calculating traffic signal settings. Traffic Engineering & Control, (1971).

ALLSOP, Richard E. Sigcap: A computer program for assessing the traffic capacity of signal- controlled road junctions. Traffic Engineering & Control, 17 (1976).

CORMAN, Francesco, et al. Evaluation of green wave policy in real-time railway traffic management. Transportation Research Part C: Emerging Technologies. 17 (2009); pp. 607–616.

DABIRI, Sina and ABBAS, Montasair. Arterial traffic signal optimization using particle swarm optimization in an integrated vissim-matlab simulation environment, in Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. IEEE, 2016; pp. 766–771.

DOMINGUEZ, Patricia y CORTINEZ, Victor. Un enfoque de optimización para el control de semáforos, in XII Congreso de ingeniería del transporte. 7, 8 y 9 de Junio, Valencia (España). Editorial Universitat Politècnica de Valencia. 2016; pp. 2456–2470.

EBERHART, Russell, SHI, Yuhui and KENNEDY, James. Swarm intelligence, Elsevier. 1 (2001).

GONG, Yue and ZHANG, Jun. Real-time traffic signal control for roundabouts by using a pso-based fuzzy controller, in Evolutionary Computation (CEC). 2012 IEEE Congress on,

IEEE.2012; pp. 1–8

GUTIERREZ, Ivan F. Control de tráfico urbano. Bogota, 2013. Trabajo de investigacion (Ingeniero electronico). Universidad de los Andes. Facultad de ingenieria.

HERNANDEZ ORTIZ, Mariana. Estudio de impactos ambientales producido por el parque automotor de la ciudad de montecristi. Montecristi, 2018. B.S. tesis. Universidad Estatal del Sur de Manabí.

HOFBAUER, Josef and SIGMUND, Karl. Evolutionary games and population dynamics. Cambridge University press. 1 (1998).

HOMCHAUDHURI, Baisravan, VAHIDI, Ardan and PISU, Pierluigi. Fast model predictive control- based fuel efficient control strategy for a group of connected vehicles in urban road conditions. IEEE Transactions on Control Systems Technology. 25 (2017); pp. 760– 767.

JAMSHIDNEJAD, Anahita, et al. Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothening methods. IEEE Transactions on Control Systems Technology. 26 (2018); pp. 813–827.

JOVANOVIC, Aleksandar, NIKOLIC, Milos and TEODOROVIC, Dusan. Area-wide urban traffic control: A bee colony optimization approach. Transportation Research Part C: Emerging Technologies. 77 (2017); pp. 329–350.

KENNEDY, James. The particle swarm: social adaptation of knowledge. in Evolutionary Computation, 1997. IEEE International Conference on, IEEE. 1997; pp. 303–308.

KHOOBAN, Mohammad H, et al. An optimal general type-2 fuzzy controller for urban traffic network, ISA transactions. 66 (2017); pp. 335–343.

KRAJZEWICZ, Daniel, et al. Sumo (simulation of urban mobility); an open-source traffic

simulation, 4th Middle East Symposium on Simulation and Modelling (MESM2002). (2002); pp. 183–187.

PEREZ, Javier y GOMEZ, Jaime. Análisis discriminante de algunas variables que influyen en la contaminación acústica debida al tráfico urbano en una gran ciudad. *Revista Ingenierías Universidad de Medellín*. 11 (2012); pp. 13– 22.

QUIJANO, Nicanor. Honeybee social foraging for urban traffic control. in *Control in Transportation Systems*. 15(2009); pp. 588–593.

OWEN, Larry, et al. Street and traffic simulation: traffic flow simulation using corsim, in *Proceedings of the 32nd conference on Winter simulation*. Society for Computer Simulation International. 2000; pp. 1143– 1147.

PANTOJA, Andres, OBANDO, German and QUIJANO, Nicanor. Distributed optimization with information-constrained population dynamics, *Journal of the Franklin Institute*. 356 (2019); pp. 209–236.

PAPAGEORGIOU, Markos, et al. Review of road traffic control strategies, *Proceedings of the IEEE*. 91 (2003); pp. 2043–2067.

PARK, Byungkyu and SCHNEEBERGER, John. Microscopic simulation model calibration and validation: case study of vissim simulation model for a coordinated actuated signal system. *Transportation Research Record*. 1856 (2003); pp. 185–192.

PEDRAZA, Luis, HERNANDEZ, Cesar y LOPEZ, Danilo. Control de tráfico vehicular usando anfis. *Ingeniare. Revista chilena de ingeniería*. 20 (2012); pp. 79– 88.

CELIS PEÑARANDA, Jose, et al. Control adaptativo para optimizar una intersección semafórica basado en un sistema embebido. *Ingeniería y ciencia*. 12 (2016); pp. 169–193.

PEREZ, Fernando, et al. Análisis del flujo de tráfico vehicular a través de un modelo

macroscópico. Dyna. 81 (2014).

