Universidad de la República

PROYECTO DE GRADO

Ingeniería en Computación

# Algoritmos evolutivos aplicados a la sincronización de semáforos en el Corredor Garzón

Alvaro Acuña
Efraín Arreche

2014

Supervisor: Sergio Nesmachnow

Centro de Cálculo - Instituto de Computación
Montevideo, Uruguay

Algoritmos Evolutivos aplicados a la sincronización de semáforos en el Corredor Garzón
Acuña, Alvaro - Arreche, Efraín
Proyecto de Grado
Instituto de Computación - Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay, Julio de 2015

# Algoritmos evolutivos aplicados a la sincronización de semáforos en el Corredor Garzón


## Resumen

**Este proyecto propone el estudio de la sincronización de semáforos como problema de optimización multiobjetivo, y el diseño e implementación de un algoritmo evolutivo para resolverlo con alta eficacia numérica y desempeño computacional. Se plantea como caso de estudio la sincronización de semáforos en el Corredor Garzón (Montevideo, Uruguay), un escenario urbano muy interesante por su complejidad, relacionada con el número de cruces, calles y semáforos, y por la problemática del tráfico en la zona. Además, las autoridades responsables admitieron la existencia de problemas relacionados con la sincronización de los semáforos, por lo que todavía hay espacio para la mejora de los tiempos promedio de los viajes en el Corredor Garzón.**

**El problema de sincronización de semáforos es un problema de optimización NP-difícil por lo que los métodos computacionales exactos sólo son útiles en instancias de tamaño reducido. Este trabajo propone utilizar un algoritmo evolutivo para calcular una configuración eficiente de los semáforos, maximizando la velocidad media del transporte colectivo y de otros vehículos. El enfoque seguido comprende la obtención de datos reales relacionados con la red vial, el tráfico y la configuración de los semáforos, y la utilización del simulador de tráfico SUMO para generar los datos requeridos por el algoritmo evolutivo.**

**El análisis experimental compara los resultados numéricos del algoritmo evolutivo con el escenario base que modela la realidad actual. Complementariamente se desarrolla un escenario alternativo cuyo objetivo es mejorar la velocidad promedio de ómnibus y otros vehículos realizando modificaciones sobre el escenario base. Los resultados demuestran que el algoritmo evolutivo propuesto logra mejoras significativas en la calidad de servicio al comparar con la realidad actual, mejorando hasta 15.3 % la velocidad promedio de ómnibus y 24.8 % la velocidad promedio de otros vehículos en el escenario base, mientras que al aplicar el algoritmo evolutivo en el escenario alternativo se obtienen mejoras de hasta 49.9 % en la velocidad promedio de los ómnibus y de hasta 26.74 % en la velocidad promedio de otros vehículos.**



**Palabras clave: Algoritmo Evolutivo, Sincronización de semáforos, Optimización de tráfico, Corredor Garzón**

# Índice general

**1. Introducción** **1**
1.1. Motivación y contexto . . . . . 1
1.2. Objetivos . . . . . 2
1.3. Enfoque . . . . . 3
1.4. Limitaciones y alcance . . . . . 3
1.5. Aportes . . . . . 3
1.6. Estructura del documento . . . . . 4

**2. Marco Teórico** **5**
2.1. Problema del tránsito vehicular . . . . . 5
2.2. Corredores urbanos de tráfico . . . . . 6
2.3. Corredor Garzón . . . . . 8
2.4. Sincronización de semáforos . . . . . 11
2.5. Algoritmos Evolutivos . . . . . 13
2.5.1. Algoritmos Genéticos . . . . . 14
2.5.2. Algoritmos evolutivos multiobjetivo . . . . . 17
2.5.3. Algoritmos evolutivos paralelos . . . . . 19
2.5.4. Implementación y entornos de desarrollo . . . . . 19
2.6. Simulación de tráfico . . . . . 21
2.7. Trabajos relacionados . . . . . 23
2.7.1. Resumen . . . . . 27

**3. Estrategia de resolución** **29**
3.1. Modelo del problema . . . . . 29
3.2. Simulaciones de tráfico . . . . . 29
3.3. Diseño del mapa de la zona de estudio . . . . . 31
3.4. Recolección de datos de semáforos y vehículos en trabajo de campo. . . . 33
3.5. Desarrollo de instancias realistas del problema . . . . . 34
3.5.1. Configuración de semáforos . . . . . 35
3.5.2. Generación de tráfico vehicular . . . . . 35
3.5.3. Generación de recorridos de ómnibus . . . . . 35
3.5.4. Resumen . . . . . 36
3.6. Formulación del problema . . . . . 37
3.7. Arquitectura de la solución . . . . . 37

**4. Implementación de la solución** **39**
4.1. Implementación: Biblioteca Malva . . . . . 39
4.2. Especificación del AG utilizado . . . . . 41
4.2.1. Representación de individuos . . . . . 42
4.2.2. Generación de la población inicial . . . . . 44
4.2.3. Función de *fitness* . . . . . 44
4.2.4. Operadores evolutivos . . . . . 44
4.3. Modelo de paralelismo e implementación . . . . . 46

**5. Análisis experimental** **49**
5.1. Plataforma de ejecución y desarrollo . . . . . 49
5.2. Experimentos de calibración paramétrica del AE . . . . . 49
5.2.1. Tiempo de simulación . . . . . 50
5.2.2. Criterio de parada . . . . . 50
5.2.3. Tamaño de la población . . . . . 50
5.2.4. Probabilidades de cruzamiento y mutación . . . . . 51
5.3. Descripción de escenarios . . . . . 51
5.3.1. Caso base o realidad actual del corredor . . . . . 51
5.3.2. Escenario alternativo . . . . . 52
5.3.3. Detalles del escenario alternativo . . . . . 52
5.4. Resultados . . . . . 54
5.4.1. Resultados de la simulación del caso base. . . . . 54
5.4.2. Resultados numéricos de la evaluación . . . . . 55
5.4.3. Valores numéricos al aplicar los cambios (escenario alternativo) . . 56
5.4.4. Resultados de la evaluación sobre el escenario alternativo . . . . . 57
5.4.5. Variación de la función de *fitness* . . . . . 58
5.4.6. Eficiencia computacional . . . . . 60

**6. Conclusiones y trabajo futuro** **63**
6.1. Conclusiones . . . . . 63
6.2. Trabajo futuro . . . . . 65

**Bibliografía** **65**

# Índice de figuras


2.1. Evolución de la venta de automóviles en Uruguay . . . . . 6
2.2. Gráfica de costos de transporte en función de la gente transportada. . . . 7
2.3. Localización del Corredor Garzón en Montevideo . . . . . 9
2.4. Perfil propuesto para el corredor Garzón. . . . . 9
2.5. Distintas fases de semáforos para dos cruces en una red de tránsito. . . . . 12
2.6. Esquema de la evolución del valor del fitness en un algoritmo evolutivo . 15
2.7. Representación binaria de un cromosoma. . . . . 15
2.8. Cruzamiento de un punto . . . . . 17
2.9. Cruzamiento uniforme . . . . . 17
2.10. Mutación por inversión binaria . . . . . 17
2.11. Modelo Maestro-Esclavo . . . . . 19

3.1. Simulación de tráfico en el simulador SUMO. . . . . 31
3.2. Mapa del Corredor Garzón . . . . . 32
3.3. Ejemplo de planilla electrónica para el conteo manual de tráfico. . . . . 34
3.4. Mapa diseñado en el TrafficModeler para la generación de tráfico. . . . . 36
3.5. Arquitectura de la solución propuesta. . . . . 38

4.1. Ejemplo de representación de las soluciones para el problema. . . . . 42
4.2. Representación de la configuración de semáforos en Sumo . . . . . 43
4.3. Representación de una fase de los semáforos para un cruce. . . . . 43
4.4. Ejemplo del operador de cruzamiento utilizado . . . . . 44
4.5. Función para la evaluación de individuos en paralelo utilizando threads, agregada al código provisto por Malva. . . . . 46
4.6. Modificaciones de la función evaluate_parents. . . . . 47

5.1. Resultados del análisis de la probabilidad de cruzamiento ($p_C$) y mutación ($p_M$) . . . . . 51
5.2. Gráfico de las paradas alternativas. . . . . 54
5.3. Comparación de la duración de los viajes en el escenario base y al aplicar el AE. . . . . 56
5.4. Comparación de la duración de los viajes en minutos entre el escenario base y el algoritmo evolutivo sobre el escenario alternativo. . . . . 58
5.5. Comparación de los *speedup* promedios para cada tipo de tráfico. . . . . 61

# Índice de tablas

2.1. Resumen de los principales trabajos relacionados relevados. . . . . 28

3.1. Resumen del revelamiento del tráfico en la zona del Corredor Garzón. . . 33

5.1. Resultados del análisis de diferentes valores para el tamaño de población. 50
5.2. Resultados numéricos del caso base. . . . . 55
5.3. Resultados numéricos del caso base. . . . . 55
5.4. Valores numéricos del escenario alternativo . . . . . 56
5.5. Mejoras del escenario alternativo. . . . . 57
5.6. Valores numéricos al aplicar el AE sobre el escenario alternativo. . . . . 57
5.7. Valores obtenidos al modificar los pesos de la función de *fitness*. . . . . 59
5.8. Análisis de la eficiencia computacional. . . . . 61

# Índice de Algoritmos

1. Algoritmo Genético . . . . . 14
2. Algoritmo Evolutivo multiobjetivo . . . . . 18
3. Algoritmo Genético de Malva. . . . . 41

# Capítulo 1

# Introducción

*He llamado a este principio, por el cual cada pequeña variación, si útil, es preservada, con el término de Selección Natural*

— Charles Darwin, El origen de las especies

Este capítulo pretende introducir al lector en el contexto general donde se enmarca el problema de sincronización de semáforos en el Corredor Garzón. Inicialmente se describen las motivaciones y el enfoque seguido para el desarrollo del proyecto. A continuación se especifican los objetivos propuestos, las limitaciones y su alcance. Para finalizar se delinea la estructura del documento, presentando una breve descripción del contenido de cada capítulo.

## 1.1. Motivación y contexto

En gran parte del mundo, el parque automotor está creciendo de forma sostenida desde hace varios años. Este crecimiento provoca serios problemas relacionados con la congestión del tráfico, afectando el desarrollo de las ciudades y la calidad de vida de las personas (Bull, 2003). Las congestiones de tráfico producen una progresiva disminución en la velocidad media de circulación de los vehículos y esto hace que disminuya el nivel de aceptación del transporte público, cuyo servicio en general es ineficiente. Para resolver este problema las autoridades suelen optar por instrumentar sistemas de transporte costosos como los *metros*, pero también se ha demostrado que existen otras opciones viables como los sistemas de Ómnibus de Tránsito Rápido (en inglés, *Bus Rapid Transit*, BRT) (Bañobre y Romero, 2009).

Uruguay, y en particular su capital Montevideo, no escapa al fenómeno global de congestión del tráfico. Aunque la situación en Montevideo no sea tan crítica como en otras ciudades del mundo, las autoridades municipales han tomado medidas para reducir el impacto de este problema, implementando un Plan de Movilidad Urbana, con el objetivo de mejorar la eficiencia del transporte público (IMM, 2010).

Uno de los puntos principales del Plan de Movilidad Urbana de la ciudad de Montevideo involucra la construcción de corredores urbanos de tránsito con carriles exclusivos para ómnibus. El primer corredor exclusivo implementado fue el Corredor Garzón, ubicado en la ciudad de Montevideo, con una extensión de 6.5 km, incluyendo 24 cruces semaforizados y vías exclusivas para ómnibus.

Desde su inauguración en el año 2012 el Corredor Garzón ha recibido críticas por no cumplir con uno de sus principales objetivos: el de agilizar el transporte público. La Intendenta de Montevideo Ana Olivera admitió que la duración de los viajes en el Corredor Garzón aumentó considerablemente. En palabras al Diario El País (2015) indicó:

> "En el momento más crítico llegaron a haber 24 minutos más, lo medimos, porque hubo un momento que se confundió mucho o se partidizó la crítica y nosotros queríamos tener los datos objetivos. Y aquí el dato objetivo era que los ciudadanos de la zona y en particular los de Lezica, no era que se sentían perjudicados, era que habían sido perjudicados. Y nosotros lo asumimos"

Los métodos para optimizar el tráfico se pueden dividir en dos categorías. Por un lado, los métodos pueden enfocarse en la modificación de las rutas, por ejemplo agregando nuevas vías de circulación, o ensanchando las existentes. Esta alternativa permite lograr mejoras en la circulación del tránsito, pero como punto negativo exigen un alto costo monetario (Litman, 2009) y disponer de espacio físico para implementar las modificaciones viales. Por otro lado se encuentran los métodos orientados a influir en el comportamiento de los conductores, que incluyen a las técnicas para configurar los semáforos y/o agregar señalizaciones de tránsito, entre otros (McKenney y White, 2013). Estos métodos son muchas veces la única opción viable cuando no se dispone de espacio físico o del capital de inversión que exigen la modificación de las vías de tránsito. Por este motivo, el estudio de estrategias para la sincronización eficiente de semáforos con el objetivo de mejorar la velocidad promedio de los viajes se presenta como un aporte importante y necesario para el desarrollo ordenado de las ciudades.

El problema de sincronización de semáforos es un problema de optimización NP-difícil (Yang y Yeh, 1996), por lo que para resolver instancias realistas es necesario utilizar heurísticas o metaheurísticas que encuentren buenas soluciones en un tiempo aceptable. Por este motivo se propone el uso de algoritmos evolutivos para resolver el problema, ya que han demostrado su utilidad para hallar soluciones de buena calidad eficientemente (Rouphail et al., 2000; Sánchez et al., 2008).

Este trabajo presenta un Algoritmo Evolutivo (AE) aplicado al problema de gestión del tráfico en la zona del Corredor Garzón mediante la sincronización de los semáforos, con el objetivo de aportar una solución eficiente e innovadora para el desarrollo de la ciudad y mejorar la calidad de vida de los ciudadanos. Las particularidades del Corredor Garzón lo convierten en un reto complejo desde el punto de vista de la investigación, como consecuencia de la extensión del tramo, de la cantidad de cruces, del número de semáforos, de las reglas de exclusividad y los diferentes tipos de tráfico, entre otros. Los resultados experimentales demuestran que el AE propuesto es capaz de mejorar la velocidad promedio de los ómnibus y otros vehículos al considerar el escenario real abordado.

## 1.2. Objetivos

Los objetivos que se plantearon al inicio del proyecto incluyeron:

1. Estudio del problema del tráfico y la sincronización de semáforos.
2. Relevamiento de información sobre trabajos relacionados en el ámbito de control de tráfico y sincronización de semáforos.

3. Diseño e implementación de un algoritmo evolutivo (AE) multiobjetivo, capaz de resolver eficientemente el problema de sincronización de semáforos en la zona del Corredor Garzón.
4. Creación de instancias realistas del problema, incluyendo un mapa y datos precisos sobre configuración de semáforos, tráfico y reglas de tránsito obtenidos de la realidad actual.
5. Aplicación de técnicas de computación de alto desempeño para mejorar el rendimiento computacional de la solución implementada.

## 1.3. Enfoque

La metodología de resolución del problema de sincronización de semáforos en el Corredor Garzón propone como primer punto el modelado del problema. En este modelo se incluye: la creación de instancias realistas basadas en datos recabados de la realidad del Corredor Garzón (tráfico vehicular, configuración de semáforos, etc) y el uso de un simulador de tráfico para analizar el comportamiento de variables útiles para el modelado, como son la velocidad promedio de ómnibus y de otros vehículos. Se plantea como un objetivo importante recabar datos precisos de la realidad, por lo tanto se incluyen tres fuentes para obtener y validar la información: datos recabados in-situ, reuniones con responsables y técnicos de la Intendencia Municipal de Montevideo (IMM), y finalmente información disponible públicamente. Como segundo punto de la metodología, se plantea el diseño y desarrollo de un AE multiobjetivo que basado en el modelo mencionado anteriormente y modificando la configuración de los semáforos, sea capaz de mejorar la velocidad promedio de ómnibus y de otros vehículos en la zona del Corredor Garzón. El tercer punto de la metodología propone la aplicación de técnicas de alto desempeño para mejorar la eficiencia computacional de la solución, al considerarse que el problema y las instancias desarrolladas son complejas y requiere un tiempo importante de ejecución.

## 1.4. Limitaciones y alcance

El alcance geográfico del presente proyecto comprende la zona del Corredor Garzón, que incluye toda la extensión del Corredor y dos caminos paralelos a cada lado del mismo. Se pretende desarrollar un modelo preciso de la realidad pero se tiene en cuenta que algunos de sus elementos serán eliminados o simplificados, por lo tanto, se buscará que estas simplificaciones no afecten los resultados de la solución propuesta.

El modelado incluye el relevamiento manual de datos en el Corredor Garzón. Debido a los errores intrínsecos que pueden ocurrir en el relevamiento manual de datos in-situ, no se puede asegurar la exactitud de los mismos. Por este motivo, se realizarán verificaciones con el objetivo de minimizar el impacto que estos errores pudieran tener.

## 1.5. Aportes

Los aportes del proyecto incluyen: *i)* el estudio del estado del arte y el marco teórico para el tratamiento del problema del tráfico vehicular; *ii)* la creación de instancias realistas del problema utilizando datos recabados in-situ y *iii)* el diseño e implementación de un AE multiobjetivo que resuelve el problema de sincronización de semáforos en el

Corredor Garzón. El AE utiliza técnicas de computación de alto desempeño y se enfoca en hallar configuraciones de semáforos que permitan mejorar la velocidad promedio de ómnibus y otros vehículos para distintas densidades de tráfico (baja, media y alta).

Estos aportes se complementan con el desarrollo de un sitio web (`http://www.fing.edu.uy/inco/grupos/cecal/hpc/AECG`), donde los interesados pueden consultar información sobre el proyecto y sus resultados.

Adicionalmente, se redactó un artículo en idioma español de 6 páginas para presentar en el congreso CICos (Congreso Internacional de Cómputo en Optimización y Software) 2015 en México y un artículo de síntesis en idioma inglés de 10 páginas, con el objetivo de tomarlo como base para la redacción de un artículo de investigación para presentar en una conferencia internacional.

Finalizando, se destaca la participación del proyecto en Ingeniería de Muestra 2014 (`https://www.fing.edu.uy/ingenieriademuestra`), donde el póster descriptivo del mismo captó la atención y fue bien recibido por parte del público presente.

## 1.6. Estructura del documento

El capítulo 2 presenta los fundamentos teóricos necesarios para comprender el resto del trabajo. Comienza describiendo el problema del tráfico vehicular y algunas formas de atacarlo, entre las que se encuentra la creación de planes de movilidad y la sincronización de semáforos. El capítulo incluye información sobre corredores urbanos de tránsito y en concreto sobre el Corredor Garzón, así como una descripción acerca de algoritmos evolutivos y simuladores de tráfico. Complementariamente, se ofrece una reseña de los principales trabajos relacionados, haciendo especial foco en algoritmos evolutivos para la sincronización de semáforos.

El capítulo 3 explica la estrategia de resolución del problema de sincronización de semáforos implementada en este trabajo. Se presenta el modelado del problema basado en: la creación de un mapa digital de la zona del Corredor Garzón compatible con el simulador de tráfico SUMO, el trabajo de campo realizado para obtener datos de la realidad actual (que serán usados en las simulaciones) y el uso de técnicas de alto desempeño para mejorar el rendimiento del algoritmo. Adicionalmente el capítulo presenta al simulador de tráfico SUMO y finaliza describiendo la arquitectura propuesta para el diseño e implementación del AE multiobjetivo.

El capítulo 4 describe la implementación del AE, presentando la biblioteca Malva y las especificaciones del AE desarrollado. Se detalla la representación del cromosoma utilizado, la inicialización de la población, la función de *fitness* y los operadores evolutivos. Para finalizar, el capítulo explica la implementación del modelo de paralelismo utilizado.

El capítulo 5 presenta el análisis experimental del AE, describe los escenarios utilizados en su evaluación y los resultados numéricos obtenidos. Se especifica el escenario que modela la situación actual del Corredor Garzón y el escenario alternativo que presenta modificaciones sobre el escenario anterior con el objetivo de mejorar la velocidad promedio de circulación. Se ofrece un análisis de los resultados obtenidos al variar la función de *fitness* y se realiza un breve estudio de la eficiencia computacional del AE.

El capítulo 6 plantea las conclusiones finales del trabajo, tanto sobre la metodología utilizada como de los resultados obtenidos. Para finalizar, el capítulo describe las posibles líneas de investigación para desarrollar en el futuro.

# Capítulo 2

# Marco Teórico

Este capítulo aborda el marco teórico necesario para comprender el desarrollo de los capítulos posteriores. Se analiza el problema del tráfico en general y las soluciones propuestas para manejarlo, incluyendo la construcción de corredores exclusivos para el transporte público. En este contexto, se presenta la descripción del Corredor Garzón y sus problemas. Luego, se presentan los simuladores de tráfico y la teoría detrás de los algoritmos evolutivos. Para finalizar, se incluye un relevamiento de trabajos relacionados para mostrar y comentar otras variantes del problema y soluciones propuestas en la literatura del área.

## 2.1. Problema del tránsito vehicular

El constante crecimiento del parque automotor ocasiona problemas relacionados con las congestiones vehiculares que afectan la calidad de vida de las personas (Bull, 2003). Este problema tiene un gran impacto en el desarrollo de las ciudades, por lo que es un componente principal en los planes estratégicos para el crecimiento de las mismas.

La congestión ocasiona una progresiva merma de la velocidad promedio de circulación, con la consecuencia del incremento en la duración de los viajes y del consumo de combustible. Este problema repercute en la contaminación atmosférica y sonora que impacta directamente en la salud de las personas; además, se genera una exigencia en las vías de tránsito que produce un deterioro mayor de calles y rutas.

Uruguay, y en particular Montevideo, no escapa a la problemática global de la congestión del tráfico. El aumento del parque automotor en la capital del país está en ascenso constante desde el 2005 (INE, 2014), y según proyecciones el crecimiento seguiría en un promedio de 4.5 % anual hasta el 2020 (BBVA Research, 2013). Este crecimiento viene de la mano con el sostenido aumento de las ventas de vehículos en el país desde el 2003 como se aprecia en la Figura 2.1. Los expertos indican que la situación de congestión ya está instalada en la ciudad y la infraestructura vial no acompasó este crecimiento. Montevideo es la ciudad con más semáforos por automóvil en Latinoamérica, con más de 620 cruces semaforizados, algunos de los cuales no están coordinados (Subrayado, 2013).

En un contexto global, el aumento en la circulación de automóviles provoca que disminuya el nivel de aceptación del transporte público, cuyo servicio en general es ineficiente. Para resolver este problema, las autoridades suelen instrumentar sistemas de transporte costosos como los *metros*, pero se ha demostrado que existen otras opciones viables como el BRT (Bus Rapid Transit / ómnibus de tránsito rápido)(Bañobre y Romero, 2009).

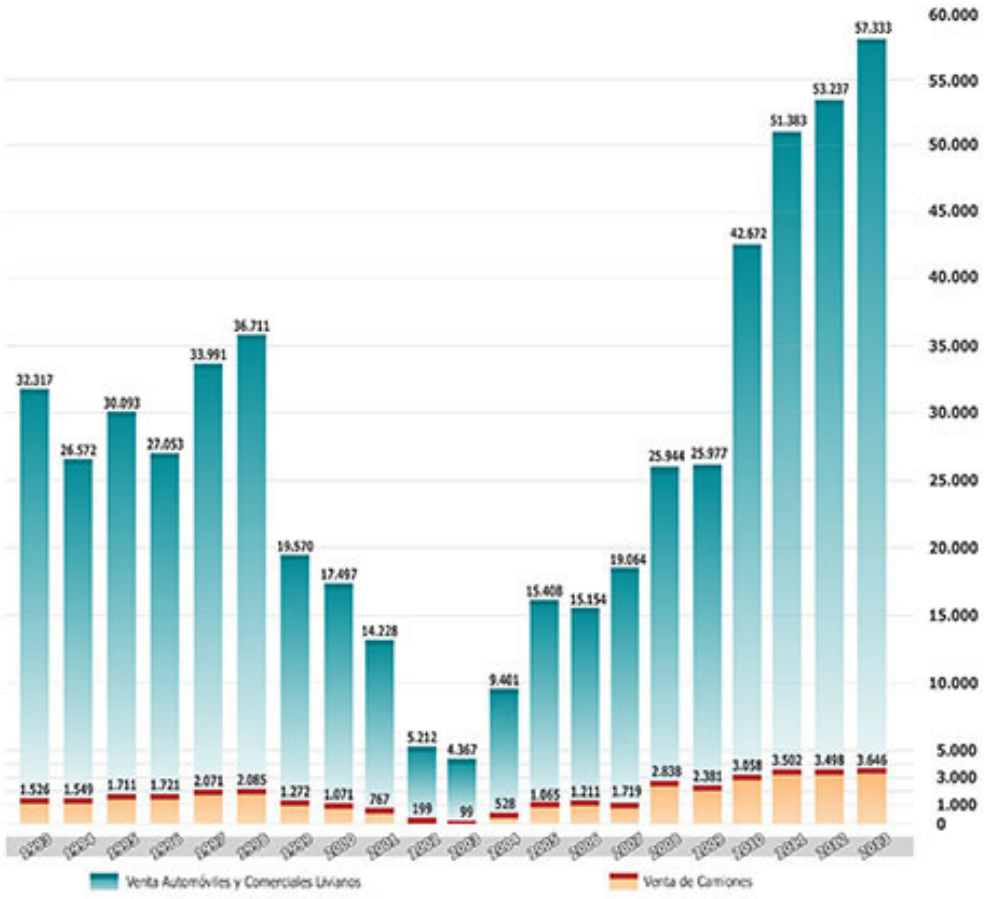


Figura 2.1: Evolución de la venta de automóviles por año. El valor más bajo corresponde al año 2003, y el más alto 2013. Imagen extraída de http://www.autoanuario.com.uy.

En el caso de la ciudad de Montevideo, para solucionar este problema se está implementando el Plan de Movilidad Urbana (IMM, 2010), con el objetivo de mejorar la eficiencia del transporte público y democratizar el acceso al mismo. El sistema de transporte está inspirado en un BRT, con la construcción de varios corredores exclusivos en la ciudad. En la siguiente sección se presenta más información sobre los BRT, corredores urbanos y en particular el Corredor Garzón.

## 2.2. Corredores urbanos de tráfico

El corredor urbano de tráfico, también llamado *corredor segregado*, se caracteriza por una separación física entre el carril de circulación de los ómnibus y los carriles para el resto del tráfico. Ésta es la principal diferencia con un concepto similar llamado *carril de sólo bus*, en donde se separan los carriles por líneas horizontales pintadas en la calle, indicando que sólo pueden circular ómnibus. Los carriles de *sólo bus* suelen ser poco exitosos como medida para agilizar el tráfico de transporte colectivo por falta de control que evite que otros vehículos los invadan.

En el caso de Montevideo el carril *sólo bus* está sobre la derecha, por lo que los vehículos privados lo invaden al virar a la derecha y los taxis lo suelen invadir para

levantar o dejar pasajeros. Al ubicar los corredores alineados en el medio de las vías de tráfico, se evitan esos problemas.

Las ciudades de países en vías de desarrollo, que tengan como objetivo fundamental conseguir mayores velocidades en el sistema vial, deben estudiar la factibilidad de instalar corredores, dado que existen muchas posibilidades de implementación y no todas pueden ser aplicables por el contexto zonal, cultural o de inversión necesaria.

BRT es una solución innovadora, de alta capacidad y de menor costo para el transporte público que puede alcanzar el rendimiento y los beneficios de sistemas ferroviarios, con un costo significativamente menor como se puede apreciar en la Figura 2.2. Se trata de un sistema integrado de movilidad basado en ómnibus para el transporte de los pasajeros a sus destinos de manera rápida y eficiente. Al mismo tiempo, ofrece la flexibilidad necesaria para satisfacer una variedad de condiciones locales. Los elementos del sistema de BRT pueden ser fácilmente personalizados a las necesidades de la comunidad e incorporan tecnologías de última generación de bajo costo que atraen a más pasajeros y en última instancia ayudan a reducir la congestión de tráfico en general (Institute for Transportation & Development Policy (ITDP), 2007).

