



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Diseño y Aplicación de Técnicas Metaheurísticas para el Control de Tráfico

Trabajo Fin de Máster

**Máster en Inteligencia Artificial, Reconocimiento de
Formas e Imagen Digital**

Autor: Marcelo Fernando Valdiviezo Condolo

Tutor: Antonio Garrido Tejero

2016-2017

Agradecimientos

Aprovecho este espacio para agradecer a aquellas personas que de una u otra forma han formado parte en el desarrollo del presente trabajo, el mismo que no podría haberse realizado sin los conocimientos adquiridos en el presente Máster; aportándome todas sus asignaturas las nociones necesarias para llevar a cabo este estudio.

La realización del presente Trabajo Final de Máster es fruto de las orientaciones y estímulo del profesor Antonio Garrido Tejero, quien me ha conducido durante estos meses con una actitud abierta y generosa, guiándome y mostrando en cada momento una inmejorable disposición ante las dudas surgidas durante la realización del mismo.

Deseo agradecer profundamente a la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación de la República del Ecuador, por el auspiciante apoyo que me brindó como parte del programa de becas Convocatoria Abierta 2015 – Primera Fase.

Y, por supuesto a mis padres Rosa y Camilo, mis hermanos Argenys, Rebeca y Luis, mi novia Cristina y mis amigos que supieron apoyarme y motivarme en todo momento durante el proceso de estudios y el desarrollo del presente trabajo, convirtiéndose en los pilares fundamentales de cada uno de los logros de mi vida.

Resumen

El presente trabajo se centra en la aplicación de técnicas metaheurísticas, específicamente un algoritmo genético, para la optimización de la duración de los ciclos de los semáforos, para topologías de entornos reales de las ciudades. Se presenta una función de fitness basada en los tiempos de espera de los vehículos, que conjuntamente con la ayuda de la evaluación a través de un simulador de microtráfico, minimiza el consumo de combustible, la duración de los viajes, los tiempos de espera y la emisión de CO₂.

Palabras clave: tráfico, optimización, algoritmo genético, inteligencia artificial, semáforos.

Abstract

The present work focuses on the application of metaheuristic techniques, specifically a genetic algorithm, for the optimization of the duration of the cycles of the traffic lights, for topologies of real environments of the cities. There is a fitness function based on the waiting times of the vehicles, which together with the help of the evaluation through a microtraffic simulator, minimizes fuel consumption, travel time, waiting times and emission of CO₂.

Keywords: traffic, optimization, genetic algorithm, artificial intelligence, traffic lights.

Tabla de contenidos

1.	Introducción	9
1.1.	Antecedentes.....	9
1.2.	Motivación.	12
1.3.	Descripción del Problema.....	13
1.4.	Objetivos y Alcances.	14
1.5.	Organización de la Memoria.	15
2.	Marco Teórico.....	17
2.1.	Elementos Técnicos.	17
2.1.1.	Semáforo.	17
2.1.2.	Ciclo de un Semáforo.	18
2.1.3.	Fase de un Semáforo.....	18
2.1.4.	Estado de un Semáforo.	19
2.1.5.	Simulador de Tráfico SUMO (Simulator Urban MObility).....	20
2.2.	Técnicas de Inteligencia Artificial.	25
2.2.1.	Enjambre de Partículas.....	26
2.2.2.	Colonias de Hormigas.....	26
2.2.3.	Búsqueda Tabú.	27
2.2.4.	Alineamiento Simulado.	27
2.3.	Algoritmos Genéticos.....	28
2.3.1.	Términos de un Algoritmo Genético.	29
2.3.2.	Operadores Genéticos.....	30
2.3.3.	Diagrama de Flujo del Algoritmo.	33
3.	Propuesta de Solución.....	35
3.1.	Procedimiento.....	35
3.2.	Codificación.	36
3.3.	Obtención de los Datos.....	38
3.4.	Población Inicial.	42
3.5.	Función de Evaluación.	42
3.6.	Función de Evaluación en SUMO.	44
3.7.	Operadores Genéticos Propuestos.	45



3.7.1.	Selección.....	46
3.7.2.	Cruce.	46
3.7.3.	Mutación.	47
3.7.4.	Algoritmo Genético.	47
4.	Experimentación y Pruebas	49
4.1.	Escenario.....	49
4.2.	Parámetros del Algoritmo Genético.	50
4.2.1.	Tamaño de la población inicial.	50
4.2.2.	Número de Generaciones.	51
4.2.3.	Probabilidad de Mutación.	52
4.3.	Evaluación de Soluciones.	53
4.4	Análisis de Resultados.	55
4.4.1.	Fitness	56
4.4.2.	Duración del Viaje	56
4.4.3.	Tiempo de Espera.	57
4.4.4.	Emisión de CO ₂	58
4.4.5.	Consumo de Combustible.....	58
4.4.6.	Tiempo de Cómputo.	59
5.	Conclusiones y Trabajos Futuros.	61
5.1.	Conclusiones.....	61
5.2.	Trabajos Futuros.....	61
	Bibliografía	63

Índice de Figuras

Figura 1: Congestión en Ciudad de México (México, 2017).	9
Figura 2: Sistema de Comunicación de Datos en el Control de Transporte Inteligente (Mazek, y otros, 2016).	13
Figura 3: Ciclo y Fases para un conjunto semafórico	18
Figura 4: Fases para una intersección.	19
Figura 5: Intersección con cruce por prioridad.	20
Figura 6: Entorno del Simulador Gráfico SUMO.	21
Figura 7: Parámetros estáticos para vehículo por defecto.....	23
Figura 8: Parámetros dinámicos de un vehículo en SUMO	24
Figura 9: Parámetros de la vía en SUMO	25
Figura 10: Representación de términos usados en un Algoritmo Genético (Castrovation, 2017).	29
Figura 11: Operador de Cruce en un Punto.....	32
Figura 12: Operador de Cruce en n Puntos.....	32
Figura 13: Cruce Uniforme.....	33
Figura 14: Diagrama de Flujo Algoritmo Genético.....	34
Figura 15: Diagrama del Procedimiento para la Solución del Problema	35
Figura 16: Intersección en el Simulador SUMO, duración de las fases y codificación.....	36
Figura 17: Representación de la codificación de los tiempos y los estados de los semáforos.	37
Figura 18: Captura del mapa en la página web OpenStreetMap	38
Figura 19: Mapa importado al simulador SUMO	39
Figura 20: Estructura del fichero mapa.rou.xml.....	39
Figura 21: Captura del fichero de configuración mapa.sumo.cfg.....	40
Figura 22: Extracto del fichero mapa.net.xml con la configuración de semáforos de la topología	41
Figura 23: Generación de la población inicial.	42
Figura 24: Ruta para un vehículo y cálculo de waitsteps.	44
Figura 25: Gráfica de Selección por método de ruleta	46
Figura 26: Operación de Cruce	47
Figura 27: Diagrama de Flujo del Algoritmo Genético.....	48
Figura 28: Mapa de las calles utilizadas en los experimentos.....	49
Figura 29: Algoritmo Genético Variando el Tamaño de la Población.....	51
Figura 30: Algoritmo Genético Variando el número de Generaciones.....	52
Figura 31: Algoritmo Genético Variando la Probabilidad de Mutación.....	53
Figura 32: Evolución del Algoritmo Genético con los parámetros seleccionados.	54

Figura 33: Comparación de fitness de las soluciones generadas por diferentes técnicas. 56
Figura 34: Comparación Duración del Viaje 57
Figura 35: Comparación tiempo de espera. 57
Figura 36: Comparación de Emisión de CO2. 58
Figura 37: Comparación Consumo de Combustible. 59
Figura 38: Tiempo de cómputo. 60

Índice de Tablas

Tabla 1: Descripción de los estados de los semáforos. 19
Tabla 2: Parámetros de Simulación 50
Tabla 3: Parámetros Variando Tamaño de la Población Inicial..... 51
Tabla 4: Parámetros Variando el Número de Generaciones 51
Tabla 5: Parámetros Variando la Probabilidad de Mutación 52
Tabla 6: Parámetros seleccionados para el Algoritmo Genético. 53
Tabla 7: Evaluación soluciones Algoritmo Genético 54
Tabla 8: Evaluación soluciones aleatorias. 54
Tabla 9: Evaluación solución SCPG 55
Tabla 10: Evaluación de Algoritmo Genético con función de Evaluación SUMO
..... 55

1. Introducción

1.1. Antecedentes.

En este documento se presenta el Trabajo Final de Master desarrollado para el control de tráfico vehicular, mediante la aplicación de técnicas metaheurísticas.

Durante las últimas décadas uno de los principales problemas en las diferentes ciudades del mundo, es el congestionamiento vehicular y los efectos que esto ocasiona.

Grandes urbes como Ciudad de México, Bangkok, Jakarta, Detroit, Londres, Bruselas, Los Ángeles, París, entre otras (TomTom, 2017), luchan diariamente con el grave problema de la congestión vehicular y sus efectos como la contaminación ambiental, desgaste de los vehículos por constante aceleración y frenado, retrasos y prolongadas horas de espera. Los mencionados efectos acarrearán una disminución en la calidad de vida debido a problemas de estrés al que constantemente se someten conductores y peatones al circular por las diferentes ciudades. La reducción de la productividad es otra consecuencia, debida a los tiempos perdidos en el camino a los lugares de trabajo. En claro ejemplo es Ciudad de México que presenta el mayor índice de congestión con un 66% de tiempo de viaje adicional que los conductores deben permanecer atrapados en el tráfico a cualquier hora del día, llegando a perderse hasta 227 horas por año, pudiendo apreciar un ejemplo en la figura 1.



Figura 1: Congestión en Ciudad de México (México, 2017).

La contaminación ambiental provocada por los diferentes tipos de vehículos motorizados se debe principalmente a la emisión del monóxido de carbono, uno de los gases contaminantes que más abunda en la atmósfera de las ciudades, llegando incluso a convertirse en uno de los principales problemas de salud pública. Este gas ocasiona graves daños sobre nuestra salud y en algunos casos su acción sobre nuestro organismo puede llegar a ser mortal. Por otra parte, los gases de escape de los vehículos también contienen óxidos de nitrógeno que son asimismo contaminantes para el medioambiente (DGT, 2017).

La búsqueda de soluciones a este problema ha cobrado notable interés, sobre todo por la relación directa existente con los altos niveles de contaminación presentados en los últimos años, por el incremento en la población mundial, el crecimiento de las ciudades y la multiplicación de automotores que circulan por las vías. Estamos ante un problema de índole global, que afecta diariamente a millones de personas. Sin embargo, la optimización del tráfico es una tarea bastante compleja por la gran cantidad de parámetros involucrados en su análisis. Una de las alternativas de solución a este problema es la optimización de la duración de los ciclos de los semáforos, ya que se consideran una opción de bajo costo, al no requerir infraestructura adicional.

Con el crecimiento de las ciudades, el número de los semáforos se incrementa, y la programación por parte de los ingenieros de tráfico se vuelve una tarea más compleja y caótica. Esta tarea cobra mayor dificultad debido al gran número de combinaciones posibles, las variables que deben ser consideradas, y las restricciones a tomar en cuenta para garantizar la seguridad de conductores, peatones y vehículos.

A diferencia de técnicas como semáforos inteligentes que requieren de nueva infraestructura, ubicación de sensores, y modificaciones sobre la obra civil ya estructurada, el uso de técnicas de inteligencia artificial en la optimización de los ciclos de los semáforos se presenta como una herramienta viable por ser rápidas, eficientes y de bajo costo (Alba, 2014).

El uso de simuladores de tráfico ha sido otro de los factores que ha permitido realizar el estudio y la investigación de las diferentes técnicas. Resulta conveniente realizar las pruebas bajo entornos simulados que permiten obtener parámetros de medida y evaluación del tráfico, con valores muy aproximados a los reales, y garantizando la seguridad durante las etapas de pruebas.

En 1998 De Schutter propuso un modelo de control y optimización de semáforos para una intersección, desarrollando modelos que describen la

longitud de la cola formada por los vehículos en cada carril y generando esquemas óptimos y sub-óptimos de conmutación de los semáforos con la posibilidad de ciclos de longitudes variables (De Schutter & De Moor, 1998).

Wiering describió el uso de sistemas multi-agente en el aprendizaje reforzado para controladores de semáforos inteligentes, buscando minimizar el tiempo de espera total de los vehículos en las ciudades, buscando de esta forma reducir la congestión vehicular (Wiering, 2000).

Lim, G., Kang, J., & Hong, Y. en el año 2001 proponen un algoritmo genético tomando como entrada la longitud de la cola de espera ante un semáforo y generando como solución el tiempo en verde optimizado para la intersección, además de que se utiliza el flujo de tráfico entrante en la intersección, durante la fase del color rojo (Lim, Kang, & Hong, 2001).

La propuesta de Tubaishat, Shang y Shi en 2003, se trata de un sistema de control basado en tres capas; una red de sensores inalámbricos para detectar la presencia de vehículos en las intersecciones y extraer datos relevantes como la velocidad y la cantidad de vehículos, una política del modelo del flujo del tráfico y una capa de coordinación de alto nivel (Tubaishat, Shang, & Shi, 2007).

Inicialmente las investigaciones se centraban en sencillas redes que no correspondían directamente a las complejas topologías de calles, intersecciones y semáforos que existen en las ciudades. Sin embargo, la aplicación de simuladores de microtráfico permitió desarrollar nuevos modelos como el propuesto por García-Nieto y Enrique en el 2013, donde plantean la optimización de ciclos de los semáforos mediante un optimizador de enjambre de partículas (PSO), utilizando el simulador SUMO para su evaluación. Esta investigación permitió encontrar soluciones óptimas para diferentes áreas metropolitanas centrándose en dos objetivos principales: la cantidad de vehículos que llegan y la duración total del viaje de los vehículos (García-Nieto, Olivera, & Alba, 2013).

Para el año 2014, el trabajo propuesto por D Rahbari hace uso de los algoritmos genéticos para la optimización de las señales de tráfico en las intersecciones, tomando en una primera etapa los datos estadísticos recogidos en las intersecciones y generando una población inicial, demostrando que mediante el uso de esta técnica es posible reducir el tiempo de espera ante los semáforos en color rojo (Rahbari, 2014).

El trabajo de Sattari, M. centra su investigación en la optimización basada en colonias de hormigas y haciendo uso de semáforos inteligentes, tomando en cuenta la condición de carga de las calles. Se propone dividir la topología de las calles en celdas, en donde cada vehículo guiará a través de la ruta con menor tráfico (Sattari, Malakooti, Jalooli, & Noor, 2014).



Como podemos revisar en la bibliografía citada la mayor parte de los estudios se centran en el análisis de una o pocas intersecciones cuyas características no necesariamente representan la estructura de la topología de las calles como las vemos en la vida real.

Los estudios recientes ya se centran en Topologías que abarcan mayor área y representan topologías de ciudades reales. Sin embargo, no se toman en cuenta ciertas características en la duración de los ciclos de los semáforos, como la duración del color amarillo que es recomendable tomarlo como una constante para que éste no afecte a la psicología del conductor que conoce el tiempo necesario para su acción de frenado (DGT, 2017).

1.2. Motivación.

Este trabajo se ha enfocado en la topología de las calles de ciertos sectores de la ciudad de Valencia, aunque se puede utilizar para optimizar los ciclos de los semáforos en cualquier ciudad del mundo. Es importante considerar que el problema del tráfico no es exclusivo de las grandes urbes, sino que incluso ciudades pequeñas con baja densidad poblacional presentan graves problemas de congestión.