QIAN, Yong-Sheng and WANG, Chun-Lei. Urban traffic signal control based on three swarms cooperative-particle swarm optimization algorithm., Jisuanji Gongcheng yu Yingyong (Computer Engineering and Applications). 43 (2007); pp. 187–189.

QUIJANO, Nicanor, et al. The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory. IEEE Control Systems. 37 (2017); pp. 70–97.

ROBERTSON, Dennis. 'tansyt' method for area traffic control. Traffic Engineering & Control. 8 (1969).

ROBLES, Daniel, ÑAÑEZ, Pablo y QUIJANO, Nicanor. Control y simulación de tráfico urbano en Colombia: Estado del arte. Revista de ingeniería. (2009); pp. 59–69.

SANTAMARIA, Maria Violeta y RAMOS, Mario Fernando. Semáforos inteligentes para la regulación del tráfico vehicular. Ingeniería: Ciencia, Tecnología e Innovación. 1 (2014); pp. 37–37.

TAYLOR, Peter and JONKER, Leo. Evolutionary stable strategies and game dynamics, Mathematical biosciences. 40 (1978); pp. 145–156.

TETTAMANTI, Tamas, et al. Iterative calibration of vissim simulator based on genetic algorithm. Acta Technica Jaurinensis. 8 (2015); pp. 145–152.

TETTAMANTI, Tamas and VARGA, Istvan. Development of road traffic control by using integrated vissim-matlab simulation environment. Periodica Polytechnica Civil Engineering. 56 (2012); pp. 43–49.

TETTAMANTI, Tamas, et al. Model predictive control in urban traffic network management. in Control and Automation, 2008 16th Mediterranean Conference on, IEEE. 2008; pp. 1538–1543.

WEBSTER, F.V. Traffic signal settings, road research technical paper no. 39. Road

Research Laboratory. 39(1958).

YANG, Hui, et al. Study of traffic flow adjustment methods to congestion area based on transmodeler, Science Technology and Engineering. 8 (2011); pp. 022.

ZHOU, Zhao, et al. Two-level hierarchical model- based predictive control for large-scale urban traffic networks. IEEE Transactions on Control Systems Technology. 25 (2017); pp. 496–508

ANEXOS

ANEXO A

REPOSITORIO DE CÓDIGOS Y ARCHIVOS

En el siguiente enlace se puede visitar los diferentes códigos utilizados en este trabajo.

<https://github.com/AndresInsuasty/Control-de-traffic.git>

Resultados informe

Sincronización

Redes de tráfico en VISSIM

F.V. Webster Código

PSO Código

Replicator Dynamics Codigo

Guide para visualización de resultados

El algoritmo de MPC y la actualización de la API para algoritmos de versiones anteriores a la 7.0 de VISSIM puede ser encontrada en el siguiente enlace <https://github.com/AndresInsuasty/API-Vissim-Matlab-Algortimos.git>

ANEXO B

ARTICULO CONGRESO IEEE

En la siguiente hoja se presenta el paper sometido al congreso internacional 4th IEEE COLOMBIAN CONFERENCE ON AUTOMATIC CONTROL, a desarrollarse en octubre de 2019 en Medellín, Colombia.

Control Strategies for Urban Traffic Tested in a Co-Simulation Framework

Andres Insuasty, Gabriel Coral, and Andrés Pantoja

Departamento de Electrónica

Universidad de Nariño

Pasto, Colombia

andresinsuastyd10@gmail.com, gabrielcoralc95@gmail.com, ad_pantoja@udenar.edu.co

Abstract—The management of urban traffic is a challenging task due to its complexity and importance in modern cities. This work proposes two control strategies for a study case developed in a co-simulation platform between a specialized traffic simulator (VISSIM) and Matlab to implement the control methods. The proposed strategies (Particle Swarm Optimization and Replicator Dynamics) are compared with a fixed-time technique (Webster) to analyze the influence of the adaptive methods on some performance indexes such as queue lengths, vehicle fluxes, and fuel consumption in a real network.

Index Terms—Co-simulation, Particle Swarm Optimization, Replicator Dynamics, Urban Traffic Control.

I. INTRODUCTION

Urban transportation as a whole is a high-impact system to people's daily life and cities development. The negative consequences of saturated traffic levels include significant time losses and increasing pollution due to the high fuel consumption. As an additional consequence, the excess of traffic produces health issues related to unacceptable sound environments and low air quality, with direct and indirect implications on general welfare, sleep disorders, anxiety, depression, and even, heart diseases [1].

The development of advanced traffic-management strategies and novel technologies on smart signal lights [2] allow the transportation system to reduce the traffic congestion. These new techniques are based on a coordinated system of lights, sensors, and controllers in charge of a growing vehicle density over almost all cities. The different methodologies for traffic control can be divided in two groups. The open-loop strategies are implemented with fixed times in the lights, taking into account historical data and statistics on the vehicle fluxes. On the other hand, the closed-loop strategies are more sophisticated and sensitive to the traffic demand in each moment.