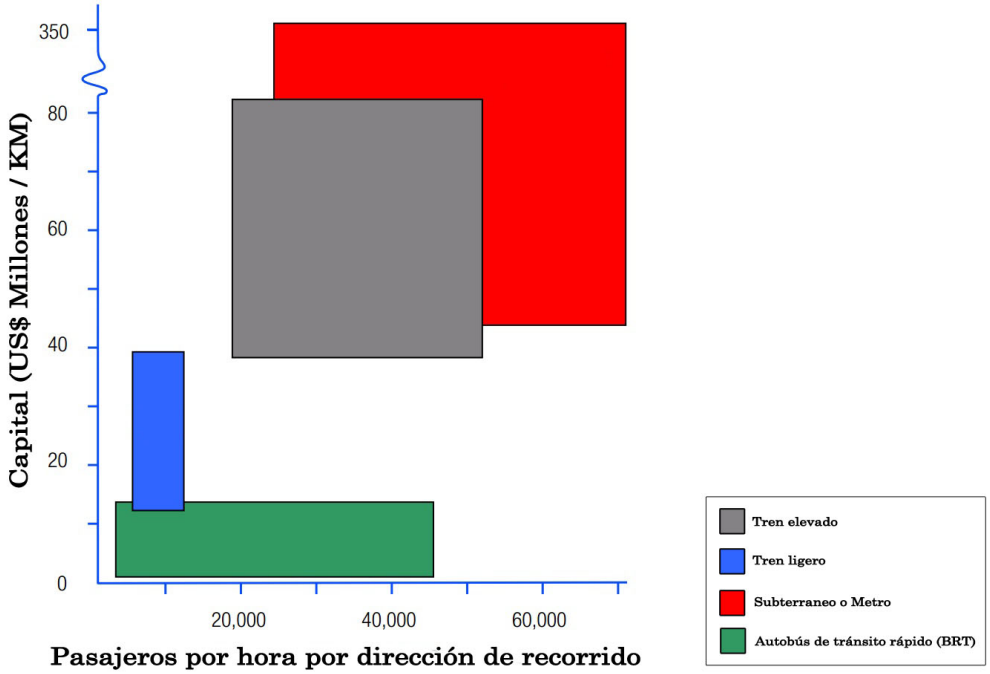


Figura 2.2: Gráfica de costos de transporte en función de la gente transportada.- Imagen original extraída de Institute for Transportation & Development Policy (ITDP) (2007)

Un BRT contiene características similares al tren ligero o al sistema de metro, por lo que es mucho más confiable, conveniente y rápido que los servicios regulares de ómnibus. Entre sus características fundamentales se destacan el uso de carriles exclusivos para el ómnibus, un alineamiento central del carril y un largo mínimo de 3 km. Además de estás características, se recomienda incluir el uso de ómnibus de gran capacidad para transportar un número mayor de personas, la compra de pasajes fuera del ómnibus (agilizando la entrada de pasajeros) y que el ómnibus tenga prioridad sobre otros vehículos en las intersecciones viales. Con las especificaciones adecuadas, un BRT es capaz de evitar las

causas de los retrasos que suelen tener los servicios regulares de ómnibus (como estar atrapado en el tráfico y hacer cola para pagar a bordo).

Para crear una definición común de BRT y calificar los sistemas existentes alrededor del mundo evaluando su funcionamiento basado en las mejores prácticas internacionales, existe el standard BRT creado por The Institute for Transportation and Development Policy (ITDP) (2014). Su principal objetivo es brindar una mejor experiencia a los pasajeros, con un costo económico acorde e impacto ambiental positivo. Este standard cuenta con un método para calcular el puntaje del corredor y determinar su nivel de calidad.

En la siguiente sección se presenta un análisis exhaustivo de varios puntos presentados en el standard BRT en relación con el caso de estudio del proyecto: el Corredor Garzón.

## 2.3. Corredor Garzón

El Corredor Garzón fue construido como parte del Plan de Movilidad Urbana en la ciudad de Montevideo (IMM, 2010). El corredor está ubicado en el centro geográfico de Montevideo como se puede apreciar en la Figura 2.3 y conecta los barrios de Colón, Sayago, Belvedere y Paso Molino. Tiene una extensión de 6.5 km con 24 cruces semaforizados, en donde se encuentran calles importantes como Millán, que conecta con una autopista (Ruta 5), y Bulevar Batlle y Ordoñez, que tiene una gran densidad de tráfico. Es importante aclarar que el Corredor Garzón no es sólo una conexión de extremo a extremo, ya que se encuentra en una zona densamente poblada cuyos barrios tienen al corredor como la principal vía de movilidad.

Como se aprecia en la Figura 2.4, el corredor consiste básicamente en tres calles paralelas e independientes. Dos calles cuentan con dos carriles de una sola mano y entre medio de éstas se encuentra una calle de doble vía con un carril para cada vía, que es exclusivamente usado por ómnibus urbanos durante el día y por ómnibus urbanos y suburbanos en la noche.

Basándose en el standard antes mencionado, el Corredor Garzón cumple con la definición de BRT por las siguientes características: *i)* el largo del carril exclusivo para los ómnibus es de más de 3 km de longitud, *ii)* más del 90 % de la extensión del corredor está segregado físicamente, impidiendo el traspaso de carril de otros vehículos hacia el carril exclusivo para ómnibus, *iii)* presenta dos carriles centrales para los ómnibus que se encuentran en medio de los carriles para los demás vehículos constituyendo un corredor alineado centralmente, *iv)* incluye prohibiciones de algunos virajes a la izquierda y el carril de ómnibus tiene prioridad en la mayoría de las intersecciones.

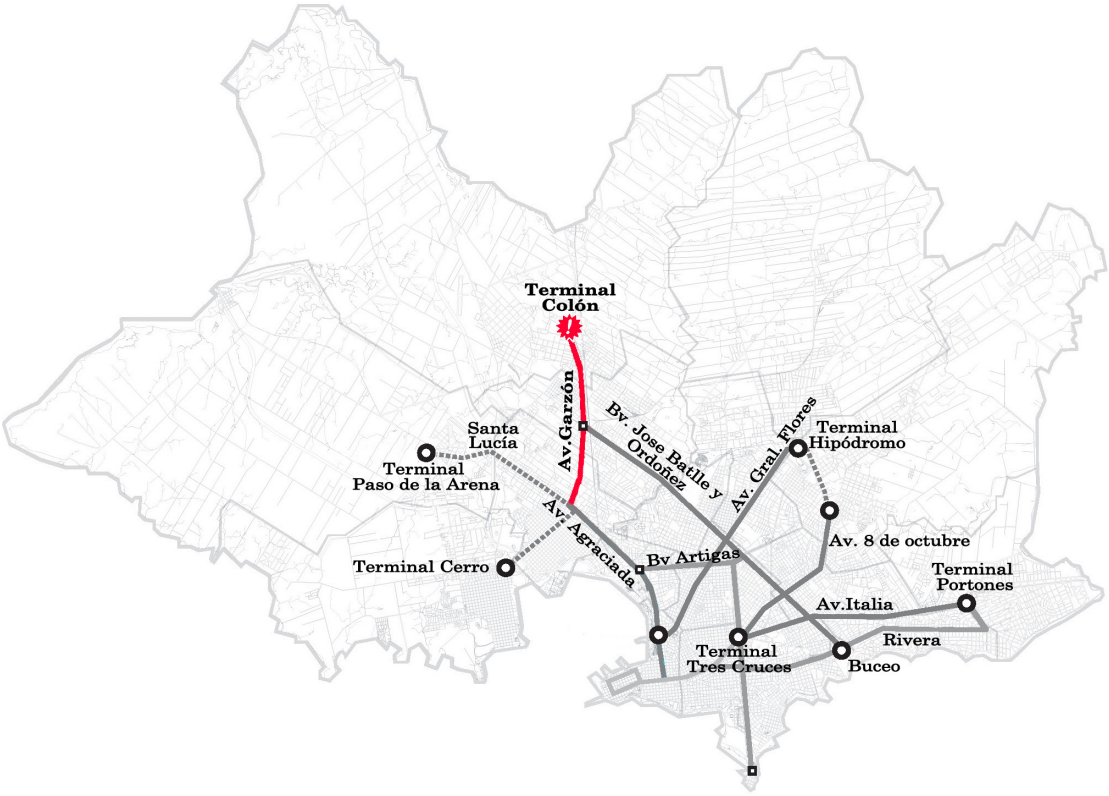


Figura 2.3: Localización del Corredor Garzón en Montevideo, referenciado en color rojo - Imagen original extraída de www.montevideo.gub.uy

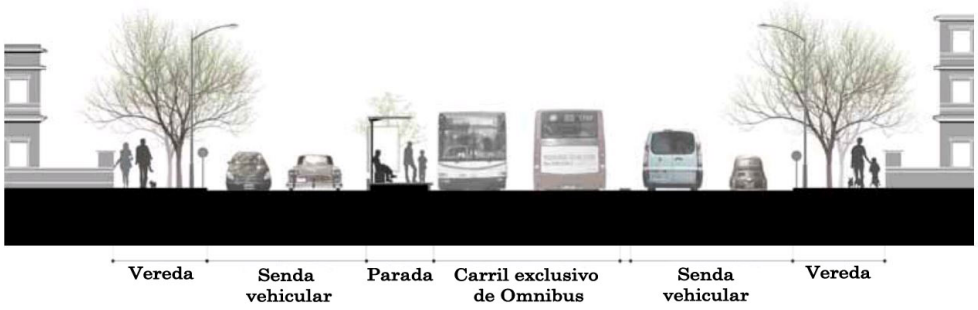


Figura 2.4: Perfil propuesto para el corredor Garzón desde San Quintín a Camino Colman - Imagen original extraída de (IMM, 2010)

Analizando los puntos presentados en el standard como recomendaciones para la implementación de corredores urbanos, hay algunas características que no están totalmente aplicadas al Corredor Garzón, entre las que se destacan:

1. Cobrar el boleto fuera del ómnibus: uno de los factores más importantes para mejorar la experiencia del usuario, así como también la velocidad en zonas de mucha carga, es que existan al menos algunas estaciones (no sólo paradas de ómnibus) donde el boleto se cobre (o se use tarjeta de transporte) al entrar a la estación.

2. Ofrecer servicios expresos o limitados: una forma de mejorar las velocidades de operación consiste en crear líneas que no se detengan en todas las paradas (evitando aquellas paradas con menor demanda de pasajeros) y poner más ómnibus en la calle.

3. Instrumentar un carril extra para adelantarse en paradas: este carril resulta crítico en sistemas de transporte colectivo de gran porte para poder manejar los servicios expresos. En sistemas de baja demanda es una buena inversión y en el caso del Corredor Garzón podría permitir que los servicios suburbanos funcionaran durante todo el día por el corredor, mejorando así el transporte público y privado.

4. Establecer un mínimo de distancia entre paradas e intersecciones: según el standard, la mínima distancia entre la intersección y la parada es de 26 m, pero idealmente deberían ser 40 m para evitar retrasos. En el Corredor Garzón las paradas están sobre las intersecciones y hay varias paradas donde se detiene más de una línea de ómnibus. Esto puede ocasionar problemas, dado que si uno o dos ómnibus sólo esperan por el semáforo en rojo, los demás que lleguen generarán una fila aguardando para detenerse en la parada y así poder levantar o dejar pasajeros y posiblemente al llegar al cruce ya no los habilitará la luz verde que dejó pasar a los primeros coches, debiendo tener que esperar nuevamente.

5. Tener estaciones centrales o conexión entre paradas: la ausencia de paradas en el centro de los dos carriles de ómnibus hace que el diseño tenga una construcción más cara (hay que hacer dos paradas). La no existencia de una conexión física para que el pasajero pueda cambiar de recorrido sin tener que cruzar la calle, lo hace menos eficiente y más inseguro.

6. Contar con una distancia entre estaciones que cumpla el standard: las paradas deberían estar a una distancia de entre 300 m y 800 m, siendo 450 m la distancia óptima tanto para el pasajero como para el transporte. Si bien el Corredor Garzón cumple la distancia óptima en promedio (464 m), individualmente existen casos como las cuatro paradas que se encuentran entre Emancipación y Avenida Islas Canarias cuyas distancias son menores al mínimo sugerido por el standard.

7. Utilizar varias puertas para el ingreso/egreso de pasajeros en los ómnibus: con el fin de mejorar el flujo y volumen de pasajeros, sería conveniente contar con dos puertas anchas o al menos tres puertas comunes.

8. Mejorar la calidad de las paradas: las paradas deberían proteger al usuario de las inclemencias climáticas, contar con puertas corredizas que se abran cuando hay un ómnibus (para evitar que alguien caiga al corredor) y brindar información al pasajero en tiempo real, etc. Estas especificaciones no hacen al corredor más rápido pero si más seguro, cómodo y confiable.

Desde su inauguración en el año 2012, el Corredor Garzón ha recibido críticas debido a que su principal objetivo de agilizar el transporte público no fue cumplido. Luego de varios intentos de mejoras al respecto, se ha vuelto a una situación que implica la misma velocidad promedio para el transporte público que existía antes de realizar el corredor (El País, 2015).

Las autoridades municipales admitieron que se han cometido errores en el diseño del corredor, y que no se ha logrado sincronizar los semáforos en las vías de tránsito que lo componen (El País, 2013). Un buen funcionamiento de los semáforos es fundamental para asegurar que el tráfico circule con eficiencia y a la vez aporte seguridad a los peatones. A continuación se tratará específicamente el tema de la sincronización de semáforos, que da la motivación para este proyecto.

## 2.4. Sincronización de semáforos

Los métodos utilizados para la optimización del tráfico tienen como objetivo mejorar el flujo de vehículos en una red vial. Estos métodos se pueden clasificar en dos categorías: los que influyen en el comportamiento de los conductores (mediante la configuración de semáforos, introducción de señalizaciones, etc) o los que realizan modificaciones en las vías de tráfico (agregando nuevos carriles, ensanchando calles, etc). Las modificaciones de infraestructuras pueden producir mejoras drásticas, pero requieren una inversión monetaria y un espacio físico que muchas veces no está disponible. Por esta razón, los métodos destinados a influir en el comportamiento de los conductores se presentan como una mejor opción o inclusive como única opción en muchos escenarios.

Los métodos para la sincronización de semáforos se encuentran entre los más efectivos para agilizar el tránsito y no generar congestiones. Estas técnicas, permiten aumentar la velocidad promedio de los viajes y mejorar las perspectivas de desarrollo de la ciudad así como la calidad de vida de sus habitantes.

Un concepto importante en el manejo de semáforos es el de *fase*, que se refiere a una configuración específica de luces de semáforos en una intersección de calles, que permiten el movimiento de ciertos flujos de tráfico. Como se ve en la Figura 2.5 cada intersección puede tener diferente número de fases y también distintas duraciones. Las fases suelen ser configuradas y establecidas manualmente por técnicos especializados basados en su experiencia, aunque en ciertas ocasiones se utilizan simulaciones computacionales para obtener configuraciones apropiadas.

Existen tres parámetros a tener en cuenta, que determinarán el comportamiento del sistema: la duración de fase, los ciclos de las luces y los valores de *offset*. A continuación se explica cada uno de ellos:

- Duración de fase: Se refiere a la duración en que una configuración específica de las luces de los semáforos está activa en una intersección. Un conjunto de las luces estará en verde y otras en rojo, determinando qué vía se encuentra habilitada para cruzar en ese momento. La elección correcta de este parámetro es fundamental para lograr fluidez en el tránsito; las calles con una densidad de tránsito mayor tendrían que tener más tiempo asignado para cruzar. Realizar la optimización individual de cada intersección no tiene por qué conducir a una solución óptima de toda la red vial, ya que puede ocasionar cambios en el flujo vehicular que afecte otras secciones de la red. Por esta razón es conveniente utilizar un enfoque global al modelar una solución.

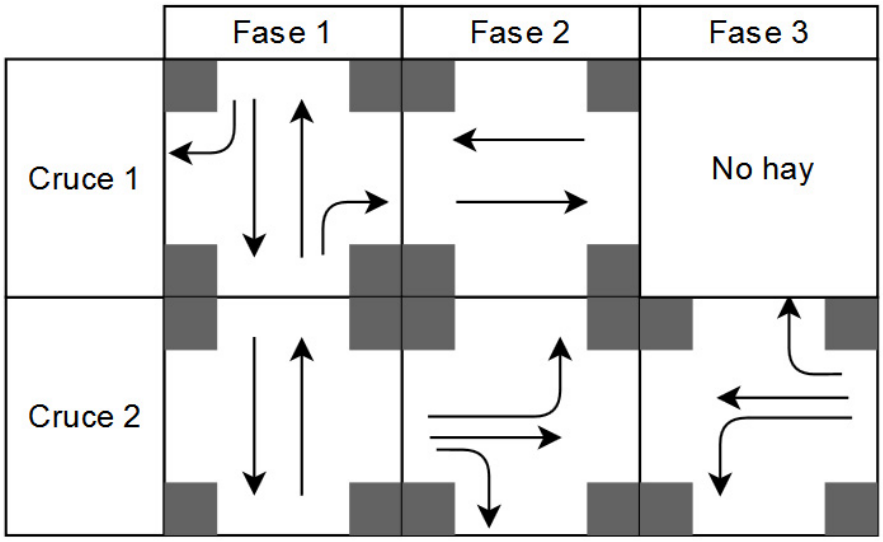


Figura 2.5: Distintas fases de semáforos para dos cruces en una red de tránsito.

- Duración del ciclo: Un ciclo representa un conjunto de fases. En general la duración de un ciclo es la suma de las duraciones de las fases. Como indica su nombre, el ciclo se repetirá una vez que se completa. La duración se puede incrementar o disminuir para permitir mayor cantidad de repeticiones de las fases.

- *Offset*: Indica en qué fase comienza el ciclo o en qué instante de tiempo, permitiendo que las intersecciones comiencen su ciclo en diferentes momentos. Este concepto es muy importante para sincronizar un flujo de tráfico en lo que se conoce como *línea verde*, en donde los vehículos logran pasar todas las intersecciones sin detenerse.

Los parámetros mencionados anteriormente pueden ser utilizados a la hora de sincronizar los semáforos de una zona, buscando una optimización global de la red vial. También se debe tener en cuenta la implementación de soluciones seguras desde el punto de vista de las ordenanzas de tránsito, siendo lo más básico que no existan intersecciones donde los flujos vehiculares estén habilitados para cruzar al mismo tiempo.

Los métodos para lograr la coordinación necesaria entre semáforos incluyen desde simples mecanismos de reloj hasta sistemas computarizados que se ajustan en tiempo real con la ayuda de sensores en la calle. Estos métodos pueden clasificarse como estrategias de tiempo fijo o de tiempo dinámico auto-ajustable.

Se considera al problema de sincronización de semáforos como un problema de optimización NP-difícil (Yang y Yeh, 1996), lo que provoca que los métodos computacionales exactos lo resuelvan en forma eficiente, solamente en instancias de tamaño reducido. Por este motivo, deben utilizarse métodos heurísticos y metaheurísticos para resolver instancias realistas del problema, algunos de los cuales se describen en la sección de trabajos relacionados. Entre los métodos más desarrollados y efectivos para resolver el problema están los algoritmos evolutivos, en particular los algoritmos genéticos, que serán explicados a continuación.

## 2.5. Algoritmos Evolutivos

Los Algoritmos Evolutivos (AE) son un conjunto de técnicas metaheurísticas para la resolución de problemas complejos que se inspiran en la evolución natural. Los AE trabajan sobre una población de individuos que representan una solución y utilizan mecanismos de selección, reproducción y técnicas para mantener la diversidad con el objetivo de encontrar soluciones de buena calidad para el problema (Spears, 2000).

Un AE se describe como una técnica iterativa que busca en cada paso mejorar las soluciones por medio de operadores de exploración y explotación, basado en un criterio predefinido a maximizar o minimizar.

Se pueden destacar cuatro etapas en la ejecución del AE:

- Evaluación: Para cada individuo de la población se determina un valor de aptitud (*fitness*) en relación a su capacidad para resolver el problema.
- Selección: Proceso en donde se eligen cuales son los individuos que sobrevivirán a la siguiente generación y sobre los cuales se aplicarán los operadores evolutivos.
- Operadores evolutivos: Se aplican combinaciones entre individuos (cruzamiento), y modificaciones aleatorias de individuos (mutación). Los operadores generan nuevos individuos que sustituirán a los existentes en la población.
- Reemplazo: Se produce el recambio generacional, sustituyendo a la antigua población por una nueva, que podría tener sobrevivientes de la anterior o solamente nuevos individuados generados en la etapa de aplicación de los operadores evolutivos.

Se han desarrollado enfoques de algoritmos de optimización que aplican estos conceptos, las más relevantes se describen a continuación:

- Estrategias de Evolución: Propuesto por Rechenberg (1973), fue originalmente presentado como un método de optimización utilizando individuos que codifican números reales en problemas relacionados al diseño de ingeniería. Su característica principal es que utilizan el operador de mutación como motor para la evolución. Sin embargo, modelos más recientes agregan otro tipo de operadores, incluyendo cruzamiento.
- Programación genética: Intenta generar un programa de computación para resolver una tarea específica aplicando técnicas evolutivas. Cada individuo puede ser un programa que es representado en forma de árbol, donde cada nodo del árbol tiene una operación y los nodos terminales un operando, de esta manera se facilita la evaluación de expresiones matemáticas. La función de aptitud se refiere a qué tan cerca de resolver la tarea se encuentra el individuo (Koza, 1992).
- Algoritmo Genético (AG): Es considerado el más popular de los AE, dado su versatilidad a la hora de resolver problemas de optimización. El operador de cruzamiento es el principal operador evolutivo, siendo el de mutación un operador secundario. Estos operadores se aplican sobre la población de soluciones potenciales en cada generación. En la siguiente sección, se presentan en detalle las características de los algoritmos genéticos, que es la técnica que se utiliza en este trabajo para resolver el problema de sincronización de semáforos.

### 2.5.1. Algoritmos Genéticos

El Algoritmo 1 muestra la formulación básica de un AG que se basa en el esquema general de un AE. Comienza con una población inicial de individuos a los cuales en cada generación se le aplican operadores de cruzamiento y mutación, seleccionando a los mejores en base a su aptitud para resolver el problema. El operador de cruzamiento guía la búsqueda, mientras el operador de mutación se encarga de aportar diversidad a la exploración. Por lo tanto, el objetivo es que con el paso de las generaciones se obtengan mejores soluciones (como se muestra en la Figura 2.6) hasta detenerse usando un criterio de parada, ya sea el número de iteraciones o cuando ya no se puede mejorar más la solución. Los trabajos de Goldberg (1989) y Mitchell (1998) brindan más detalles sobre los AE.

#### Esquema del algoritmo

El esquema básico de funcionamiento se presenta en el Algoritmo 1:

**Algoritmo 1** Algoritmo Genético

```
1: Inicializar(Pob(0))
2: generación = 0
3: mientras no se cumple el criterio de parada hacer
4:     Evaluar (Pob(generación))
5:     Padres = Seleccionar(Pob(generación))
6:     Hijos = Cruzamiento(Padres)
7:     Hijos = Mutación(Hijos)
8:     NuevaPob = (Reemplazar Pob(generación), Hijos)
9:     generación++
10: fin
11: retornar mejor solución encontrada
```

Un individuo es una codificación de una solución que resuelve el problema. La población inicial puede generarse aleatoriamente o basándose en heurísticas que utilizan conocimiento específico sobre el problema. El mecanismo de selección determina cuales individuos son adecuados para integrar la población de la siguiente generación. Este mecanismo utiliza el valor de la función de evaluación o *fitness* para definir qué tan buena o apta es una solución en comparación con las demás. En cada iteración, la cual se conoce como generación, se aplican operadores de cruzamiento y mutación sobre los individuos. El cruzamiento permite combinar a dos individuos para obtener otros que potencialmente sean una mejor solución. La mutación aplica cambios aleatorios sobre los individuos. Estos operadores y el mecanismo de selección son probabilísticos, es decir, que su aplicación depende de una una tasa de probabilidad asociada al operador.

Por lo tanto, se van seleccionando, combinando y cambiando las mejores soluciones en un proceso, que si el AG está bien diseñado, permite ir obteniendo mejores soluciones. El criterio de parada indica cuando termina este proceso, ya sea porque se alcanzó un número de generaciones predefinidos o porque la mejora no es evidente. Al final se devuelve la mejor solución encontrada en todo el proceso.

A continuación se presentan las principales características de los AG.

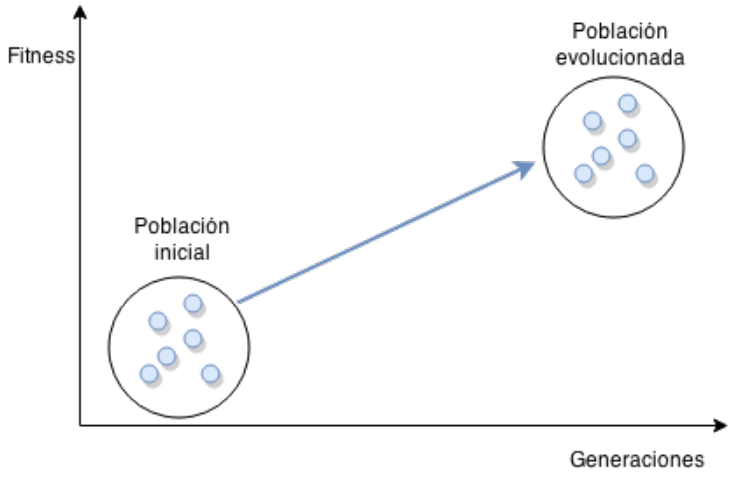


Figura 2.6: Esquema de la evolución del valor del *fitness* en un AE a través de las generaciones.

### Representación de soluciones

Los AG no trabajan directamente sobre las soluciones del problema, sino que utilizan una abstracción de las soluciones llamada cromosoma. Un cromosoma es un vector de genes donde el valor de cada gen se denomina alelo; nombres inspirados en la evolución natural biológica. En general los AG codifican las soluciones utilizando un vector de números binarios o reales de largo fijo.

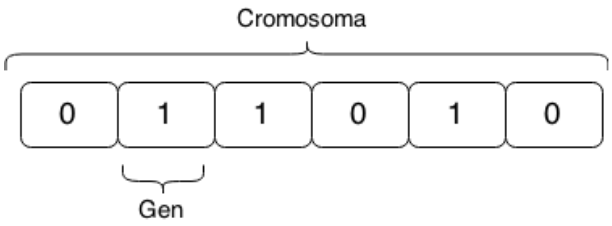


Figura 2.7: Representación binaria de un cromosoma.

### Función de Evaluación

La función de evaluación indica qué tan bueno es un individuo para resolver el problema en cuestión, utilizando un valor conocido como *fitness*, por lo que también es llamada función de *fitness*. Este valor se utiliza para definir cuales son los mejores individuos y de esta forma guiar la exploración hacia la región que incluya las mejores soluciones. En general, la función de *fitness* es la que insume la mayor cantidad de tiempo del AG, en comparación con los demás operadores.

### Operador de Selección

Existen diversos operadores de selección, cuya función es mantener las mejores características de los individuos en las siguientes generaciones. Entre ellos se encuentran:

- Selección proporcional: Elige aleatoriamente individuos donde la probabilidad de selección es proporcional al valor del *fitness*. Los mejores individuos son elegidos con mayor probabilidad, pero los peores individuos también pueden ser elegidos, lo que permite mantener la diversidad en la población. Sea N la cantidad de individuos en la población y $f_i$ el valor de *fitness* del i-ésimo individuo, la probabilidad asociada a su selección está dada por la la ecuación 2.1 .
- Torneo: Se eligen aleatoriamente un determinado número de individuos y se selecciona un subconjunto de los individuos con mejor *fitness*.
- Rango: Se ordenan los individuos por el valor de *fitness* y se selecciona un determinado porcentaje (rango) de los mejores individuos.

$$F(x) = p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{2.1}$$

### Operador de cruzamiento

La función del operador de cruzamiento es combinar individuos con el objetivo de preservar las mejores características de los progenitores y así lograr construir mejores soluciones. El operador de cruzamiento se aplica de acuerdo a una tasa de probabilidad prefijada.

En general, los operadores de cruzamiento se pueden clasificar en:

- Cruzamiento de un punto: A partir de dos padres se selecciona un punto al azar de los cromosomas obteniendo dos trozos que se combinan para obtener dos hijos. Se explica en la Figura 2.8
- Cruzamiento multipunto: El método anterior se puede generalizar para obtener más puntos de corte y más recombinaciones.
- Cruzamiento uniforme: Para cada posición en el cromosoma se intercambian genes según una tasa de probabilidad, este mecanismo se muestra en la Figura 2.9 .

### Operador de mutación

El operador de mutación es el método utilizado para modificar un individuo, con el objetivo de mantener y mejorar la diversidad y otorgar al AG un mecanismo que le permita escapar de óptimos locales. En general la mutación aplica una modificación aleatoria en el cromosoma, por ejemplo, en una representación binaria se invertiría aleatoriamente el valor de un alelo, como se muestra en la Figura 2.10.

### Criterio de reemplazo

El criterio de reemplazo indica cual es el mecanismo utilizado para realizar el recambio generacional. Se podría reemplazar todos los padres por los hijos o seleccionar sólo algunos padres a reemplazar, que en general son los que tienen peor valor de *fitness*.