Para este fin, y de acuerdo a lo comentado anteriormente, las técnicas de inteligencia artificial y específicamente las técnicas metaheurísticas son herramientas que nos permitirán obtener soluciones factibles sobre un problema de optimización, como es la asignación de tiempos a la duración de los ciclos de los semáforos. Estas técnicas permiten reducir el tiempo empleado por especialistas a las tareas de planificación y se adapta con facilidad a las diferentes topologías de las calles de las ciudades en el mundo.

Teniendo en cuenta que más allá de la congestión como problema central, tenemos el grave problema de la contaminación por emisión de gases y las consecuencias en el deterioro de la salud. La contaminación nos afecta tanto a conductores como peatones, lo cual es razón suficiente para centrar nuestra atención en la búsqueda de soluciones factibles para reducirla, y específicamente aquella producida por el tráfico vehicular.

Haciendo uso de simuladores de microtráfico que nos permiten extraer características de una topología y de la misma forma evaluar diferentes parámetros como son la contaminación por diferentes gases, tiempos de espera, duración de rutas, consumo de combustible, entre otros, podemos llegar a obtener soluciones factibles y en tiempos relativamente cortos.

1.3. Descripción del Problema.

Uno de los principales problemas que genera el tráfico vehicular son los tiempos perdidos frente a los semáforos, las constantes acciones de frenado y arranque que desgastan los componentes de los vehículos e incrementan los volúmenes de consumo de combustible con la consecuente contaminación.

Antiguamente los semáforos se programaban para que cada color tenga un tiempo constante de duración, y este ciclo se repetía a lo largo de todo el día, sin importar las condiciones variantes del tráfico. Estos programas estaban basados en reglas y estudios de la conducta de conducción y parámetros físicos necesarios para que se ajusten a los tiempos de reacción de los conductores ante maniobras de frenado y arranque.

Los semáforos modernos responden a programas conectados a complejos sistemas de sensores que permiten determinar el flujo vehicular en diferentes sentidos, y establecer los colores y su duración de acuerdo a la demanda del tráfico. En la figura 2 podemos observar una intersección con una serie de sensores que se comunican mediante redes inalámbricas para gestionan la comunicación instantánea con un servidor central que contiene programas de gestión del tráfico. Cabe resaltar que a pesar de que los semáforos modernos se han implementado en diferentes ciudades, aún existen muchos de ellos que no responden a la demanda de tráfico en tiempo real y requieren de programas optimizados, que se encuentran a cargo de los ingenieros de tráfico y especialistas en los temas de vialidad.

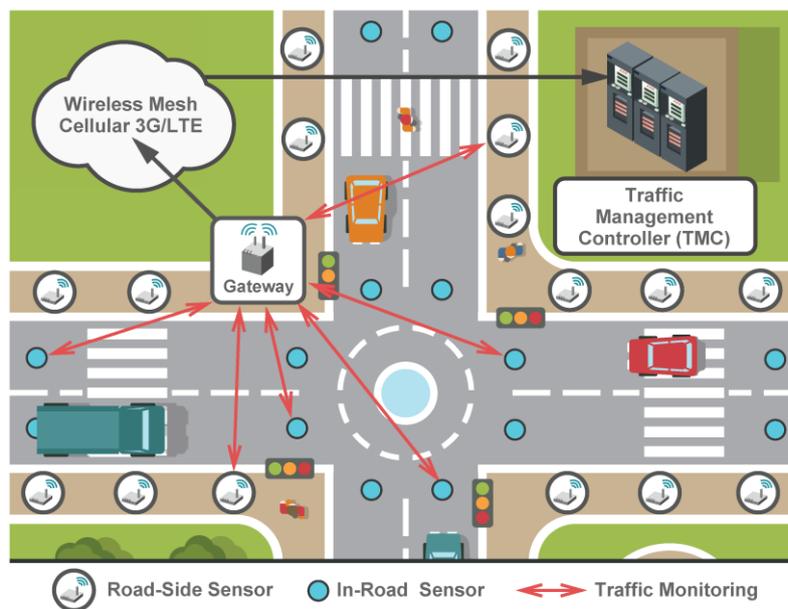


Figura 2: Sistema de Comunicación de Datos en el Control de Transporte Inteligente (Mazek, y otros, 2016).

Ante lo expuesto, el problema central abordado en el presente trabajo es la optimización de los ciclos de los semáforos, para topologías de ciudades reales, orientados a reducir los tiempos perdidos, la duración de las rutas de cada vehículo, consumo de combustible y niveles de contaminación mediante el uso de técnicas de inteligencia artificial y un simulador de microtráfico.

Hemos tomado en cuenta que la mayor parte de las investigaciones se centran en intersecciones sencillas, o en el uso de complejos sistemas dotados de sensores, para lo cual sería necesario reemplazar o modificar infraestructura ya existente. Además, las propuestas más recientes, generan soluciones en donde no se toma en cuenta las recomendaciones sobre duración de los ciclos que se ajusten apropiadamente a los tiempos de reacción de conductores.

1.4. Objetivos y Alcances.

Con la motivación descrita, el trabajo pretende aplicar las técnicas de inteligencia artificial, específicamente un algoritmo genético en la resolución del problema de la optimización de la duración de los ciclos de los semáforos.

Como datos de entrada disponemos de la topología de las calles del sector analizado que contiene información de los semáforos y sus fases, la cantidad de vehículos a simular, las rutas generadas para los vehículos y el tiempo de simulación.

Los datos de salida es un vector de números enteros positivos, que corresponden a la duración de las fases de los ciclos de los semáforos que conforman la topología ingresada, dicha solución puede ser evaluada para obtener datos de análisis como tiempo de espera, duración de los viajes, consumo de combustible, emisión de CO₂.

Para lo descrito anteriormente nos hemos planteado los siguientes objetivos:

- Desarrollar una herramienta que permita obtener soluciones factibles en la optimización de la duración de los ciclos de los semáforos.
- Emplear técnicas de inteligencia artificial que faciliten su uso en los diferentes escenarios y que se adapten a las topologías de las calles de las ciudades.

- Diseñar una función de fitness que permita evaluar las soluciones generadas con base en los parámetros propuestos por el simulador.
- Analizar los resultados obtenidos y comparar con base en los parámetros utilizados para su evaluación.

Los beneficios obtenidos se centran en la herramienta que constituya una alternativa eficaz a la optimización de ciclos de los semáforos, minimizando parámetros de contaminación producida por CO₂, consumo de combustible, duración y tiempo perdido en la ruta. Podemos resaltar el desarrollo de una opción válida al uso del simulador para la evaluación de las soluciones obtenidas por el algoritmo genético, que reduzca el tiempo de cómputo.

1.5. Organización de la Memoria.

Esta memoria del Trabajo de Fin de Máster se ha estructurado en tres partes principales. En la primera parte abordamos una introducción acerca del tema y la revisión del estado del arte para conocer el desarrollo de diferentes técnicas aplicadas a la optimización de los semáforos para el control del tráfico vehicular, analizando los problemas con los que cuentan las técnicas actuales. La segunda parte está formada por el planteamiento de una solución basada en los algoritmos genéticos y el desarrollo de pruebas experimentales para evaluar la validez de nuestro planteamiento. Finalmente, la tercera parte se centra en las conclusiones obtenidas y las líneas de trabajo que surgen como resultado del presente estudio.

La primera parte abarca los siguientes capítulos:

CAPITULO 1. En este capítulo se realiza una descripción general del problema desde el estado del arte de la optimización de los semáforos para el control del tráfico vehicular, la motivación para el desarrollo del presente trabajo, los objetivos planteados y los beneficios obtenidos.

CAPITULO 2. Se centra en una revisión sobre términos técnicos utilizados en la teoría del tráfico vehicular, además aborda conceptos sobre el marco teórico de las técnicas metaheurísticas, los algoritmos evolutivos y específicamente los algoritmos genéticos que son el tema central de estudio del presente trabajo.



Los capítulos que corresponden a la segunda parte son los siguientes:

CAPITULO 3. En este capítulo se abarca el diseño algorítmico y experimental, el planteamiento de la codificación utilizada y la selección de los diferentes parámetros del algoritmo genético, así como de la función de evaluación.

CAPÍTULO 4. En el presente capítulo se evalúa el algoritmo previamente implementado y se lleva a cabo el análisis de los diferentes resultados obtenidos en las pruebas elaboradas.

La tercera parte está compuesta por el siguiente capítulo.

CAPÍTULO 5. Consta de las conclusiones que se desprenden del trabajo realizado, principalmente del análisis y evaluación de las pruebas, también se realiza propuestas para el desarrollo de nuevos trabajos derivados del presente estudio.

2. Marco Teórico

Para la optimización de los ciclos de los semáforos mediante la aplicación de técnicas metaheurísticas, es necesario previamente aclarar algunas definiciones básicas de los diferentes términos empleados en el desarrollo de esta memoria.

2.1. Elementos Técnicos.

Partimos de la definición de uno de los elementos esenciales en la gestión del tráfico como es el semáforo, los diferentes tipos y los elementos necesarios para la definición de los parámetros utilizados en nuestro trabajo, también se presenta la descripción y características de uno de los principales simuladores de microtráfico.

2.1.1. Semáforo.

Se trata de un dispositivo utilizado para el control del tráfico, cuya función es permitir el paso alternativo de los vehículos desde las distintas vías que confluyen en una determinada intersección en intervalos de tiempo establecidos. El semáforo funciona a través de la gestión de lámparas de colores generalmente Verde, Ámbar y Rojo, visibles a los conductores, las mismas que representan señales de tráfico establecidas. Los semáforos generalmente no poseen un programa de gestión individual, sino que para un conjunto semafórico que se encuentra ubicado en una intersección, existe un regulador central que garantiza la sincronización de todos los semáforos para evitar accidentes en cada una de las intersecciones.

Existen diferentes tipos de semáforos que responden a las diferentes necesidades del tráfico entre los que podemos listar:

- Semáforos de Tiempos Fijos. En este caso la duración de las fases y los ciclos permanecen invariables y responden a un tiempo previamente establecido de acuerdo a condiciones generales del tráfico en la intersección.
- Semáforos totalmente accionados por el tránsito. La duración de los ciclos y las fases responden a la demanda del tráfico mediante sensores ubicados en las calles y que determinan el volumen del tráfico en el tramo comprendido entre intersecciones.

- Semáforos semi-accionados por el tránsito. En este caso se dispone de medios para ser accionados en accesos secundarios, ya sea por conductores o peatones. Generalmente se ubican en accesos de poco tráfico dando prioridad a los accesos principales.
- Semáforos controlados por computador. Este tipo de semáforos más modernos permite gestionar de forma remota la duración de los ciclos y las fases, además de permitir generar rutas rápidas para vehículos de emergencia (Valencia V. G., 2000).

2.1.2. Ciclo de un Semáforo.

Se denomina ciclo de un semáforo al periodo de tiempo transcurrido desde el cambio de un grupo semafórico hasta que dicha configuración se vuelva a repetir, es decir al conjunto de secuencias completas de paso permitido de los vehículos que se pueden dar al mismo tiempo en los semáforos conectados a un mismo programa regulador. También podemos entender a un ciclo como el conjunto de fases sucesivas de un semáforo hasta la repetición de las mismas como podemos apreciar en la Figura 3.

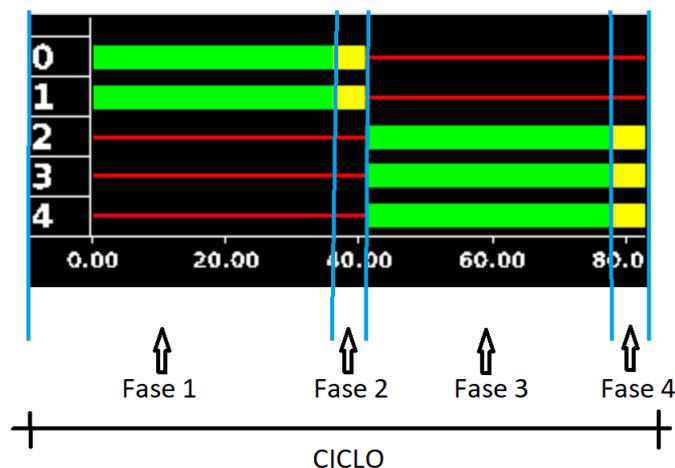


Figura 3: Ciclo y Fases para un conjunto semafórico

2.1.3. Fase de un Semáforo.

Se conoce como fase a cada una de las divisiones del ciclo, durante la cual la duración de los colores de todos los grupos semafóricos permanece invariable. Una fase está a su vez formada por diferentes estados que corresponden a cada uno de los semáforos que integran el grupo semafórico.

La fase es la que nos indica la cantidad de movimientos que están permitidos al mismo tiempo como podemos apreciar en la figura 4, en donde:

En la Fase 1 se permite los movimientos del punto C hacia los puntos B y D de forma simultánea.

En la Fase 2 se permite los movimientos del punto C hacia los puntos B y D, si y solo si los vehículos se encuentran demasiado cerca de la intersección y una acción de frenado brusca podría poner en peligro su seguridad y las de los demás conductores.

En la Fase 3 está permitido el movimiento desde los carriles del punto A hasta los respectivos carriles del punto B, además desde el carril izquierdo del punto A hacia el punto D.

En la Fase 4, los movimientos permitidos en la fase anterior se restringen únicamente a la condición dada de si se encuentran demasiado cerca de la intersección.

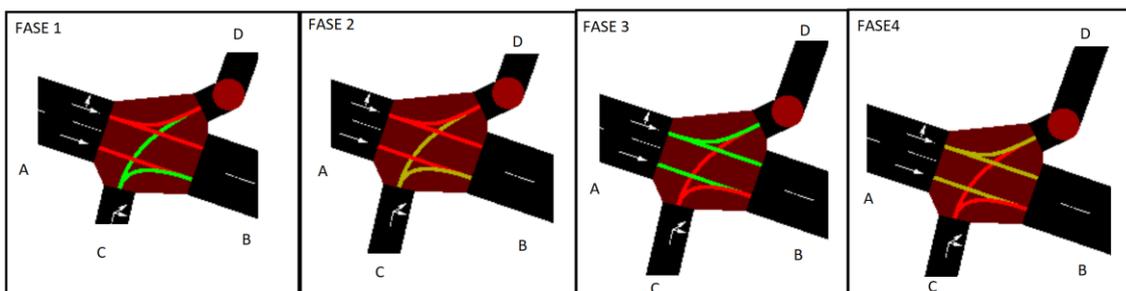


Figura 4: Fases para una intersección.

2.1.4. Estado de un Semáforo.

Un estado se representa como cada uno de los colores que indican la acción a realizar por parte de los conductores, existen tres colores utilizados normalmente con su respectivo significado. Sin embargo, pueden existir algunas variaciones que permitan organizar la prioridad del tráfico en determinado sentido de las vías, como se presenta en la tabla 1.

Tabla 1: Descripción de los estados de los semáforos.

Color	Carácter	Descripción
Rojo	r	Luz roja que indica que los vehículos están obligados a detenerse.
Amarillo	y	Luz ámbar o amarilla que indica que los vehículos deben empezar a desacelerar si se encuentran demasiado lejos de la intersección, en caso de ser inevitable la parada por la

		proximidad al cruce, puede pasar.
Verde	g	Luz verde indica que los vehículos que se encuentran en un carril no prioritario pueden cruzar con la precaución de que no existan vehículos en el sentido de prioridad. Por ejemplo, como se aprecia en la figura 5, si un vehículo quiere realizar un giro en U de C hacia B, debe reducir su velocidad y esperar que no haya un vehículo en cruce de A hacia B ya que este tiene mayor prioridad.
Verde	G	Luz verde indica que los vehículos pueden cruzar la intersección.