The closed-loop control is a complex algorithm based on real-time measurements of the vehicle inputs to each intersection. For instance, an ANFIS model (Adaptive Network Based Fuzzy Inference Systems) is used in [3] to optimize the flux in a system with 8 street intersections, by controlling synchronization and phasing times. As a result, the vehicular density and the average number of queued vehicles are reduced by 42.6% and 61.7%, respectively, in comparison to the

results for open-loop strategies. In a similar way, [4] presents evolutionary algorithms to solve a multiobjective optimization problem in a road corridor, obtaining improvements for the speed of traffic in buses (15.3%) and private cars (24.8%). Moreover, researches on advanced control strategies show significant improvements over several performance indices. Control techniques such as optimal control (MPC) [5], [6], adaptive control [7], bioinspired strategies [8], [9], and intelligent control [10], present better fluxes, fuel consumption, and delays, among other indicators, in comparison to fixed-time techniques.

Due to the complexity of a real traffic system, the validation of the control strategies is not carried out in actual transportation networks or general-purpose simulators. The stochastic behavior of vehicles are emulated by specialized platforms that take into account many characteristics of individual cars and groups of vehicles, such as constraints in fluxes, allowed directions, and geometry of the ways, just to name a few. In this work, we use VISSIM as a platform with large available technical information, high precision in simulations, and extended use in research [11]. To program the interaction between the controller and the traffic system, we use a co-simulation environment, where the plant is simulated in VISSIM to provide the data about vehicle fluxes to the control scheme programmed in MATLAB, in charge of processing the information and feeding the control signals (light times in the semaphores) back to the traffic system.

In complement, as the main contribution of the work, we propose two control strategies (a modified Particle Swarm Optimization and Replicator Dynamics) over a real scenario of two street intersections. In these algorithms, the time of each signal is a resource that is allocated in fixed and variable cycles in response to the stochastic changes in the shared fluxes of the interconnected lanes. The results of the control methods are compared with an open-loop controller through several performance indices.

The rest of the work presents the description of the proposed strategies in the next section. Then, the co-simulation structure is presented in Section III to introduce the study case. Finally, the simulation results are analyzed and compared in Section IV.

II. TRAFFIC-CONTROL STRATEGIES

Traffic-management systems use coordinated semaphores in the street intersections to direct the vehicle flux, which is the amount of cars crossing by a point in a time unit. The flux depends on the signals, corresponding to a single light or light combinations at the same time, such as the ones required for straight-and-turn signaling.

The term phase is defined as the time associated with the signaling of one or more independent movements that receive simultaneously the right-of-way. A phase begins with the loss of the right-of-way of the vehicles that are in conflict with the ones that win the right to cross. With several traffic lights in a set of intersections, a cycle is defined as the time required to complete a sequence of all the phases in the semaphores.

Some of the most used measures to evaluate the performance of traffic systems include the queue length in a street, which is the distance in meters of the number of vehicles pending for the right-of-way. The saturation percentage at an intersection is a measure of the current demand related to the total capacity that depends mainly on the geometry of the streets. Moreover, other indices can be calculated based on the previous measures such as the fuel consumption, waiting times, delays, and polluting emissions, among others.

A general classification of traffic-control strategies defines the fixed-time and variable-time controllers. The first one refers to a control where the cycle, duration, and sequence of intervals are invariable and defined previously. The calibration depends on historic data that must be evaluated every certain time. The controllers with variable time are more complex and allow the system to improve the performance of the network according to measurements of the vehicle flux at each intersection. They are usually implemented in three basic steps, starting with the collection of information (vehicle amounts, queue lengths, and saturation values, among other measures) to calculate some indicators that must be optimized. Then, an algorithm is executed for the adjustment of the times in the traffic lights (i.e., the control variables), and finally, adjust the parameters to improve the performance in the next cycle [7].

A. A Fixed-Time Control Strategy

The control methods with fixed times determine the phases of each traffic light based on data and statistical measurements prior to the implementation [12], [13]. Based on field observations and simulation of traffic conditions, F. V. Webster in [14] determines an optimal cycle time length given by

$$C = \frac{1.5L + 5}{1 - \sum_{i=1}^{\varphi} Y_i}, \quad (1)$$

where L is the loss time in seconds per cycle, φ is the number of phases, and Y_i is the maximum value of the ratio between the average and saturation flows for lane i . The loss time is given by the duration of the phases in yellow and red, which are respectively defined by

$$T_y = t + \frac{v}{2a} \quad \text{and} \quad T_r = \frac{W + L_v}{v}, \quad (2)$$

where t represents the perception or reaction time of the driver, v is the average velocity of the vehicles, a is the deceleration rate, W is the width of the intersection, and L_v is the queue length of the vehicles. Then, the loss time in each cycle is $L = T_r + T_y$.