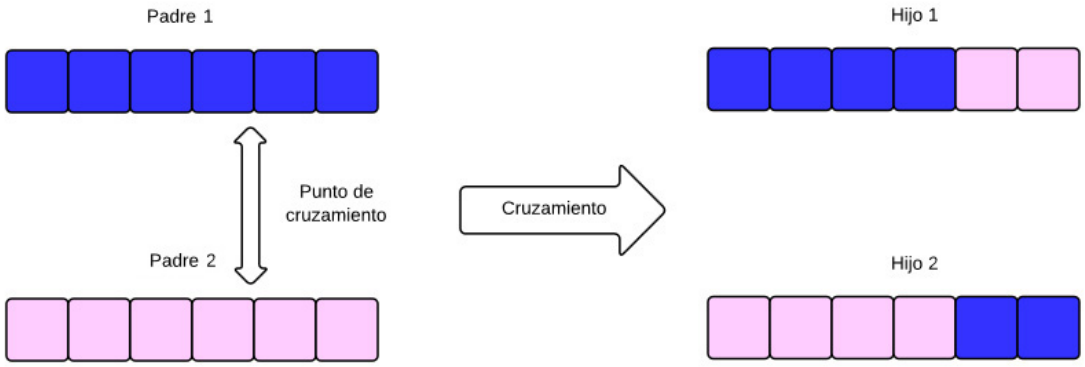


Figura 2.8: Cruzamiento de un punto

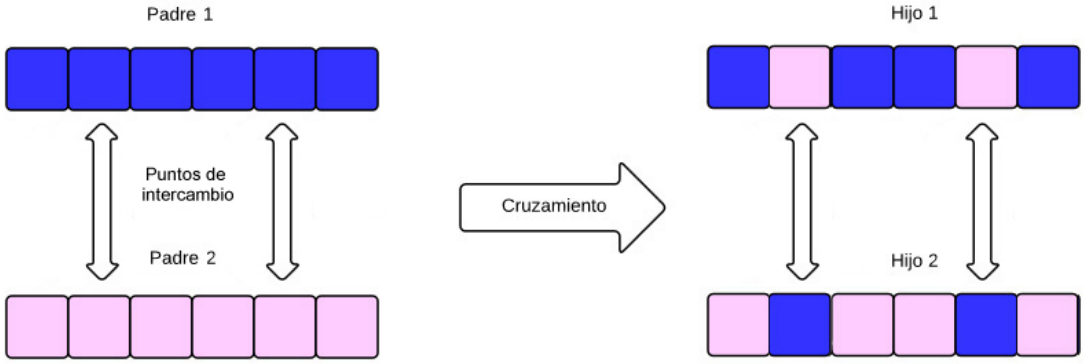


Figura 2.9: Cruzamiento uniforme

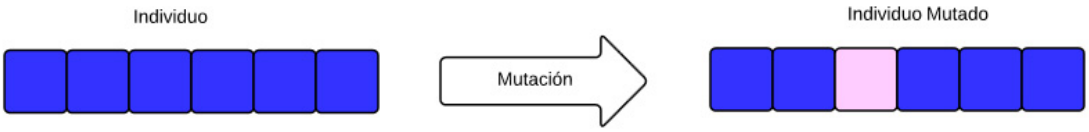


Figura 2.10: Mutación por inversión binaria

### Criterio de parada

El criterio de parada indica cuándo debe terminar la fase iterativa del algoritmo. Puede definirse un número prefijado de generaciones o determinar si el proceso se estanca, por ejemplo cuando no se generan mejoras significativas en el valor de *fitness*.

## 2.5.2. Algoritmos evolutivos multiobjetivo

Los problemas de optimización multiobjetivo trabajan sobre un espacio multidimensional de funciones y no tienen una única solución. Por este motivo, el significado de óptimo cambia. Una solución es un *óptimo de Pareto* si ninguna de las funciones objetivo puede mejorar su valor sin degradar otro de los valores objetivo. Todas estas soluciones

son consideradas igualmente buenas, ya que los vectores no se pueden ordenar completamente y representan diferentes valores de compromiso entre las funciones objetivo. Al conjunto de los valores funcionales de los óptimos de Pareto se les llama frente de Pareto.

Existen algoritmos evolutivos para resolver problemas de optimización multiobjetivo. Éstos son los llamados MOEA, por sus siglas en inglés *MultiObjetive Evolutionary Algorithm*. Sus propósitos son aproximarse al frente de Pareto y lograr obtener una gama de diferentes compromisos entre las funciones a optimizar para luego poder tomar la decisión de cual elegir (Deb, 2001).

El Algoritmo 2 describe el esquema básico de un MOEA, donde se aprecia que existen dos operadores característicos: el operador de diversidad y el operador de asignación de *fitness*. El primero se aplica para evitar la convergencia a un sector en particular del frente de Pareto, mientras que el segundo intenta brindar mayor chance de perpetuar a los individuos con mejores características.

**Algoritmo 2** AE multiobjetivo. En negrita se indican las diferencias con el AE genérico.

```
1: Inicializar(Pob(0))
2: generación = 0
3: mientras no se cumple el criterio de parada hacer
4:     Evaluar (Pob(generación))
5:     Operador diversidad(Pob(generación))
6:     Asignar fitness(Pob(generación))
7:     Padres = Seleccionar(Pob(generación))
8:     Hijos = Cruzamiento(Padres)
9:     Hijos = Mutación(Hijos)
10:    NuevaPob = (Reemplazar Pob(generación), Hijos)
11:    generación++
12: fin
13: retornar frente de pareto
```

Los MOEA se pueden clasificar por el método de asignación de *fitness*. Por un lado se tienen los que no son basados en Pareto, que utilizan métodos sencillos de asignación de *fitness*. Estos métodos no reflejan la independencia entre los objetivos y tampoco garantizan una resolución correcta del problema. Un método popular entre los que no utilizan dominancia de Pareto es la combinación lineal de los objetivos, que es relativamente sencilla de implementar y adecuada para problemas de optimización con no más de tres funciones objetivo y donde el frente de Pareto del problema es convexo (Coello et al., 2002). El problema es tratado como una optimización monoobjetivo, no reflejando el carácter multiobjetivo del problema. En este método, el *fitness* se calcula como una suma ponderada con los pesos fijados a priori, de acuerdo a la expresión en la Ecuación 2.2.

$$F(x) = \sum_{i=1}^{n} w_i f_i(x) \tag{2.2}$$

Por otro lado, se encuentran los métodos de asignación de fitness que utilizan explícitamente la dominancia de Pareto y garantizan la independencia entre los objetivos del problema.

### 2.5.3. Algoritmos evolutivos paralelos

Resolver problemas complejos suele insumir gran cantidad de tiempo, por lo que paralelizar el AE es útil para lograr tiempos de ejecución menores. Pero éste no es el único objetivo que se puede conseguir con un AE paralelo, entre otros existen: encontrar soluciones alternativas al mismo problema, búsqueda más eficiente aún sin *hardware* paralelo, facilidad en la cooperación con otros métodos y búsqueda paralela de múltiples puntos en el espacio (Alba y Tomassini, 2002).

En el caso de los AG, gran parte del tiempo se ocupa en la etapa de evaluación de *fitness*, por esta razón es un buen método distribuir la carga en varios procesadores para que las evaluaciones se realicen en paralelo.

Un modelo de AE paralelo muy utilizado por su facilidad conceptual y facilidad de implementación es el *maestro-esclavo*, que brinda la posibilidad de utilizar racionalmente los recursos computacionales disponibles en todos los computadores modernos. El proceso maestro es el encargado de ejecutar los operadores básicos del algoritmo y distribuir a procesos esclavos la evaluación de *fitness* para un conjunto de individuos. El esclavo devuelve el resultado y luego el maestro es el encargado de continuar con la evaluación, ejecutando los operadores. De este modo, aumenta la eficiencia computacional del algoritmo, ya que las múltiples ejecuciones de una de las funciones más costosas es distribuida entre varios procesos que ejecutan concurrentemente.

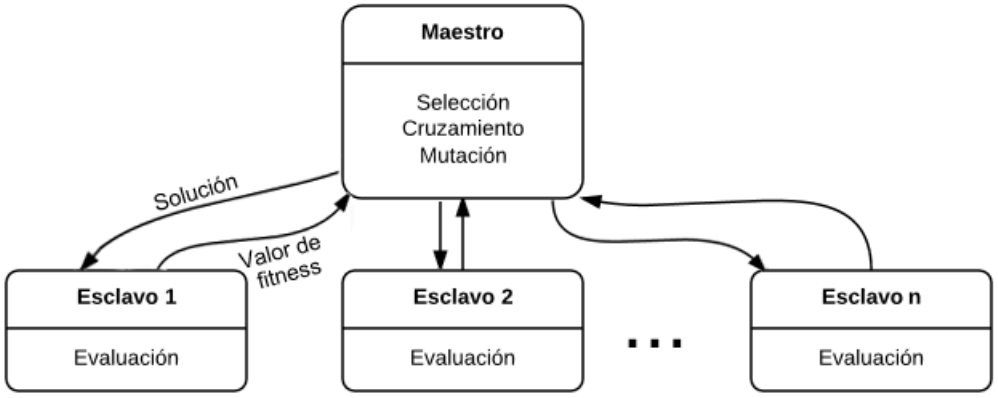


Figura 2.11: Modelo Maestro-Esclavo

### 2.5.4. Implementación y entornos de desarrollo

A la hora de implementar un AG es importante tener en cuenta ciertos aspectos que dotarán a la solución de una mayor flexibilidad, eficiencia y robustez. En general, se recomienda usar el paradigma de orientación a objetos, ya que aporta múltiples ventajas entre las que se encuentran: reusabilidad del código, abstracción de problemas, modularidad y legibilidad. Sin embargo, la orientación a objetos puede ocasionar pérdidas de eficiencia computacional, por lo que hay que balancear estos aspectos.

En el trabajo de Alba y Cotta (1997) se indican con más detalle los elementos importantes de una implementación de AG, entre las que destacan: no utilizar tamaño fijo para las poblaciones pues afectan la flexibilidad, evitar las múltiples evaluaciones de una misma solución, dar al usuario la posibilidad de extender las funcionalidades de manera

modular, al implementar bibliotecas genéricas usar lenguajes orientados a objetos que permiten su extensión en forma sencilla. Por estas razones se han desarrollado entornos de desarrollo o *frameworks* para trabajar en la resolución de problemas utilizando AE y otras técnicas metaheurísticas. Los *frameworks* encapsulan de manera transparente al usuario los aspectos antes mencionados, lo que permite un desarrollo más rápido, sencillo y seguro.

Existe una gran variedad de *frameworks* para AE; a continuación se describen algunos de los más populares junto con sus principales características.

- JMetal: Es un *framework* orientado a objetos basado en Java para la resolución de problemas de optimización multiobjetivo utilizando metaheurísticas. Su arquitectura permite experimentar con técnicas embebidas y también desarrolladas por el usuario. Entre sus características se destaca que al estar implementado en Java puede ser usado tanto en sistemas Windows como Linux. Posee una amplia variedad de algoritmos multiobjetivo listos para usar, y también algoritmos paralelos. Está en constante desarrollo y cuenta con una documentación detallada (Durillo y Nebro, 2011).

- Galib: Es una biblioteca escrita en C++ que incluye objetos y herramientas útiles para la resolución de problemas usando AG. Es gratuita y de código abierto, pudiendo ser ejecutada tanto en Linux como en Windows. No tiene un desarrollo sostenido, siendo su última actualización en el año 2000, esto no quita que todavía se siga usando por su sencillez y flexibilidad (Wall, 1996).

- Paradiseo: Es un *framework* orientado a objetos que utiliza C++. Su objetivo es el diseño de metaheurísticas tanto paralelas como distribuidas. Cuenta con AEs, búsquedas locales y optimización basada en enjambres. Funciona tanto en plataformas Windows como Linux y posee varios módulos destinados a extender sus funcionalidades. Tiene una buena documentación y su ultima versión data del año 2012 (Cahon et al., 2004).

- Open Beagle: Es un *framework* en C++ que utiliza AE. Brinda un entorno de alto nivel para trabajar con distintas técnicas, soporta programación evolutiva, AG y estrategias evolutivas. Su arquitectura permite utilizar los principios de la programación orientada a objetos para lograr un código reutilizable y eficiente. Sus características apuntan a brindar un entorno amigable, eficiente, multi-plataforma y gratuito (Gagné y Parizeau, 2002).

- Mallba: Es una biblioteca escrita en C++ que brinda esqueletos de algoritmos exactos, metaheurísticas, e híbridos para la resolución de problemas de optimización. Mallba maneja el paralelismo de forma simple para el usuario. Posee una arquitectura flexible y extensible lo que permite agregar nuevos esqueletos de forma sencilla. No cuenta con documentación extensa pero si con varios ejemplos completos de implementaciones de los algoritmos. Mallba funciona tanto en ambiente Linux como Windows (Alba et al., 2006).

- Malva: Surge como un *branch* de la biblioteca Mallba, con modificaciones para ser ejecutado en ambientes actuales. Cuenta con las mismas características que Mallba aunque al estar en fase de desarrollo soporta sólo algunos algoritmos, entre los que se encuentra un AG (Fagúndez, 2014).

Los *frameworks* son una herramienta importante a la hora de desarrollar un AE. Al enfocarse en el problema de la sincronización de semáforos también serán necesarias otras utilidades entre las que se encuentra los simuladores de tráfico, los cuales serán explicados en la siguiente sección.

## 2.6. Simulación de tráfico

Los simuladores de tráfico son programas que simulan el movimiento del flujo vehicular sobre una red terrestre, marítima o aérea. Son usados en proyectos de investigación, estudio de congestiones y análisis de impacto de obras. Existen varias razones para optar por esta herramienta, entre las que se encuentran: la rapidez en la obtención de resultados y el costo asociado de implementación.

Los simuladores se pueden dividir en dos grandes categorías, macroscópicos y microscópicos. En algunos casos se considera una tercera categoría híbrida de estas dos llamada mesoscópicos. En los simuladores *macroscópicos* el tráfico es modelado como un flujo continuo y es descrito de manera agregada usando características como la velocidad o densidad del flujo. En los simuladores *microscópicos* el tráfico se considera compuesto de partículas discretas. Cada partícula es actualizada según las propiedades de la red en ese momento, como límites de velocidad, vehículos cercanos y caminos a seguir. Los simuladores microscópicos utilizan un modelo de decisiones del conductor, lo que permite crear una distribución heterogénea del comportamiento de los vehículos. En general, los simuladores *mesoscópicos* representan a los individuos con alto nivel de detalle pero sus interacciones y actividades con un bajo nivel de detalle, por ejemplo, agrupando vehículos en paquetes que se mueven por una red, considerándolos como una sola entidad.

Se considera que las simulaciones microscópicas se aproximan más a la realidad y que obtienen un nivel de granularidad mayor. Estas características pueden ser útiles cuando se asignan propiedades sobre cada vehículo y se quiere observar el comportamiento cuando éstas cambian. Sin embargo, no implica que se dejen de usar simulaciones macroscópicas, pues aunque no poseen tanto detalle son más rápidas y en determinadas circunstancias podrían ser una mejor opción.

En la actualidad existe una gran variedad de simuladores disponibles, que se encuentran en las diferentes categorías que fueron mencionadas anteriormente. A continuación se ofrece una breve descripción de algunos de los simuladores presentados en el trabajo de Kotushevski y Hawick (2009) que presenta una lista detallada de simuladores de tráfico.

- SUMO: Simulador abierto, portable y microscópico, diseñado para soportar grandes redes de tránsito. Es de los más populares, y utiliza una serie de archivos de configuración para representar las rutas, los vehículos y el tráfico (Krajzewicz et al., 2002).

- Quadstone Paramics Modeller: Simulador modular y microscópico capaz de modelar un amplio rango de problemas de tránsito y transporte.

- Aimsun: Paquete de simulación que integra varios tipos de modelos de transporte, por ejemplo herramientas para el tráfico estático y un simulador microscópico

- Trafficware SimTraffic: Es un simulador que forma parte del paquete *Trafficware's Syncro Studio*, que cuenta con una herramienta para la sincronización de semáforos.

- Corsim Trafvu: Parte del paquete *Tsis Corsim*, presenta las animaciones y los gráficos estáticos para las redes de tráfico utilizando *Corsim* como entrada.

Cabe destacar que de los cinco simuladores de tráfico mencionados anteriormente, solamente SUMO es abierto y gratuito, el resto son programas propietarios.

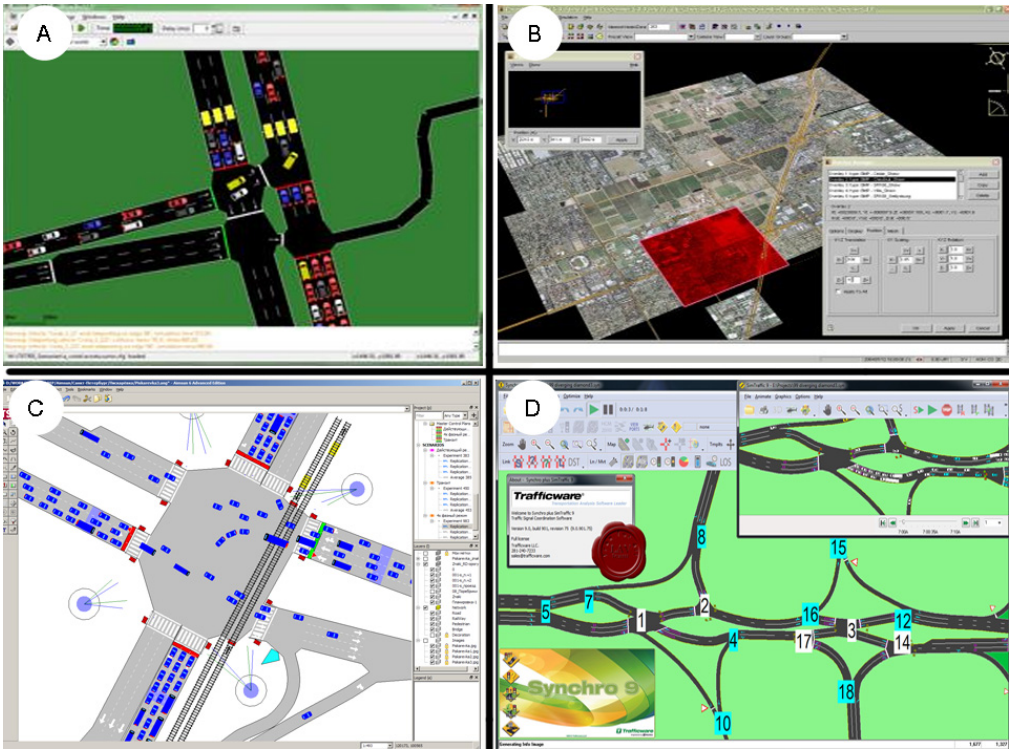


Figura 2.12: Algunos simuladores de tráfico. (A) Sumo, (B) Quadstone Paramics, (C) Aimsun, (D) Trafficware SimTraffic

Los simuladores requieren de la creación de una *red de tránsito*. En general, la red refiere a las propiedades que tendrán las vías de tránsito, por ejemplo: la cantidad de carriles, el límite de velocidad, la ubicación, el largo, las conexiones entre calles, etc. Algunos simuladores obtienen esta entrada desde archivos de texto, lo cual puede ser un proceso lento y propenso a errores; otros pueden importar redes viales de servicios, como Open Street Map (Haklay y Weber, 2008). También necesitan conocer las *rutas seguidas por los vehículos*. Los simuladores pueden tomar esta información de manera explícita, indicando para cada vehículo la ruta seguida, o utilizando otros métodos dinámicos indicando sólo los puntos de inicio y final del recorrido, cuya ruta será generada en tiempo de simulación. El procesamiento manual para generar los recorridos vehiculares puede ser un proceso verdaderamente complejo, por este motivo muchos simuladores brindan herramientas destinadas a facilitar esta tarea.

Un aspecto importante a tener en cuenta es la salida que genera el simulador, ya que es fundamental a la hora de analizar resultados y sacar conclusiones. En general, la información básica reportada incluye: la velocidad y tiempos de recorrido, aunque también puede contener más detalles incluyendo la duración de detenciones, la densidad de tráfico, uso de combustible y/o la cantidad de emisiones contaminantes.

## 2.7. Trabajos relacionados

La investigación del estado del arte se realizó con dos objetivos en mente: el primero fue analizar las distintas soluciones que existen actualmente para el problema de sincronización de semáforos y el segundo fue encontrar nuevas prácticas, algoritmos o utilidades que pudieran fortalecer la solución a implementar.

El problema del tráfico optimizando las luces de los semáforos se puede resolver por diferentes métodos, incluyendo redes neuronales (López et al., 1999), lógica difusa (Lim et al., 2001), Redes de Petri (Di Febbraro et al., 2002), entre otros. El número de artículos encontrados en el relevamiento de trabajos relacionados fue abundante y los tipos de propuesta fueron variadas. Por este motivo se decidió enfocar la búsqueda en los trabajos que proponen soluciones cercanas al proyecto propuesto. En particular, se relevaron aquellos trabajos que resolvían el problema de sincronización de semáforos utilizando un AE.

Una reseña de los principales trabajos relacionados se presenta a continuación:

- J. Sánchez, M. Galán, y E. Rubio. Genetic algorithms and cellular automata: a new architecture for traffic light cycles optimization. En *Congress on Evolutionary Computation, 2004. CEC2004*, volumen 2, p. 1668–1674. IEEE, 2004

  Este trabajo se basa en tres ideas fundamentales: *i)* el uso de un AG para la sincronización de los semáforos con el objetivo de mejorar el tráfico en una red vial simple, *ii)* la simulación de autómatas celulares para la función de evaluación del tráfico y *iii)* la utilización de una infraestructura cluster para realizar ejecuciones del algoritmo en paralelo. El caso de estudio presentado es pequeño, con 5 calles de 2 vías que se intersectan. La codificación del cromosoma utiliza un vector de números enteros, donde se codifica para cada intersección cuál es la calle habilitada en cada ciclo. El AG utiliza una estrategia de selección elitista, donde los dos mejores individuos se clonan a la siguiente generación y el resto son generados aplicando un operador de cruzamiento de dos puntos.

  Para la evaluación del *fitness*, se considera el tiempo que transcurre desde el momento en que un vehículo entra en la red hasta que sale (llega a su destino). Se ejecuta sobre una infraestructura cluster en forma paralela con una estrategia maestro-esclavo. El maestro envía los cromosomas a los esclavos para que ejecuten la función de *fitness* y devuelvan el resultado, luego el maestro se encarga de generar la siguiente población. Se comparan los resultados del AG con los valores obtenidos de simular el escenario con una configuración de semáforos aleatoria y otra fija. El AG logra una reducción del 56 % comparando con la configuración de semáforos fija, y una reducción de 84 % comparando con la configuración aleatoria en el tiempo promedio que los vehículos permanecen en la red vial.

  El mismo grupo de trabajo realizó otros aportes similares, expandiendo esta investigación, los cuales se presentan a continuación.

- J. Sánchez, M. Galán, y E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: "Las Ramblas" area in Santa Cruz de Tenerife. *IEEE Transactions on Evolutionary Computation*, 12(1):25–40, 2008

  Este estudio aplica lo presentado en el trabajo anterior a un escenario real en Santa Cruz de Tenerife, utilizando un AG para sincronizar los semáforos de la zona con

el objetivo de mejorar el flujo de circulación del tráfico. Algunas mejoras introducidas en el modelo del problema incluyen una nueva codificación del cromosoma utilizando código de Gray, que según los autores permite mejorar el desempeño computacional de los operadores de mutación y cruzamiento. La población inicial se compone de nueve soluciones provistas por la Alcaldía de la ciudad. Las estrategias de selección, cruzamiento y mutación son similares a las aplicadas en el trabajo anterior, así como la función de *fitness* que evalúa el tiempo promedio de permanencia de los vehículos en la red vial simulada.

El escenario discretizado de la zona estudiada lo componen 42 semáforos, 26 vías de entrada y 20 de salida. Las soluciones provistas por la Alcaldía se ejecutaron en el simulador y los resultados obtenidos se compararon con los valores alcanzados por el AE, concluyendo que logra una mejora de 26 % en el valor del *fitness*.

- J. J. Sánchez, M. J. Galán-Moreno, y E. Rubio-Royo. Traffic signal optimization in "La Almozara" district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):132–141, 2010

  Este trabajo contiene puntos de contacto con el presentado anteriormente. Un cambio introducido fue el análisis de cuatro funciones de *fitness* diferentes, evaluando: *i)* la cantidad de vehículos que llegaron a destino, *ii)* el tiempo de viaje promedio, *iii)* el tiempo de ocupación promedio y *iv)* la velocidad promedio global.

  El trabajo incorpora nuevas métricas correspondientes al gas total emitido por los vehículos, que tienen relación con la velocidad a la que circulan. El modelo discretizado de la zona de *La Almozara* cuenta con 17 semáforos, 7 intersecciones, 16 entradas y 18 salidas. El análisis experimental incluye 10 escenarios que van desde baja congestión de tráfico hasta alta. Se analizan los resultados para las distintas funciones de fitness concluyendo que las mejoras más importantes en el valor de *fitness* ocurren cuando la congestión de tráfico es más alta.

- R. H. J. Penner y C. Jacob. Swarm-based traffic simulation with evolutionary traffic light adaptation. En *Proceedings of Artificial Intelligence and Soft Computing*, p. 192–197, 2002

  Este trabajo se centra en un modelo de simulación basado en enjambres. Utiliza un AG con el objetivo de sincronizar los semáforos y así minimizar el tiempo de espera de los vehículos en toda la red vial. El cromosoma contiene la secuencia y duración de los semáforos, así como la relación con los semáforos complementarios. La mutación tiene en cuenta esa relación para no generar inconsistencias, por ejemplo: que ocurran dos luces verdes en la misma intersección. Existe mayor probabilidad de cruzamiento entre semáforos que se encuentran en la misma intersección. La función de *fitness* calcula la relación entre el tiempo total de viaje y el tiempo de espera de todos los vehículos.

  El primer escenario considerado en el análisis experimental es pequeño; cuenta con una ruta de tres carriles y tres intersecciones. En este escenario el AG logra mejoras significativas. Luego se resuelve un segundo escenario más complejo, de 28 semáforos y 9 intersecciones, logrando mejoras de hasta 26 % en el tiempo de espera total.

- D. H. Stolfi. Optimización del tráfico rodado en ciudades inteligentes. Master's thesis, Universidad de Málaga, 2012

  Este trabajo propone el concepto de ciudad inteligente enfocado en la movilidad, indicando que la congestión de tráfico provoca tanto pérdidas económicas como contaminación ambiental. Plantea el desarrollo de un algoritmo inteligente, que toma en cuenta el estado de congestión de las rutas y sugiere al usuario la ruta más rápida a su destino con el objetivo de minimizar los tiempos de viaje de los vehículos que circulan por la red vial. Para detectar el nivel de congestión, utiliza un dispositivo en el automóvil que se enlaza por *wifi* con los semáforos, los cuales cuentan con sensores. El trabajo no se basa en la sincronización de los semáforos, sino que presenta un sistema de búsqueda de la mejor ruta.

  El escenario experimental presentado incluye una zona de la ciudad de Málaga, donde el entramado vial se compone de ocho entradas y ocho salidas. Para la simulación del tráfico utiliza el simulador SUMO. Los vehículos modelados son: turismo, monovolumen, furgoneta y camión. Cada vehículo posee características diferentes, como la longitud, la velocidad y la probabilidad para entrar en la red vial.

  Implementa un AG, donde los cromosomas representan sensores que incluyen la información de los destinos y rutas posibles. La estrategia de selección consiste en tomar los dos peores individuos y reemplazarlos por los dos mejores hijos encontrados. La función de *fitness* tiene en cuenta los valores obtenidos de la simulación del tráfico, incluyendo la cantidad de viajes completados, el tiempo medio de viaje y el retraso medio.

  Para el análisis experimental se obtienen los resultados iniciales simulando 64 itinerarios diferentes y se comparan con la aplicación del algoritmo inteligente. Las simulaciones se realizan hasta con un máximo de 800 vehículos. El trabajo concluye que al aumentar la cantidad de vehículos (más de 400) en el sistema, el algoritmo mejora sustancialmente los resultados iniciales.

- K. T. Teo, W. Y. Kow, y Y. Chin. Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. En *2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM)*, p. 172–177. IEEE, 2010

  Este trabajo desarrolla un AG que intenta optimizar los tiempos de las luces de los semáforos para lograr mejorar el tráfico vehicular en un escenario simple de una intersección. El cromosoma representa los tiempos de las luces verdes, mientras que la función de *fitness* se calcula teniendo en cuenta el largo de las colas generadas por los vehículos en las intersecciones. La ejecución de la simulación utiliza un tiempo fijo de 600 segundos por generación, sin embargo no se describe el tipo de simulador utilizado. El análisis experimental compara los resultados de la aplicación del AG cuando el escenario tiene en cuenta un flujo continuo de vehículos, y cuando no. Entre las conclusiones del trabajo se indica que la optimización de las luces de los semáforos utilizando un AG es una buena opción para resolver el problema de la congestión del tráfico.