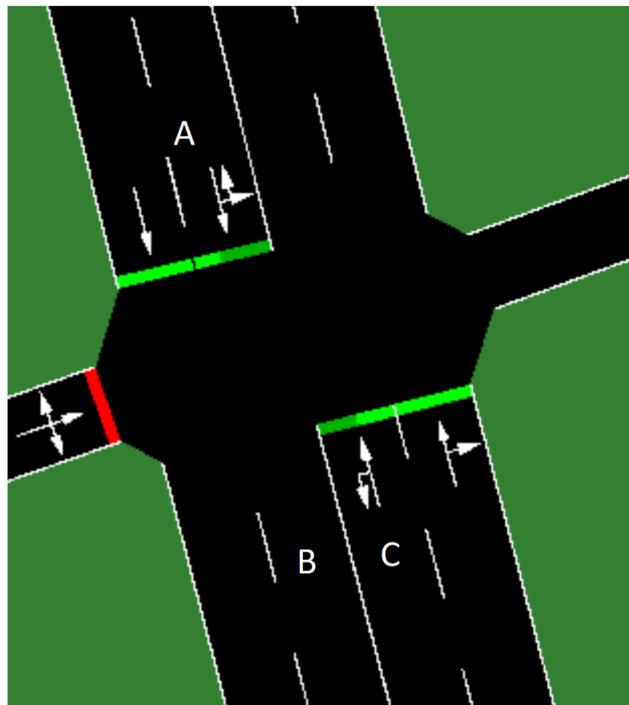


Figura 5: Intersección con cruce por prioridad.

2.1.5. Simulador de Tráfico SUMO (Simulator Urban MObility).

Realizar la planificación de la duración de los ciclos de los semáforos requiere una gran cantidad de pruebas que permitan determinar la factibilidad de una configuración dada como solución. El desarrollo de los diferentes experimentos en un entorno real acarrearía un sinnúmero de inconvenientes y los consecuentes peligros a la seguridad de conductores y peatones, por lo que para solventar esta necesidad se utilizan herramientas de simulación. En este caso, se ha optado por el simulador SUMO (DLR, 2017), por su clara aceptación en la simulación de microtráfico vehicular, sus características de código abierto y su elevada potencia en las tareas de

simulación y evaluación. Además, está dotado de una potente interfaz gráfica para visualizar las redes y simular el comportamiento del tráfico de acuerdo a lo requerido como se aprecia en la figura 6.

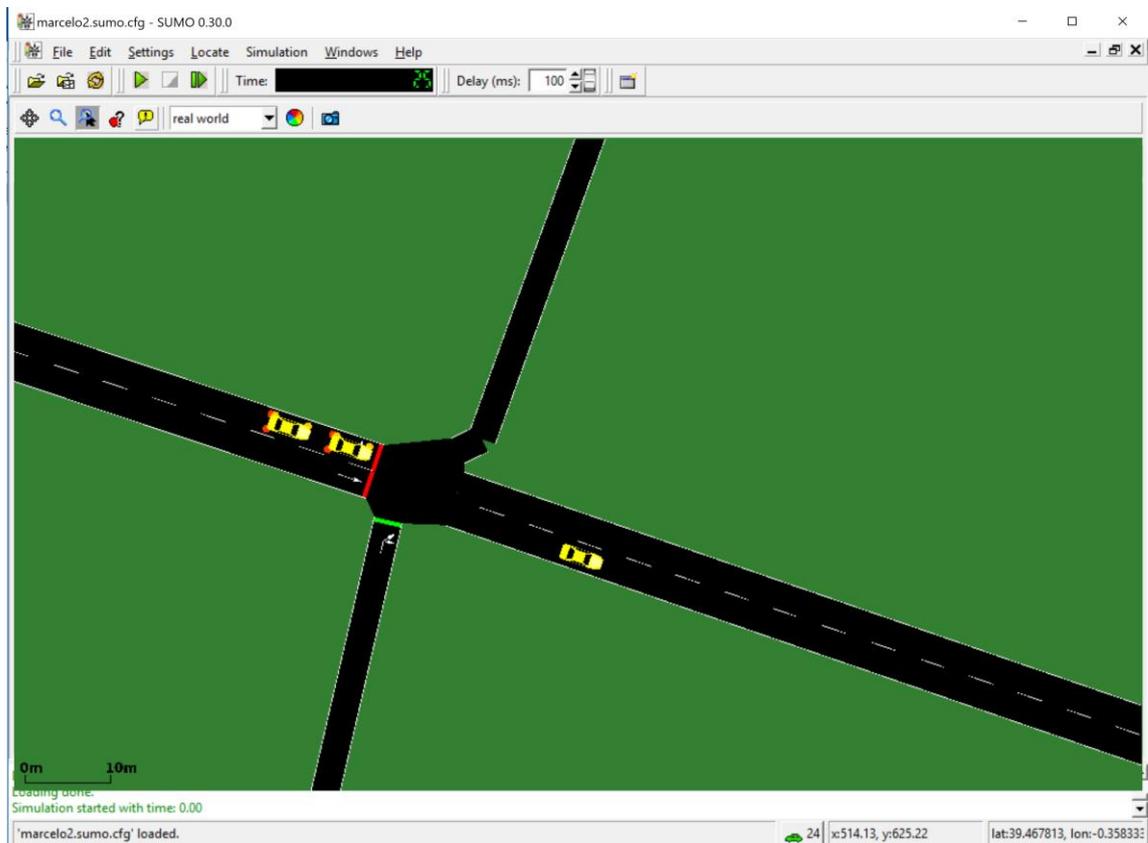


Figura 6: Entorno del Simulador Gráfico SUMO.

Entre las características principales del simulador podemos destacar las siguientes:

- Permite la simulación microscópica de los diferentes elementos involucrados en el tráfico: vehículos, peatones, transporte público, entre otros.
- Interacción en línea, mediante TraCI (Traffic Control Interface).
- Simulación de tráfico multimodal, por ejemplo, vehículos, transporte público y peatones.
- Planificación de programas de control de semáforos, los mismos que pueden ser importados o generados automáticamente por SUMO.
- Variedad de formatos para la importación de redes, entre los que podemos destacar OpenStreetMap, VISUM, VISSIM y NavTeq.
- Implementado en C++ y hace uso de bibliotecas portátiles. (DLR, 2017)
- Algunos de los principales componentes utilizados en el desarrollo del presente trabajo son:

- SUMO: esta herramienta ha sido utilizada para la simulación en línea de comandos, es en sí la parte central del simulador que se ejecuta una vez que se hayan configurado los parámetros de simulación y se hayan generado los diferentes ficheros que representan las características de la red y del tráfico.
- GUI SIM: interfaz de usuario para simulación gráfica, que permite visualizar todos elementos y cada uno de los detalles de la red, nos da la posibilidad de avanzar es pasos de tiempo, detener la simulación y analizar los parámetros de vehículos, semáforos y del tráfico en general.
- NETCONVERT: herramienta para la importación de la topología de la red a través de redes definidas por nodos y ejes, en nuestro caso desde el formato OSM de OpenStreetMaps.
- RANDOMTRIPS: herramienta programada en Python para la generación aleatoria de rutas para la evaluación del tráfico generado por los vehículos.
- NETEDIT: herramienta de edición visual para las diferentes redes de calles, semáforos y otros elementos de la topología de la red, nos permite construir programas de ciclos de los semáforos, y agregar o quitar elementos de los mapas previamente importados.

Los ficheros de configuración, así como los que contienen la información de evaluación de la red, se encuentran en formato XML, y se dispone de las librerías necesarias para su conversión a formato CSV, para su mejor evaluación y análisis.

Las características de simulador de microtráfico de las que dispone sumo nos permiten analizar diferentes parámetros a nivel de vehículos, ya sea a través de la interfaz gráfica o consultando los diferentes ficheros de salida que nos presenta. Así podemos destacar parámetros estáticos como los que se observan en la figura 7 tomada de SUMO para el modelo de vehículo generado por defecto, en donde encontramos las dimensiones físicas del automotor, la clase de vehículo, el modelo de emisión de gases utilizado para calcular CO₂, CO, HC, NO_x, PM_x, consumo de combustible, etc. El *car follow model* o modelo de seguimiento de los vehículos se utiliza para determinar la forma en que se comportan cuando tienen otro vehículo por delante, parámetros de velocidad y aceleración, entre otros.

Name	Value	Dynamic
Type Information:		
type [id]	DEFAULT_VEHTYPE	×
length	5.00	×
width	1.80	×
height	1.50	×
minGap	2.50	×
vehicle class	passenger	×
emission class	HBEFA3/PC_G_EU4	×
car follow model	Krauss	×
maximum speed [m/s]	55.56	×
maximum acceleration [m/s ²]	2.60	×
maximum deceleration [m/s ²]	4.50	×
imperfection (sigma)	0.50	×
reaction time (tau)	1.00	×
person capacity	4	×
boarding time	0.50	×
container capacity	0	×
loading time	90.00	×
type parameters [key:val]		×

Figura 7: Parámetros estáticos para vehículo por defecto.

Otra de las características que podemos observar son los parámetros dinámicos, es decir aquellos que varían de acuerdo a las características de la simulación, los mismo varían para cada time step (instante de tiempo configurado en la simulación) y podemos observar en la figura 8, algunos de los más relevante como la velocidad, aceleración, tiempo de espera, tiempo de espera acumulado, tiempo perdido, las cantidades de gases emitidos, la cantidad de combustible consumido y el ruido generado por el vehículo en determinado instante de la simulación. Destacamos que la información que se presenta de forma gráfica se puede obtener en los ficheros de salida de la simulación, o en los ficheros generados a través de la simulación mediante consola de comandos.

Name	Value	Dynamic
lane [id]	10377690#3_1	✗
position [m]	133.28	✓
lateral offset [m]	0.00	✓
speed [m/s]	1.42	✓
acceleration [m/s^2]	-3.09	✓
angle [degree]	108.56	✓
slope [degree]	0.00	✓
time gap on lane [s]	1.78	✓
waiting time [s]	0.00	✓
waiting time (accumulated, 100.00s) [s]	0.00	✓
time loss [s]	9.38	✓
impatience	0.00	✓
last lane change [s]	11.00	✓
desired depart [s]	11.00	✗
depart delay [s]	0.00	✗
remaining [#]	0	✗
stop info		✗
line		✗
CO2 [mg/s]	0.00	✓
CO [mg/s]	0.00	✓
HC [mg/s]	0.00	✓
NOx [mg/s]	0.00	✓
PMx [mg/s]	0.00	✓
fuel [ml/s]	0.00	✓
electricity [Wh/s]	0.00	✓
noise (Harmonoise) [dB]	43.40	✓
devices	tripinfo_	✗
persons	0	✓
containers	0	✓
parameters [key:val]		✗

Figura 8: Parámetros dinámicos de un vehículo en SUMO

SUMO también nos permite observar a través de su interfaz gráfica los parámetros asociados a las vías, así podemos destacar en la figura 9 la velocidad máxima permitida, longitud y ancho del tramo, porcentaje de ocupación, tipo de vía y los tipos de vehículos permitidos y restringidos para la misma.

Name	Value	Dynamic
maxspeed [m/s]	27.78	✗
length [m]	82.58	✗
width [m]	3.20	✗
street name		✗
stored traveltime [s]	-1.00	✓
loaded weight	-1.00	✓
routing speed [m/s]	1.#J	✓
brutto occupancy [%]	0.00	✓
netto occupancy [%]	0.00	✓
edge type	highway.primary	✗
priority	9	✗
allowed vehicle class	iclc custom1 custom2	✗
disallowed vehicle class	an rail rail_electric ship	✗
permission code	29237247	✗

Figura 9: Parámetros de la vía en SUMO

2.2. Técnicas de Inteligencia Artificial.

Existen muchos problemas en que la complejidad de los mismos, o la gran cantidad de factores que se ven involucrados en su resolución, no permiten encontrar un método algorítmico o alguna técnica exacta que nos lleve a obtener una solución. También existen casos en que, dada una solución, resulta complicado determinar si se trata de la mejor dentro de un conjunto de posibles soluciones, por lo que la misma debe ser generada y mejorada a través de un proceso de búsqueda automática, valiéndonos de las herramientas de la inteligencia artificial.

Las técnicas de inteligencia artificial se basan en el modelado del comportamiento humano, una aproximación a las características que nos permiten a las personas resolver diferentes problemas complejos. Son una alternativa rápida y precisa a los problemas de optimización respecto a otras técnicas convencionales existentes.

Generalmente las técnicas de inteligencia artificial se basan en un conjunto de soluciones a partir de las cuales se obtiene una solución más óptima, algunas de las características más relevantes de estas técnicas son las siguientes:

- Toman en cuenta resultados obtenidos anteriormente.
- Aprenden y adaptan su desempeño ante variaciones del problema inicial.

- Pueden planificar su camino de búsqueda de soluciones hacia adelante.
- Actúan de manera inteligente basados en la inteligencia humana o comportamientos sociales (Badar, Umre, & Junghare, 2014).

En el análisis del caso de la optimización de los tiempos de duración de los ciclos de los semáforos, visto como un problema de optimización combinatorial, por las diferentes opciones de configuración, así como la gran cantidad de restricciones existentes, hemos decidido, dentro de las técnicas de inteligencia artificial, utilizar algoritmos genéticos. Estos algoritmos nos permiten utilizar una representación vectorial de las soluciones, que se adecuan a los requerimientos del simulador, además de ser de fácil implementación.

Entre las principales técnicas de inteligencia artificial enfocadas en dar soluciones a problemas de optimización, podemos destacar los Enjambres de Partículas, Colonias de Hormigas, Búsqueda Tabú, Alineamiento Simulado y Algoritmos Genéticos.

2.2.1. Enjambre de Partículas.

La optimización por enjambre de partículas conocido como PSO por sus siglas en inglés (Particle Swarm Optimization) es una técnica de computación inteligente desarrollada en 1995 por Kennedy y Eberhart, quienes se basaron en el comportamiento social de los individuos inspirados en los bancos de peces o bandadas de aves. En esta técnica el sistema se inicializa con un conjunto de soluciones generadas aleatoriamente y cada solución está asociada a una velocidad aleatoria, aquí las soluciones potenciales son conocidas como partículas. El concepto de esta técnica se basa en que en cada instante de tiempo se cambia la velocidad de una partícula hacia su mejor posición y al mismo tiempo hacia la mejor posición global del enjambre.

El enjambre de partículas es una técnica que resulta de mucha utilidad para la optimización de soluciones de un problema con un amplio rango de funciones (Eberhart & Kennedy, 1995).

2.2.2. Colonias de Hormigas.

Las hormigas son insectos que presentan un comportamiento social, viven en colonias y enfocan sus acciones en el desarrollo de la colonia, basados en

esto Marco Dorigo propuso esta técnica de inteligencia artificial para problemas de optimización combinatoria.

Las colonias de hormigas se centran en cada individuo como un mecanismo probabilístico que encuentra soluciones al problema, usando rastros de una feromona artificial que cambia con el tiempo para simular la disipación de la misma en el ambiente. De esta forma las hormigas que localizan el camino más corto hacia el hormiguero emanan una mayor cantidad de feromona que atrae al resto, y por el contrario, aquellas que han tomado un camino menos eficiente, concentran menor cantidad de feromona, por lo tanto ese camino tiene menos probabilidad de ser seguido por las demás. Esta técnica se presenta como una buena alternativa para problemas de optimización multidimensionales (Dorigo, 1996).

2.2.3. Búsqueda Tabú.

La Búsqueda Tabú es un procedimiento adaptativo utilizado para resolver problemas de optimización combinatoria que hace uso de otros métodos como algoritmos de programación lineal y heurísticas especializadas intentado superar las limitaciones de los óptimos locales (Glover, 1989).

Esta técnica es una búsqueda por vecindario, empieza generando una solución aleatoria y se continúa con el vecindario de la solución actual. En cada iteración del algoritmo se genera un movimiento o cambio en el valor de la solución y así se va variando la solución actual conjuntamente con su vecindario, limitándose al conjunto de soluciones y a las restricciones del problema de optimización.