To calculate the minimum delay at the intersection, the total effective green time of the cycle ($C - L$) is proportionally distributed according to the saturation ratio values Y_i

$$G_i = \frac{Y_i(C - L)}{\sum_{j=1}^{\varphi} Y_j}, \quad (3)$$

expression in which G_i is the effective green time for the phase i . With the previous formulae, the optimal times are established for each phase, which are permanently programmed in the traffic-control system.

B. Particle Swarm Optimization (PSO)

This technique is based on models of social behaviors, such as the movements of flocks of birds or schools of fish. This algorithm optimizes a problem with a population of individuals that represents the candidate solutions. Individuals can move around the search space according to simple criteria and restrictions, looking for a better solution [9], [15].

The update criterion for the movement of the i^{th} individual takes into account the best personal position ($pBest_i$) and the best global position ($gBest$) of an agent in the group. These two parameters, including a random factor, influence the speed of movement of each individual in the search space at the next time interval. In each iteration of the algorithm, the velocity and position of the agents are updated according to [15]

$$V_i(k+1) = wV_i(k) + c_1r_1(pBest_i - X_i(k)) + \dots \\ \dots + c_2r_2(gBest - X_i(k))$$

and

$$X_i(k+1) = X_i(k) + V_i(k+1). \quad (4)$$

In these equations, $X_i(k)$ is the position of the i^{th} individual at the instant k , w is the inertial weight, c_1 and c_2 are positive coefficients that determine the relative influence of $pBest_i$ and $gBest$, respectively, and r_1 , r_2 are random values in the interval $[0, 1]$ to provide the stochastic component of the search. The position of each individual is evaluated by a fitness function. If the new position of the individual is better than the previous one, $pBest_i$ is replaced by X_i , and if a better global position is discovered, $gBest$ is also updated.

To use the PSO algorithm in traffic control, it is assumed that there exist φ movement phases, including all the traffic lights. Each phase corresponds to one individual belonging to the PSO swarm, where the position of each agent X_i , represents the green time of its phase. According to the problem, the objective function or *fitness* that each individual should minimize is defined by

$$f_i = \frac{Qlen_i}{Cap_i}, \quad (5)$$

which represents the relationship between the current queue length Q_{len_i} and the maximum capacity of the lane Cap_i of the semaphore i . Clearly, the minimization of these fitness functions represents an improvement in the network and is a direct consequence of the implementation of the green time of each phase X_i .

The algorithm, at the beginning of each cycle C , evaluates the queue length for each lane to obtain the value of the fitness (5). Then, the speed and position are calculated using (4), taking into account the more recent values of $pBest_i$ and $gBest$. These values are updated if the fitness decreases with respect to the previous iteration, replacing the old values with the new best X_i . Thus, the algorithm at each iteration evaluates the traffic conditions and updates the control trying to converge to a minimum queue-length value.

However, there are some drawbacks related to this classic PSO configuration, given that the traffic conditions (lengths and flows) change constantly over time. The problem lies in the updating criteria for $pBest$ and $gBest$, values that only change if the fitness at time k is less than that the one obtained in $k - 1$. If the conditions of the route worsen in consecutive cycles, $pBest_i$ and $gBest_i$ do not change, causing the system to stagnate in an unfavorable allocation (i.e., a good road condition is assumed when in reality it is worse). To solve this condition, we propose to modify the algorithm by resetting the values of $pBest_i$ and $gBest$ more frequently. For this purpose, $pBest_i$ is updated if the value of the fitness has not improved during two consecutive cycles. If the new fitness does not provide a value better than the stored $pBest_i$ in these two intervals, this parameter is replaced by the current X_i . Regarding $gBest$, this parameter is updated by the minimum $pBest$ of all the population when $gBest$ achieve a value less than a 10% of the saturation level. In addition, it is worth noting that the value of c_2 is considerably less than c_1 , since the problem prioritizes the value of the best personnel over the best global.

1) *Fixed-Cycle PSO*: In a fixed cycle time, the summation of all phases remains constant, i.e., $\sum_{i=1}^{\varphi} X_i = C$. However, in PSO each X_i is the result of an evolutionary process, so in general, the restriction is not fulfilled. To satisfy the constraint, at the end of each iteration, the positions of the individuals are adjusted by calculating the difference between the cycle C and $\sum_{i=1}^{\varphi} X_i \neq C$. Then, the values of X_i are proportionally to obtain the required fixed cycle.

2) *Variable-Cycle PSO*: To allow the cycle length to change according to the road conditions and provide more flexibility in the optimization of green times, we propose that the sum of the phases may vary within a range $C - \frac{a}{2} < \sum_{i=1}^{\varphi} x_i < C + \frac{a}{2}$, where a is a positive tuning constant. If the limits are exceeded, the surplus (or scarce) time is cut (or added) to each individual proportionally. Consequently, the algorithm determines an optimal cycle time for the current system situation.