- D. J. Montana y S. Czerwinski. Evolving control laws for a network of traffic signals. En *Proceedings of the First Annual Conference on Genetic Programming*, p. 333–338. MIT Press, 1996

  Esta propuesta plantea un enfoque adaptativo que utiliza sensores para analizar el tráfico en tiempo real. Un sensor contabiliza los autos que circulan y otro detecta el largo de la cola de vehículos generada en la intersección. A partir de los datos obtenidos por los sensores se modifican los tiempos de las luces de los semáforos, con el objetivo de mejorar la circulación del tránsito.

  Se aplica un algoritmo híbrido entre programación genética, específicamente STGP (strongly typed genetic programming) (Montana, 1995) y un AG, para resolver el problema de sincronización se semáforos. La medida básica de efectividad en la función de evaluación es el *delay*, que representa el total de tiempo perdido causado por las señales de tráfico.

  El análisis experimental se desarrolló en un escenario simple con cuatro intersecciones y tres tipos de tráfico diferente, utilizando una versión especial del simulador TRAF-NETSIM (Rathi, 1990). En los experimentos se comparan los valores obtenidos utilizando una configuración fija de los semáforos, cuyos valores fueron obtenidos al aplicar un AG, contra el algoritmo adaptativo desarrollado. Los resultados indican que se aprecian mejoras en el flujo de tránsito (hasta 40 % en el valor de *fitness*) y destaca la buena adaptabilidad del algoritmo implementado en diferentes circunstancias. Sin embargo, se recalca que el escenario es simple y de tamaño reducido, siendo una incógnita como el algoritmo funcionará en problemas más complejos.

- A. Vogel, C. Goerick, y W. Von Seelen. Evolutionary algorithms for optimizing traffic signal operation. En *Proceedings of the European symposium on intelligent techniques (ESIT)*, p. 83–91, 2000

  Este trabajo utiliza un enfoque auto-adaptativo para mejorar el tráfico, tanto en el corto como en el largo plazo, a través de la optimización de las señales de tráfico en las intersecciones de una red de rutas.

  Se presenta la idea de que una configuración de semáforos particular, aún siendo optimizada usando simulaciones, tiene poca probabilidad de ser la mejor en todas las situaciones o en casos extremos (horas picos). Para solucionar ese problema, propone un sistema auto-adaptable que toma la información del tráfico actual usando detectores de vehículos y espacios disponibles.

  El trabajo propone el desarrollo de un AE donde cada individuo representa un sistema de fases de los semáforos, mientras la función de *fitness* tiene en cuenta las demoras en el tráfico que se obtiene utilizando simulaciones. El escenario es relativamente pequeño, con una intersección de dos calles cada una con tres líneas, donde la ruta principal tiene el doble de densidad vehicular. Los resultados obtenidos de este trabajo indican que la ventaja de usar conocimiento experto para configurar los parámetros iniciales es mínimo, ya que el algoritmo llega rápidamente a resultados similares. No se realizaron comparaciones con otros escenarios, sólo se estudió el comportamiento del AE utilizando diferentes parámetros.

- N. Rouphail, B. Park, y J. Sacks. Direct signal timing optimization: Strategy development and results. En *XI Pan American Conference in Traffic and Transportation Engineering*, p. 195–206, 2000

  Este trabajo estudia una pequeña red de tráfico con nueve intersecciones semaforizadas en la ciudad de Chicago (USA); el escenario incluye: la red vial, el tráfico de vehículos y las paradas de ómnibus. Se toman valores reales de tráfico en horas pico, comprobando que las colas de vehículos generadas en la simulación coinciden con la realidad. Utiliza el programa TRANSYT-7F (Wallace et al., 1998) que permite visualizar mapas y un simulador de tráfico comercial llamado CORSIM (Halati et al., 1997). El objetivo es resolver el problema de sincronización de semáforos, utilizando un AG cuya función de evaluación tiene en cuenta las demoras en la red y el largo de las colas producidas por los vehículos. El análisis experimental compara los valores de la realidad, con los resultados obtenidos por el AG. Los resultados indican que el AG mejora los valores del caso base, reduciendo las demoras simuladas en un 44 %.

### 2.7.1. Resumen

Esta sección presenta un breve análisis sobre los trabajos evaluados y como se relacionan con el presente proyecto.

El trabajo de Sánchez et al. (2004) posee algunos puntos de contacto con el presente trabajo al utilizar un AE paralelo con una arquitectura master-slave. La principal diferencia es que el escenario que evalúan es muy pequeño en comparación y no se estudia un escenario real. El siguiente trabajo de Sánchez et al. (2008) expande su trabajo previo y lo aplica a un escenario real, en Santa Cruz de Tenerife. Los resultados obtenidos son muy positivos, logrando mejoras de hasta 26 % en el valor de *fitness*. El trabajo de (Sánchez et al., 2010) prueba diferentes funciones de fitness teniendo en cuenta diversos factores como tiempo de viaje o velocidad promedio. Este trabajo inspiró la realización de una función multiobjetivo en el proyecto actual que tuviera en cuenta la velocidad promedio de los vehículos .

Las propuestas de Sánchez et al. (2008) y Rouphail et al. (2000) guardan puntos de contacto con el presente trabajo, al utilizar algoritmos evolutivos para resolver el problema de sincronización de semáforos sobre escenarios realistas. Sin embargo, el tamaño del escenario considerado en el presente trabajo (Corredor Garzón) es mayor al de los trabajos relacionados por su extensión (6.5 km), la cantidad de intersecciones semaforizadas (28) y el número de paradas de ómnibus (28) y reviste una complejidad mayor al incluir la lógica aplicada para el funcionamiento del carril exclusivo para ómnibus.

El trabajo de Stolfi (2012) no se enfoca en la sincronización de los semáforos, pero plantea una posibilidad interesante para mejorar el tráfico de una ciudad indicando a los conductores la mejor ruta a seguir. Esta novedosa opción se podría tomar como un elemento en trabajos futuros. Tanto Teo et al. (2010) como Stolfi (2012) plantean la simulación con un tiempo fijo, característica que se consideró para el presente proyecto.

Montana y Czerwinski (1996) y Vogel et al. (2000) proponen algoritmos adaptables en tiempo real, siendo un aporte interesante a tener en cuenta en trabajos a futuro.

En conclusión el estudio de los trabajos relacionados permitió conocer más en profundidad distintas soluciones y métodos que fueron tenidos en cuenta en menor o mayor medida en la solución propuesta.

Tabla 2.1: Resumen de los principales trabajos relacionados relevados.

| Autor | Año | Línea de trabajo |
|---|---|---|
| Montana y Czerwinski | 1996 | Sincronización de semáforos en tiempo real utilizando un algoritmo híbrido entre programación genética y algoritmo genético en un escenario simple. |
| Vogel et al. | 2000 | Sincronización de semáforos en tiempo real utilizando un algoritmo evolutivo un escenario simple. |
| Rouphail et al. | 2000 | Sincronización de semáforos utilizando un algoritmo genético en un escenario real. |
| J. Penner y Jacob | 2002 | Sincronización de semáforos utilizando un algoritmo genético en un escenario complejo. |
| Sánchez et al. | 2004 | Sincronización de semáforos utilizando un algoritmo genético en un escenario simple. |
| Sánchez et al. | 2008 | Sincronización de semáforos utilizando un algoritmo genético en un escenario real. |
| Sánchez et al. | 2010 | Sincronización de semáforos utilizando un algoritmo genético en un escenario real. Análisis de cuatro funciones de fitness diferentes. |
| Teo et al. | 2010 | Sincronización de semáforos utilizando un algoritmo genético en un escenario simple. |
| Stolfi | 2012 | Propone el concepto de ciudad inteligente, utilizando un algoritmo genético para sugerir la mejor ruta a los conductores en tiempo real. |

# Capítulo 3

# Estrategia de resolución

Este capítulo explica la estrategia de resolución seguida para resolver el problema de sincronización de semáforos en el Corredor Garzón. La metodología se basa en tres puntos: el modelado del problema, el desarrollo de un AE multiobjetivo y el uso de técnicas de alto desempeño para mejorar el rendimiento del algoritmo. El capítulo comienza presentando el modelo del problema, que incluye el diseño del mapa de la zona y la creación de instancias realistas que serán usadas para simular el tráfico. El capítulo describe al simulador de tráfico SUMO y el trabajo de campo realizado para recabar datos precisos de la realidad. Para finalizar se detalla la arquitectura propuesta para el diseño e implementación del AE multiobjetivo.

## 3.1. Modelo del problema

El problema se modela mediante una simplificación de la realidad, para considerar el uso de aquellos elementos útiles para resolver la sincronización de semáforos. El modelado se basa en el diseño de un mapa de la zona geográfica a estudiar y la construcción de instancias realistas, incluyendo datos recabados in-situ, que serán usadas en el simulador de tráfico SUMO. A continuación se presenta el simulador de tráfico, el proceso seguido para el diseño del mapa y el método de recolección de datos para desarrollar el modelo propuesto.

## 3.2. Simulaciones de tráfico

Como se explicó en el capítulo de marco teórico, SUMO es un simulador de tráfico gratuito y de código abierto que permite modelar redes de calles, vehículos, transporte público, movimiento de peatones y semáforos. Este simulador sigue un modelo microscópico, ya que realiza la simulación individual explícita de cada elemento en el escenario. SUMO incluye un conjunto de herramientas destinadas a facilitar la generación de tráfico y la construcción de mapas (SUMO, 2014).

Al existir un amplio abanico de posibilidades a la hora de elegir un simulador de tráfico, se efectuó un análisis relacionado con las posibilidades y objetivos del proyecto.

A continuación se presentan las razones que sustentan la elección de SUMO como el simulador de tráfico utilizado en el marco del proyecto:

- Portabilidad: SUMO puede ser ejecutado tanto en Windows como en Linux. Esta característica es requerida, dado que en Windows se realizan algunas porciones del diseño del mapa y Linux (CentOs) es el sistema operativo utilizado por la infraestructura computacional de alto desempeño (Cluster FING) donde se plantea desarrollar el análisis experimental.
- Disponibilidad de modos de ejecución: SUMO puede ejecutarse sin interfaz gráfica, utilizando solamente la línea de comando, lo que aumenta sensiblemente la velocidad de ejecución. Esta funcionalidad es fundamental a la hora de diseñar el AE, ya que la interacción con el simulador de tráfico debe ser eficiente. Por otro lado SUMO, presenta la opción de ejecutarse con interfaz gráfica, lo que es indispensable a la hora de visualizar la simulación. Esta opción es útil sobre todo en el inicio del desarrollo, cuando se realizan ajustes y pruebas sobre el modelo y el algoritmo.
- Software gratuito y abierto: SUMO se presenta como un proyecto de código abierto por lo que permite implementar características especificas, y no tiene costos asociados, que no podrían ser cubiertos por un proyecto estudiantil.
- Documentación y mantenimiento: El simulador incluye una detallada documentación que hace más fácil su utilización. SUMO cuenta con una comunidad muy activa que responde dudas en los foros, lo cual es útil a la hora de buscar soporte en caso de problemas o errores. SUMO también cuenta con un desarrollo activo que permite su mejora por medio de actualizaciones frecuentes.
- Configuración simple: SUMO presenta un sencillo sistema de configuración, basada en archivos XML para la ejecución de las simulaciones. La modificación de estos archivos permiten alterar la configuración de los semáforos, las propiedades y rutas de los vehículos, entre otras opciones. Complementariamente cuenta con herramientas para importar mapas de servicios como OSM (Haklay y Weber, 2008).
- Información de salida: SUMO ofrece la opción de generar una amplia variedad de datos, producida por la simulación ejecutada. Entre estos datos se encuentran la velocidad de los vehículos y el largo del recorrido, datos que son requeridos por la solución propuesta.
- Eficiencia: SUMO soporta redes de tránsito muy grandes y está diseñado para ejecutar simulaciones a gran velocidad, siendo una característica deseable por la complejidad del problema a solucionar. Un ejemplo de escenario de grandes proporciones es el *"TAPAS Cologne"*, disponible públicamente en www.sumo.dlr.de/wiki , donde se describe el tráfico de la ciudad de Colonia (Alemania) para un día entero, simulando cerca de 1.2 millones de recorridos de vehículos.

SUMO tiene un funcionamiento sencillo; toma como entradas archivos de configuración que representan la red vial, los vehículos, el tráfico y los semáforos y genera archivos de salida con información útil del escenario simulado: el tiempo de simulación, la cantidad de vehículos, la velocidad de los vehículos, la duración de los viaje, las emisiones de gases contaminantes, etc. Una instancia de ejecución del simulador SUMO se muestra en la Figura 3.1, donde se aprecia la red vial, los autos, ómnibus y paradas de ómnibus.

Dada la complejidad del problema a enfrentar en el marco del proyecto, se propone utilizar otras herramientas para realizar ciertas tareas de manera más eficiente y con menos errores. *NetConvert* viene integrado con SUMO y es utilizado para generar la red vial a partir del mapa obtenido de Open Street Map, transformándolo al formato que SUMO reconoce. Otra herramienta integrada con SUMO es *DUaRouter* que permite generar recorridos de vehículos basado en dinámicas definidas por el usuario. *Traffic Modeler*, creado por Papaleondiou y Dikaiakos (2009), es útil para la generación de tráfico de manera visual que puede ser exportado al formato reconocido por SUMO.

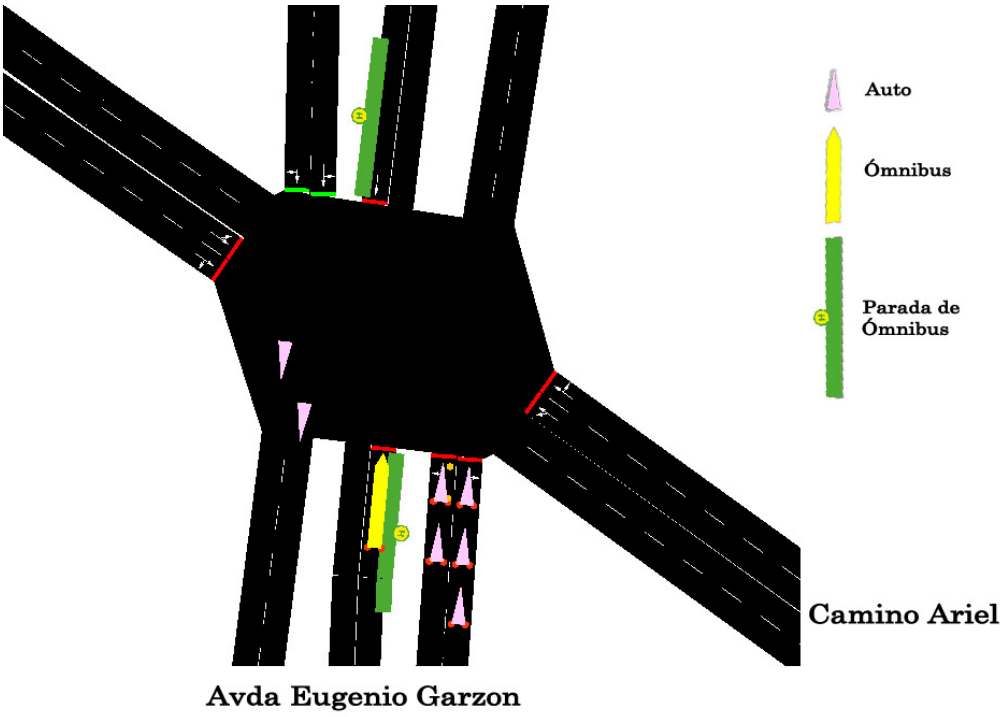


Figura 3.1: Simulación de tráfico en el simulador SUMO en el cruce entre Bulevar Batlle y Ordoñez y el Corredor Garzón.

## 3.3. Diseño del mapa de la zona de estudio

El primer paso del modelado consistió en diseñar un mapa de la zona del Corredor Garzón que sea compatible con el simulador SUMO. Se utilizó el servicio de Open Street Map (OSM) (Haklay y Weber, 2008), que cuenta con mapas libres y actualizados por una comunidad muy activa, para importar la zona de interés en el mapa y el programa Java Open Street Map Editor para adecuarlo a los elementos del problema. Para validar la exactitud del mapa importado desde OSM se utilizaron otros servicios (Google Maps y Bing Maps) y un estudio in-situ de la zona, recorriendo el Corredor Garzón y analizando en cada intersección las reglas de cruce y las direcciones de circulación. Como resultado de la verificación in-situ fueron detectadas algunas inconsistencias entre la realidad y el mapa importado. Todas las inconsistencias detectadas fueron corregidas en la versión del mapa a utilizar en la investigación.

La zona de estudio comprende al Corredor Garzón en toda su extensión y dos caminos paralelos a cada lado (véase el mapa en la Fig. 3.2). No existen calles paralelas reales, por lo que el mapa se diseñó seleccionando las calles que construyen dos caminos paralelos a cada lado del Corredor Garzón y las calles internas entre las paralelas, verificando que cada camino paralelo incluya calles doble mano o dos calles de una sola mano.

Dado los cambios realizados en el mapa, como la eliminación de calles que no formaban parte de la zona modelada, no se pudo contribuir con la comunidad OSM subiendo el mapa con las modificaciones que corregían algunas inconsistencias encontradas.

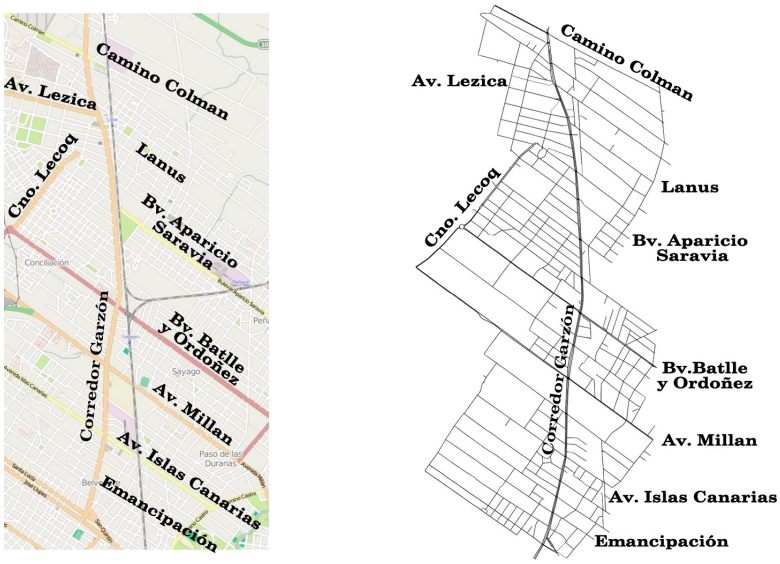


Figura 3.2: Mapas de la zona de estudio: a la izquierda el mapa extraído de OSM, a la derecha el mapa procesado para ser usado en SUMO.

La herramienta *Netconvert* fue utilizada para transformar el mapa obtenido desde OSM al formato XML que reconoce el simulador SUMO. Esta herramienta no reconoce algunos datos incluidos en el formato OSM, por ejemplo los números de las casas, las vías de tren, las vías de peatones, los caminos, las plazas, las paradas de ómnibus y los separadores que tiene el corredor de las vías para autos, entre otros. Se destaca como punto importante que NetConvert no reconoce correctamente a los corredores, ya que confunde y superpone el corredor central con las vías para los autos, seguramente por no reconocer los separadores físicos que presenta el corredor. Para solucionar este problema se utilizó el programa JOSM para eliminar los datos no reconocidos por NetConvert y editar el mapa, separando aproximadamente un metro las calles del corredor central.

La modificación realizada permitió que *NetConvert* interpretara correctamente el funcionamiento del corredor, para que los vehículos no se trasladaran entre la vía de los autos y la del corredor de ómnibus. Sin embargo, el cambio implementado generó un problema, ya que cada cruce de la realidad pasó a ser representado por tres cruces distintos (uno por cada calle del corredor). Para solucionar el problema anterior y otros errores relacionados a las reglas de circulación en las intersecciones, se realizaron ajustes manuales, modificando los archivos XML relacionados. Entre otros cambios, se especificaron manualmente las reglas de movilidad correctas en algunos cruces, por ejemplo cuando estaba prohibido girar a la izquierda.

Al finalizar el proceso de diseño se obtuvo un mapa fiel a la realidad, delimitado en la zona del Corredor Garzón, que es compatible con el simulador de tráfico SUMO. Al importar el mapa desde OSM y transformarlo con *NetConvert* se genera una configuración de semáforos por defecto, que debe ser corregida utilizando los datos relevados in situ; proceso que se explica a continuación.

## 3.4. Recolección de datos de semáforos y vehículos en trabajo de campo.

Existen datos públicos sobre la cantidad de ómnibus y sus frecuencias en el sitio web de la Intendencia de Montevideo (IMM), pero no se tienen datos sobre el tráfico vehicular en la zona del Corredor Garzón. Por este motivo se realizó un revelamiento in-situ en cinco cruces representativos de la Avenida Garzón, con características útiles para modelar variantes en la cantidad de vehículos circulantes. Los cruces seleccionados fueron: Camino Ariel, Batlle y Ordoñez, Plaza Vidella, Camino Colman y Aparicio Saravia.

El relevamiento de tráfico aplicó las recomendaciones de los textos sobre el conteo vehicular (Smith y McIntyre, 2002): se realizó un miércoles, con clima soleado y en el horario de 15:00 a 17:00 horas, para evitar sesgos que se producen los fines de semana, en horas pico, o en días de lluvia. Se realizaron filmaciones de entre 15 y 30 minutos en los cruces y luego se analizaron los vídeos para efectuar el conteo manual con la posibilidad de enlentecer el vídeo para mayor facilidad del conteo. Luego se completó una planilla electrónica con el formato presentado en la Figura 3.3, donde se reportan la cantidad de vehículos que recorren el Corredor Garzón, los que circulan por la calle que cruza y los que doblan en cada intersección.

Se decidió contabilizar sólo automóviles, en base a un relevamiento preliminar de la zona, que permitió constatar que la cantidad de otros vehículos (camiones, motos, bicicletas, etc.) no es significativa en el total del tráfico (menos del 10 % de los vehículos). La Tabla 3.1 muestra la cantidad de automóviles en cada cruce.

Tabla 3.1: Resumen del revelamiento del tráfico en la zona del Corredor Garzón.

| *intersección* | *dirección de circulación* | | | |
|---|---|---|---|---|
| | *Sur-Garzón* | *Norte-Garzón* | *Oeste* | *Este* |
| Camino Colman | 410 | 390 | 140 | 230 |
| Plaza Vidiella | 400 | 444 | 292 | 0 |
| Aparicio Saravia | 390 | 450 | 40 | 90 |
| Batlle y Ordoñez | 510 | 390 | 470 | 300 |
| Camino Ariel | 436 | 226 | 177 | 203 |

Complementariamente al conteo manual, se recorrió el corredor en automóvil a una velocidad constante, para tomar datos realistas de demora (duración total del recorrido y el tiempo que un vehículo permanece detenido en los cruces semaforizados), a ser utilizados al desarrollar las simulaciones de tráfico.

Para la configuración de los semáforos, se realizó un recorrido (en bicicleta) por toda la extensión del Corredor Garzón. Se cronometró la duración de las luces de los semáforos en cada esquina recorriendo el camino de ida, y para validar los tiempos también se relevaron

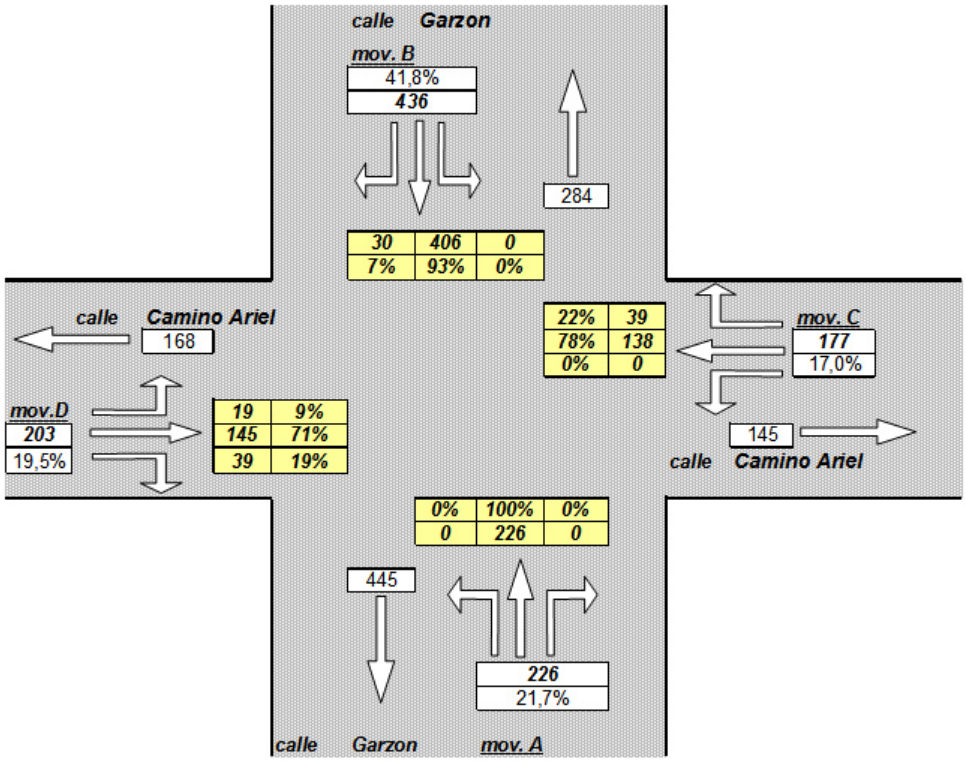


Figura 3.3: Ejemplo de planilla electrónica para el conteo manual de tráfico en la intersección entre el corredor Garzón y Camino Ariel

las duraciones de las luces en el camino de vuelta. Estos tiempos fueron verificados por los vídeos obtenidos en el conteo del tráfico.

## 3.5. Desarrollo de instancias realistas del problema

Una vez diseñado el mapa de la zona de estudio y relevados los datos de la realidad, se crearon instancias realistas del problema. El caso base representa la situación actual (tráfico, red vial y sincronización de semáforos) del Corredor Garzón. Para configurar la simulación del tráfico en SUMO se utilizan tres archivos en formato XML: *i*) la *configuración de los semáforos*, que detalla su ubicación en el mapa, la duración de las luces y el comienzo de fase (informacion obtenida de los datos relevados in-situ); *ii*) la *ruta de los vehículos*, con el recorrido de cada vehículo, creado con el software *Traffic Modeler* (Papaleondiou y Dikaiakos, 2009), que permite generar visualmente modelos de tráfico complejos; se optó por un modelo de movilidad entre áreas, que permite una buena granularidad al especificar la densidad de tráfico; y *iii*) el *recorrido de los ómnibus*, que indica el recorrido, la frecuencia, la ubicación de las paradas, y el tiempo que se demora en la parada; las frecuencias y los recorridos de los ómnibus, fueron obtenidos de datos públicos de la IMM para las líneas urbanas *G*, *409*, *2*, *D5* y *148*. Estos tres elementos serán explicadas en detalle a continuación.

### 3.5.1. Configuración de semáforos

El diseño del mapa genera una configuración de semáforos por defecto que no es correcta, por tanto, se corrige utilizando los datos relevados previamente sobre las duraciones de las luces y las fases de los semáforos de cada intersección del Corredor Garzón. La modificación se realiza sobre un archivo XML que representa la configuración de semáforos de todo el Corredor Garzón, que será utilizado como archivo de entrada a la hora de realizar las simulaciones de tráfico.

### 3.5.2. Generación de tráfico vehicular

Existen varios modelos disponibles para representar la circulación de los vehículos. El modelo *aleatorio* genera recorridos al azar para los vehículos. En el modelo *basado en áreas* se especifican las zonas donde el tráfico se origina y donde finaliza. Un modelo más complejo es el *basado en actividad*, donde se definen la cantidad de casas, el número de vehículos y la población de una zona determinada. Luego, el modelo genera la densidad de tráfico basado en los tipos de actividades que se realizan comúnmente, como ir al trabajo, hacer las compras, ir a la escuela, etc. Finalmente, un modelo muy utilizado, sobre todo en escenarios de tamaño reducido, es el *definido por el usuario*, que permite determinar la ruta de cada vehículo con mayor detalle.

Se utilizó el programa *Traffic Modeler* que se caracteriza por generar modelos de tráfico complejos de manera visual como se ve en la Figura 3.4. Se optó por un modelo de movilidad entre áreas, lo que permite una buena granularidad al especificar la densidad de tráfico. Para generar múltiples instancias con distintas configuraciones de tráfico, en el *Traffic Modeler* se especifican la cantidad de autos y las áreas entre las cuales se movilizan, utilizando el mapa base diseñado previamente, luego el programa genera los recorridos que son exportados en un archivo. Este archivo es procesado con la herramienta *duaRouter* (incluido en SUMO) para generar otro archivo que contiene las rutas especificas de cada recorrido de los autos. El archivo resultante será incluido al ejecutar las simulaciones de tráfico.