2.2.4. Alineamiento Simulado.

Esta técnica se basa en los principios físicos de la termodinámica, cuya inspiración nace en la observación de los procesos de formación de cristales en el enfriamiento de sólidos. También se conoce como recocido simulado y es se trata de obtener el estado de baja energía de un sólido.

En el Alineamiento simulado la temperatura del solido se eleva hasta conseguir que este se funda y posteriormente el elemento se enfría hasta conseguir que las partículas se dispongan en un estado de energía mínimo. El algoritmo empieza con una solución con un determinado valor de evaluación y se mueve a través de su vecindario mediante diferentes tipos de movimientos permitidos. Es una técnica que se adapta fácilmente a diferentes problemas de optimización, es de fácil implementación y dispone



de gran facilidad para superar óptimos locales (Badar, Umre, & Junghare, 2014).

2.3. Algoritmos Genéticos.

Los algoritmos genéticos fueron introducidos en la década de 1970 por John Holland en la Universidad de Michigan, inspirándose inicialmente en los procesos de evolución de los seres vivos, que en su lucha por la supervivencia y adaptación mantienen en sus cromosomas las características que a priori le brindan a cada individuo, una mayor probabilidad de sobrevivir.

Basados en la definición original podemos definir a los algoritmos genéticos como sistemas de búsqueda que se basan en los mecanismos de selección y genética natural. Toman en cuenta los principios de supervivencia de los más aptos y compatibles entre las estructuras de cadenas, que contienen información aleatorizada, sometida a procesos de intercambio, reproducción y mutación (Goldberg, 1985).

Los algoritmos Genéticos se pueden describir básicamente en cuatro etapas. La primera etapa de inicialización de una población de individuos; una segunda etapa de evaluación y selección de acuerdo a las características de cada individuo y su aptitud respecto a ciertos parámetros de evaluación; reproducción para generar nueva descendencia y finalmente la mutación aleatoria de la nueva descendencia (Jones, 2005).

Por lo descrito anteriormente respecto a las técnicas de inteligencia artificial y en especial a los algoritmos genéticos, el presente trabajo se centra en estos últimos basado en los siguientes aspectos:

- Los algoritmos genéticos son intrínsecamente paralelos, respecto a la mayor parte de las otras técnicas que solo permiten explorar el espacio de soluciones en una sola dirección al mismo tiempo, existiendo la posibilidad de encontrar soluciones que lleven a óptimos locales, algo que se solventa mediante la mutación en los genéticos, que permite explorar el espacio de soluciones en direcciones múltiples a la vez.
- Una de las fortalezas de los algoritmos genéticos es que nos permiten trabajar con problemas con un espacio de soluciones relativamente grande, debido al paralelismo implícito que explora diferentes direcciones en el espacio de soluciones, encontrando resultados óptimos en periodos de tiempo cortos.
- En problemas con espacios de soluciones que a priori se presentan imposibles de resolver exhaustivamente, los genéticos son una de las mejores alternativas, sobre todo cuando nos encontramos con

funciones de aptitud discontinuas, ruidosas, que cambian mucho con el tiempo o poseen gran cantidad de óptimos locales, los genéticos consiguen mejores resultados gracias a los operadores genéticos, cuyo objetivo es escapar de estos óptimos locales.

- Otra característica importante a la hora de utilizar los algoritmos genéticos es su gran capacidad de trabajar con problemas que involucran muchos parámetros simultáneamente, es decir, su paralelismo permite obtener conjuntos de soluciones en donde una solución puede minimizar o maximizar un parámetro y otra solución puede hacer lo mismo con otro parámetro diferente, siendo las dos, igual de óptimas (Mateos, 2004).

2.3.1. Términos de un Algoritmo Genético.

Hemos descrito anteriormente la estrecha relación existente entre los algoritmos genéticos y los procesos evolutivos biológicos, por lo que muchos términos utilizados responden a sus equivalentes en la biología evolutiva, como los descritos a continuación:

- Cromosomas. Se refiere a la estructura básica de codificación con la cual se construye el conjunto de soluciones para el problema. Generalmente una población está compuesta por más de un cromosoma.
- Genotipo. Se denomina de esta forma al conjunto completo de cromosomas.
- Fenotipo. Se trata de un individuo en particular dentro del Genotipo.
- Genes. Cada uno de los valores individuales que conforman un Cromosoma.

En la figura 10 podemos apreciar la representación de los términos mencionados.

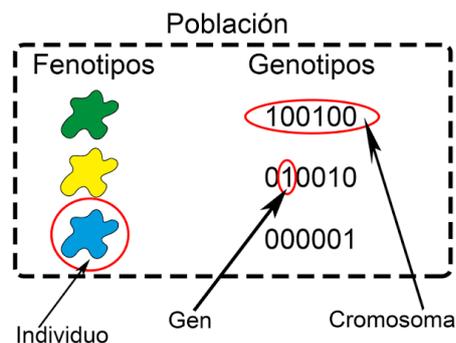


Figura 10: Representación de términos usados en un Algoritmo Genético (Castrovation, 2017).

2.3.2. Operadores Genéticos.

Un algoritmo genético puede utilizar diferentes técnicas para cada una de las etapas que lo conforman, algunas veces se realiza la combinación de las técnicas para cada uno de los procesos del algoritmo. A continuación, describimos los principales operadores para cada una de las etapas.

2.3.2.1. Selección.

La selección consiste en tomar los individuos más aptos de la población, de los cuales se elegirá el más apto para que forme parte de la nueva población y de esta forma asegurarnos una mejora constante en la descendencia.

Existen diferentes variantes para el operador de selección entre los cuales tenemos:

- Selección Elitista. Se garantiza la selección de los miembros más aptos de cada generación. También se suele utilizar ciertas variantes de este tipo de selección, en donde los mejores individuos de cada generación pasan a la siguiente si no existe ninguno mejor.
- Selección Proporcional a la Aptitud. Aquí se asigna una probabilidad de selección acorde a la aptitud del individuo, dando mayores opciones a los individuos que sean más aptos, lo que no necesariamente garantiza su selección.
- Selección por Ruleta. Es una forma proporcional de selección en donde su probabilidad de ser elegidos no depende únicamente de su aptitud, sino que también depende de la diferencia respecto a la aptitud de los demás individuos.
- Selección Escalada. La agresividad de la selección aumenta al incrementarse la aptitud media de la población, es decir este operador se hace más selectivo conforme a los individuos los distinguen características más pequeñas.
- Selección por Torneo. Se basa en una clasificación en subgrupos de la población, de donde para cada subgrupo un individuo es elegido.
- Selección por Rango. A cada individuo se le asigna un rango numérico en función de su aptitud y se selecciona en función de este rango, con lo que se busca reducir la invariabilidad genética que los individuos más aptos puedan ganar al principio en función de los menos aptos.

- Selección Generacional. No se conservan individuos entre las generaciones. Es decir, la descendencia de los individuos seleccionados en cada generación forma se convierte de forma total en la siguiente generación.
- Selección por Estado Estacionario. Se conservan algunos individuos entre generaciones. Es decir, se conserva el acervo genético, que son ciertas características presentes en todas las generaciones.
- Selección Jerárquica. Se realizan múltiples rondas de selección en cada generación. Siendo las primeras evaluaciones rápidas y menos discriminatorias y las generaciones más avanzadas son sometidas a criterios de evaluación más ajustados (Rodríguez-Piñero, 2003).

2.3.2.2. *Cruce.*

Luego de la etapa de selección en donde los individuos a que a priori presentan una mayor aptitud, es necesario realizar ciertas modificaciones sobre ellos, con el objetivo de que las siguientes generaciones varíen y mejoren su aptitud respecto a la generación anterior.

El operador de cruce permite generar variaciones en los cromosomas a partir de los cromosomas padres que han sido elegidos en la etapa de selección. Podemos describir algunos operadores de cruce que serán más aptos de acuerdo a las características de la representación de nuestros cromosomas. Entre los principales tenemos cruce en un punto, cruce de n puntos y cruce uniforme.

- Cruce en un punto. Se genera aleatoriamente un punto que se encuentre entre uno y la longitud del cromosoma, se dividen los padres en el punto generado y se forman hijos intercambiando partes de los cromosomas de los padres como se observa en la figura 11. La desventaja de este operador es que depende de la codificación utilizada, y más específicamente de la posición en la que aparecen los genes en los cromosomas.



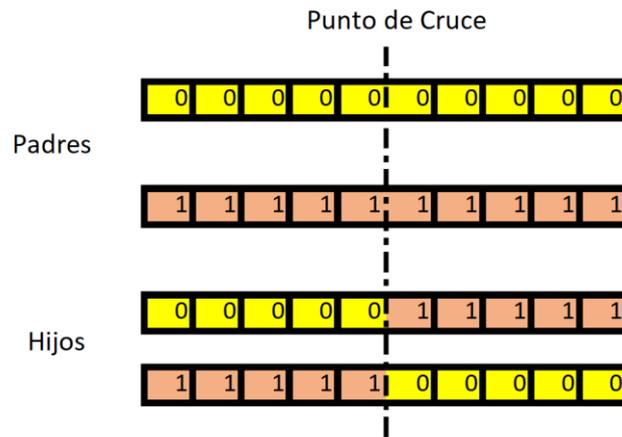


Figura 11: Operador de Cruce en un Punto

- Cruce en n Puntos. En este caso se generan aleatoriamente n puntos de cruce y se juntan las partes alternadas de los padres para generar los hijos como se aprecia en la figura 12.

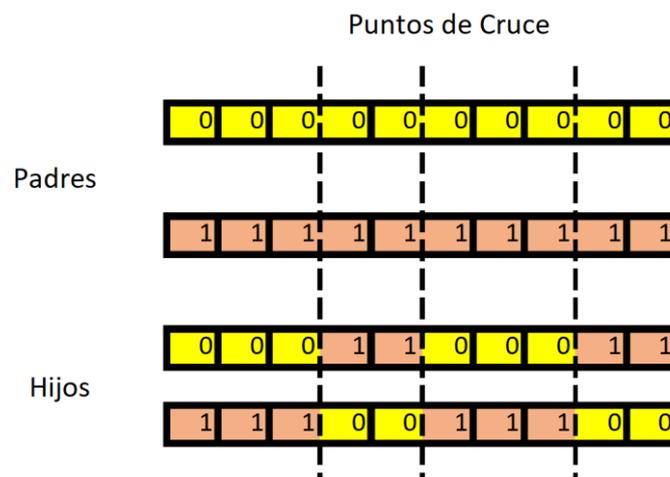


Figura 12: Operador de Cruce en n Puntos

- Cruce Uniforme. Este operador implica la generación de una máscara con valores binarios, y la selección al azar de uno de los padres, entonces si en la máscara aparece un valor igual a 1, se toma el gen del padre seleccionado, caso contrario, si aparece un 0, el gen pertenece al otro de los padres. Para producir otro descendiente, se intercambia el papel de los padres o se invierte el valor de la máscara binaria. Este proceso podemos observarlo en la figura 13.

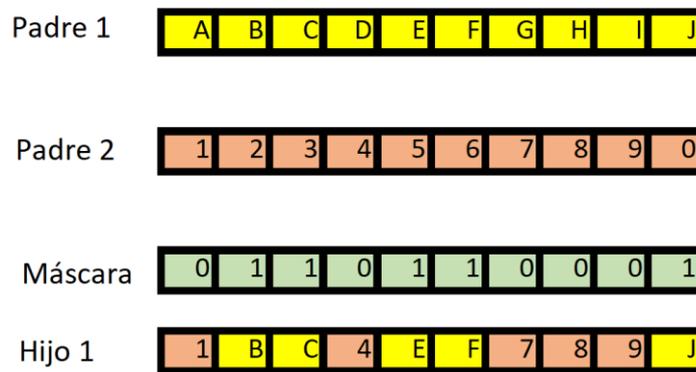


Figura 13: Cruce Uniforme

También se suelen plantear el uso de operadores que crucen los cromosomas de tres padres, de acuerdo a la naturaleza del problema, aunque los mismos no correspondan necesariamente a la naturaleza de la evolución biológica (Berzal , 2017).

2.3.2.3. Mutación.

La mutación se considera un operador básico que nos permite agregar un elemento de aleatoriedad sobre el entorno de los individuos de la población, lo que permite la posibilidad de generar nuevas soluciones cuando la población de individuos va convergiendo (Lawrence, 1985).

La selección de la probabilidad de mutación es un aspecto básico en el desarrollo de los algoritmos genéticos, eligiéndose por lo general probabilidades muy bajas, ya que de lo contrario estaríamos ante una búsqueda aleatoria. Sin embargo, un valor demasiado bajo puede llevar a que la población converja en un óptimo local. Holland propone utilizar una probabilidad de mutación igual a $\frac{1}{l}$ donde l es la longitud del cromosoma (Holland, 1992).

2.3.3. Diagrama de Flujo del Algoritmo.

Un Algoritmo Genético de forma general posee la estructura descrita a continuación. En primer lugar, se lleva a cabo la inicialización, es decir se

genera una población inicial de forma aleatoria a partir de la cual se realizarán las operaciones necesarias para obtener nuevas y mejores soluciones, luego se procede a evaluar cada uno de los individuos a través de una función que nos permita obtener la aptitud en función de uno o varios parámetros. En este punto se evalúa también cierta condición de término, que de no cumplirse llevará a la ejecución de los operadores genéticos seleccionados. Es decir, se seleccionan los individuos que se convertirán en los padres de la siguiente generación, se generan nuevos individuos mediante los operadores de cruce, se aplica la mutación para proveer de cierta variabilidad sobre la descendencia y se aplica la función de evaluación para determinar la aptitud de los individuos de la nueva generación. El algoritmo podemos apreciarlo en el siguiente diagrama de flujo en la figura 14.