C. Replicator Dynamics (RD)

The RD model describes how certain characteristics of a population can evolve through the mutual interaction of individuals and the evaluation of their welfare state [16]. In analogy, urban traffic can be seen as a population system, since each vehicle has an individual behavior and all of them should be organized pursuing certain objectives (e.g., taking the shortest path or getting the lower fuel consumption). In these scenarios, the population is characterized by presenting replicable behaviors among individuals in search of common goals.

To describe the RD strategy, we assume that there is a set of n habitats in a natural environment, where a large number of individuals of a species must be distributed. In this case, $p_i(t) \geq 0$ is the number of individuals that choose the habitat i , while $p = [p_1, \dots, p_n]^T$ is the population state, and P_{tot} is the total population in the environment. The changes in each population define the RD model given by

$$\frac{dp_i}{dt} = p_i(f_i(p) - \bar{f}(p)), \forall i = 1, \dots, n, \quad (6)$$

where $f_i(p)$ represents the fitness function perceived in the i^{th} habitat and $\bar{f}(p)$ is the weighted average of the fitness functions,

$$\bar{f}(p) = \frac{1}{P_{tot}} \sum_{i=1}^n p_i f_i(p). \quad (7)$$

An interesting consequence of the definition in (7), is that the total population is kept constant, i.e., $\sum_{i=1}^n p_i = P_{tot}$ for all the time. [17].

In addition, observing (6), the steady state is achieved when $p_i^*(f_i(p^*) - \bar{f}(p^*)) = 0$, where $p^* = [p_1^* \ p_2^* \ \dots \ p_n^*]^T$ is an equilibrium point. If $p_i^* > 0 \ \forall \ i$, the condition $f_i(p^*) = \bar{f}^*$ is satisfied, where $\bar{f}^*$ is the average fitness in equilibrium. Then, at steady state, all habitats have the same welfare. Appropriate analogies with the population schemes have shown that certain choices of fitness functions lead to find a stable equilibrium point that solves various engineering optimization and control problems [18].

1) *Fixed-Cycle RD*: Assuming that the total number of habitats n represents the number of phases φ of the traffic network, the number of individuals in each habitat p_i represents the time in green for the phase i . Thus, the constant population condition is $\sum_{i=1}^{\varphi} p_i = P_{tot} = C$, where C is the cycle time, and the fixed-cycle restriction is respected by the RD for all the time. Now, an appropriate fitness function for the traffic system is given by

$$f_i(p) = \frac{w_1 q_i + w_2 Q_i}{w_1 S_{q_i} + w_2 S_{Q_i}}, \quad (8)$$

where, for the time cycle C , q_i is the vehicle flow, Q_i is the queue length, and S_{q_i} and S_{Q_i} are the saturation values of flow and queue for the phase i , respectively. Moreover, w_1 and w_2 are adjustable weights to determine the influence of flows or queues, taking into account that $w_1 + w_2 = 1$.

The fitness function is determined by the queue length and the vehicular flow in a proportional relationship given that larger fitnesses require longer green times. The combination of these two measures in the fitness definition allows the RD to take advantage of each one. For instance, estimating the vehicle flow requires to count the number of vehicles entering the network in the next cycle. On the other hand, the queue length estimation only takes into account the current status of the network and has slow changes compared to the flow when the system is saturated. In these cases, the rate of vehicles entering the network is very low and the calculation of the fitness only with the flow would estimate that the network is not saturated. This is a wrong indication and it is solved by adding the queue-length control, which is able to identify the clogging and can give priority to the controlled street. In this way, these two parameters together can be adapted to different states of the transit network. Finally, the weight parameters w_1 and w_2 are adjusted to determine the priority of the variables. In this case, $w_1 > w_2$, offering better performance under normal traffic conditions.

2) *Variable-Cycle RD*: To adjust the replicator dynamics to a variable cycle, the restriction $\sum_{i=1}^{\varphi} p_i = C$ must be relaxed to allow $\sum_{i=1}^{\varphi} p_i < C$. In this case, the time is a resource with an upper bound to be distributed in each phase. If one habitat is added to the environment in (6), we have $\varphi + 1$ equations and a new phase in the traffic system, obtaining $\sum_{i=1}^{\varphi+1} p_i = C$. The additional phase is a “fictitious agent” to [16], which is not assigned to any semaphore but acts as a surplus variable storing the time that is not used in the real phases. The fitness function for the fictitious phase is defined by

$$f_{\varphi+1}(p) = \frac{w_1 \sum_{i=1}^{\varphi} q_i + w_2 \sum_{i=1}^{\varphi} Q_i}{w_1 \sum_{i=1}^{\varphi} S_{q_i} + w_2 \sum_{i=1}^{\varphi} S_{Q_i}}, \quad (9)$$

which takes into account the fitness functions of all the real phases by monitoring the complete state of the system. The average fitness function must consider the new phase in the form

$$\bar{f}(p) = \frac{1}{P_{tot}} \sum_{i=1}^{\varphi+1} p_i f_i(p). \quad (10)$$

With the addition of the new phase, all fitnesses at equilibrium are still the same (including the fictitious one), maintaining the stability and optimal properties of the equilibrium point [16]. Furthermore, the control with an upper limit for the cycle time of the intersections can be better adapted to the dynamic conditions of a real traffic flow, without changing the structure of the controller.