### 3.5.3. Generación de recorridos de ómnibus

Se plantea el uso de dos tipos de vehículos: autos y ómnibus. Cada tipo de vehículo posee diferentes características asociadas, como el tamaño, la aceleración y la velocidad máxima. No se especifican otros tipos de vehículos como motos o camiones, ya que su proporción respecto al total de tráfico en la zona del Corredor Garzón no lo justificaba. Las frecuencias y los distintos recorridos de los ómnibus, fueron obtenidos de datos públicos de la Intendencia Municipal de Montevideo (IMM). Las frecuencias incluyen las líneas líneas urbanas: *G*, *409*, *2*, *D5* y *148*. Las líneas de ómnibus suburbanas no utilizan el carril exclusivo central, sino las calles adyacentes provocando un comportamiento diferente al de las líneas urbanas. Por este motivo las velocidades de las líneas suburbanas no serán tomadas en cuenta en la evaluación de la función de *fitness* del AE, para no generar distorsiones en los resultados.

La ubicación y líneas de cada parada se obtuvo del Servicio de Información Geográfica de Montevideo (http://sig.montevideo.gub.uy). Existen 14 paradas de líneas urbanas por el Corredor Garzón para el recorrido de ida y la misma cantidad para el recorrido de vuelta. Las detenciones de los ómnibus en cada parada fueron calculadas utilizando los vídeos del trabajo de campo para el relevamiento del tráfico, donde se cronometraron los

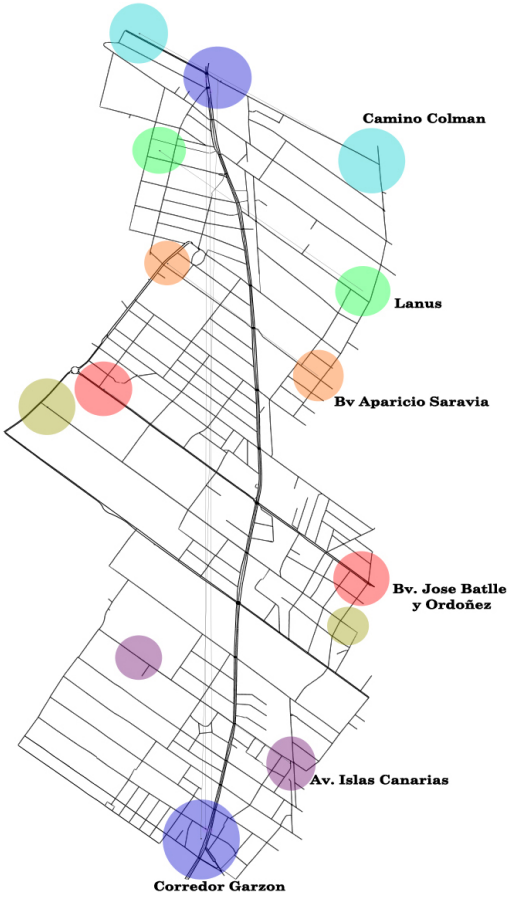


Figura 3.4: Mapa diseñado en el TrafficModeler para la generación de tráfico. Los círculos del mismo color especifican el tráfico entre esas áreas.

tiempos que demoraban las detenciones en las paradas. De este estudio se calcularon los tiempo de detención en las paradas, que se encuentran en el rango de 15 a 35 segundos.

Se realizó un estudio basado en datos de GPS proporcionados por la IMM. Estos datos cuentan con la posición exacta, la velocidad instantánea y la referencia del ómnibus, para un conjunto de líneas seleccionadas tomadas en un período de una semana (del 01/09/14 al 08/09/14). Para este procesamiento se utilizó QGIS (www.qgis.org), un sistema de información geográfico capaz de visualizar, editar y realizar operaciones sobre elementos geográficos. Adicionalmente fue necesario el desarrollo de *scripts* para obtener las estadísticas necesarias, ya que la cantidad de información estudiada era muy grande (1.5Gb). El uso de QGIS permitió seleccionar, filtrar y relacionar los datos de posición y velocidad con las líneas que circulan por el Corredor Garzón. Luego de procesar los datos se constató una velocidad media para los ómnibus de 14.5 km/h en el Corredor Garzón, lo que permitió ajustar mejor el modelo.

### 3.5.4. Resumen

Una vez obtenidos los archivos con la configuración de los semáforos, el recorrido de los vehículos y el recorrido de los ómnibus, se cuenta con todos los elementos necesarios para la creación de instancias realistas del problema, que serán utilizadas en el simulador

para modelar la realidad. En el análisis experimental presentado en el capítulo 5 se describen los escenarios que hacen uso de estas instancias, modificando algunas variables como la densidad del tráfico para ajustarlo a las necesidades de la experimentación.

## 3.6. Formulación del problema

El problema de optimización a resolver es el de maximizar la velocidad promedio de ómnibus y vehículos en la zona del Corredor Garzón. Se plantea optimizar dos objetivos simultáneamente, utilizando un enfoque de optimización multiobjetivo. Para implementarlo, se utiliza una combinación lineal de los objetivos (velocidad de ómnibus y de vehículos), una técnica que proporciona un método sencillo y adecuado para resolver problemas de optimización que involucran menos de tres objetivos y el frente de Pareto del problema es convexo (Coello et al., 2002).

Los objetivos del problema, la velocidad promedio de los ómnibus ($vpb$) y la velocidad promedio del resto de los vehículos ($vpv$), fueron seleccionados por ser las principales métricas de calidad de servicio para los usuarios del Corredor Garzón, y por su utilidad a la hora de realizar comparaciones con la realidad.

La formulación multiobjetivo del problema se presenta en la Ecuación 3.1. Se plantea optimizar una combinación lineal de los objetivos, donde $w_1$ y $w_2$ son pesos que pueden ser utilizados para dar mas prioridad a un tipo de vehículo sobre otro. Inicialmente se busca optimizar con igual importancia ambas velocidades, por lo cual se utilizan los valores $w_1 = w_2 = 1$, en la sección de análisis experimental se realizarán experimentaciones con otros valores.

$$f = w_1.vpb + w_2.vpv \tag{3.1}$$

La optimización de la velocidad promedio de los vehículos se realiza en toda la red vial considerada en el mapa estudiado, de manera global. Se busca mejorar la velocidad promedio en vehículos que circulan por Garzón y por el resto de las calles del escenario geográfico estudiado. Aplicar una técnica de optimización global es fundamental para lograr una mejora sostenible en la velocidad de los vehículos, ya que si cada cruce se optimizara individualmente, se podrían generar problemas de congestión en otras intersecciones o sub-zonas del mapa.

## 3.7. Arquitectura de la solución

El esquema de la Figura 3.5 muestra la arquitectura propuesta para la resolución del problema de sincronización de semáforos. La arquitectura se puede dividir en dos grandes bloques que interactuan entre sí, por un lado el algoritmo de optimización y por otro la evaluación de la solución.

El AE implementa paralelismo con un enfoque maestro-esclavo. El primer paso consiste en generar la población inicial de individuos. Estos individuos representan una configuración de semáforos para todo el Corredor Garzón, que incluye la duración de las fases y el *offset*. Luego, por cada individuo se genera un archivo compatible con el simulador, que representa la configuración de semáforos. Este archivo se utiliza como entrada para ejecutar el simulador al momento de evaluar la función de *fitness*, como se aprecia en la flecha azul hacia la derecha de la Figura 3.5. Estas evaluaciones se realizan

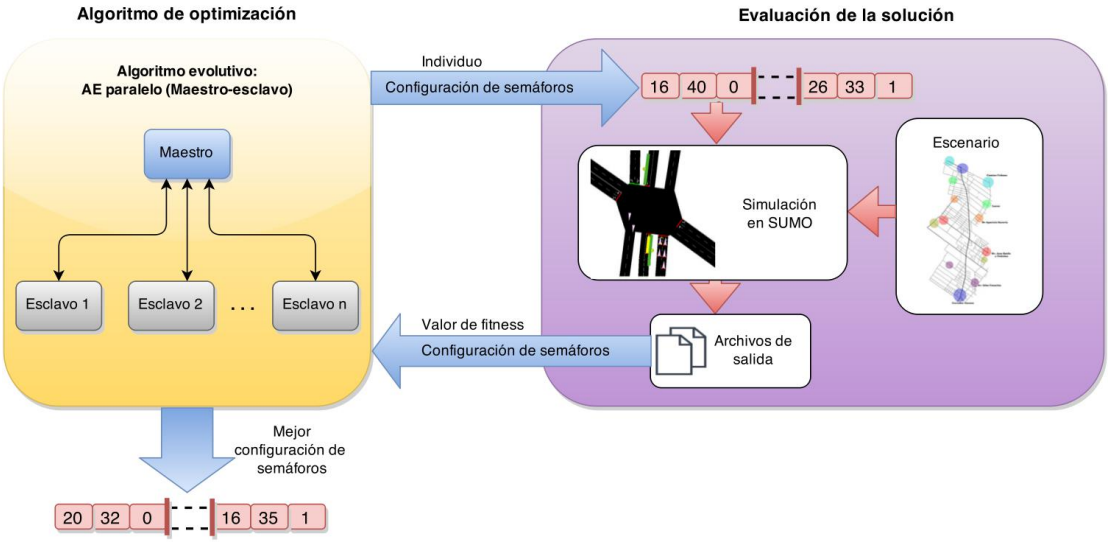


Figura 3.5: Arquitectura de la solución propuesta.

en paralelo, por lo que pueden existir tantas evaluaciones ejecutando al mismo tiempo como individuos en la población. Al finalizar la ejecución del simulador SUMO se generan archivos de salida que son enviados hacia el AE (específicamente hacia cada uno de los esclavos) como indica la flecha azul que apunta hacia la izquierda en la Figura 3.5. Los datos de salida incluyen información necesaria para calcular el *fitness* de cada individuo, como la velocidad media de ómnibus y otros vehículos. El proceso maestro consolida los datos enviados por los esclavos y aplica los operadores evolutivos continuando con la siguiente iteración. El AE se detiene al alcanzar un número de generaciones determinado, en ese momento se crea un archivo con la mejor configuración de semáforos encontrada, que corresponde a la mejor solución del problema de optimización planteado.

# Capítulo 4

# Implementación de la solución

Este capítulo detalla la implementación del AE desarrollado. Comienza describiendo a la biblioteca Malva, utilizada para la implementación y continúa especificando el AE desarrollado donde se explica: la representación del cromosoma, la generación de la población, la función de *fitness* y el funcionamiento de los operadores evolutivos. Para finalizar se detalla el modelo de paralelismo utilizado.

## 4.1. Implementación: Biblioteca Malva

Dada la complejidad del problema, se contempló desde el inicio del proyecto la utilización de un *framework* para el desarrollo del AE. De esta forma, se facilita el desarrollo y la implementación de una solución robusta y flexible. Como se detalló en el marco teórico, existen varias opciones disponibles de *framework* para el desarrollo de algoritmos evolutivos. Por las particularidades del problema planteado, existen ciertos requerimientos a la hora de seleccionar un *framework*, entre los que se destacan:

- Código abierto y gratuito: Al ser un proyecto de investigación de índole académico es conveniente que no existan costos económicos asociados. Además, es importante que se cuente con el código abierto con el fin de introducir modificaciones en el código base o corregir errores.

- Algoritmo Genético: El *framework* debe facilitar el desarrollo de un AG ya que es la base en la que se sustenta el proyecto.

- Algoritmo paralelo: La complejidad del problema y de las instancias que se estudian, justifican la necesidad de que el *framework* incluya la opción de ejecución en paralelo para reducir los tiempos de ejecución. Si no cuenta con esta funcionalidad nativa, es deseable que al menos exista la posibilidad de modificar el código base para agregarla.

- Plataforma: Es requerido que el *framework* se pueda ejecutar en la plataforma de computación de alto desempeño Cluster FING cuyo sistema operativo es Linux CentOs, ya que es donde se realiza el análisis experimental del trabajo.

- Confiabilidad: Es deseable que el *framework* sea lo suficientemente estable como para tener confianza de que el algoritmo funcionará de manera correcta. Se valorará positivamente que el *framework* cuente con documentación accesible, ejemplos de código desarrollado, y casos de éxito que lo utilicen.
- Lenguaje: Es deseable que el *framework* esté desarrollado en C++ o Java, por la experiencia que el equipo de desarrollo cuenta en estos lenguajes.
- Multiobjetivo: Es deseable que el *framework* soporte algoritmos multiobjetivo pero no requerido, ya que la función que se quiere implementar es sencilla y puede usarse una agregación lineal de los objetivos.

Luego de analizar los puntos anteriores, se seleccionó la biblioteca Malva para la implementación del AE. Malva cumple con la mayoría de los puntos: es de código abierto y gratuito, facilita el desarrollo de un AG, funciona en la plataforma Cluster FING, utiliza el lenguaje C++ y el punto de mayor peso en la decisión fue la confiabilidad que daba saber que ya se habían realizado implementaciones exitosas de soluciones similares utilizando AE paralelos en la plataforma Cluster FING. Los dos puntos donde Malva no satisface totalmente los requerimientos son: el funcionamiento en paralelo y el soporte de algoritmos multiobjetivo. Estos problemas se pueden resolver de forma aceptable y sencilla como se explica a continuación. Para el caso del funcionamiento en paralelo al ser Malva de código abierto, es posible modificar el código para soportar la creación de hilos de ejecución, este procedimiento se explica en más detalle en la última sección de este capítulo que trata sobre el modelo de paralelismo y su implementación. El segundo punto acerca del soporte de algoritmos multiobjetivo por parte del *framework*, no supone un gran inconveniente, ya que la función de *fitness* que se plantea utilizar es una combinación lineal sencilla de objetivos y puede ser utilizada en Malva. A continuación se detalla su funcionamiento y características.

Como se explicó anteriormente, Malva (Fagúndez, 2014) surge como una variante del proyecto Mallba (Alba et al., 2006). Malva propone la actualización, mejora y desarrollo de Mallba como un proyecto de código abierto colaborativo. Su objetivo es proveer varios esqueletos de heurísticas de optimización que puedan ser utilizados y extendidos de manera fácil y eficiente. Los esqueletos se basan en separar dos conceptos: el problema concreto que se quiere resolver y el método utilizado para resolverlo. Por lo tanto, un esqueleto se puede ver como una instancia de una plantilla genérica para resolver un problema particular, manteniendo todas las funcionalidades genéricas.

Malva utiliza el lenguaje C++, con el objetivo de brindar modularidad y flexibilidad. Los esqueletos se ofrecen como un conjunto de clases *requeridas* que el usuario deberá modificar para adaptarlo a su problema, y clases *provistas* que incluyen todos los aspectos internos del esqueleto siendo independientes del problema particular. Entre los algoritmos provistos se encuentran el AG, el algoritmo CHC (Eshelman, 1991) y otros.

## 4.2. Especificación del AG utilizado

Se utiliza un AG tradicional (Goldberg, 1989) provisto por la biblioteca de optimización Malva (Fagúndez, 2014), implementada en C++. El código del AG fue modificado para incluir las características del problema a resolver y para proveer la funcionalidad de ejecución en paralelo siguiendo el modelo maestro-esclavo para evaluar las posibles soluciones del problema, generando un hilo de ejecución por cada simulación. El siguiente esquema describe el funcionamiento del algoritmo utilizado:

**Algoritmo 3** Algoritmo Genético de Malva.

```
1: t = 0
2: Inicializar( P(t))
3: Evaluar estructuras en ( P(t))
4: while No terminar do
5:    t++
6:    Seleccionar C(t) de P(t-1)
7:    Recombinar estructuras en C(t) formando C'(t)
8:    Mutar estructuras en C'(t) formando C"(t)
9:    Evaluar estructuras en C"(t) generando un hilo de ejecucion por cada una
10:   Consolidar valores de la evaluacion
11:   Reemplazar P(t) de C"(t) y P(t-1)
12: end while
```

A continuación se realiza un resumen de las características del algoritmo implementado, que en la siguiente sección serán descritas en detalle:

- Se aplica un modelo de AE paralelo, utilizando el esquema maestro-esclavo, donde en cada iteración el maestro genera un hilo por cada evaluación de la función de *fitness* y luego espera a la terminación de todos los hilos para consolidar los datos.

- Se considera una formulación multiobjetivo del problema, que intenta optimizar tanto la velocidad promedio de vehículos como la de ómnibus, teniendo cada componente un peso específico en la combinación lineal utilizada para definir la función de *fitness*.

- La representación de soluciones: es un vector de números naturales, que representan la duración de las fases de los semáforos y el *offset* para todos los semáforos del Corredor Garzón.

- Respecto a los operadores evolutivos: se implementa una variante del cruzamiento de un punto específico para este problema y se utilizan dos tipos de mutación para modificar la duración de fase y el *offset* de los semáforos.

- Respecto a los criterios de selección y reemplazo, el AE propuesto utiliza un modelo generacional y reemplaza padres por hijos. La selección de los padres se realiza por el método de torneo de tres individuos y la selección de hijos por el método de selección proporcional.

### 4.2.1. Representación de individuos

La solución propuesta contempla dos elementos clave de la planificación de semáforos: la *duración de fase* y el *offset* (en segundos). La Fig. 4.1 presenta un ejemplo de la representación utilizada para las soluciones del problema: la información se agrupa lógicamente en cruces, el valor de cada *gen* es la duración en segundos de una fase en un cruce (se omiten las fases de luces amarillas, que no modifican los tiempos reales de paso de vehículos) y el *offset* se encuentra al final de cada cruce representado. El largo de una representación depende de la cantidad de cruces y de la cantidad de fases que incluye. Con la representación propuesta se busca una optimización global de todo el sistema y no individualmente para cada cruce, lo cual es fundamental para mejorar la velocidad promedio del Corredor Garzón y del resto de las calles que lo cruzan.

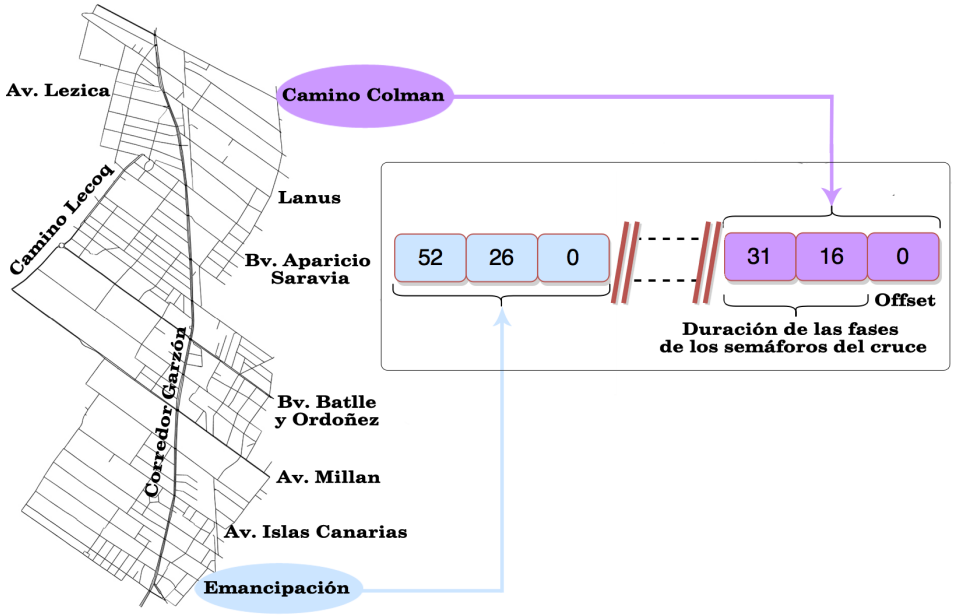


Figura 4.1: Ejemplo de representación de las soluciones para el problema.

Toda la información del cromosoma se guarda y carga en el formato *XML* del archivo de configuración de semáforos, que utiliza el simulador SUMO. La Figura 4.2 muestra como se modela la esquina de Garzón y Emancipación en el formato antes mencionado. Para las distintas fases de los semáforos de una determinada esquina se utiliza la etiqueta *phase*, donde *state* contiene información sobre la secuencia de luces. Por ejemplo, la fase *GGGGrrrGGGGrrr* indica la secuencia de luces, *G* indica la luz verde con preferencia, *g* indica la luz verde sin preferencia, *r* la luz roja, *y* luz amarilla. La etiqueta *duration* especifica la duración de la fase y corresponde a los valores del cromosoma. El tiempo indicado en la etiqueta *offset* especifica el inicio de la fase y será el correspondiente al valor del gen de *offset* en el cromosoma para el cruce de Garzón y Emancipación.

```
<!-- EMANCIPACION -->
<tlLogic id="cluster_1894127318_917520037_917579892" type="static" programID="1" offset="0">
    <phase duration="52" state="GGGGrrrGGGGrrr"/>
    <phase duration="3" state="yyyyrrryyyyrrr"/>
    <phase duration="26" state="rrrrGGgrrrrGGG"/>
    <phase duration="6" state="rrrryyyrrrryyy"/>
</tlLogic>
```

Figura 4.2: Representación de la configuración de semáforos en Sumo

El simulador representa en las esquinas (mediante el uso de flechas blancas) la elección de carriles habilitados para los vehículos que transitan por un determinado carril; en el caso de un giro, la flecha apunta hacia donde se encuentra el carril del giro. La Figura 4.3 muestra como se modela la esquina de Garzón y Emancipación en el simulador, a la que se le añadieron los números del 1 al 14 para indicar todas las flechas blancas de la esquina y establecer la correspondencia con la representación en SUMO en la Figura 4.2. Por lo tanto, para el estado *GGGGrrrGGGGrrr* se tiene que: *i) GGGG* se corresponde con las flechas 1, 2, 3 y 4, *ii) rrr* se corresponde con 5, 6 y 7, *iii) GGGG* se corresponde con 8, 9, 10 y 11 y *iv)rrr* se corresponde a 12, 13 y 14. Estableciendo esta correspondencia para cada cruce, se modelan las secuencias y duraciones de luces originales de manera que represente lo relevado in-situ.

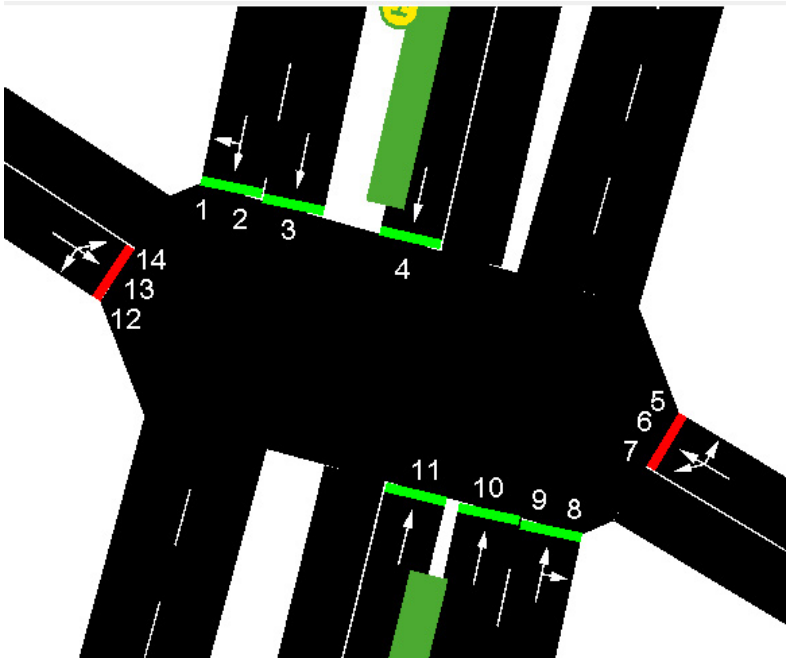


Figura 4.3: Representación de una fase de los semáforos para el cruce de Emancipación y Garzón.

### 4.2.2. Generación de la población inicial

Para inicializar la población se toma como referencia la configuración obtenida a partir de los datos de la realidad, recolectadas en el relevamiento realizado. Luego, para cada cruce se varían las duraciones de las fases de manera aleatoria entre un rango de valores configurable. Además, la fase inicial se selecciona aleatoriamente entre la cantidad de fases del cruce.

### 4.2.3. Función de *fitness*

La función de *fitness* utilizada corresponde a la formulación presentada en el capítulo 4. Se considera una formulación multiobjetivo del problema, que intenta optimizar tanto la velocidad promedio de vehículos como la de ómnibus, teniendo cada componente un peso específico en la combinación lineal utilizada para definir la función de *fitness*.

### 4.2.4. Operadores evolutivos

#### Operador de cruzamiento

Se utiliza cruzamiento de un punto. Se selecciona hasta un determinado cruce que se decide aleatoriamente como punto de corte, y se producen dos hijos seleccionando las soluciones de un individuo, desde el primer cruce hasta el punto de corte y se une con las soluciones del otro individuo desde el punto de corte en adelante; el otro hijo se forma de manera inversa. En el escenario que plantea la Figura 4.4, los padres cuentan con tres cruces. Se elige un punto de corte aleatoriamente entre dos cruces, cortando a los padres e intercambiando los trozos del cromosoma para generar los hijos como se ve en la Figura 4.4 .

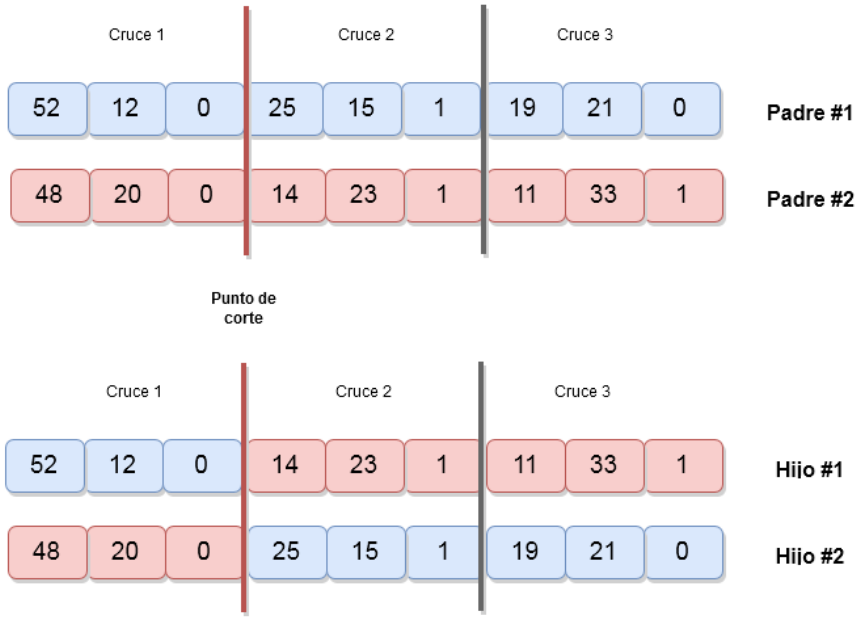


Figura 4.4: Ejemplo del operador de cruzamiento utilizado

### Operador de mutación

Se implementaron dos tipos de mutación:

- Mutación de duración de fase: para cada fase de cada cruce se modifica su duración sumando o restando una cantidad dada de segundos. Se valida que el nuevo valor obtenido se encuentre dentro de un rango especificado de entre 5 y 60 segundos (valores que se pueden modificar), para que no se produzcan casos alejados de la realidad, por ejemplo una intersección donde los vehículos tengan menos de 5 segundos para cruzar.
- Mutación del *offset*: Para cada *offset* del cromosoma se suma o resta una cantidad de segundos dada y se valida que esté dentro de un rango prefijado de entre 0 y el máximo valor de *offset* para ese cruce.

Ambas mutaciones son aplicadas de acuerdo a una tasa de probabilidad prefijada.

### Selección y reemplazo

Se utilizan los operadores y criterios provistos por el AG de Malva. La selección de padres se realiza por el método de torneo de tres individuos y la selección de hijos por el método de selección proporcional. Respecto a la política de reemplazo, se utiliza un modelo "+", en el cual los hijos compiten con los padres por la supervivencia en la siguiente generación.

### Parámetros del algoritmo

Los parámetros del algoritmo son: el número de generaciones que se realizan hasta detener el algoritmo, el tamaño de la población que indica la cantidad de individuos incluidos en el AE, la tasa de probabilidad de cruzamiento y la de mutación que se utiliza para determinar cuando aplicar los operadores de cruzamiento y de mutación. Los valores específicos de los parámetros del AE se ajustan en el capítulo 5 donde se realiza el análisis de configuración paramétrica.

## 4.3. Modelo de paralelismo e implementación

Uno de los requisitos planteados al inicio del presente trabajo era la creación de un AG que soportara paralelismo, con el objetivo de reducir los tiempos de ejecución de las simulaciones necesarias para evaluar la función de *fitness* de los individuos. El motivo principal fue que los escenarios planteados se consideraron complejos por lo que la ejecución del AE insumiría un tiempo considerable. El código base del AE fue modificado para que soportara la ejecución en paralelo. Específicamente, se desarrolló una nueva forma de evaluar a los individuos, generando un hilo de ejecución por cada individuo.