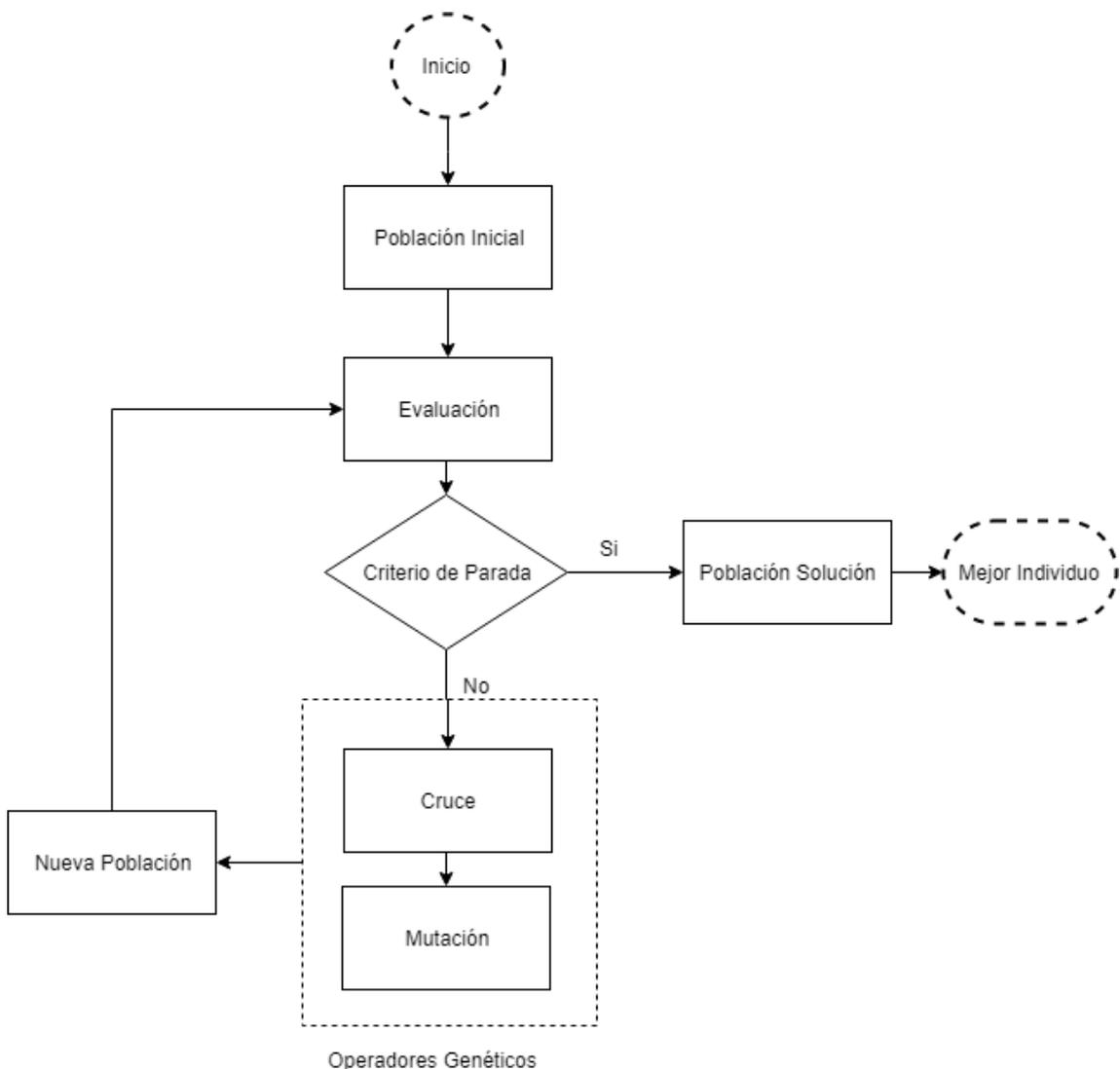


Figura 14: Diagrama de Flujo Algoritmo Genético

3. Propuesta de Solución

Para dar solución al problema planteado en la sección 1.3. acerca de la optimización de la duración de los ciclos de los semáforos, se aplicará un algoritmo genético el cual describimos en el presente capítulo.

3.1. Procedimiento.

Los pasos a seguir para la solución del problema son en primer lugar la codificación de nuestro problema, posteriormente la obtención de los datos para determinada área de la ciudad, generación de los parámetros para dicha área. La segunda parte corresponde a la generación de la población inicial y finalmente el algoritmo genético para la optimización de los ciclos de los semáforos, como se puede observar en la figura 15.



Figura 15: Diagrama del Procedimiento para la Solución del Problema

3.2. Codificación.

Para nuestro problema se ha codificado la duración de las fases de los ciclos de cada conjunto semafórico para la topología de las calles de un sector determinado de la ciudad. Hemos adoptado esta representación formada por un vector de números enteros positivos, ya que es la misma utilizada por el simulador SUMO para los programas de control de semáforos y garantiza la interrelación con las intersecciones adyacentes, como podemos apreciar en la figura 16.

Un aspecto importante de esta representación es que restringe los estados de las fases a valores válidos para evitar combinaciones no permitidas que produzcan colisiones en las intersecciones. Además, simplifica los estados de los semáforos de cada carril reduciendo la cantidad de datos ya que no es necesario representar el valor de cada semáforo de forma individual.

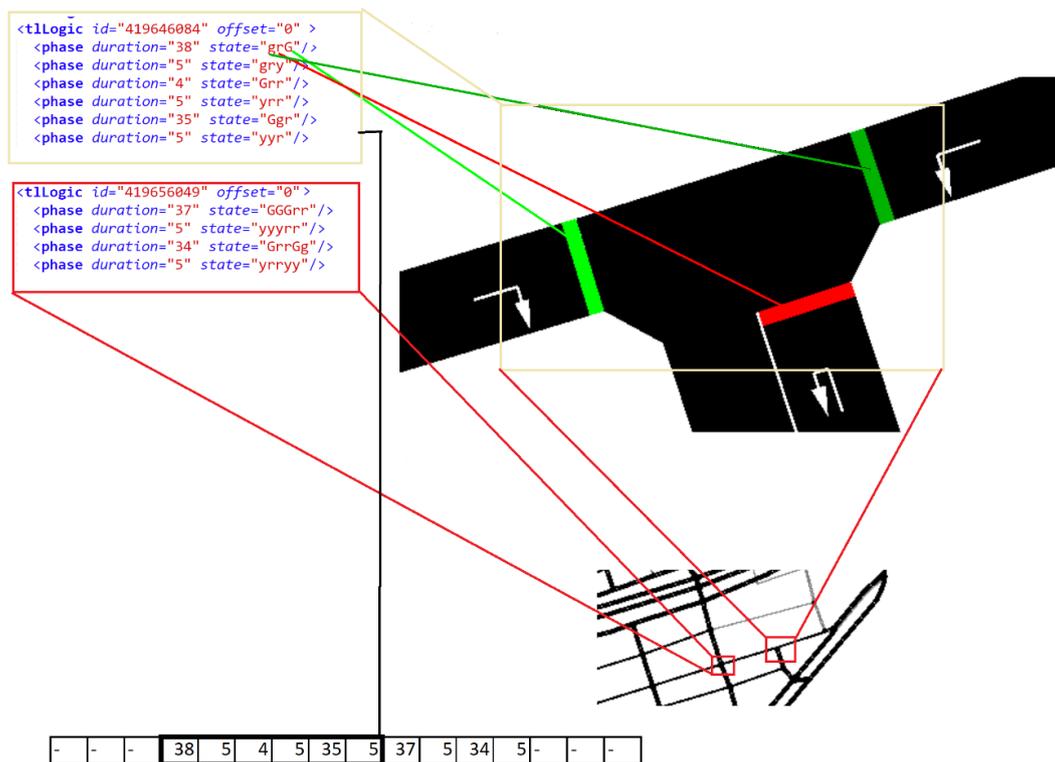


Figura 16: Intersección en el Simulador SUMO, duración de las fases y codificación

Cabe mencionar que SUMO realiza una configuración de la duración de los ciclos de los semáforos basado en su propio algoritmo SUMO Cycle Program Generator (SCPG), que consiste en asignar determinados valores dentro de

un intervalo de tiempo, con base en la proporción de estados en verde de las fases, la cantidad de vías de entrada existentes en las intersecciones y el tiempo de frenado de los vehículos cuando se acercan a los semáforos (García Nieto & Alba, 2013).

Para nuestra codificación, cada gen corresponde al tiempo de duración de cada una de las fases que conforman un ciclo semafórico, teniendo así un cromosoma de longitud igual a la cantidad de fases de cada ciclo por el número de controladores de tráfico existentes en la topología.

En la figura 17 se observa un fragmento de la codificación para una intersección, en donde el número 37 corresponde a la duración en segundos de la fase en los estados Green-Green-Green-red-red para los semáforos numerados del 0 al 4. A partir del segundo 37, se realiza el cambio de fase a yellow-yellow-yellow-red-red con una duración de 5 segundos, el cambio posterior es a una fase red-red-red-Green-Green con una duración de 36 segundos y finalmente el ciclo termina con la fase red-red-red-yellow-yellow con una duración de 5 segundos, para dar un ciclo total de duración de 83 segundos .



Figura 17: Representación de la codificación de los tiempos y los estados de los semáforos.

3.3. Obtención de los Datos.

Para llevar a cabo la generación de la población inicial, una vez que se ha encontrado la representación adecuada para los cromosomas del algoritmo genético, procedemos a extraer los datos de la topología de la red, mediante la descarga del mapa seleccionado desde la web de OpenStreetMap (OpenStreetMap, 2017). Esta web nos permite descargar mapas en formato libre de cualquier ciudad del mundo, los mismo que tienen información acerca de ciudades, caminos, carreteras, vías, semáforos, estaciones de buses y ferroviarias, entre otras, como podemos observar en la figura 18. Exportamos determinada área, la misma que se archiva en formato .osm.

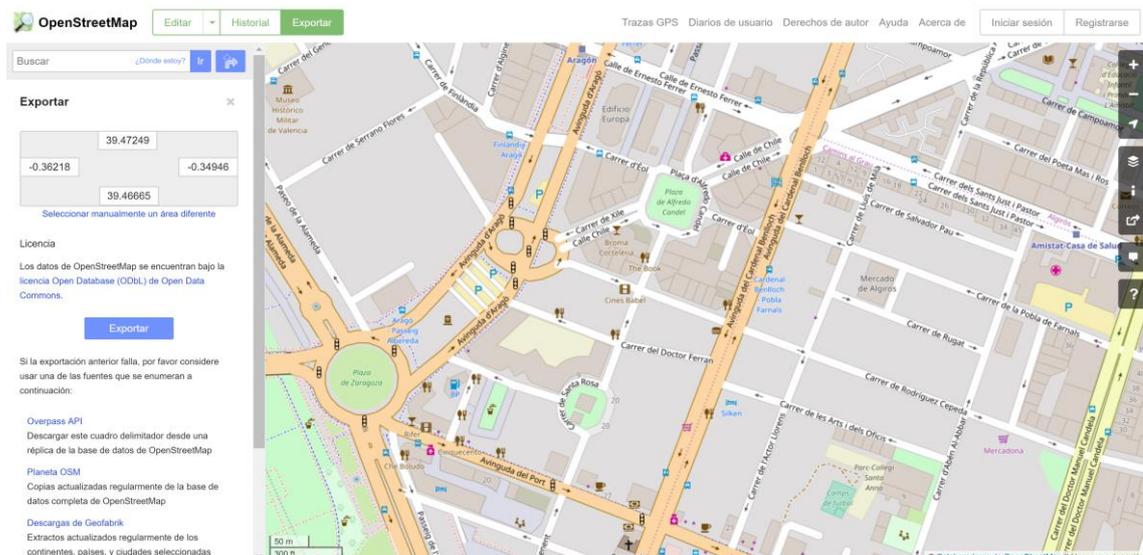


Figura 18: Captura del mapa en la página web OpenStreetMap

Una vez exportado el mapa, empezamos a hacer uso de las herramientas de SUMO, en primer lugar, convertimos el mapa al formato requerido por el simulador, mediante la herramienta NETCONVERT generamos los ficheros con extensión .net.xml a partir del mapa.osm. En este punto se utiliza la herramienta NETEDIT que nos permite editar el mapa, para agregar o quitar elementos de acuerdo a los requerimientos de la simulación. En la figura 19 podemos observar el mapa.net.xml cargado en el entorno gráfico del simulador.

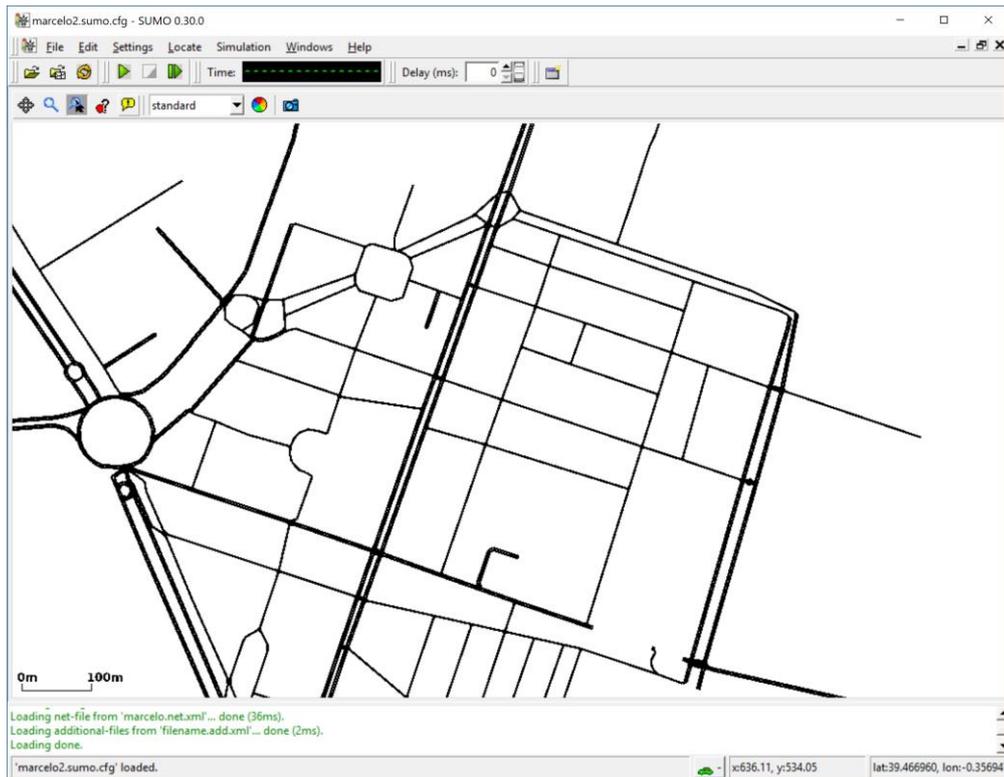


Figura 19: Mapa importado al simulador SUMO

Luego de obtener la topología de las calles sobre las cuales se va a evaluar la optimización de los ciclos de los semáforos, generamos las rutas que tomarán los diferentes vehículos mediante la herramienta randomTrips.py. Esta herramienta nos genera un fichero en formato .rou.xml en donde se listan las rutas para cada vehículo, para lo cual es necesario definir como dato de entrada la cantidad de vehículos a simular. La estructura del fichero la podemos analizar en la figura 20.

```

<vehicle id="0" depart="0.00">
  <route edges="158013564#0 158013564#1 23290336#6 186342046#1"/>
</vehicle>
<vehicle id="1" depart="1.00">
  <route edges="23396934#2 23160312 23160310#1 186347183#1 23160076#1 23160076#2
23160076#3 10377690#6 23396933#0 23396933#1 23396933#2 186342046#1 23113278#0
23113278#1 23113287#1 23113287#2 23113287#3 23113287#4 23113287#5"/>
</vehicle>

```

Figura 20: Estructura del fichero mapa.rou.xml

El fichero mapa.rou.xml nos presenta para cada ruta planeada, el identificador del vehículo que toma esa ruta, el tiempo en el cual se realiza la partida, y los ejes que atraviesa el vehículo desde el punto de partida hasta su destino.

Posteriormente creamos un fichero de configuración en donde se incluirá la información necesaria para ejecutar una simulación ya sea en el modo

gráfico o en el modo de consola, la estructura de dicho fichero se muestra en la figura 21.

```
<configuration>
  <input>
    <net-file value="marcelo.net.xml"/>
    <route-files value="marcelo.rou.xml"/>
  </input>

  <output>
    <tripinfo-output value="output-tripinfos.xml"/>
  </output>

  <time
    begin="0"
    end="500"
    time-to-teleport="-1"
    srand="13423"
    route-steps="-1"
  />

  <reports
    print-options="false"
  />
</configuration>
```

Figura 21: Captura del fichero de configuración *mapa.sumo.cfg*

En el fichero de configuración dentro de la etiqueta *input* se asocia con el fichero que guarda la topología de la ciudad dentro del atributo *net-file*, también se asocia con el fichero de rutas generado previamente en el atributo *route-files*, dentro de esta etiqueta también se puede agregar cualquier fichero adicional como es el caso de los ficheros que contendrán los programas con la duración de los ciclos optimizados. Dentro de la etiqueta *output* debemos agregar la lista de todos los ficheros con reportes que se requieran, aunque también se lo puede hacer desde la consola de comandos, finalmente en la etiqueta *time* establecemos todos los parámetros referidos al tiempo de simulación, como duración de la simulación, además de que podemos establecer una semilla fija que nos permita replicar las simulaciones y los parámetros que de otra forma toman valores aleatorios.