III. SIMULATION ENVIRONMENT

In this work, we propose a co-simulation environment similar to the one in [11], with the plant in VISSIM as the specialized traffic simulator, and Matlab in charge of processing the information and sending the control signals back to the system.

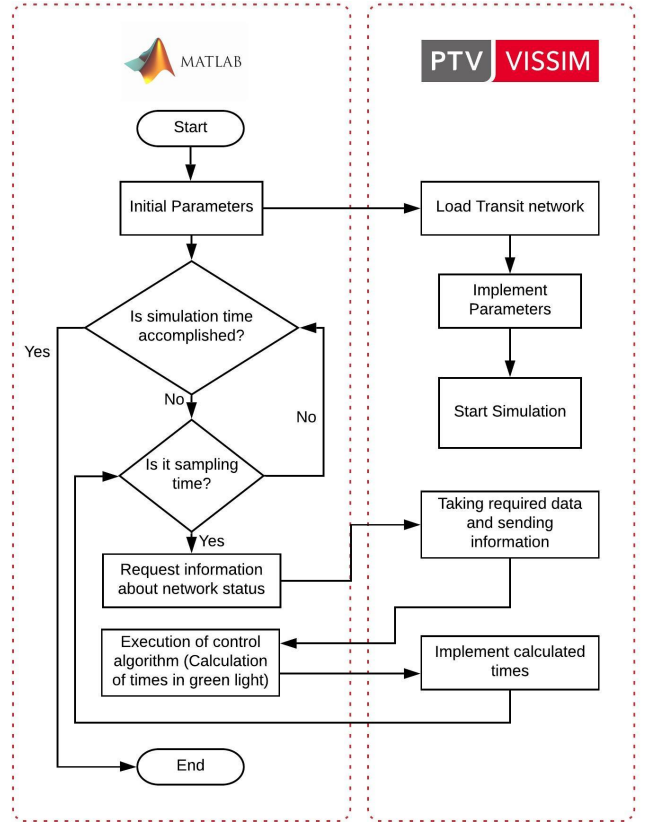


Fig. 1. Flow diagram describing the operation of the co-simulation strategy between Matlab and VISSIM for traffic control.

To communicate the two platforms, we use the VISSIM-COM interface, where the functions and parameters of the simulator can be manipulated by programming commands in different languages (handling COM objects), such as C++, Visual Basic, Java, Python, and Matlab, among others. With this interface, the user can manipulate directly the attributes of the internal objects of the traffic simulation. The program in Matlab uses the COM port to import and export information, while the implementation of control algorithms is eased by the well-known control and optimization toolboxes.

Figure 1 shows a general scheme of the proposed co-simulation operation. First, the initial parameters are established for the traffic system (simulation time, traffic network to be used, vehicular inflow, and seed of randomness, among others) that define the work scenario. These parameters are set in Matlab and then, sent to VISSIM to initiate the simulation. Within the Matlab environment, the algorithm waits until the time for data sampling is completed, requesting the relevant information from the traffic simulator (e.g., queue length, flows, delays, and fuel consumption). With this information, the control strategy calculates the green time for each traffic light, and the control variables are fed back and implemented in VISSIM, repeating the sequence continuously until the simulation time is reached.

IV. SIMULATION RESULTS AND ANALYSIS

Next results are obtained in a network of two connected intersections (inspired by a real street network in the city of Pasto, Colombia), analyzing the behavior of the algorithms in cases where the output flow of an intersection becomes the inflow to the other. The analysis of the connection effects of several lane intersections is important since it represents most of the real traffic systems. In these cases, the influence of the road capacity on the system's performance is not relevant if an inefficient control strategy is implemented.

The study case is composed by two connected intersections with 8 traffic lights, as it is shown in Figure 2. It is worth noting that the intersections share vehicular flow, so the performance of the system depends on an implicit coordination between the controllers of each semaphore. The simulation time is 18000 seconds (5 hours), with a sampling time of $T = 115$ s, which corresponds to the optimal cycle calculated by the Webster's technique. The roads shown in Figure 2 have similar geometric characteristics.