El código del AG provisto por la biblioteca Malva (newGA.pro.hh) fue modificado agregando dos nuevas funciones encargadas de llamar a la función de *fitness* y asignar el valor obtenido al individuo tanto para padres (*evaluate_parent_threads*) como para hijos (*evaluate_offpring_threads*). En la Figura 4.5 se presenta el código de la función *evaluate_parent_threads* que se explica a continuación.

```
372    void *Population::evaluate_parent_threads(void *data) {
373        struct threadData *d = (struct threadData *) data;
374        int i = (int) d->id_i;
375        Population *p = (Population *) d->poblacion;
376        int index = p->_fitness_values[i].index;
377
378        if (p->_fitness_values[i].change) {
379            p->_fitness_values[i].change = false;
380            p->_fitness_values[i].fitness = p->_parents[index]->fitness(i);
381            p->_evaluations++;
382        }
383    }
```

Figura 4.5: Función para la evaluación de individuos en paralelo utilizando threads, agregada al código provisto por Malva.

En el código de la Figura 4.5 el parámetro *data* contiene información sobre la población, el individuo a evaluar, su índice y un vector con los valores de *fitness* de la población. En la línea 378 se hace un chequeo para saber si el valor de *fitness* ya fue obtenido para el individuo, de esta forma no se repiten evaluaciones, en la línea 380 se ejecuta la función de *fitness* y se aplica el valor de *fitness* al individuo, luego en la línea 381 se incrementa el número de evaluaciones realizadas en la población. La función *evaluate_offpring_threads* es análoga a *evaluate_parent_threads* pero específica para los hijos, por lo que no se considera necesario mostrar su código.

La función *evaluate_parents()* provista por Malva se encarga de iterar sobre los padres de la población y ejecutar la evaluación sobre cada uno de ellos para obtener el valor de *fitness* asociado. Esta función fue modificada para generar un hilo de ejecución por cada evaluación, como se aprecia en la porción de código de la Figura 4.6. En la línea 434 se genera el hilo llamando a la función agregada anteriormente *evaluate_parent_threads*. Luego en la línea 504 se espera a que todos los hilos finalicen su ejecución. Análogamente se realizan modificaciones similares en la función *evaluate_offsprings* que itera sobre los hijos, para que genere un hilo por cada evaluación de los hijos, llamando a la función de *evaluate_offpring_threads*.

```
484     .
485     .
486     for (int i = 0, k = 0; i < _fitness_values.size(); i++, k++) {
487         if (_fitness_values[i].change) {
488 
489             threadDataArray[k].id_i = i;
490             threadDataArray[k].id_j = 0;
491             threadDataArray[k].poblacion = this;
492             threadDataArray[k].resultado = 0.0;
493             rc = pthread_create(&threads[k], NULL, &Population::evaluate_parent_threads,
494                                 &threadDataArray[k]);
495             if (rc) {
496                 printf("ERROR; codigo de error de pthread_create() es %d\n", rc);
497                 exit(-1);
498             }
499             real_threads++;
500         }
501     }
502     //Esperar que finalicen los threads
503     void * status;
504     for (k = 0; k < real_threads; k++) {
505         rc = pthread_join(threads[k], &status);
506         if (rc) {
507             printf("ERROR; codigo de error de pthread_create() es %d\n", rc);
508             exit(-1);
509         }
510     }
511     .
```

Figura 4.6: Modificaciones de la función evaluate_parents.

Se utilizó el esquema maestro-esclavo para el modelo de paralelismo. El proceso maestro se encarga de la mayoría de las etapas del AE, al comenzar, inicializa la población y se encarga de distribuir la evaluación de los individuos hacia los esclavos, creando un hilo de ejecución por cada esclavo. Luego el proceso maestro espera a que las evaluaciones terminen para obtener los valores de *fitness* de los esclavos. Una vez obtenidos los valores, selecciona a los mejores individuos y le aplica los operadores de cruzamiento y mutación. Para finalizar ejecuta el criterio de reemplazo para generar la siguiente población. Mientras, los esclavos se encargan solamente de obtener el individuo enviado por el maestro y de efectuar la evaluación del individuo, que corresponde a ejecutar el simulador y obtener los datos de salida.

El modelo de paralelismo no cambia el comportamiento del algoritmo secuencial, sólo se enfoca en mejorar la eficiencia computacional para poder abordar escenarios realistas y complejos.

En el capítulo siguiente, en la sección de análisis experimental se realiza un estudio de la eficiencia computacional del AG paralelo, para comparar los tiempos de ejecución entre la versión paralela y la secuencial.

# Capítulo 5

# Análisis experimental

Este capítulo presenta la evaluación experimental del AE multiobjetivo propuesto para resolver el problema de sincronización de semáforos en el Corredor Garzón. Se describe la plataforma de ejecución utilizada en la evaluación experimental y se estudia la configuración paramétrica del AE con el objetivo de obtener una mejor calidad de resultados, A continuación se describen los escenarios utilizados en la evaluación del AE y los experimentos de validación, reportando los resultados numéricos. Finalmente se ofrece un breve análisis de la eficiencia computacional del AE propuesto.

## 5.1. Plataforma de ejecución y desarrollo

Dada la complejidad del problema, el análisis experimental se realizó en la plataforma de cómputo de alto desempeño Cluster FING, de la Universidad de la República (Nesmachnow, 2010). Para acelerar el tiempo de procesamiento se ejecuta el AE en paralelo, distribuyendo la carga de procesar las simulaciones de tráfico en múltiples núcleos de cómputo. El Cluster FING cuenta con procesadores AMD Opteron 6272 de 2.09GHz, con 64 núcleos, 48GB RAM, y sistema operativo CentOS Linux 6.5.

## 5.2. Experimentos de calibración paramétrica del AE

Se llevó a cabo un análisis previo para determinar el mejor valor de los parámetros del AE propuesto. Este análisis tiene como objetivo obtener una mejor calidad de resultados. El análisis consistió en determinar valores adecuados para el *tiempo de simulación* de los escenarios planteados en SUMO, el *criterio de parada* y el *tamaño de la población* del AE. Posteriormente, se realizó un ajuste de las *probabilidades de cruzamiento* y de *mutación* en el AE.

Los experimentos de ajuste paramétrico se realizaron sobre tres escenarios de tráfico: *i*) *tráfico bajo*, con 30 ómnibus y 500 vehículos; *ii*) *tráfico medio*, con 60 ómnibus y 1000 vehículos; y *iii*) *tráfico alto*, con 120 ómnibus y 2000 vehículos. Estos escenarios no incluyen datos recabados de la realidad (como el tráfico vehicular o la frecuencia de los ómnibus), para no ajustar los parámetros a un caso particular.

En el ajuste paramétrico, los primeros parámetros que se definen son: el tiempo de simulación, el criterio de parada y el tamaño de población. Luego de establecidos los valores para los parámetros mencionados, se realizan pruebas para todas las combinaciones

de probabilidad de cruzamiento y mutación, buscando los valores con los cuales el AE obtiene una mejor calidad de resultados.

### 5.2.1. Tiempo de simulación

El tiempo de simulación refiere a la duración de la ejecución de una simulación usando SUMO. Es un parámetro que se puede configurar y permite tener un mejor control sobre los tiempos totales de ejecución del algoritmo.

Se establece un tiempo de simulación de 4000 *steps* (medida interna de tiempo del simulador) que representa 66 minutos en la realidad. El tiempo de ejecución real de cada simulación depende de la plataforma computacional utilizada y de la cantidad de vehículos considerados en el escenario, variando entre los 5 y 20 segundos. Para cada escenario, se valida que más del 80 % de los vehículos completen la simulación (lleguen a sus destinos).

### 5.2.2. Criterio de parada

Se decidió utilizar como criterio de parada un determinado número de generaciones. Este criterio permite estandarizar las pruebas y evaluar una comparación justa entre los distintos escenarios de tráfico. Para determinar el criterio de parada del AE se busca un compromiso entre los resultados numéricos y el tiempo de ejecución. Se realizaron 10 ejecuciones del AE para cada tipo de tráfico, y los resultados indicaron que el mejor valor de *fitness* no varía significativamente luego de la generación 400, por lo que se decidió utilizar un criterio de parada de 500 generaciones, que involucra un tiempo de ejecución del AE en un margen razonable (entre 1 y 24 horas).

### 5.2.3. Tamaño de la población

Para determinar el número de individuos de la población se tiene en cuenta el valor de *fitness* obtenido por el AE, su tiempo de ejecución y los recursos computacionales. Teniendo en cuenta la distribución de recursos computacionales para realizar las simulaciones de tráfico, el número máximo de individuos (64) queda determinado por la cantidad de núcleos en un nodo del Cluster FING. Además, se estudian poblaciones de 32 y 48 individuos.

La Tabla 5.1 reporta los resultados de 10 ejecuciones del AE por cada tipo de tráfico estudiado, con diferente número de individuos en la población. Los resultados indican que no existen diferencias significativas en los valores de fitness, por lo cual se elige un tamaño de población de 32 individuos, teniendo en cuenta el menor tiempo de ejecución del AE y el menor uso de recursos computacionales.

Tabla 5.1: Resultados del análisis de diferentes valores para el tamaño de población.

| *#población* | *fitness* | | *tiempo de ejecución (m)* |
|---|---|---|---|
| | *mejor* | *promedio*$\pm\sigma$ | |
| 32 | 17.28 | 16.37$\pm$0.5 | 4853$\pm$155 |
| 48 | 16.19 | 15.84$\pm$0.3 | 6772$\pm$256 |
| 64 | 17.27 | 16.46$\pm$0.6 | 10184$\pm$526 |

### 5.2.4. Probabilidades de cruzamiento y mutación

Para el análisis paramétrico de la probabilidad de cruzamiento ($p_C$) se consideraron tres valores candidatos: {0.5, 0.8, 1} y para la probabilidad de mutación ($p_M$) los valores {0.01, 0.05, 0.1}. De las nueve combinaciones posibles, se realizaron tres ejecuciones independientes del AE para cada uno de los tres tipos de tráfico (bajo, medio y alto).

| $p_C$ | $p_M$ | *fitness prom.*$\pm\sigma$ |
|---|---|---|
| 0.5 | 0.01 | 16.09±0.30 |
| 0.5 | 0.05 | 15.60±0.17 |
| 0.5 | 0.1 | 16.16±0.42 |
| 0.8 | 0.01 | 16.04±0.55 |
| 0.8 | 0.05 | 15.85±0.32 |
| 0.8 | 0.1 | 16.08±0.34 |
| 1 | 0.01 | 16.08±0.45 |
| 1 | 0.05 | 15.82±0.34 |
| 1 | 0.1 | 16.04±0.25 |

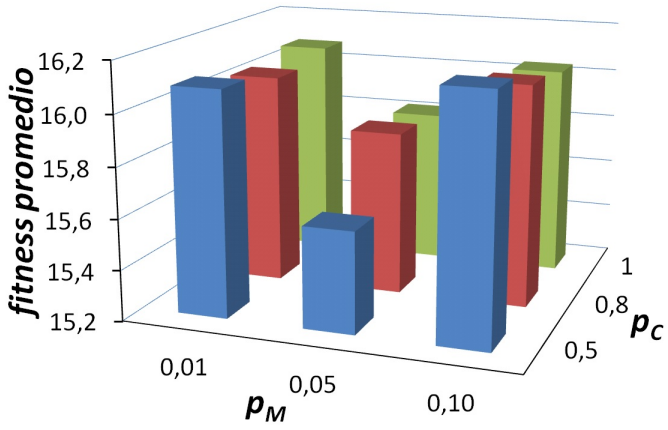


Figura 5.1: Resultados del análisis de la probabilidad de cruzamiento ($p_C$) y mutación ($p_M$)

El análisis estadístico muestra que los resultados siguen una distribución normal, siendo posible aplicar el test de Student para estudiar las distribuciones. Al comparar los resultados de las combinaciones que obtienen el mejor (0.5–0.1) y el peor (0.5–0.05) promedio de fitness, el test de Student reporta un valor $t(x) = 0{,}07$, que permite rechazar la hipótesis nula con un nivel de significancia de 0.1, indicando que existe evidencia estadística para elegir la combinación con el mejor promedio.

Entre las dos mejores combinaciones (0.5–0.1) y (0.5–0.01) no existe una diferencia significativa, siendo posible utilizar cualquiera de ambas. En la evaluación experimental se utiliza la combinación de parámetros (0.5-0.01) por su buen promedio y baja desviación estándar.

## 5.3. Descripción de escenarios

Esta sección presenta los escenarios que serán evaluados experimentalmente. El primero es el escenario base que representa la realidad actual y el segundo es un escenario alternativo que presenta modificaciones respecto al caso base, con el objetivo de mejorar las velocidades promedio de los vehículos.

### 5.3.1. Caso base o realidad actual del corredor

El caso base representa la situación actual en términos de tráfico, red vial y sincronización de semáforos del corredor Garzón. Se simulan las instancias realistas desarrolladas en el capítulo 4 para obtener las velocidades promedio de ómnibus y otros vehículos y poder comparar con los resultados numéricos luego de aplicar el AE propuesto.

Sobre el mapa base se generan tres escenarios de tráfico: 1) *tráfico medio*, que representa los datos obtenidos en el trabajo de campo (2000 vehículos y 70 ómnibus); 2) *tráfico bajo*, que se construye reduciendo en un 50 % la cantidad de vehículos y el tiempo de espera en las paradas de ómnibus, ya que existirán menos personas utilizando el transporte público (1000 vehículos y 70 ómnibus); y 3) *tráfico alto*, en el que se aumenta 50 % la cantidad de vehículos y el tiempo de espera en la parada de los ómnibus (3000 vehículos y 70 ómnibus). Las frecuencias de ómnibus se mantienen, ya que en la realidad no son alteradas cuando cambia la densidad de tráfico. El valor de 50 % para el aumento y reducción del número de vehículos se infirió al analizar datos de la zona de Garzón de años anteriores, proporcionados por la IMM.

### 5.3.2. Escenario alternativo

Para mostrar la utilidad que tienen las simulaciones de tráfico, se genera un escenario alternativo basado en el escenario base al que se le modifican los aspectos que se consideran influyen negativamente en la circulación del tráfico. Una de las ventajas principales de las simulaciones de tráfico es que no requiere gran inversión monetaria ni de tiempo y que no modifica la infraestructura del lugar, por lo que se pueden generar distintas pruebas para encontrar aquellas que logren el objetivo buscado, por ej. la mejora en la circulación del tráfico.

Analizando los puntos que se entienden podrían atentar contra el buen funcionamiento del corredor, se realizan algunas modificaciones al escenario base para intentar mejorarlo.

Entre los cambios propuestos para el escenario alternativo se encuentran: eliminación de paradas y pasajes peatonales, alternancia de paradas y modificación de reglas de los semáforos.

### 5.3.3. Detalles del escenario alternativo

Los cambios propuestos para el escenario alternativo incluyen la eliminación de paradas, semáforos y pasajes peatonales y la posibilidad de alternar paradas. Se estudiaron otras propuestas como por ejemplo construir calles paralelas a Garzón o incorporar nuevas reglas de tráfico en los cruces (por ej. permitir doblar a la derecha con roja) como existen en otros países pero fueron descartadas por la poca viabilidad real de implementarlas.

#### Eliminación de paradas

Se consideraron eliminar paradas que cumplieran con las siguientes características: que no estuvieran próximas a una calle principal y que existiera otra parada cercana para no afectar a los pasajeros realizando un traslado excesivo. Se seleccionaron solamente dos paradas para que su eliminación no tuviera un impacto negativo en el funcionamiento diario del Corredor y que cumplieran los criterios mencionados anteriormente. Las paradas elegidas están ubicadas en las intersecciones de Garzón con las calles Ariel y Casavalle.

### Eliminación de pasajes peatonales

En la actualidad, existen en el corredor Garzón tres pasajes peatonales con semáforos que detienen el tráfico. El objetivo buscado al proponer su eliminación es aumentar la velocidad promedio de los vehículos circulantes. Dos de los pasajes peatonales controlan el flujo peatonal de las esquinas en la Plaza Vidiella y Camino Besnes e Irigoyen (sin pulsador en funcionamiento). En el escenario alternativo estos dos pasajes fueron cambiados por una señalización de PARE en la calle transversal al corredor Garzón. El tercer pasaje es netamente peatonal y se encuentra ubicado frente a la Facultad de Agronomía, entre las calles Millán y Orticochea, el cual fue eliminado en el escenario alternativo. Una opción para no eliminar físicamente los pasajes peatonales y mantener los resultados obtenidos, es implementar el pasaje peatonal mediante puentes por encima del corredor Garzón.

### Alternar paradas

Uno de los problemas del ómnibus es su baja aceleración, por lo que cada vez que se detiene en un semáforo o en una parada, demora en retornar a una velocidad aceptable. Al reducir la cantidad de paradas donde un ómnibus tiene que detenerse se mejora la velocidad promedio. La línea G recorre el corredor Garzón de punta a punta y es cubierta por las empresas COETC y CUTCSA. Una posibilidad para alternar paradas consiste en dividir las paradas por empresa y compartir las ganancias generadas equitativamente u otro método para equiparar el pasaje transportado.

La Figura 5.2 brinda un ejemplo donde una empresa se detendrá en las paradas pares, la otra en las impares y algunas paradas con más cantidad de pasajeros serán realizadas por las dos. Cada empresa viajará por el corredor con una frecuencia de 4 minutos (como en la actualidad). Si se reduce el número de paradas donde se detiene un ómnibus, aumentará su velocidad promedio y no se deberá resentir en demasía el servicio, ya que la disminución de la frecuencia en una parada se contrarresta con el aumento promedio de velocidad.

### Cambio básico de semáforos

Al hacer el relevamiento de los datos obtenidos de la configuración de los semáforos del Corredor Garzón, se constató que en todas las intersecciones en donde una línea de ómnibus que circula por el corredor tiene un viraje a la izquierda, se hace detener el tránsito de la derecha de la misma, cada vez que el corredor central tiene la luz verde. Esto no parece tener mucho sentido, ya que los vehículos podrían seguir circulando por el corredor sin ningún tipo de problema. El carril que hay que detener cuando una línea dobla a la izquierda es el de la izquierda del ómnibus, pero no los dos carriles al mismo tiempo. No se tiene conocimiento si esta decisión corresponde a un error en la configuración, un tema de costos o facilidad para manejar los dos semáforos de los carriles paralelos juntos. Como esto ocurre en varias intersecciones y en ambos sentidos, el cambio mejora la velocidad promedio de los autos que circulan por los dos carriles.

Este cambio se aplicó en las siguientes intersecciones: *i*) *Islas Canarias*, dobla línea 409 hacia la izquierda, orientado a Colón (Norte); *ii*) *Camino Ariel*, doblan líneas como la 2 y la 148 hacia la izquierda, orientado a Paso Molino (Sur); y *iii*) *Camino Casavalle*, dobla línea 174 hacia la izquierda, orientado a Paso Molino (Sur).

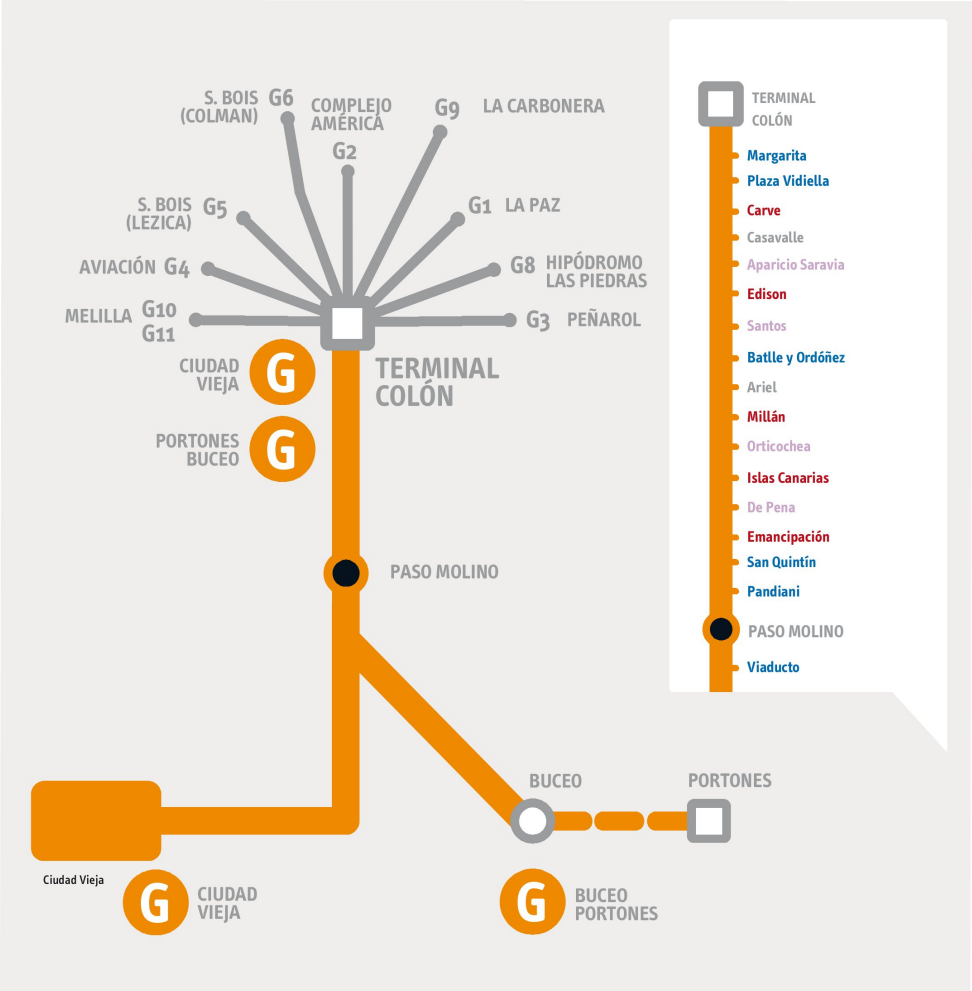


Figura 5.2: Gráfico de las paradas alternativas. Gris: Parada Eliminada. Azul: línea G de COETC y de CUTCSA. Rojo: línea G de COETC. Violeta: G de CUTCSA. - Imagen original extraída de montevideo.gub.uy

## 5.4. Resultados

Esta sección muestra los resultados obtenidos en la simulación del caso base, y los resultados numéricos del AE. El objetivo es comparar la velocidad promedio de ómnibus y vehículos con la realidad actual

### 5.4.1. Resultados de la simulación del caso base.

Los resultados de las simulaciones para los diferentes tráficos en la realidad actual, tomada como caso base para comparar los resultados del AE propuesto, se reportan en la Tabla 5.2. Los resultados representan la realidad actual del Corredor Garzón: la velocidad promedio en tráfico medio es de 14.59 km/h para los ómnibus y de 28.81 km/h para los vehículos. Estos valores se corresponden con los que surgen del análisis de datos reales (trazas de GPS de ómnibus) proporcionados por la IMM(14.5 km/h), verificando que el modelo propuesto se aproxima significativamente a la realidad.

Tabla 5.2: Resultados numéricos del caso base: velocidad promedio para ómnibus (*vpb*) y vehículos (*vpv*) para los distintos tipos de tráfico.

| *escenario* | *vpb*(km/h) | *vpv*(km/h) | *fitness* |
|---|---|---|---|
| tráfico bajo | 15.89 | 32.45 | 13.42 |
| tráfico medio | 14.59 | 28.81 | 12.05 |
| tráfico alto | 14.31 | 26.36 | 11.30 |

### 5.4.2. Resultados numéricos de la evaluación

Los resultados de aplicar la optimización de las velocidades de ómnibus y otros vehículos con el AE propuesto se reportan en la Tabla 5.3. Se presentan los valores promedio de las velocidades para ómnibus y otros vehículos, el valor de *fitness* resultante y la mejora porcentual del valor de *fitness* obtenida en comparación con el valor obtenido en el caso base.

Se realizaron 20 ejecuciones independientes del AE para cada tipo de tráfico. Al aplicar el AE se generan nuevas configuraciones de semáforos, por lo que se obtiene un valor diferente de la velocidad promedio y del valor de *fitness*, calculándose la desviación estándar ($\sigma$) para evaluar la robustez de los resultados.

Tabla 5.3: Resultados numéricos del AE: velocidad promedio para ómnibus (*vpb*) y vehículos (*vpv*) para los distintos tipos de tráfico.

| *instancia* | *vbp*(km/h) | *vpv*(km/h) | *fitness* | | mejora *fitness* (%) | |
|---|---|---|---|---|---|---|
| | | | promedio±$\sigma$ | mejor | promedio | mejor |
| *tráfico bajo* | 17.92±0.18 | 34.30±0.40 | 14.50±0.14 | 14.88 | 8.04 | 10.8 |
| *tráfico medio* | 16.95±0.32 | 33.29±0.29 | 13.95±0.15 | 14.19 | 15.7 | 17.7 |
| *tráfico alto* | 16.51±0.61 | 32.90±0.25 | 13.72±0.17 | 14.04 | 21.4 | 24.2 |

Los resultados reportados en la Tabla 5.3 indican que la optimización utilizando AE permite mejorar la velocidad promedio de los ómnibus y de otros vehículos en los tres tipos de tráfico estudiados. Además, las soluciones cuentan con mayor robustez, ya que al variar el tráfico la velocidad media de los vehículos se mantiene en un rango mucho más ajustado que en el caso original. Las mejoras logradas por el AE con respecto al caso base son de hasta **24 %** en el *fitness* y las mejoras en las velocidades son de hasta **15.3 %** en la velocidad promedio de ómnibus y **24.8 %** en la velocidad promedio de otros vehículos, al comparar con la realidad actual.

El análisis estadístico indicó que los resultados siguen una distribución normal. Por lo tanto se puede aplicar el criterio de *significancia estadística de comparación de medias* para validar los resultados: un método $A$ (en este caso, la optimización con AE) es mejor que otro método $B$ (en este caso, el escenario base) si los resultados de $A$ y B cumplen que $|f_{avg}(A) - f_{avg}(B)| > \max(\sigma(f_A), \sigma(f_B))$. El análisis numérico de los resultados indica que el criterio de significancia se cumple para todos los casos estudiados, por lo que se puede afirmar que existe evidencia estadística para afirmar que los resultados obtenidos por el AE son mejores a los valores del escenario base para todos los tipos de tráfico.

En la Figura 5.3 se comparan las duraciones de los viajes recorriendo el Corredor Garzón en toda su extensión (6.5 km) para el caso base y las duraciones luego de aplicar la optimización con el AE propuesto. Se comprueba que las duraciones de los viajes luego de aplicar el AE son menores tanto para ómnibus como para otros vehículos en los tres tipos de tráfico estudiados.

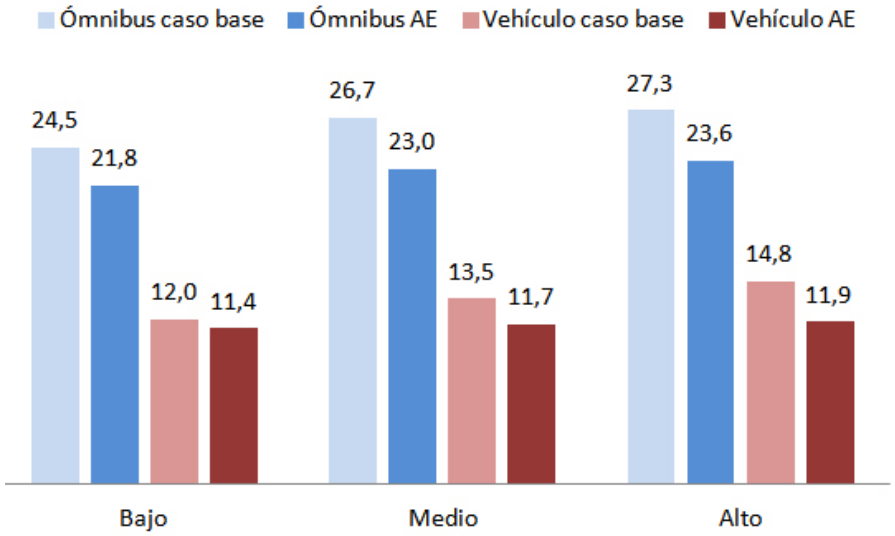


Figura 5.3: Comparación de la duración de los viajes (ómnibus y otros vehículos, en minutos) en el recorrido completo del Corredor Garzón para los diferentes tipos de tráfico.

### 5.4.3. Valores numéricos al aplicar los cambios (escenario alternativo)

Para determinar cuales son los cambios que logran mejores rendimientos se elabora la Tabla 5.4. Esta tabla se basa en el tráfico medio, ya que sólo se quiere realizar una comparación sencilla de las mejoras realizadas dado que luego se realizaran las evaluaciones aplicando todas las modificaciones para cada uno de los tipos de tráfico estudiados. Las modificaciones presentadas se aplican en el orden que aparece en la tabla y sus valores son acumulativos. Se puede apreciar que la modificación que logra una mayor diferencia es la utilización de paradas alternadas.