A partir de este momento se dispone de todos los elementos básicos para ejecutar una simulación. Sin embargo, para nuestro trabajo es necesario recurrir al fichero *mapa.net.xml* y extraer la información relativa a los semáforos y su respectivo plan de duración de los ciclos, a partir del cual posteriormente se generará la codificación para el algoritmo genético.

En el fichero antes mencionado encontramos la cantidad de controladores para cada grupo semafórico denominado *tLogic*, el mismo que posee un atributo *id* para su identificador, un atributo *offset* para establecer un tiempo de inicio del programa de control, y un identificador *program* para indicar a

que programa de control pertenece la configuración. Por cada controlador existe un ciclo semafórico dividido en fases con sus atributos *duration* que indica la duración de cada fase y *state* para indicar los estados o colores de los semáforos, como se muestra en la siguiente figura 22.

```
<tlLogic id="cluster_249528572_3941598522" offset="0" programID="1" type="static">
  <phase duration="36" state="GGrrr"/>
  <phase duration="5" state="yyrrr"/>
  <phase duration="37" state="rrGGG"/>
  <phase duration="5" state="rryyy"/>
</tlLogic>
<tlLogic id="cluster_252108755_252108758" offset="0" programID="1" type="static">
  <phase duration="27" state="GGGrrrGGgg"/>
  <phase duration="5" state="yyyrrryygg"/>
  <phase duration="6" state="rrrGrrrrrGG"/>
  <phase duration="5" state="rrryrrrrryy"/>
  <phase duration="28" state="rrrrGGGrrrr"/>
  <phase duration="5" state="rrrryyrrrr"/>
</tlLogic>
<tlLogic id="cluster_252108760_252108761" offset="0" programID="1" type="static">
  <phase duration="37" state="GGrrGGg"/>
  <phase duration="5" state="yyrryyy"/>
  <phase duration="37" state="rrGGrrr"/>
  <phase duration="5" state="rryyrrr"/>
</tlLogic>
```

Figura 22: Extracto del fichero *mapa.net.xml* con la configuración de semáforos de la topología

A continuación, procedemos a ejecutar el simulador en su modo consola mediante el comando *sumo* con los siguientes parámetros:

-c mapa.sumo.cfg. este parámetro asocia la actual simulación con los parámetros de configuración.

--device.emissions.probability 1. Este parámetro permite generar un modelo de emisión de gases y consumo de combustible para los vehículos, que para nuestro trabajo se basa en el modelo del Manual de Factores de Emisión para Transporte terrestre HBEFA (Handbook Emission Factors for Road Transport) que provee factores de emisión para todas las categorías de vehículos actuales, dividido en diferentes clases para las distintas situaciones de tráfico.

--fcd-output fcd_out.xml. Este parámetro genera un fichero de salida que contiene por cada *step* de la simulación parámetros como velocidad, posición y eje sobre el que se encuentra el vehículo en determinado tiempo, la misma que nos servirá para determinar los instantes en que el vehículo se detiene, observando que su velocidad sea menor que 0.07 metros por segundo.

Una vez realizada esta primera simulación, se generan automáticamente los ficheros que contienen la información de los viajes realizados por cada

vehículo, de donde podemos extraer información relevante como las rutas, la longitud de las rutas, el momento en que se han detenido frente a un semáforo, la misma que nos servirá más adelante para generar nuestra función de evaluación para el algoritmo genético.

3.4. Población Inicial.

La población inicial se ha obtenido a partir de valores de la duración de las fases generadas por el algoritmo SCPG de SUMO, a partir de los cuales hemos generado valores aleatorios en el rango de ± 3 segundos para cada uno de los tiempos, con excepción de las fases que contienen el estado amarillo que se las ha dejado constantes, como se aprecia en la figura 23. En nuestro trabajo se ha empezado con una población inicial de 100 individuos.

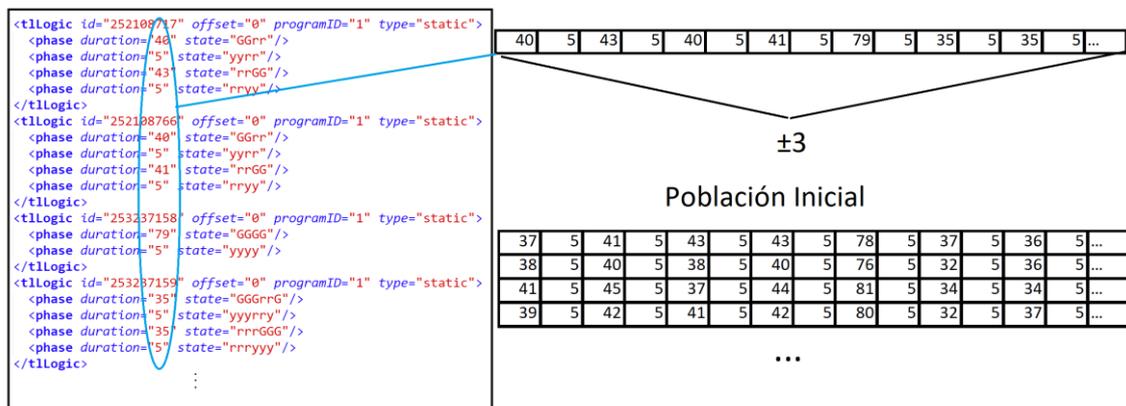


Figura 23: Generación de la población inicial.

3.5. Función de Evaluación.

La función de evaluación o fitness es aquella que nos permite determinar cuan apto es un individuo, respecto a diferentes factores. En nuestro trabajo, proponemos utilizar principalmente una función de bajo costo computacional. Esta función busca reducir el tiempo de ejecución del algoritmo genético, con resultados aceptables que minimicen los parámetros de duración promedio de los viajes de los vehículos, tiempo de espera promedio de los vehículos debido a conducir por debajo de la velocidad óptima y la emisión de gases contaminantes, específicamente CO₂.

Para establecer la aptitud de las soluciones generadas para la optimización de los ciclos de los semáforos, se propone la siguiente función de fitness, que se ha obtenido mediante el siguiente proceso.

Em primer lugar, se ha determinado el tiempo medio que le toma a un vehículo llegar a una intersección en donde se encuentra un semáforo y tomando en cuenta si el estado le permite o restringe el paso, se ha calculado el tiempo de espera en la intersección. Este proceso se ha repetido para todos los vehículos para obtener el parámetro *waitSteps* que nos permite estimar el resto de parámetros que forman la función de evaluación, tomando en cuenta las siguientes expresiones.

$$Duration = waitSteps + t_l + t_{mv}$$

$$t_l = t_a - t_{mv}$$

$$t_{mv} = \frac{length}{allowedSpeed}$$

$$allowedSpeed = 13.89 \frac{m}{s}, 19.44 \frac{m}{s}, 25 \frac{m}{s}$$

Donde t_l es el tiempo perdido por conducir por debajo de la velocidad ideal debido a encontrarse en una intersección o a cualquier parada no programada.

El parámetro t_a corresponde al tiempo actual en que se realiza el cálculo

El parámetro t_{mv} es el tiempo mínimo de viaje, es decir el tiempo que le toma al vehículo completar un tramo si lo hace a la máxima velocidad permitida y se calcula mediante el cociente entre la longitud del tramo sobre la velocidad permitida, que para las rutas urbanas utilizadas es de $13.89 \frac{m}{s}$ cómo lo indica el simulador SUMO y concuerda con lo establecido por la Dirección General de Tráfico de España que establece un equivalente de $50 \frac{km}{h}$ para vías urbanas y travesías, $70 \frac{km}{h}$ en vías fuera de poblado y $90 \frac{km}{h}$ en autopistas y autovías (Tráfico, 2015).

De esta forma hemos podido obtener los parámetros de duración y de tiempo perdido para nuestra función de fitness.

Por ejemplo, para la ruta que se presenta en la figura 24, en donde se ha marcado con letras las intersecciones en donde se encuentran ubicados los semáforos. El vehículo parte del punto A en el segundo 10 de simulación, y a una velocidad media de $11.93 \frac{m}{s}$, recorre 82.23m hasta llegar a la intersección B. Aplicando el cálculo para obtener el tiempo empleado, tenemos que el vehículo ha tardado alrededor de 7 segundos en llegar a la intersección. Comparando con el semáforo que se encuentra en esa intersección en el estado Green con una duración de 36 segundos, podemos



concluir que el vehículo no se detiene. Realizando el mismo procedimiento, se ha determinado que el vehículo para llegar a la intersección F ha tomado 17 segundos, los cuales sumados al momento en que partió el vehículo tenemos 27 segundos y se encuentra con el semáforo en estado red, con una duración de 39 segundos. De lo anterior obtenemos que el tiempo que el vehículo está detenido es la diferencia entre la duración de la fase en red y el tiempo que tarde en llegar el vehículo a la intersección, dándonos como resultado 12 segundos. De esta forma se va sumando el tiempo que el vehículo pasa detenido, denominados waitsteps.

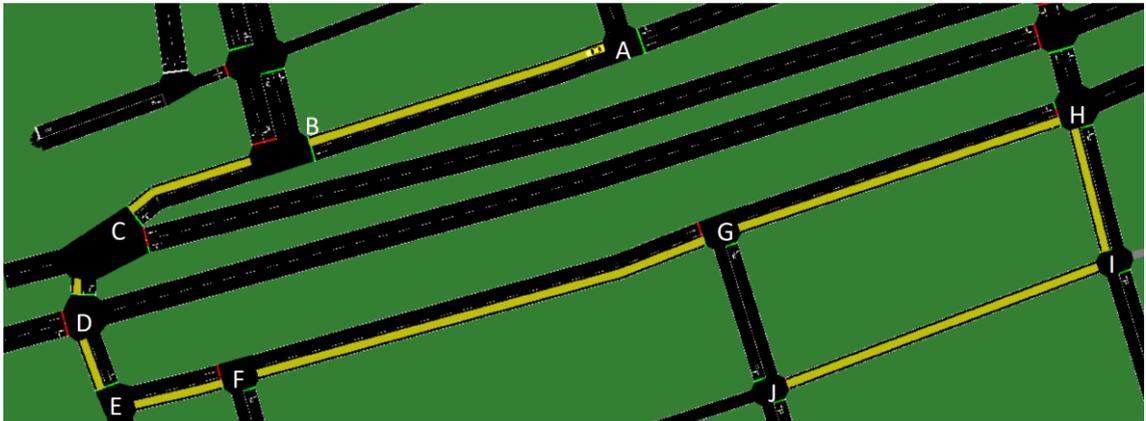


Figura 24: Ruta para un vehículo y cálculo de waitsteps.

3.6. Función de Evaluación en SUMO.

Una vez que se ha realizado la evaluación mediante la función descrita en el apartado anterior, se selecciona una cantidad reducida de los individuos considerados más aptos a través de la mencionada función, los mismos que ahora se volverán a evaluar a través del simulador SUMO, en donde, podemos extraer los siguientes parámetros que nos permitirán terminar la bondad de los individuos.

- duration (segundos) [D]: El tiempo necesario para que cada uno de los vehículos complete la ruta generada.
- routeLength (metros) [rL]: La longitud de la ruta del vehículo.
- waitSteps (steps) [W]: El número de steps en que cada vehículo reduce su velocidad por debajo de 0.1m/s, para nuestro caso la duración da cada step se encuentra fijada a un segundo.
- timeLoss (segundos) [tL]: El tiempo perdido por un vehículo debido a estar por debajo de la velocidad ideal.

- CO₂ (miligramos) [CO₂]: La cantidad de CO₂ emitida por el vehículo durante el viaje. Basada en el modelo HBEFA.
- Fuel (mililitros) [fuel]: La cantidad de combustible consumida por el vehículo durante el viaje.
- CVh (--)[CVh]: La razón dada por el cociente entre la cantidad de vehículos que completan su viaje antes de que concluya la simulación y la cantidad de vehículos programados para su simulación.

Con los parámetros obtenidos en el simulador, elaboramos una función de fitness que nos permita minimizar la duración de los viajes de los vehículos (D), el consumo de combustible (fuel), y el tiempo perdido debido a estar por debajo de la velocidad ideal (tL), además se prioriza y se maximiza la cantidad de vehículos que completan su viaje programado por sobre la cantidad de vehículos simulados.

$$fitness = \frac{\sum_{v=0}^V \alpha \cdot D_v + \beta \cdot fuel_v + \gamma \cdot tL_v}{CVh}$$

En la fórmula de fitness se ha ponderado algunos parámetros utilizando los valores α, β, γ , para controlar la importancia que le damos a cada uno de ellos dentro de la evaluación, y se realiza una sumatoria por cada uno de los vehículos que han completado el viaje.

Esta función de fitness se caracteriza por proporcionar parámetros lo más realistas posibles, con base en modelos estadísticos y a un análisis para cada vehículo en cada unidad de tiempo denominada timeStep.

La combinación de las dos funciones de evaluación nos permite generar soluciones equilibrando la rapidez de cómputo del algoritmo con la eficiencia de las mismas.

3.7. Operadores Genéticos Propuestos.

Dentro de lo revisado acerca de los algoritmos genéticos se ha elaborado un diseño que se plantea con los siguientes operadores genéticos y la estructura que se mostrará a continuación.



3.7.1. Selección.

Para esta parte de nuestro algoritmo genético hemos optado por aplicar una selección por regla de la ruleta, a través del siguiente proceso:

- **Suma Total.** Calcular la suma total acumulada de los fitness de todos los individuos de la población actual.
- **Elegir un número aleatorio r.** Generar un número aleatorio entre 0 y la Suma Total.
- **Recorrer.** Recorrer la población acumulando nuevamente los fitness. Cuando la suma que se lleve sea mayor o igual a r seleccionamos el individuo donde se vaya recorriendo.

En la figura 25 podemos observar que las soluciones con menor fitness obtienen mayor probabilidad de ser elegidas, ya que estamos minimizando la función.



Figura 25: Gráfica de Selección por método de ruleta

Además, en la evaluación con la segunda función de fitness se ha combinado con Elitismo para garantizar que los individuos más aptos pasen a la siguiente generación.

3.7.2. Cruce.

La reproducción se lleva a cabo mediante el cruce en un punto, que consiste en seleccionar un punto aleatoriamente, a continuación, la parte anterior al punto se toma de un cromosoma padre, y se completa con la parte posterior al punto del otro cromosoma padre, simulando el intercambio de material

genético para de esta forma obtener dos nuevos individuos que mantienen ciertas características de los padres, como se observa en la figura 26.

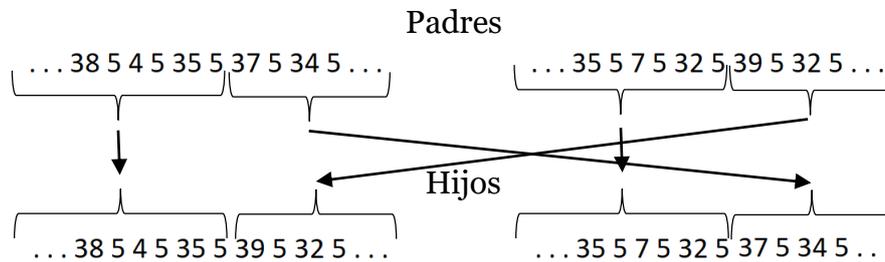


Figura 26: Operación de Cruce

Hay que mencionar que este operador ha sido seleccionado porque nos permite generar soluciones que varían una sección completa de los grupos semafóricos de la topología utilizada, permitiendo conservar las características de los padres.