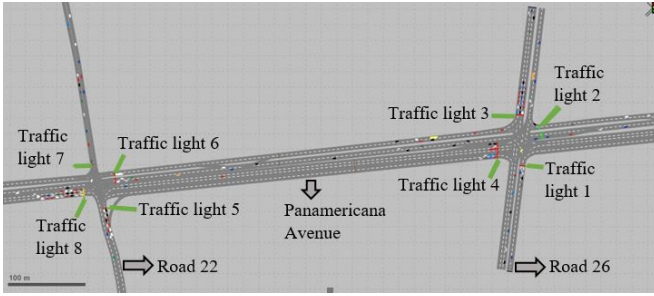


Fig. 2. Street network of two intersections with 8 traffic lights inspired by the system made up of Panamericana Avenue between streets 22 and 26, Pasto, Colombia.

The results for the accumulated average queue length obtained with the proposed algorithms are shown in Figure 3. The data show that the adaptive algorithms have a better performance against the technique of fixed time. In particular, the reductions in queue lengths for PSO with fixed and variable cycle are about 31% and 39%, respectively. The results for RD are even better, obtaining an improvement of 36% and 43% at the end of the simulation (for fixed and variable cycles). This results also imply a decrease in fuel consumption and waiting time for the vehicles, results that are shown in Table I. The indices presented in the table are the averages for mean maximum queue lengths in meters (Qlen and QlenMax, respectively), the number of vehicles passing through the intersections (VehsAll), the delay for each vehicle measured in seconds (VehsDelayAll), the stop delays measured in seconds (StopDelayAll), the number of stops (StopAll), the emissions of carbon monoxide, nitrogen oxide, and volatile organic compounds measured in grams (EmissionCO, EmissionNOX and EmissionVOC), and the fuel consumption measured in gallons (FuelConsumption). It is remarkable that all the indices have a similar behavior, with the best results for the RD algorithm followed by the PSO, both with variable cycle.

TABLE I
ACCUMULATED AVERAGE PERFORMANCE INDICES FOR THE FIXED-TIME SIMULATIONS (WEBSTER), PSO, AND RD WITH FIXED AND VARIABLE CYCLES (SUBSCRIPTS FC AND VC, RESPECTIVELY).

	Qlen	QlenMax	VehsAll	VehsDelayAll	StopDelayAll
PSO_{FC}	8081.5	35424.8	36055.5	14495.1	11873.7
PSO_{VC}	7160.1	33028.5	36051.5	12943.3	10400.7
RD_{FC}	7410	30370.8	36061.5	13554.8	10986.4
RD_{VC}	6147	26513.9	36092.5	11378.3	8846.3
Webster	11718	53501.1	36590	13852.8	11398.3
	StopsAll	EmissionCO	EmissionNOX	EmissionVOC	FuelConsumption
PSO_{FC}	283.7	48626	9460.9	11269.5	695.6
PSO_{VC}	273.9	45525.3	8857.5	10550.9	651.2
RD_{FC}	276.6	46667.3	9079.8	10815.6	667.6
RD_{VC}	272.9	42933.4	8353.3	9950.2	614.2
Webster	269.6	47267.2	9196.5	10954.6	676.2

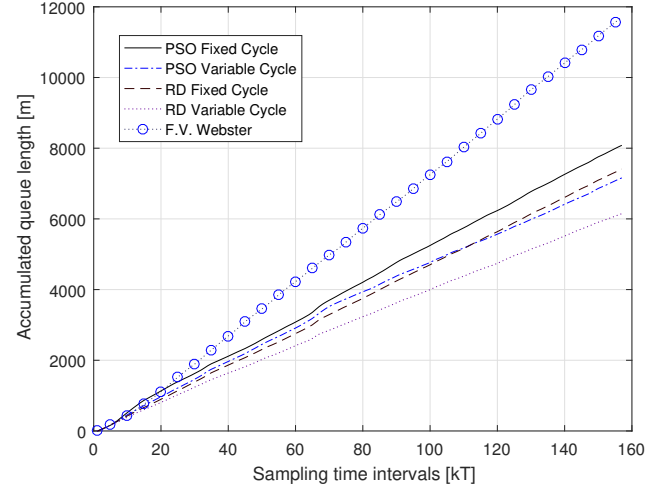


Fig. 3. Accumulated queue length in time for the network of two intersections.

Based on the results, we can observe that the control with fixed cycle (in PSO and RD) is restrictive, since the system does not react appropriately to high saturation conditions of the streets. In other words, regardless of the system's conditions, the entire time resource must be assigned, even though some of the intersections are not occupied and some others are very saturated. On the other hand, in the strategies with variable cycles the total cycle can be reduced depending on the saturation, meaning that all phases can last just a few to allow a faster evolution in the traffic lights, and giving right-of-way to small flows, more frequently, at each saturated lane.