Tabla 5.4: Valores numéricos del escenario alternativo con su velocidad promedio ómnibus (vpb) y velocidad promedio vehículos(vpv) comparando el *fitness* para el tráfico medio

| *modificación del escenario* | $vpb(km/h)$ | $vpv(km/h)$ | *fitness* | *mejora(%)* |
|---|---|---|---|---|
| caso base | 14.59 | 28.81 | 12.05 | - |
| eliminar paradas | 15.44 | 29.03 | 12.35 | 2.4 |
| eliminar peatonales | 16.02 | 29.32 | 12.59 | 4.4 |
| paradas alternadas | 19.17 | 28.88 | 13.34 | 10.7 |
| cambiar reglas de semáforos | 18.50 | 29.70 | 13.39 | 11.1 |

Una vez que se aplican todos las modificaciones sobre el escenario, se realiza un análisis para los demás tipos de tráfico. Como se reporta en la Tabla 5.5, se obtienen mejores rendimientos en todos los tipos de tráfico estudiados y el mejor rendimiento se obtiene cuando el tráfico es alto.

Tabla 5.5: Mejoras del escenario alternativo para las velocidades promedio de los ómnibus *(vpb)* y de otros vehiculos *(vpv)* en el escenario alternativo para distintos tipos de tráficos

| *escenario* | *vpb*$(km/h)$ | *vpv*$(km/h)$ | *fitness* | *mejora fitness*(%) |
|---|---|---|---|---|
| tráfico bajo | 20.72 | 33.18 | 14.97 | 11.5 |
| tráfico medio | 18.50 | 29.70 | 13.39 | 11.1 |
| tráfico alto | 18.60 | 27.17 | 12.70 | 12.4 |

Luego de obtener los valores para el escenario alternativo se procede a aplicar el AE, cuyos resultados se presentan a continuación.

### 5.4.4. Resultados de la evaluación sobre el escenario alternativo

El escenario alternativo supuso una mejora sustancial en comparación con el caso base (11 % en el valor de *fitness*). Se procedió a aplicar el AE para determinar si aún hay posibilidad de mejorar los valores de velocidad y *fitness*.

Los resultados obtenidos se reportan en la Tabla 5.6 y permiten concluir que se mejora claramente el rendimiento del escenario alternativo y por supuesto del caso base en todos los tipos de tráfico. Comparando con la realidad actual, se logran mejoras de hasta 37 % en el valor de *fitness*.

Tabla 5.6: Mejoras obtenidas al aplicar el algoritmo evolutivo sobre el escenario alternativo, comparando las velocidades de ómnibus *(vpb)*, otros vehículos *(vpv)* y el *fitness* con cada tipo de tráfico contra el escenario base.

| *instancia* | *vpb*$(km/h)$ | *vpv*$(km/h)$ | *fitness* | | mejora *fitness* (%) | |
|---|---|---|---|---|---|---|
| | | | promedio$\pm\sigma$ | mejor | promedio | mejor |
| *tráfico bajo* | 23.15±0.36 | 34.43±0.33 | 15.99±0.08 | 16.10 | 19.1 | 19.9 |
| *tráfico medio* | 21.83±0.50 | 33.89±0.22 | 15.47±0.09 | 15.65 | 28.3 | 29.8 |
| *tráfico alto* | 21.46±0.54 | 33.41±0.38 | 15.24±0.19 | 15.50 | 34.8 | 37.1 |

En la gráfica de la Figura 5.4 se comparan las duraciones de los viajes al aplicar el AE sobre el escenario alternativo. Se produce una gran reducción en la duración de los viajes de los ómnibus en los tres tipos de tráfico, mientras para el caso de los vehículos la mayor diferencia ocurre cuando el tráfico es alto.

Al aplicar el criterio de significancia estadística para comprobar que la mejora es significativa tanto al comparar con los valores del caso base como con los del escenario alternativo se determina que efectivamente existe evidencia estadística que soporta esta hipótesis.

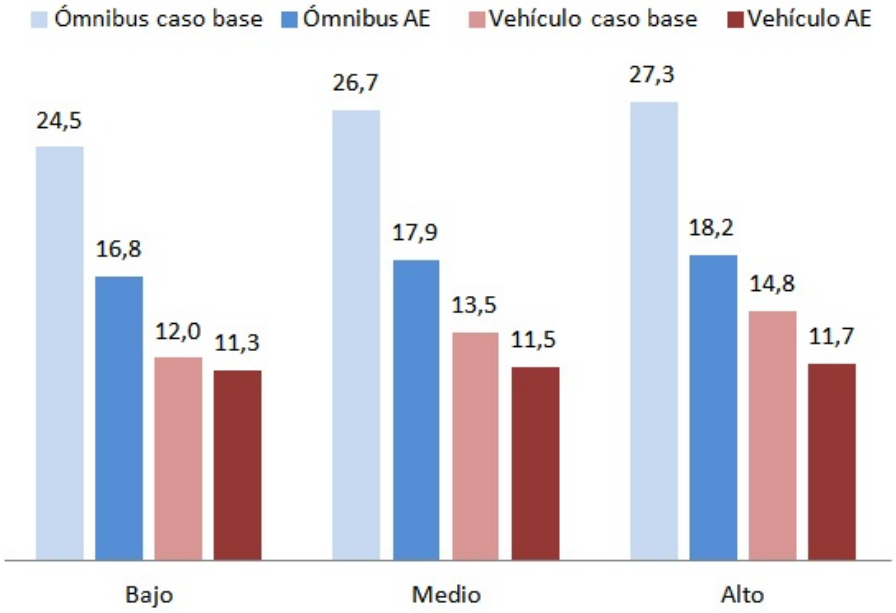


Figura 5.4: Comparación de la duración en minutos de los viajes realizados sobre el escenario base y al aplicar el algoritmo evolutivo al escenario alternativo, de ómnibus y otros vehículos en el recorrido completo del corredor Garzón, para los diferentes tipos de tráfico.

### 5.4.5. Variación de la función de *fitness*

La función de *fitness* propuesta (Ecuación 3.1) utiliza los pesos $w_1 = w_2 = 1$, lo que representa un balance equitativo para ómnibus y vehículos. Estos pesos pueden ser modificados en función de lo que se necesite de acuerdo a la realidad de la zona estudiada, por lo que se realizaron experimentos con dos configuraciones de pesos para comparar como varían las velocidades cuando se otorga más prioridad a un tipo de vehículo sobre el otro.

#### Prioridad para los ómnibus

Este caso brinda más prioridad a los ómnibus, y se justifica por el hecho que uno de los objetivos buscados por la IMM es que se utilice más el transporte colectivo como parte de su Plan de Movilidad Urbana. La premisa es que al mejorar la duración del viaje en ómnibus en relación al viaje en auto por el corredor, las personas optarán por el transporte colectivo. Se experimentó cambiando los pesos de la función de *fitness* con un peso de 70 % para los ómnibus y 30 % al resto de los vehículos.

### Prioridad a otros vehículos

Este caso asigna 70 % del peso a los vehículos y 30 % a los ómnibus. Es el caso opuesto al anterior y resulta útil para comparar sus resultados. No resulta especialmente relevante en el caso de Garzón, ya que la velocidad promedio de los ómnibus es inferior a la del resto de los vehículos en todos los casos de tráfico estudiados, pero podrían existir otros escenarios donde se requiera una velocidad promedio superior para el resto de los vehículos.

### Resultados

La Tabla 5.7 compara las velocidades promedio de ómnibus y vehículos para las tres configuraciones de pesos evaluadas y por cada tipo de tráfico. El caso 50-50 es el caso base donde los pesos son iguales, 70-30 es el caso con más prioridad para los ómnibus y el 30-70 más prioridad a los otros vehículos. Se analiza cuanto varían las velocidades de ómnibus *(var. vpb)* y otros vehículos *(var. vpv)* comparando contra el caso 50-50 de cada tipo de tráfico.

Tabla 5.7: Valores obtenidos al modificar los pesos de la función de *fitness* para ómnibus (pb) y otros vehículos (pv). Se reportan las variaciones en la velocidad promedio de ómnibus (vpb), otros vehículos (vpv) y *fitness*, donde el símbolo positivo (+) y negativo(-) indican la variación porcentual de esos valores.

| *instancia* / *tráfico* | *pb( %)* / *pv ( %)* | *vpb(km/h)* | *vpv(km/h)* | *fitness* | *var. vpb( %)* | *var. vpv( %)* | *var. fitnesss( %)* |
|---|---|---|---|---|---|---|---|
| | 50-50 | 17.92±0.18 | 34.30±0.40 | 14.50±0.14 | - | - | - |
| *bajo* | 70-30 | 17.93±0.23 | 34.06±0.17 | 12.65±0.11 | +0.07 | -0.70 | -12.79 |
| | 30-70 | 17.55±0.23 | 34.71±0.21 | 16.42±0.10 | -2.06 | +1.18 | +13.21 |
| | 50-50 | 16.95±0.32 | 33.29±0.29 | 13.95±0.15 | - | - | - |
| *medio* | 70-30 | 17.29±0.27 | 33.08±0.14 | 12.24±0.12 | +2.0 | -0.62 | -12.30 |
| | 30-70 | 16.71±0.42 | 33.79±0.31 | 15.92±0.11 | -1.41 | +1.49 | +14.11 |
| | 50-50 | 16.51±0.60 | 32.90±0.25 | 13.72±0.17 | - | - | - |
| *alto* | 70-30 | 16.72±0.14 | 32.79±0.26 | 13.75±0.07 | +1.24 | -0.33 | +0.19 |
| | 30-70 | 15.48±0.42 | 33.20±0.25 | 15.49±0.16 | -6.23 | +0.92 | +12.87 |

Los resultados indican que al variar los pesos de la función de *fitness* las velocidades promedio de los vehículos se ven afectadas. Al dar más prioridad a los ómnibus se produce, como cabía esperar, un aumento en su velocidad promedio y una leve baja en la velocidad promedio del resto de los vehículos. Cuando el tráfico es bajo este cambio casi no aumenta la velocidad de los ómnibus. Una explicación posible de este comportamiento es que ya se llegó a un limite máximo de velocidad de circulación y no se puede mejorar por sobre ese valor.

Al dar más prioridad a los otros vehículos se produce un aumento en su velocidad y una disminución en la velocidad de los ómnibus, la cual es muy evidente en el caso de tráfico alto. Este resultado permite apreciar como estos valores son fuertemente afectados por la densidad de tráfico que se estudie.

En general las variaciones en las velocidades no son grandes, pero son lo suficientemente apreciables para tener cierta flexibilidad al plantear distintos objetivos que tiendan a favorecer un tipo u otro de vehículos en diferentes situaciones de tráfico.

### 5.4.6. Eficiencia computacional

Se llevó a cabo un estudio de la eficiencia computacional del AE implementado, para analizar los tiempos de ejecución cuando se usan varios procesadores y cómo se desempeña su capacidad de paralelismo.

Para estudiar la eficiencia en diferentes contextos, se evalúan nueve ejecuciones del AE; tres con cada tipo de tráfico considerado: alto, medio y bajo, para estudiarlo en diferentes contextos. El algoritmo utiliza 32 hilos de ejecución para realizar las simulaciones de tráfico, por lo que se emplea esa cantidad de núcleos.

Las pruebas fueron realizadas sobre el equipo node40 del Cluster FING, con un procesador AMD Opteron 6272 2.09GHz, 48 GB RAM y 32 núcleos utilizados.

Para evaluar la eficiencia se calcula el *speedup*. El *speedup* ($S_N$) es una métrica que evalúa la mejora de rendimiento de una aplicación paralela al aumentar la cantidad de procesadores, comparando con el rendimiento al usar un solo procesador (Ecuación 5.1). En la definición de $S_N$, $T_1$ es el tiempo de ejecución del algoritmo serial o secuencial, y $T_N$ el tiempo del algoritmo ejecutado sobre N procesadores. El resultado ideal es obtener *speedup* lineal ($S_N$ = N), pero en la práctica utilizar N procesadores no garantiza una mejora de factor N; lo habitual es obtener un *speedup* menor al ideal o *speedup* sublineal, posibles causas para este comportamiento son: que el procesamiento no sea del todo paralelizable, demoras en las comunicaciones y sincronizaciones, tiempos ociosos, etc.

$$S_N = \frac{T_1}{T_N} \tag{5.1}$$

La eficiencia computacional ($E_N$) corresponde al valor normalizado del *speedup* (entre 0 y 1) respecto a la cantidad de procesadores. Los valores cercanos a uno indican una alta eficiencia computacional.

$$E_N = \frac{T_1}{N * T_N} = \frac{S}{N} \tag{5.2}$$

La Tabla 5.8 reporta los resultados numéricos del análisis de eficiencia computacional. Los resultados indican que el algoritmo paralelo logra una mejora sustancial en los tiempos de ejecución, con un valor promedio del *speedup* de 25.7 y eficiencia promedio de 0.8 al utilizar 32 recursos de cómputo. Estos valores corresponden a un *speedup* sublineal, pero pueden considerarse como buenos valores de las métricas de eficiencia computacional.

Tabla 5.8: Análisis de la eficiencia computacional comparando los tiempos de ejecución secuencial y paralela (en minutos). Se presentan las nueve ejecuciones realizadas, tres por cada tipo de tráfico.

| # | *tipo detráfico* | *serial (m)* | *paralelo (m)* | *speedup* | *eficiencia* |
|---|---|---|---|---|---|
| 1 | bajo | 1572 | 59 | 26.64 | 0.83 |
| 2 | bajo | 1571 | 59 | 26.62 | 0.83 |
| 3 | bajo | 1183 | 44 | **26.88** | **0.84** |
| 4 | medio | 3002 | 119 | 25.22 | 0.78 |
| 5 | medio | 2195 | 82 | 26.76 | 0.83 |
| 6 | medio | 3007 | 120 | 25.05 | 0.78 |
| 7 | alto | 2920 | 110 | 26.5 | 0.82 |
| 8 | alto | 4365 | 183 | 23.85 | 0.74 |
| 9 | alto | 4276 | 177 | 24.15 | 0.75 |
| | | | promedio | 25.7±1.1 | 0.80±0.03 |

La gráfica de la Figura 5.5 muestra como al aumentar el tráfico de datos disminuye el *speedup*. Este fenómeno sucede porque el *speedup* está influenciado por los accesos al disco duro: al tener más vehículos circulando en la simulación de tráfico requiere leer y escribir más información en los archivos, lo que aumenta el tiempo de ejecución del algoritmo. El impacto del aumento del tráfico sobre el *speedup*, como se puede apreciar en los resultados reportados en la gráfica de la Figura 5.5, no es significativo.

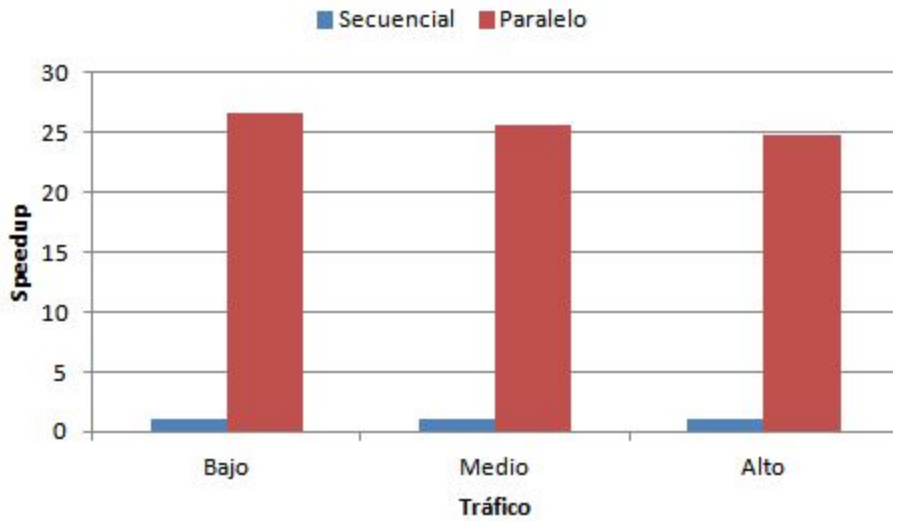


Figura 5.5: Comparación de los *speedup* promedios para cada tipo de tráfico. Se representa el caso serial que corresponde al *speedup* = 1 para fines de comparación.

# Capítulo 6

# Conclusiones y trabajo futuro

## 6.1. Conclusiones

Al analizar los objetivos planteados al inicio del proyecto, se puede afirmar que las principales metas fueron cumplidas exitosamente.

La investigación de los trabajos relacionados permitió estudiar en profundidad las técnicas de computación evolutiva utilizadas para resolver el problema de sincronización de semáforos, enfocándose en algoritmos evolutivos. En esta etapa, se relevó la información necesaria para formar una base sólida en la que se sustenta el diseño de la solución presentada. Se realizó un estudio general sobre la problemática del tráfico, que afecta tanto a la población, como al desarrollo de las ciudades. En la actualidad, se considera imprescindible la búsqueda de soluciones innovadoras y eficientes para mejorar la calidad de servicio ofrecida por las infraestructuras viales urbanas. En nuestro país (Uruguay), no existen antecedentes de propuestas relacionadas con la optimización del trafico urbano. En este contexto, este trabajo plantea un aporte interesante, orientado a estudiar el problema y demostrar que existen las herramientas y el conocimiento necesario para resolverlo.

La sincronización de semáforos es un problema difícil de abordar, en especial al trabajar sobre escenarios reales que revisten complejidad. Sin embargo, las técnicas de inteligencia computacional permiten calcular soluciones eficientes al problema. Se realizaron instancias realistas del problema que modelan la situación actual del Corredor Garzón. Estas instancias contemplan un amplio número de características del problema que proporcionan realismo y aplicación práctica a la solución aportada. El diseño del mapa se basó en información real de la zona, obteniendo el mapa base del servicio OSM. Los datos de tráfico fueron relevados in-situ, siguiendo los lineamientos recomendados en la bibliografía estudiada y las recomendaciones obtenidas de las reuniones con técnicos de la IMM. Se tuvieron en cuenta las frecuencias reales de las líneas de ómnibus que recorren el Corredor Garzón, la ubicación correcta de las paradas de ómnibus y las velocidades promedio de los vehículos que fue calculada a partir del análisis de trazas de GPS proporcionadas por la IMM.

El enfoque multiobjetivo permitió resolver el problema mediante un enfoque de optimización global, estudiando las velocidades medias de ómnibus y otros vehículos en la zona del Corredor Garzón. El análisis experimental se enfocó en realizar un análisis comparativo con el caso base que representa la realidad actual del Corredor Garzón. Los resultados obtenidos muestran la capacidad de los AE para resolver el problema: el AE propuesto logra una mejora de hasta **15.3** % en la velocidad promedio de ómnibus y **24.8** % en la velocidad promedio de otros vehículos, obteniendo evidencia estadística que valida las mejoras calculadas.

Se realizaron pruebas experimentales basadas en variar los pesos de la función de *fitness* con el objetivo de dar más prioridad a un tipo de vehículo sobre otro. El estudio del caso donde se otorga mayor prioridad a los ómnibus es interesante en el escenario del Corredor Garzón, pues uno de los objetivos del Plan de Movilidad es promover el uso del transporte público y se considera que al aumentar su velocidad relativa, aumentaría también su uso por parte de la población. Los resultados de estos experimentos mostraron como las velocidades promedio se ven afectadas, aunque las variaciones no son grandes, por ejemplo, al dar más prioridad a los ómnibus en una instancia de tráfico medio, se logró una mejora en su velocidad promedio del 2 % y una disminución del resto de los vehículos de 0.62 %, comparando con el caso donde las prioridades de ómnibus y otros vehículos son iguales.

Las simulaciones del tráfico demostraron su utilidad, al dar la flexibilidad necesaria para probar distintas variantes sobre los escenarios de forma sencilla, y poder diseñar un escenario alternativo realizando modificaciones al escenario base con el objetivo de mejorar las velocidades promedio de los vehículos. Estas modificaciones incluyen: la eliminación de paradas y pasajes peatonales, la modificación de reglas básicas de los semáforos y la alternancia de paradas. La aplicación del AE propuesto sobre el escenario alternativo logra una mejora de hasta **49.9** % en la velocidad promedio de ómnibus y **26,7** % en la velocidad promedio de otros vehículos comparando con la realidad actual.

El desarrollo de AE con capacidad de paralelización es fundamental, sobre todo en problemas complejos que requieren mucho poder de cómputo, como el abordado en este proyecto. Se implementó un modelo de paralelismo modificando el código base del AE con el objetivo de reducir el tiempo de ejecución, ya que los escenarios evaluados se consideraron complejos y que insumirían mucho tiempo de ejecución. El AE obtuvo buenas métricas de *speedup* lo que permitió reducir considerablemente el tiempo de ejecución del AE al comparar con el caso secuencial. Sin esta reducción en el tiempo de ejecución del AE hubiera sido muy difícil realizar la cantidad de pruebas presentadas.

## 6.2. Trabajo futuro

Durante el desarrollo del proyecto, se detectaron varias líneas de trabajo e investigación que sería interesante abordar como trabajo futuro.

El diseño de mapas para la simulación del tráfico requiere la realización de modificaciones para que sean reconocidos por el simulador, en algunos casos se aplicaron manualmente ya que las herramientas no brindaban la granularidad necesaria. Además, la edición de los archivos que representan la configuración de semáforos, las líneas y paradas de ómnibus supone un proceso lento y propenso a errores. Por estos motivos, para un futuro, se sugiere el desarrollo o búsqueda de nuevas herramientas que automaticen o agilicen este trabajo.

El AE desarrollado puede ser aplicado a otros lugares geográficos, modificando los datos de entrada: mapa, tráfico, configuración de los semáforos y recorridos y paradas de ómnibus. El alcance del trabajo sólo se enfocó en la zona del Corredor Garzón, pero sería interesante aplicar el AE en otros escenarios con diferentes características (en el tráfico, en las vías de tránsito, en la cantidad de semáforos, etc) con el fin de determinar su capacidad para mejorar las velocidades de los vehículos bajo esas circunstancias.

Un aspecto interesante para desarrollar en el futuro, es el uso de un enfoque multiobjetivo explícito para compararlo con el de agregación lineal utilizado en este proyecto. Para implementar este AE multiobjetivo, se podría considerar el uso de otro *framework*, por ejemplo jMetal que incluye varios tipos de algoritmos multiobjetivos.

Los trabajos de Montana y Czerwinski (1996) y Vogel et al. (2000) proponen la adaptabilidad del algoritmo en tiempo real, aunque esto requiere del agregado de sensores a la red, podría resultar en una mejora importante sobre todo, en zonas de gran densidad de tráfico. Se instalarían sensores en los semáforos que detecten la cantidad de automóviles en las intersecciones en tiempo real y un sistema que procesara los datos obtenidos en varias intersecciones para optimizar la circulación del tráfico, modificando la configuración de las luces de los semáforos. El reto de este método se encuentra en que al ser ejecutado en tiempo real, la configuración de semáforos tiene que ser obtenida relativamente rápida, por lo que el algoritmo utilizado tendría que lograr soluciones aceptables en un corto periodo de tiempo.

# Bibliografía


E. Alba y C. Cotta. On-line tutorial on evolutionary computation. En `http://neo.lcc.uma.es/TutorialEA/semEC/main.html`, 1997. Disponible online, consultado Abril 2015.

E. Alba y M. Tomassini. Parallelism and evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 6(5):443–462, 2002.

E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Diaz, I. Dorta, J. Gabarró, J. González, C. León, L. Moreno, J. Petit, J. Roda, A. Rojas, y F. Xhafa. MALLBA: A library of skeletons for combinatorial optimisation. *Parallel Computing*, 32(5-6):415–440, 2006.

E. Bañobre y Á. Romero. Los BRT en corredores segregados como sistema óptimo de transporte urbano. En *Administrando en entornos inciertos = managing in uncertain environment*, 2009.

BBVA Research. Situación Automotriz Uruguay. En `https://www.bbvaresearch.com`, 2013. Disponible online, consultado noviembre 2014.

A. Bull. *Congestión de tránsito: el problema y cómo enfrentarlo*. United Nations Publications, 2003.

S. Cahon, N. Melab, y E.-G. Talbi. Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics*, 10(3):357–380, 2004.

C. Coello, D. Van Veldhuizen, y G. Lamont. *Evolutionary algorithms for solving multi-objective problems*. Kluwer, New York, 2002.

K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

A. Di Febbraro, D. Giglio, y N. Sacco. On applying Petri nets to determine optimal offsets for coordinated traffic light timings. En *Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on*, p. 773–778. IEEE, 2002.

J. J. Durillo y A. J. Nebro. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

El País. Garzón: Olivera admitió errores. En `http://www.elpais.com.uy/`, 30 Julio 2013. Disponible online, consultado febrero 2015.

El País. Ana Olivera: "No hay justificación, nos equivocamos con Garzón". En `http://www.elpais.com.uy/`, 10 Enero 2015. Disponible online, consultado febrero 2015.

L. Eshelman. The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. En *Foundations of Genetics Algorithms*, p. 265–283. Morgan Kaufmann Publishers, Inc., 1991.

G. Fagúndez. The Malva Project: A framework for computational intelligence in C++. En `http://themalvaproject.github.io/`, 2014. Disponible online, consultado julio 2014.

C. Gagné y M. Parizeau. Open beagle: A new versatile C++ framework for evolutionary computation. En *GECCO Late Breaking Papers*, p. 161–168, 2002.

D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

M. Haklay y P. Weber. OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

A. Halati, H. Lieu, y S. Walker. CORSIM-corridor traffic simulation model. En *Traffic congestion and traffic safety in the 21st century: Challenges, innovations, and opportunities*, p. 570–576, 1997.

IMM. Plan de movilidad urbana. En `http://www.montevideo.gub.uy`, Febrero 2010. Disponible online, consultado febrero 2015.

INE. *Uruguay en cifras 2014*, capítulo Transporte y Comunicaciones (11), p. 93–99. 2014. URL `http://www.ine.gub.uy/`. Disponible online, consultado enero 2015.

Institute for Transportation & Development Policy (ITDP). Bus rapid transit planning guide. En `https://www.itdp.org`, Junio 2007. Disponible online, consultado febrero 2015.

R. H. J. Penner y C. Jacob. Swarm-based traffic simulation with evolutionary traffic light adaptation. En *Proceedings of Artificial Intelligence and Soft Computing*, p. 192–197, 2002.

G. Kotushevski y K. A. Hawick. A review of traffic simulation software. Technical Report CSTN-095, Computer Science, Massey University, Albany, North Shore 102-904, Auckland, New Zealand, 2009.

J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems).* A Bradford Book, 1992.

D. Krajzewicz, G. Hertkorn, C. Rössel, y P. Wagner. SUMO (simulation of urban mobility). En *Proc. of the 4th middle east symposium on simulation and modelling*, p. 183–187, 2002.

G. Lim, J. Kang, y Y. Hong. The optimization of traffic signal light using artificial intelligence. En *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volumen 3, p. 1279–1282. IEEE, 2001.

T. Litman. Transportation cost and benefit analysis. En `http://vtpi.org/tca/`, 2009. Disponible online, consultado febrero 2015.

S. López, P. Hernández, A. Hernández, y M. García. Artificial neural networks as useful tools for the optimization of the relative offset between two consecutive sets of traffic lights. En J. Mira y J. V. Sánchez-Andrés, editors, *IWANN (2)*, volumen 1607 of *Lecture Notes in Computer Science*, p. 795–804. Springer, 1999.

D. McKenney y T. White. Distributed and adaptive traffic signal control within a realistic traffic simulation. *Engineering Applications of Artificial Intelligence*, 26(1):574–583, 2013.

M. Mitchell. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. A Bradford Book, 1998.

D. J. Montana. Strongly typed genetic programming. *Evolutionary Computation*, 3(2): 199–230, 1995.

D. J. Montana y S. Czerwinski. Evolving control laws for a network of traffic signals. En *Proceedings of the First Annual Conference on Genetic Programming*, p. 333–338. MIT Press, 1996.

S. Nesmachnow. Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República. *Revista de la Asociación de Ingenieros del Uruguay*, 61: 12–15, 2010.

L. Papaleondiou y M. Dikaiakos. Trafficmodeler: A graphical tool for programming microscopic traffic simulators through high-level abstractions. En *IEEE 69$^{th}$ Vehicular Technology Conference*, p. 1–5. IEEE, 2009.

A. K. Rathi. Urban network traffic simulation:traf-netsim program. *Journal of Transportation Engineering*, 116(6):734–743, November-December 1990.

I. Rechenberg. *Evolution strategy: optimization of technical systems according to principles of biological evolution*. Frommann-Holzboog, 1973.

N. Rouphail, B. Park, y J. Sacks. Direct signal timing optimization: Strategy development and results. En *XI Pan American Conference in Traffic and Transportation Engineering*, p. 195–206, 2000.

J. Sánchez. *Estudio de la optimización del tráfico rodado mediante el ajuste de los ciclos de semáforos por algoritmos genéticos en dispositivos de computación paralela usando modelización discreta: ejemplos de aplicación*. PhD thesis, Universidad de Las Palmas de Gran Canaria, 2007.