3.7.3. Mutación.

La mutación consiste en la modificación aleatoria de algunos genes. En nuestra representación se seleccionará de forma aleatoria los n posibles genes a cambiar, en donde n es el número de controladores semafóricos existentes en la topología de las calles. Una vez seleccionados los genes a variar, cada uno de estos genes tienen una probabilidad muy pequeña de mutar, se elige un gen aleatoriamente y se le asigna un valor aleatorio sumando o restando valores dentro de un rango determinado, esto con el objetivo de que cada grupo semafórico tenga la probabilidad de que alguna de sus fases varíe en el proceso de mutación.

3.7.4. Algoritmo Genético.

Una vez que se han definido las características de la topología de la red como dato de entrada, así como el número de vehículos al momento de generar las rutas para la simulación, definiendo un modelo que describirá el comportamiento de los mismos.

Luego de establecer los operadores genéticos, utilizaremos la forma básica de un algoritmo genético convencional, partiendo de generar una población



inicial a partir de la solución creada por el simulador SUMO. A continuación, se evalúa cada uno de los individuos mediante nuestra función fitness que hará una evaluación rápida de toda la población, posteriormente sobre una población ya reducida, se realizará otra evaluación con el simulador SUMO que será más costosa a nivel computacional. La función de evaluación mediante SUMO es más precisa que la primera función utilizada, debido a la cantidad de parámetros que maneja. Finalmente, establecemos una condición de término basada en el número de generaciones que nos permitan llegar a la convergencia de las soluciones, de no cumplirse, permitirá la ejecución de los operadores genéticos de acuerdo a lo que se muestra en la figura 27.

Como dato de salida tenemos un vector que contiene la duración de las fases de los ciclos de los semáforos, los mismos que podrán ser utilizados para generar un programa y ser simulados mediante SUMO.

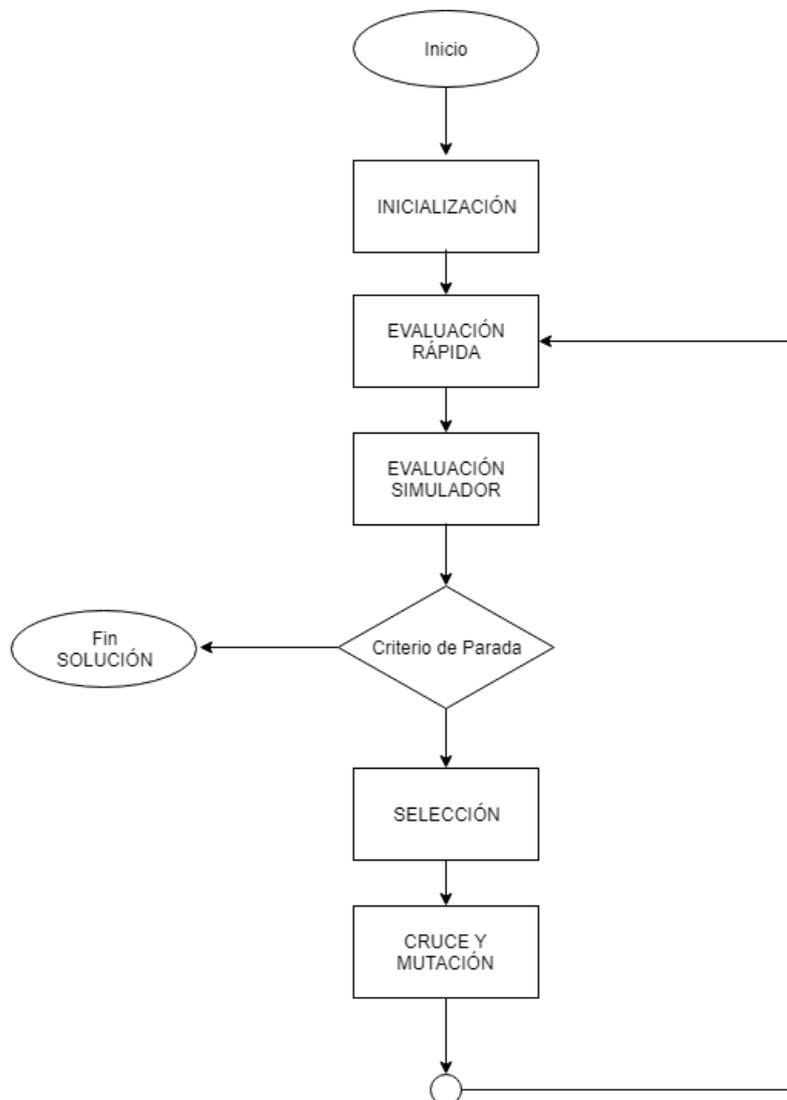


Figura 27: Diagrama de Flujo del Algoritmo Genético

4. Experimentación y Pruebas

La etapa de experimentación y pruebas nos permite establecer algunos parámetros del algoritmo Genético que, de otra forma, es muy complicado establecer a priori. Además, pretendemos realizar la comparación entre los resultados obtenidos con SUMO como única función de evaluación y la combinación de SUMO conjuntamente con nuestra función de evaluación.

4.1. Escenario.

Para nuestro trabajo el escenario corresponde a un área urbana de la ciudad de Valencia, extraída mediante los mapas de OpenStreetMap, la misma que abarca un área de 4600 m². Además, se ha establecido la cantidad de vehículos que formarán parte de la simulación a través de los datos proporcionado por la página del Ayuntamiento de Valencia que nos proporciona la intensidad de tráfico para los diferentes sectores analizados (Valencia A. d., 2017).

El mapa seleccionado está formado por 141 intersecciones como podemos apreciar en la figura 28, de las cuales 29 corresponden a nodos terminales que no se toman en cuenta. La cantidad de controladores semafóricos para esta topología es de 26, con 158 semáforos y un total de 106 fases que corresponden a la longitud del cromosoma para nuestro algoritmo genético.

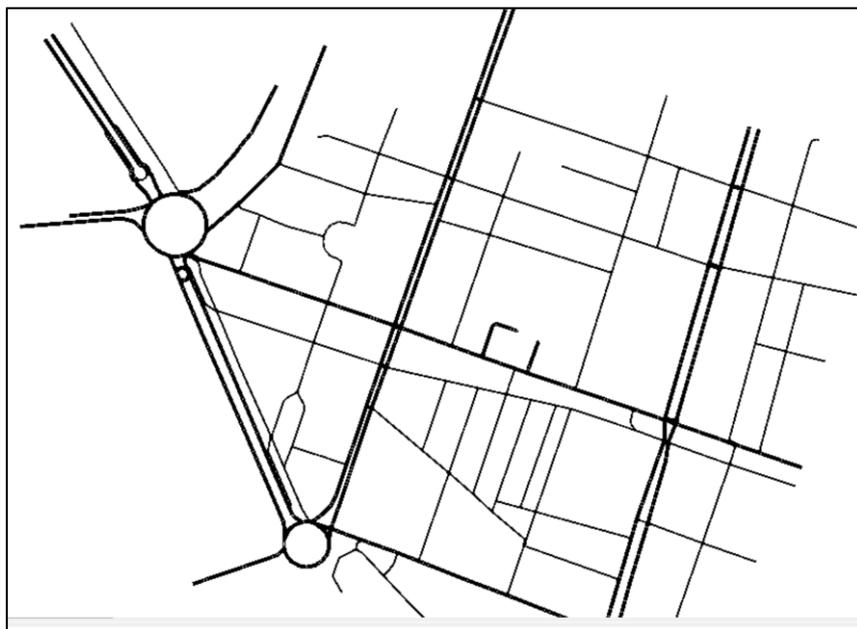


Figura 28: Mapa de las calles utilizadas en los experimentos

Las características de la zona seleccionada corresponden a un área urbana, con presencia de rotondas e intersecciones con tres o cuatro accesos. Las principales avenidas que podemos mencionar son Av. del Puerto, Av. de Aragón, Av. del Cardenal Benloch y las calles del Dr. Manuel Candela, Calle de Rodríguez Cepeda, Calle de las Islas Canarias, entre otras.

Las rutas de los viajes de cada vehículo han sido generadas de forma aleatoria, mediante la herramienta *randomTrips.py*, como se mencionó en la sección de extracción de datos. El tiempo de simulación se ha establecido en 500 segundos, que será el tiempo máximo que puede tardar un vehículo desde su origen a su destino. Los parámetros del escenario los podemos observar en la tabla 2.

Tabla 2: Parámetros de Simulación

Parámetro	Valor
Tiempo de Simulación	500 s.
Área de Simulación	4600 m ²
Número de Vehículos	100
Velocidad Máxima de Vehículo	50 km/h
Número de Intersecciones	141

El algoritmo Genético ha sido implementado en Python 3.6 y se ha utilizado el simulador SUMO para la parte de extracción de datos de la topología para implementar las funciones de evaluación.

4.2. Parámetros del Algoritmo Genético.

Para fijar cada parámetro del Algoritmo Genético, se ejecutan 10 simulaciones con los valores seleccionados y calculamos un promedio del valor de la función de fitness obtenida.

Se eligió como primer parámetro de evaluación el tamaño de la población, lo que nos permite ir incrementando el número de individuos, con el objetivo de minimizar la función de fitness, hasta llegar a un valor aceptable de tiempo de cómputo.

4.2.1. Tamaño de la población inicial.

Para las primeras pruebas ejecutaremos el algoritmo genético, variando el tamaño de la población inicial desde 10 a 300, con los parámetros que se muestran en la tabla 3.

Tabla 3: Parámetros Variando Tamaño de la Población Inicial

PARÁMETRO	VALOR
Tamaño población inicial	10-300
Probabilidad de Mutación	0.1%
Generaciones	100
Cruce	1 punto
Selección	Ruleta

En la figura 29 podemos apreciar que el tamaño de la población influye directamente con el valor de fitness obtenido, teniendo en cuenta que, para tamaños de población con pocos individuos, existe muy poca variabilidad que permita evolucionar al algoritmo genético, por lo tanto, se ha elegido un tamaño de población de 200 individuos, que permite tener una diversidad aceptable sin incrementar excesivamente el tiempo de procesamiento.

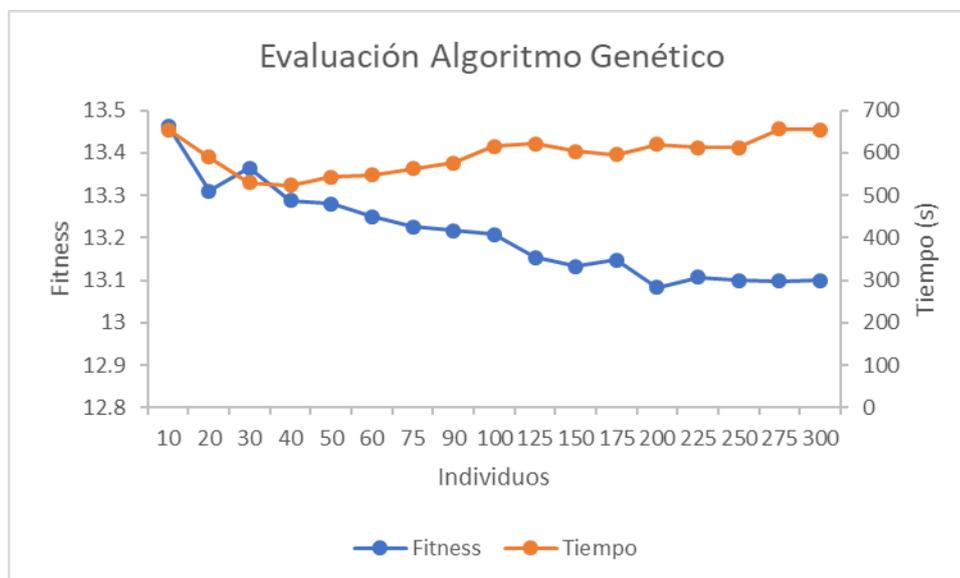


Figura 29: Algoritmo Genético Variando el Tamaño de la Población

4.2.2. Número de Generaciones.

En este experimento se tomará en cuenta el tamaño de la población inicial encontrado en el apartado anterior, y se variará el número de generaciones entre 10 y 300 para determinar un valor óptimo, teniendo en cuenta los parámetros de acuerdo a la tabla 4.

Tabla 4: Parámetros Variando el Número de Generaciones

PARÁMETRO	VALOR
Tamaño población inicial	200
Probabilidad de Mutación	0.1%
Generaciones	10-300



Cruce	1 punto
Selección	Ruleta

En este punto del algoritmo se busca un valor para el número de generaciones que nos permita establecer un fitness óptimo teniendo en cuenta las generaciones a partir de las cuales las soluciones comienzan a converger.

Como podemos apreciar en la figura 30, a medida que se incrementan las generaciones, nos encontramos con un punto en donde el fitness de las soluciones empieza a converger. Sin embargo, el tiempo de procesamiento continúa creciendo, lo que nos permite tomar una decisión respecto a cuantas generaciones se consideran un punto intermedio entre un fitness y un tiempo de procesamiento aceptables.

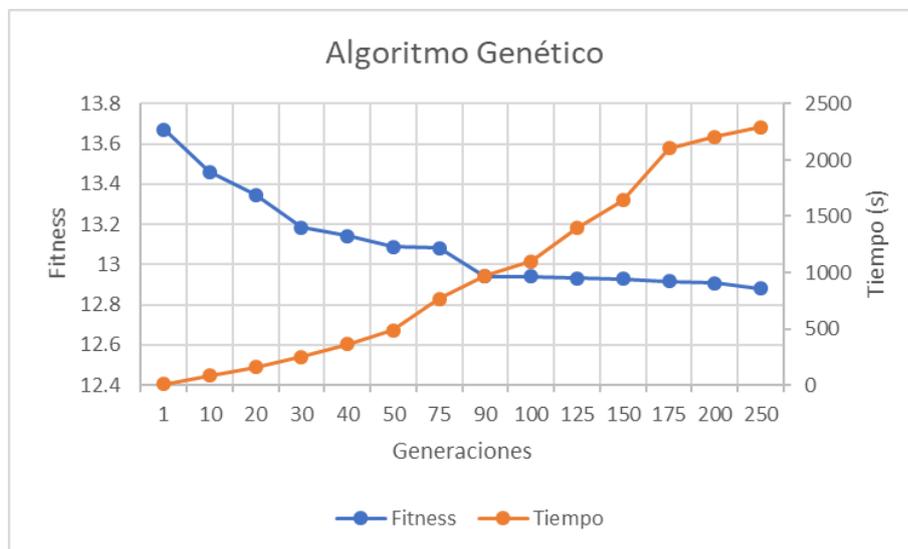


Figura 30: Algoritmo Genético Variando el número de Generaciones

4.2.3. Probabilidad de Mutación.

De acuerdo a los resultados obtenidos en los experimentos anteriores fijamos los parámetros como se muestran en la tabla 5 y variamos la probabilidad de mutación hasta encontrar un punto que minimice el fitness.

Tabla 5: Parámetros Variando la Probabilidad de Mutación

PARÁMETRO	VALOR
Tamaño población inicial	200
Probabilidad de Mutación	0.005-0.1%
Generaciones	90
Cruce	1 punto
Selección	Ruleta

En la figura 31 podemos observar la importancia de seleccionar una probabilidad de mutación apropiada, que en este caso se fija en 0.06, con lo cual podemos fijar todos los parámetros obtenidos anteriormente y realizar las pruebas finales.