Furthermore, for a variable cycle, a delay in the green-time values of the consecutive traffic lights is observed in some simulation instants, generating a phase coordination so-called green wave [19]. This coordination is related to the distance between intersections and the average speed of the vehicles. Thus, when a vehicle leaves the first intersection and arrives at the next one, it is highly probable that it will also pass. If a vehicle goes into a green wave, it will have a route with a minimum delay and possibly without stops, decreasing fuel consumption and gas emissions, among other performance indices. It is noteworthy that although the proposed algorithms carry out an independent control at each intersection (without

sharing information), it is possible to notice a coordination of phases in the simulations. This is observed by an improvement of 3% in the queue length with replicator dynamics for the network of two intersections compared to tests performed with a single intersection, corresponding to the one on the right side of the Figure 2. realree

V. CONCLUSIONS

The proposed adaptation for the PSO and RD techniques is appropriate for the control of traffic systems, taking into account the characteristics of the streets and sensor information for the feedback. In addition, the co-simulation scheme allows the system to emulate real-implementation conditions with flows adapted to actual road characteristics provided by a specialized traffic platform. Moreover, using the Matlab toolboxes facilitates the implementation of advanced controllers. As future work, we propose to carry out tests to check synchronization, study algorithms with shared information between intersections, and analyze networks with a large number of traffic lights.

ACKNOWLEDGMENTS

We would like to thank Pablo Ñáñez, Iván Gutierrez for their important inputs and discussions around the VISSIM API. We also thank to Gilberto Rueda from PTV Mexico for providing us a free student version of VISSIM for the tests in this work.

REFERENCES

- [1] J. Morales Pérez and J. A. Fernandez Gomez, "Análisis discriminante de algunas variables que influyen en la contaminación acústica debida al tráfico urbano en una gran ciudad," *Revista Ingenierías Universidad de Medellín*, vol. 11, no. 21, pp. 13–22, 2012.
- [2] M. B. Santamaría and M. R. Moscol, "Semáforos inteligentes para la regulación del tráfico vehicular," *Ingeniería: Ciencia, Tecnología e Innovación*, vol. 1, no. 1, pp. 37–37, 2014.
- [3] L. F. Pedraza, C. A. Hernández, and D. A. López, "Control de tráfico vehicular usando anfis," *Ingeniare. Revista chilena de ingeniería*, vol. 20, no. 1, pp. 79–88, 2012.
- [4] Á. Acuna, E. Arreche, and S. Nesmachnow, "Planificación de tráfico y transporte colectivo en el corredor garzón utilizando un algoritmo evolutivo," *Universidad de la República, Montevideo*, 2015.
- [5] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. De Schutter, "Sustainable model-predictive control in urban traffic networks: Efficient solution based on general smoothing methods," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 813–827, 2018.
- [6] B. HomChaudhuri, A. Vahidi, and P. Pisu, "Fast model predictive control-based fuel efficient control strategy for a group of connected vehicles in urban road conditions," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 760–767, 2017.
- [7] J. C. Peñaranda, C. E. Amado, S. S. Mora, S. C. Casadiego, B. M. Delgado, and J. R. Mateus, "Control adaptativo para optimizar una intersección semafórica basado en un sistema embebido," *Ingeniería y ciencia*, vol. 12, no. 24, pp. 169–193, 2016.
- [8] A. Jovanović, M. Nikolić, and D. Teodorović, "Area-wide urban traffic control: A bee colony optimization approach," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 329–350, 2017.
- [9] Y.-S. Qian and C.-L. Wang, "Urban traffic signal control based on three swarms cooperative-particle swarm optimization algorithm," *Computer Engineering and Applications*, vol. 43, no. 14, pp. 187–189, 2007.
- [10] M. H. Khooban, N. Vafamand, A. Liaghat, and T. Dragicevic, "An optimal general type-2 fuzzy controller for urban traffic network," *ISA transactions*, vol. 66, pp. 335–343, 2017.
- [11] T. Tettamanti, A. Csikós, I. Varga, and A. Eleőd, "Iterative calibration of vissim simulator based on genetic algorithm," *Acta Technica Jaurinensis*, vol. 8, no. 2, pp. 145–152, 2015.
- [12] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.
- [13] D. Robles, P. Ñáñez, and N. Quijano, "Control y simulación de tráfico urbano en colombia: Estado del arte," *Revista de ingeniería*, no. 29, pp. 59–69, 2009.
- [14] F. Webster, "Traffic signal settings, road research technical paper No. 39," *Road Research Laboratory*, 1958.
- [15] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence*. Elsevier, 2001.
- [16] A. Pantoja, G. Obando, and N. Quijano, "Distributed optimization with information-constrained population dynamics," *Journal of the Franklin Institute*, vol. 356, no. 1, pp. 209–236, 2019.
- [17] J. Hofbauer and K. Sigmund, *Evolutionary games and population dynamics*. Cambridge university press, 1998.
- [18] N. Quijano, C. Ocampo-Martinez, J. Barreiro-Gomez, G. Obando, A. Pantoja, and E. Mojica-Nava, "The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory," *IEEE Control Systems*, vol. 37, no. 1, pp. 70–97, 2017.
- [19] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Evaluation of green wave policy in real-time railway traffic management," *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 607–616, 2009.