J. Sánchez, M. Galán, y E. Rubio. Genetic algorithms and cellular automata: a new architecture for traffic light cycles optimization. En *Congress on Evolutionary Computation, 2004. CEC2004*, volumen 2, p. 1668–1674. IEEE, 2004.

J. Sánchez, M. Galán, y E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: "Las Ramblas" area in Santa Cruz de Tenerife. *IEEE Transactions on Evolutionary Computation*, 12(1):25–40, 2008.

J. J. Sánchez, M. J. Galán-Moreno, y E. Rubio-Royo. Traffic signal optimization in "La Almozara" district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):132–141, 2010.

D. Smith y J. McIntyre. Handbook of simplified practice for traffic studies. Technical Report 455, Iowa State University, 2002.

W. M. Spears. *Evolutionary algorithms: the role of mutation and recombination.* Springer Science & Business Media, 2000.

D. H. Stolfi. Optimización del tráfico rodado en ciudades inteligentes. Master's thesis, Universidad de Málaga, 2012.

Subrayado. «Expertos en tránsito: "el colapso está establecido" en Montevideo». En `http://www.subrayado.com.uy`, 27 Mayo 2013. Disponible online, consultado noviembre 2014.

SUMO. Sumo tools. En `http://sumo.dlr.de/wiki/Tools/Main`, Noviembre 2014. Disponible online, consultado febrero 2015.

K. T. Teo, W. Y. Kow, y Y. Chin. Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. En *2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSiM)*, p. 172–177. IEEE, 2010.

The Institute for Transportation and Development Policy (ITDP). The BRT Standard, 2014 edition. En `https://www.itdp.org/`, 2014. Disponible online, consultado noviembre 2014.

A. Vogel, C. Goerick, y W. Von Seelen. Evolutionary algorithms for optimizing traffic signal operation. En *Proceedings of the European symposium on intelligent techniques (ESIT)*, p. 83–91, 2000.

M. Wall. GAlib: A C++ library of genetic algorithm components. En `http://lancet.mit.edu/ga/`, 1996. Disponible online, consultado noviembre 2014.

C. Wallace, K. Courage, M. Hadi, y A. Gan. TRANSYT-7F user's guide. *University of Florida, Gainesville*, 1998.

C. B. Yang y Y. J. Yeh. The model and properties of the traffic light problem. En *International Conference on Algorithms. Kaohsiung, Taiwan.*, p. 19–26, 1996.

Enviado al 11 Congreso Internacional de Cómputo en Optimización y Software, Cuernavaca, México, 2015

# Planificación de tráfico y transporte colectivo en el Corredor Garzón utilizando un algoritmo evolutivo

Álvaro Acuña, Efraín Arreche, Sergio Nesmachnow

Universidad de la República, Montevideo, Uruguay
{alvaro.acuna,efrain.arreche,sergion}@fing.edu.uy

**Abstract.** Este artículo propone el estudio del problema de planificación de transporte colectivo mediante la sincronización de semáforos, y el diseño e implementación de un algoritmo evolutivo para resolverlo con alta eficacia numérica y desempeño computacional. Se plantea como caso de estudio la sincronización de semáforos en el Corredor Garzón (Montevideo, Uruguay), un escenario urbano muy interesante por el número de cruces, calles, semáforos, y por la problemática del tráfico en la zona. El problema de optimización relacionado es NP-difícil, por lo que los métodos computacionales exactos solo son útiles en instancias de tamaño reducido. Se propone utilizar un algoritmo evolutivo para calcular una configuración eficiente de los semáforos, maximizando la velocidad media del transporte colectivo y de otros vehículos. El análisis experimental compara los resultados numéricos del algoritmo evolutivo con el escenario base que modela la realidad actual. Los resultados demuestran que el algoritmo evolutivo logra mejoras significativas en la calidad de servicio al comparar con la realidad actual, mejorando hasta 15.3% la velocidad promedio de ómnibus y 24.8% la velocidad promedio de otros vehículos.

**Keywords:** planificación, transporte colectivo, corredores urbanos, algoritmos evolutivos

## 1 Introducción

El parque automotor está creciendo de forma sostenida desde hace varios años, provocando serios problemas de congestión del tráfico, que afectan el desarrollo de las ciudades y la calidad de vida de las personas [1]. Las congestiones de tráfico reducen la velocidad media de circulación de los vehiculos y esto hace que disminuya el nivel de aceptación del transporte público, cuyo servicio en general es ineficiente. Para resolver este problema las autoridades suelen optar por instrumentar sistemas de transporte costosos como los *metros*, pero también se ha demostrado que existen otras opciones viables como los sistemas de Ómnibus de Tránsito Rápido (en inglés, *Bus Rapid Transit*, BRT) [2].

La ciudad de Montevideo no escapa al fenómeno global de congestión del tráfico. Aunque la situación no sea tan crítica como en otras ciudades del mundo, las autoridades han tomado medidas para reducir el impacto de este problema, implementando un Plan de Movilidad Urbana, para mejorar la eficiencia del transporte público y democratizar el acceso al mismo [3]. La propuesta se inspira en un BRT, construyendo varios corredores exclusivos en la ciudad. El primer corredor exclusivo implementado fue el *Corredor Garzón*, ubicado en la zona norte de Montevideo, que incluye 24 cruces semaforizados y vías exclusivas para ómnibus. Desde su inauguración en el año 2012, el Corredor Garzón ha recibido críticas por no cumplir con uno de sus principales objetivos: el de agilizar el transporte público.

Los métodos para la sincronización de semáforos se encuentran entre los más efectivos para agilizar el tránsito y evitar congestiones. Estas técnicas permiten aumentar la velocidad promedio de los viajes y mejorar las perspectivas de desarrollo de la ciudad y la calidad de vida de sus habitantes. El problema de sincronización de semáforos es complejo, y las técnicas de inteligencia computacional se aplican para hallar soluciones de buena calidad eficientemente [4–7].

Este trabajo presenta un Algoritmo Evolutivo (AE) aplicado al problema de gestión del tráfico en la zona del Corredor Garzón mediante la planificación de los semáforos, con el objetivo de aportar una solución eficiente e innovadora para el desarrollo de la ciudad y mejorar la calidad de vida de los ciudadanos. Las particularidades del Corredor Garzón lo convierten en un reto complejo desde el punto de vista de la investigación, como consecuencia de la extensión del tramo, de la cantidad de cruces, del número de semáforos, de las reglas de exclusividad y los diferentes tipos de tráfico, entre otros. Los resultados experimentales demuestran que el AE propuesto es capaz de mejorar la velocidad promedio de los ómnibus y otros vehículos al considerar el escenario real abordado.

Las principales contribuciones de este trabajo incluyen: el estudio del problema de gestión del tráfico y sincronización de semáforos, con el objetivo de agilizar el transporte público; y el diseño e implementación de un AE multiobjetivo, capaz de resolver eficientemente el problema de sincronización de semáforos en la zona del Corredor Garzón.

El articulo se organiza del modo que se describe a continuación. La Sección 2 presenta la descripción y la formulación del problema de planificación de transporte colectivo en corredores urbanos. El AE propuesto para resolver el problema se describe en la Sección 3. La Sección 4 presenta el análisis experimental de la metodología de resolución propuesta, sobre casos de estudio realistas del Corredor Garzón. Finalmente, las principales conclusiones de la investigación y las líneas de trabajo actual y futuro se comentan en la Sección 5.

## 2 El problema de planificación de transporte colectivo en corredores urbanos

Esta sección presenta el problema de planificación de transporte colectivo abordado en este trabajo.

### 2.1 Modelo del problema

El problema se modela simplificando la realidad, para considerar el uso de aquellos elementos útiles para resolver la sincronización de semáforos. El modelo se basa en diseñar un mapa de la zona geográfica a estudiar y crear instancias realistas, incluyendo datos recabados in-situ, que serán usadas para simular el tráfico. A continuación se presenta el simulador de tráfico, el proceso de diseño del mapa y el método de recolección de datos.

*Simulador de tráfico.* SUMO [8] es un simulador de tráfico gratuito y de código abierto que permite modelar redes de calles, vehículos, transporte público y semáforos. SUMO sigue un modelo microscópico, realizando la simulación individual explícita de cada elemento en el escenario. El simulador toma como entrada archivos de configuración que representan la realidad y genera archivos de salida con información útil del escenario simulado: la cantidad de vehículos, la velocidad de los vehículos, la duración de los viajes, las emisiones de gases contaminantes, etc.

*Diseño del mapa de la zona de estudio.* En un primer paso se diseñó un mapa de la zona del Corredor Garzón compatible con el simulador SUMO. Se utilizó el servicio de Open Street Map (OSM) [9], que cuenta con mapas libres y actualizados, para importar la zona de interés y Java Open Street Map Editor para incluir los elementos del problema. Para validar la exactitud del mapa importado desde OSM se utilizaron otros servicios (Google Maps y Bing Maps) y un estudio in-situ de la zona, recorriendo el Corredor Garzón y analizando en cada intersección las reglas de cruce y las direcciones de circulación. Como resultado de la verificación in-situ fueron detectadas algunas inconsistencias entre la realidad y el mapa importado, que fueron corregidas en la versión del mapa a utilizar en la investigación.

La zona de estudio comprende al Corredor Garzón y dos caminos paralelos a cada lado (véase el mapa en la Fig. 1). Como no existen calles paralelas reales, el mapa se diseñó seleccionando calles que construyen dos caminos paralelos a cada lado del corredor y sus calles internas, verificando que cada camino paralelo incluya rutas en ambos sentido.

**Fig. 1:** Zona de estudio: a la izquierda el mapa extraído de OSM, a la derecha el mapa procesado para ser usado en SUMO.

*Recolección de datos de semáforos y vehículos en trabajo de campo.* Existen datos públicos sobre la cantidad de ómnibus y sus frecuencias en el sitio web de la Intendencia de Montevideo (IMM), pero no se tienen datos sobre el tráfico vehicular en la zona del Corredor Garzón. Por este motivo se realizó un relevamiento in-situ en cinco cruces representativos de la Avenida Garzón, con características útiles para modelar variantes del tráfico.

El relevamiento de tráfico aplicó las recomendaciones sobre el conteo vehicular [10]: se realizó un miércoles, con clima soleado y en el horario de 15 a 17 horas, para evitar sesgos de fin de semana, horas pico, o días de lluvia. Se filmó cada cruce (entre 15 y 30 minutos) y luego se contabilizaron sólo los automóviles, ya que un relevamiento preliminar indicó que la cantidad de otros vehículos (camiones, motos, bicicletas, etc.) no es significativa en el total del tráfico (menos del 10% de los vehículos). La Tabla 1 muestra la cantidad de automóviles en cada cruce. Para la configuración de los semáforos se recorrió el Corredor Garzón, se cronometró la duración de las luces en cada esquina en el camino de ida, y se validaron en el camino de vuelta. Los tiempos se verificaron con los vídeos del conteo del tráfico.

Tabla 1: Resumen del revelamiento del tráfico en la zona del Corredor Garzón.

| *intersección* | *dirección de circulación* | | | |
|---|---|---|---|---|
| | *Sur-Garzón* | *Norte-Garzón* | *Oeste* | *Este* |
| Camino Colman | 410 | 390 | 140 | 230 |
| Plaza Vidiella | 400 | 444 | 292 | 0 |
| Aparicio Saravia | 390 | 450 | 40 | 90 |
| Batlle y Ordoñez | 510 | 390 | 470 | 300 |
| Camino Ariel | 436 | 226 | 177 | 203 |

### 2.2 Formulación del problema

Los objetivos del problema, la velocidad promedio de los ómnibus (*vpb*) y la velocidad promedio del resto de los vehículos (*vpv*), fueron seleccionados por ser las principales métricas de calidad de servicio para los usuarios del Corredor Garzón. Se plantea optimizar los dos objetivos simultáneamente, utilizando un enfoque de optimización multiobjetivo. Para implementarlo, se aplica la técnica de combinación lineal de los objetivos, que proporciona un método sencillo y adecuado para resolver problemas de optimización que involucran menos de tres objetivos y el frente de Pareto del problema es convexo. La funcion a optimizar es $f = w_1.vpb + w_2.vpv$, donde $w_1$ y $w_2$ son pesos que pueden ser utilizados para priorizar a un tipo de vehículo sobre otro. En el estudio presentado en este trabajo se busca optimizar con igual importancia ambas velocidades, por lo cual se utilizan los valores $w_1 = w_2 = 1$.

La optimización de la velocidad promedio de los vehículos se realiza globamente en toda la red vial de la zona de estudio. La optimización global es fundamental para lograr una mejora sostenible en la velocidad de los vehículos, ya que optimizar cada cruce individualmente, podria generar problemas de congestión en otras intersecciones del mapa.

## 3 Planificación de tráfico en el Corredor Garzón utilizando algoritmos evolutivos

Esta sección describe el AE propuesto para resolver el problema de planificación de tráfico en el Corredor Garzón.

### 3.1 Metaheurísticas y algoritmos evolutivos

Las metaheurísticas son estrategias genéricas para definir algoritmos que permiten computar soluciones aproximadas de buena calidad para problemas complejos [11]. Son métodos útiles y eficientes para resolver complejos problemas de optimización sobre escenarios reales, que requieren gran esfuerzo computacional y manejan grandes volúmenes de datos. Los AE son métodos estocásticos de inteligencia computacional que emulan el proceso de evolución natural para resolver problemas de optimización, búsqueda y aprendizaje [12]. Los AE han sido exitosamente aplicados para resolver problemas de optimización subyacentes a problemas complejos del mundo real en múltiples áreas de aplicación.

Un AE es una técnica iterativa que en cada iteración (*generación*) aplica operadores estocásticos sobre un conjunto de individuos (*población*). Inicialmente, la población se genera aplicando un procedimiento aleatorio o heurístico. Cada individuo en la población codifica una solución al problema estudiado, que tiene un valor de *fitness*, evaluado por una función que determina su adecuación para resolver el problema. El AE busca mejorar el fitness de los individuos en la población, mediante *operadores evolutivos*, como la *recombinación* de partes de dos individuos y la *mutación* aleatoria de su codificación, que son aplicados a individuos seleccionados según su fitness. El criterio de parada del AE usualmente involucra un número determinado de generaciones o la detección de una situación de convergencia. El AE retorna la mejor solución hallada en el proceso iterativo, tomando en cuenta la función de fitness considerada.

En este trabajo se utiliza un algoritmo genético (AG) tradicional [13] provisto por la biblioteca de optimización Malva, modificado para incluir las características del problema a resolver y e implementar el modelo de paralelismo maestro-esclavo para evaluar las posibes soluciones del problema, generando un hilo de ejecución por cada simulación.

### 3.2 Implementación

A continuación se describen las principales características del AE implementado para resolver el problema.

*Representación de soluciones.* Las soluciones contemplan dos elementos clave de la planificación de semáforos: la *duración de fase*, que indica el tiempo en que una configuración de luces está activa en una intersección y el *offset*, que indica en qué fase comienza el ciclo. La Fig. 2 presenta un ejemplo de la representación utilizada para las soluciones del problema: la información se agrupa lógicamente en cruces, el valor de cada *gen* es la duración en segundos de una fase en un cruce (se omiten las fases de luces amarillas, que no modifican los tiempos reales de paso de vehículos). El largo de una representación depende de la cantidad de cruces y de la cantidad de fases que incluye. La representación propuesta se basa en una optimización global de todo el sistema y no individualmente para cada cruce.

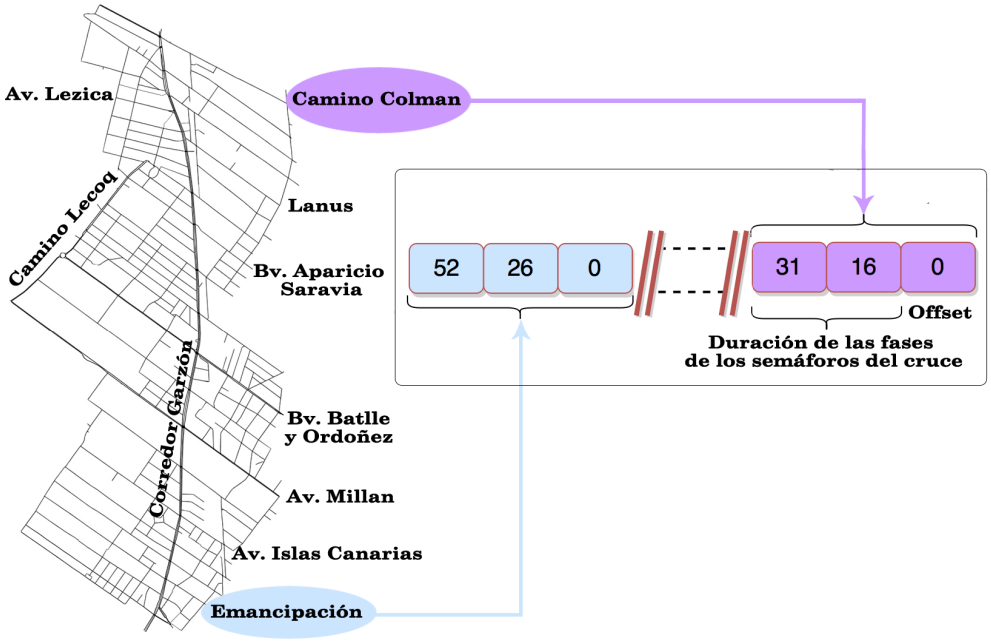


**Fig. 2:** Ejemplo de representación de las soluciones para el problema.

*Función de fitness.* La función de *fitness* corresponde a la formulación multiobjetivo presentada en la sección 2, que intenta optimizar la velocidad promedio de ómnibus y otros vehículos, teniendo cada componente un peso específico en la combinación lineal utilizada para definir la función de *fitness*.

*Inicialización de la población.* Para inicializar la población se toma como referencia la configuración obtenida a partir de los datos de la realidad, y para cada cruce se varían las duraciones de las fases de manera aleatoria entre un rango de valores configurable. Además, la fase inicial se selecciona aleatoriamente entre la cantidad de fases del cruce.

*Operador de Cruzamiento.* Se utiliza cruzamiento de un punto. Se selecciona aleatoriamente un determinado cruce como punto de corte, y se producen dos hijos seleccionando las soluciones de un individuo desde el primer cruce hasta el punto de corte y se une con las soluciones del otro individuo desde el punto de corte en adelante; el otro hijo se forma de manera inversa. De esta manera se busca mantener la sincronización de los cruces contiguos.

*Operador de Mutación.* Se implementaron dos mutaciones: *mutación de fase*, que modifica la duración de fases de un cruce sumando o restando una cantidad dada de segundos y la *mutación de offset*, que suma o resta una cantidad de segundos a un *offset* del individuo. Ambas mutaciones son aplicadas de acuerdo a una tasa de probabilidad prefijada.

*Selección y reemplazo.* Se utilizan los operadores y criterios provistos por el AG de Malva: selección de padres mediante torneo de tres individuos y selección proporcional para los hijos. Respecto a la política de reemplazo, se utiliza un modelo "+", en el cual los hijos compiten con los padres por la supervivencia en la siguiente generacion.

*Modelo de paralelismo.* El AE implementado utiliza el modelo de paralelismo maestro-esclavo: en cada generación el maestro crea un hilo de cómputo por cada evaluación de la función de *fitness*, a ejecutar por un proceso esclavo.

## 4 Análisis Experimental

Esta sección presenta la evaluación experimental del AE propuesto para resolver el problema de sincronización de semáforos en el Corredor Garzón.

### 4.1 Instancias del problema

Una vez diseñado el mapa de la zona de estudio y relevados los datos de la realidad, se crearon instancias realistas del problema. Para configurar la simulación del tráfico en SUMO se utilizan archivos en formato XML: *i*) la *configuración de los semáforos*, que detalla su ubicación, la duración de las luces y el offset (obtenidos de los datos relevados in-situ); *ii*) la *ruta de los vehículos*, con el recorrido de cada vehículo, creado con el software *Traffic Modeler* [14], que permite generar modelos de tráfico complejos (se optó por un modelo de movilidad entre áreas, que permite una buena granularidad de densidad de tráfico); y *iii*) los *datos de los ómnibus* (recorrido, frecuencia, ubicación de las paradas, tiempos de demora, etc.): las frecuencias y los recorridos fueron obtenidos de datos públicos de la IMM para las líneas urbanas *G*, *409*, *2*, *D5* y *148*.

Complementariamente, se realizó un estudio de datos de GPS (posición exacta, velocidad instantánea y referencia de cada ómnibus) para las líneas seleccionadas en un período de una semana, que permitió determinar la velocidad media para los ómnibus en el Corredor Garzón (14.5 km/h). Las detenciones en cada parada (entre 15 y 35 segundos) se calcularon utilizando los vídeos del trabajo de campo para el relevamiento del tráfico

Sobre el mapa base se generan tres escenarios de tráfico: 1) *tráfico medio*, con los datos obtenidos en el trabajo de campo (2000 vehículos y 70 ómnibus); 2) *tráfico bajo*, reduciendo un 50% los vehículos y el tiempo de espera en las paradas de ómnibus, ya que existirán menos personas utilizando el transporte público (1000 vehículos y 70 ómnibus); y 3) *tráfico alto*, aumentando 50% los vehículos y el tiempo de espera en la paradas (3000 vehículos y 70 ómnibus). Las frecuencias de ómnibus se mantienen, ya que en la realidad no se alteran cuando cambia la densidad de tráfico. El valor de 50% para aumentar/reducir el número de vehículos surge de analizar datos de la zona, de acuerdo a la IMM.

### 4.2 Resultados numéricos

La evaluación experimental se lleva a cabo sobre los escenarios descritos previamente, ejecutando en la plataforma de cómputo de alto desempeño Cluster FING, de la Universidad de la República [15] (procesadores AMD Opteron 6272 de 2.09GHz, con 64 núcleos, 48GB RAM, y sistema operativo CentOS Linux 6.5). El objetivo es comparar la velocidad promedio de ómnibus y vehículos calculadas por el AE propuesto con las de la realidad actual.

*Resultados del caso base.* La Tabla 2 reporta los resultados de las simulaciones para la realidad actual, tomada como caso base para comparar los resultados del AE propuesto. Los resultados representan la realidad actual del Corredor Garzón: la velocidad promedio en tráfico medio es de 14.59 km/h para los ómnibus y de 28.81 km/h para los vehiculos. Estos valores se corresponden con los que surgen del análisis de datos reales (trazas de GPS de ómnibus) proporcionados por la IMM (14.5 km/h), verificando que el modelo propuesto se aproxima significativamente a la realidad.

Tabla 2: Resultados del caso base: velocidad promedio de ómnibus (*vpb*) y vehículos (*vpv*) para cada tipo de tráfico.

| *escenario* | *vpb*(km/h) | *vpv*(km/h) | *fitness* |
|---|---|---|---|
| tráfico bajo | 15.89 | 32.45 | 13.42 |
| tráfico medio | 14.59 | 28.81 | 12.05 |
| tráfico alto | 14.31 | 26.36 | 11.30 |

*Resultados numéricos del AE.* Los resultados de aplicar la optimización de las velocidades de ómnibus y otros vehículos con el AE propuesto se reportan en la Tabla 3. Se presentan los valores promedio de las velocidades para ómnibus y otros vehículos, el valor de *fitness* resultante y la mejora porcentual del valor de *fitness* obtenida en comparación con el valor obtenido en el caso base. Se realizaron 20 ejecuciones independientes del AE para cada tipo de tráfico. Al aplicar el AE se generan nuevas configuraciones de semáforos, por lo que se obtiene un valor diferente de la velocidad promedio y del valor de *fitness*, calculándose la desviación estándar ($\sigma$) para evaluar la robustez de los resultados.

Los resultados reportados en la Tabla 3 indican que la optimización utilizando AE permite mejorar la velocidad promedio de los ómnibus y de otros vehículos en los tres tipos de tráfico estudiados. Además, las soluciones cuentan con mayor robustez, ya que al variar el tráfico la velocidad media de los vehículos se mantiene en un rango mucho más ajustado que en el caso original. Las mejoras logradas por el AE con respecto al caso base son de hasta **24%** en el *fitness* y las mejoras en las velocidades son de hasta **15.3%** en la velocidad promedio de ómnibus y **24.8%** en la velocidad promedio de otros vehículos, al comparar con la realidad actual.

Tabla 3: Resultados del AE: velocidad promedio de ómnibus (*vpb*) y vehículos (*vpv*) para cada tipo de tráfico.

| *instancia* | *vbp*(km/h) | *vpv*(km/h) | *fitness* | | mejora *fitness* (%) | |
|---|---|---|---|---|---|---|
| | | | promedio±$\sigma$ | mejor | promedio | mejor |
| *tráfico bajo* | 17.92±0.18 | 34.30±0.40 | 14.50±0.14 | 14.88 | 8.04 | 10.8 |
| *tráfico medio* | 16.95±0.32 | 33.29±0.29 | 13.95±0.15 | 14.19 | 15.70 | 17.7 |
| *tráfico alto* | 16.51±0.61 | 32.90±0.25 | 13.72±0.17 | 14.04 | 21.40 | 24.2 |

La Figura 3 compara las duraciones de los viajes recorriendo el Corredor Garzón en toda su extensión (6.5 km) para el caso base y luego de aplicar la optimización con el AE propuesto. Se comprueba que las duraciones de los viajes luego de aplicar el AE son menores para ómnibus y para otros vehículos en los tres tipos de tráfico estudiados.

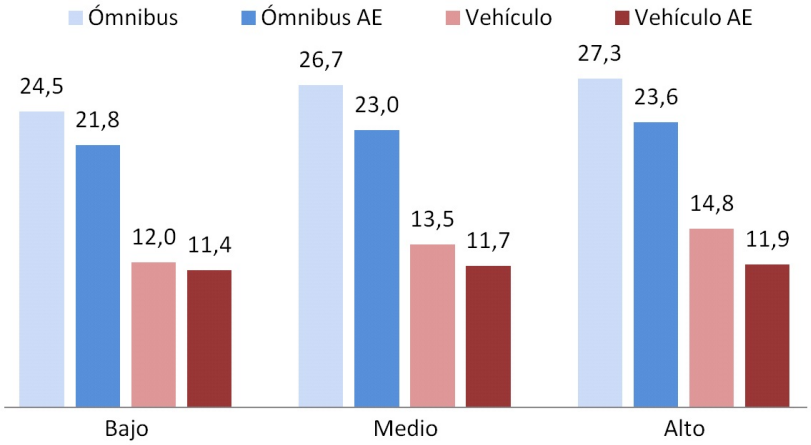


**Fig. 3:** Comparación de la duración de viajes (ómnibus y vehículos, en minutos) en el recorrido completo del Corredor Garzón.

## 5 Conclusiones y trabajo futuro

En la actualidad, se considera imprescindible la búsqueda de soluciones innovadoras y eficientes para mejorar la calidad de servicio ofrecida por las infraestructuras viales urbanas. Este trabajo plantea una solución al problema de sincronización de semáforos en el Corredor Garzón utilizando un AE. La propuesta contempla múltiples características del problema (informacion real de mapas, datos de tráfico relevados in-situ y análisis de trazas de GPS), que proporcionan realismo y aplicación práctica a la solución. El AE permitió resolver el problema mediante un enfoque de optimización global, estudiando las velocidades medias de ómnibus y otros vehículos en la zona del Corredor Garzón.

El análisis experimental compara el AE con la realidad actual del Corredor Garzón. Los resultados obtenidos muestran la capacidad de los AE para resolver el problema: se logran mejoras en la velocidad promedio de hasta **15.3**% para ómnibus y **24.8**% para otros vehículos, obteniendo evidencia estadística que valida las mejoras calculadas.

Las principales líneas de trabajo actual y futuro se orientan a generar escenarios alternativos en la zona del Corredor Garzón, teniendo en cuenta los elementos a mejorar en la realidad actual y posibles modificaciones en la función de evaluación del AE para dar prioridad a distintos tipos de vehículos. Asimismo, el AE desarrollado puede ser aplicado a otros lugares geográficos, modificando los datos de entrada del problema.

## Referencias


1. Bull, A.: Congestión de tránsito: el problema y cómo enfrentarlo. No. 87, United Nations Publications (2003)
2. Bañobre, E., Romero, Á.: Los BRT en corredores segregados como sistema óptimo de transporte urbano. In: Administrando en entornos inciertos (2009)
3. IMM: Plan de movilidad urbana (2010), online, http://www.montevideo.gub.uy, consultado febrero 2015
4. Sánchez, J.: Estudio de la optimización del tráfico rodado mediante el ajuste de los ciclos de semáforos por algoritmos genéticos en dispositivos de computación paralela usando modelización discreta: ejemplos de aplicación. Ph.D. thesis, Universidad de Las Palmas de Gran Canaria (2007)
5. Sánchez, J.J., Galán-Moreno, M.J., Rubio-Royo, E.: Traffic signal optimization in "La Almozara" district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. IEEE Trans. on Intelligent Transportation Systems 11(1), 132–141 (2010)
6. Rouphail, N., Park, B., Sacks, J.: Direct signal timing optimization: Strategy development and results. In: XI Pan American Conference in Traffic and Transportation Engineering. pp. 195–206 (2000)