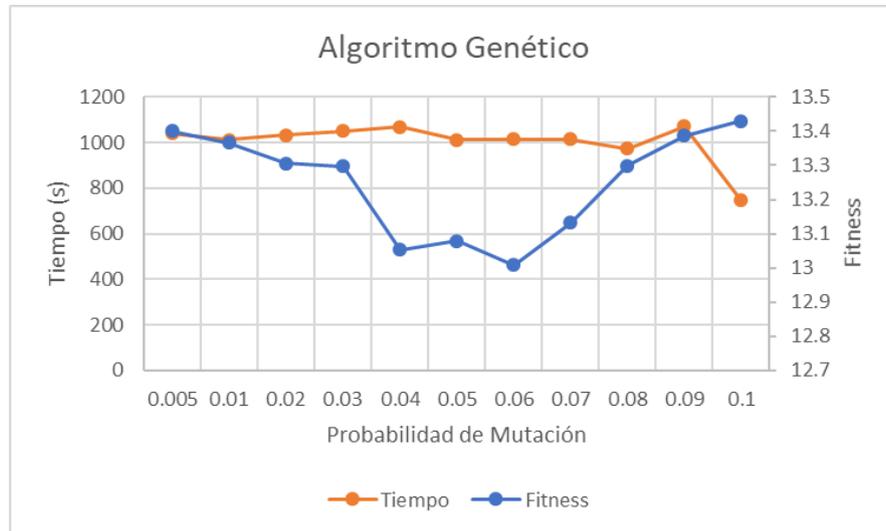


Figura 31: Algoritmo Genético Variando la Probabilidad de Mutación

4.3. Evaluación de Soluciones.

Para realizar la evaluación se han seleccionado los parámetros con los que se han obtenido mejores resultados, los mismos que los podemos en la tabla 6.

Tabla 6: Parámetros seleccionados para el Algoritmo Genético.

PARÁMETRO	VALOR
Tamaño población inicial	200
Probabilidad de Mutación	0.06%
Generaciones	90
Cruce	1 punto
Selección	Ruleta

En la figura 32 se muestra cinco diferentes ejecuciones del algoritmo genético con los parámetros seleccionados anteriormente y podemos analizar que a partir de la generación 70 todas las soluciones empiezan a converger.

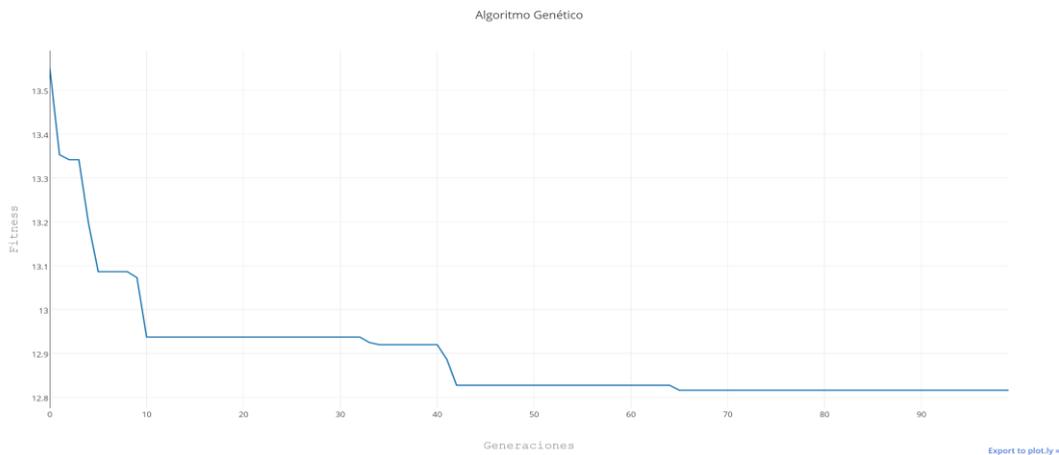


Figura 32: Evolución del Algoritmo Genético con los parámetros seleccionados.

En primer lugar, se ha evaluado las soluciones proporcionadas por 10 ejecuciones del algoritmo genético y se ha obtenido los resultados que presentamos en la tabla 7.

Tabla 7: Evaluación soluciones Algoritmo Genético

PARÁMETRO	VALOR
Fitness	12.91
Tiempo de Espera	23.48 segundos
Duración de Viaje	84.15 segundos
Vehículos Viaje Completo	84/100
Emisión de CO2	372422.0287 g
Consumo de Combustible	160.08 ml
Tiempo de Cómputo	9.70 segundos por generación

A continuación, se han evaluado 10 soluciones generadas aleatoriamente y se han obtenido el promedio de los resultados que se muestran a continuación en la tabla 8.

Tabla 8: Evaluación soluciones aleatorias.

PARÁMETRO	VALOR
Fitness	20.51
Tiempo de Espera	53.87 segundos
Duración de Viaje	136.58 segundos
Vehículos Viaje Completo	84/100
Emisión de CO2	491466.7872 g
Consumo de Combustible	222.46 ml
Tiempo de Cómputo	3.99 segundos

La tercera prueba se ha realizado con la solución generada por el simulador SUMO y su algoritmo SCPG y se han obtenido los resultados mostrados en la tabla 9.

Tabla 9: Evaluación solución SCPG

PARÁMETRO	VALOR
Fitness	19.90
Tiempo de Espera	35.12 segundos
Duración de Viaje	115.43 segundos
Vehículos Viaje Completo	84/100
Emisión de CO2	470064.2701 g
Consumo de Combustible	202.06 ml
Tiempo de Cómputo	0.8 segundos

Finalmente se realizó una evaluación del algoritmo genético solamente con el uso de SUMO como función de evaluación, cuyos resultados mostramos en la tabla 10.

Tabla 10: Evaluación de Algoritmo Genético con función de Evaluación SUMO

PARÁMETRO	VALOR
Fitness	13.104
Tiempo de Espera	23.51
Duración de Viaje	85.37
Vehículos Viaje Completo	83
Emisión de CO2	313266.8837 g
Consumo de Combustible	160.30 ml
Tiempo de Cómputo	97.81 segundos por generación

4.4 Análisis de Resultados.

Como podemos apreciar en las tablas de evaluación presentadas anteriormente nuestro algoritmo genético mejora el valor de fitness obtenido respecto a las soluciones generadas de forma aleatoria y al algoritmo utilizado por el simulador SUMO para generar los programas de duración de los ciclos de los semáforos.

También se ha realizado una evaluación de las soluciones obtenidas con el algoritmo genético, para analizar los parámetros que formaban parte de la función de evaluación y observar su comportamiento de forma individual.



4.4.1. Fitness

En primer lugar, en la figura 33 se muestra una comparación de los valores de fitness obtenidos mediante las diferentes pruebas realizadas. Aquí podemos observar que el algoritmo genético implementado con nuestra función de fitness combinada con la de SUMO minimiza los valores de fitness en comparación con los obtenidos mediante la solución generada por el simulador SUMO, la solución generada aleatoriamente y el algoritmo genético utilizando como función de fitness, los datos obtenidos por el simulador SUMO.

Cabe destacar que para el último caso se utilizaron 50 generaciones en comparación con las 100 utilizadas para el Algoritmo Genético con la función de fitness con los datos generados por el simulador en combinación con la implementada mediante las rutas de los vehículos y los tiempos de espera, esto debido al alto costo computacional.

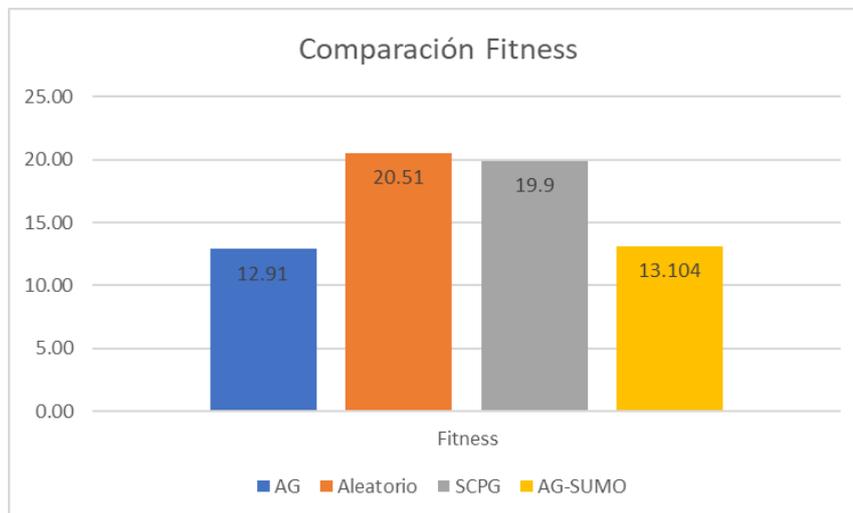


Figura 33: Comparación de fitness de las soluciones generadas por diferentes técnicas.

4.4.2. Duración del Viaje

En este caso el parámetro a evaluar es la duración del tiempo promedio que le toma a un vehículo completar su viaje, para lo cual en todas las pruebas se utilizaron las mismas rutas generadas aleatoriamente. En la figura 34 podemos observar nuevamente que el Algoritmo genético es la técnica que consigue minimizar el tiempo promedio empleado para completar una ruta específica.

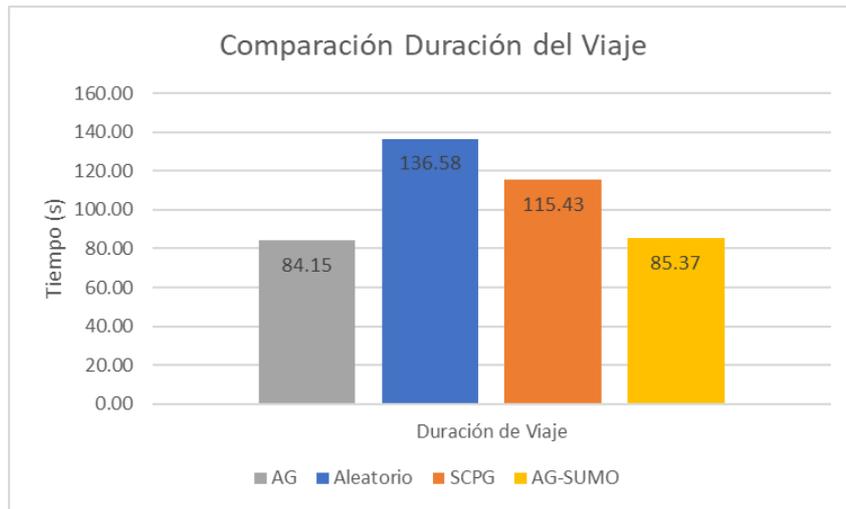


Figura 34: Comparación Duración del Viaje

4.4.3. Tiempo de Espera.

En esta sección se compara el tiempo de espera de los vehículos frente a los semáforos, en donde podemos apreciar una gran diferencia existente entre nuestro algoritmo genético y las soluciones aleatoria y SCPG, debido a que la función de evaluación rápida se basa precisamente en el tiempo que permanecen detenidos los vehículos frente a los semáforos en color rojo. Podemos observar la comparación en la figura 35.

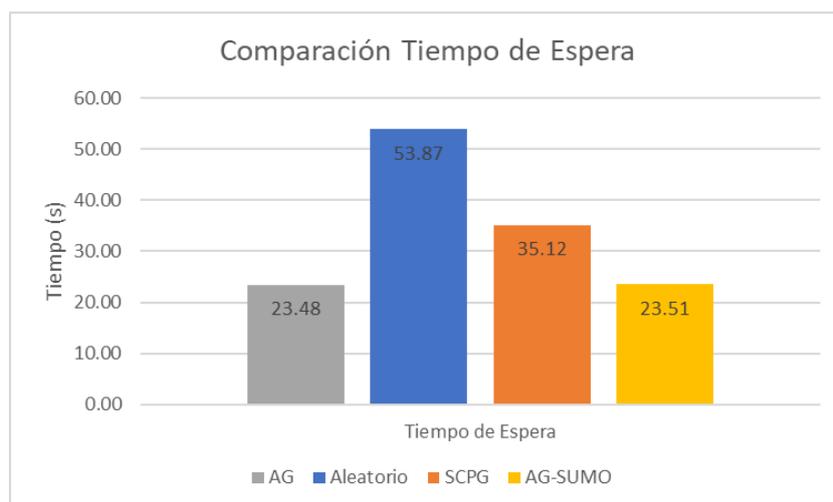


Figura 35: Comparación tiempo de espera.

4.4.4. Emisión de CO₂.

Uno de los parámetros más importantes y en torno al cual giran los objetivos de la optimización de los ciclos de los semáforos es la contaminación, que en este caso se ha tomado el CO₂ como parámetro de evaluación, en este caso la figura 36 nos muestra que el algoritmo genético con función de evaluación mediante SUMO es el que logra minimizar la cantidad de CO₂ emitido. Sin embargo, el Algoritmo Genético implementado obtiene valores aceptables de emisión de CO₂, menores que los obtenidos con soluciones aleatorias y con el algoritmo SCPG.

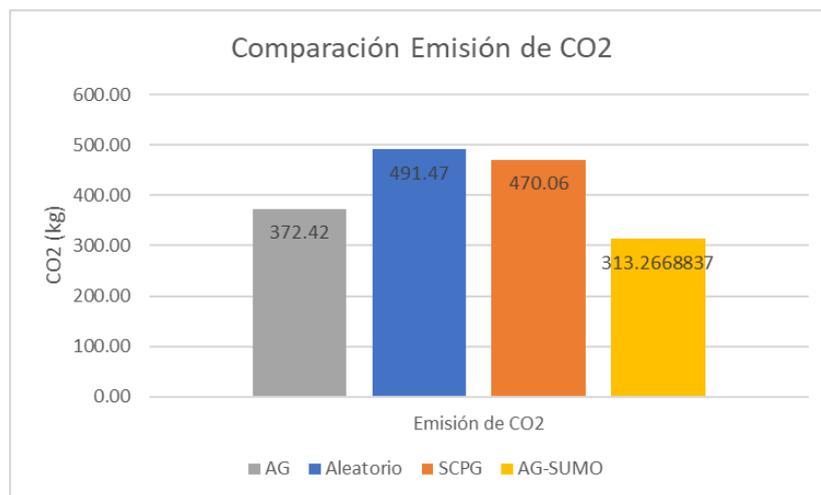


Figura 36: Comparación de Emisión de CO₂.

4.4.5. Consumo de Combustible.

Relacionado directamente con la emisión de gases tenemos el consumo de combustible cuya comparación podemos observar en la figura 37, en donde destacan que las soluciones generadas mediante los algoritmos genéticos, minimizan el consumo de combustible comparando respecto a las soluciones generadas por el algoritmo SCPG y las soluciones aleatorias.

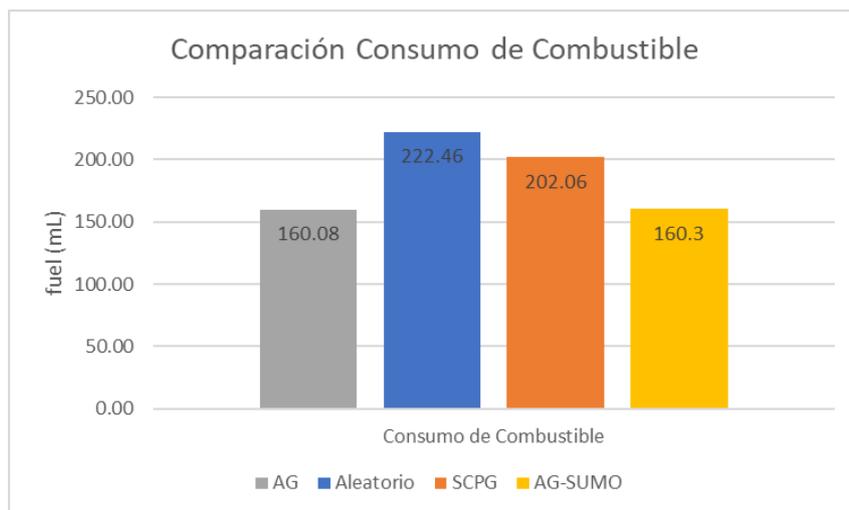


Figura 37: Comparación Consumo de Combustible.

4.4.6. Tiempo de Cómputo.

En este apartado se realiza una comparación del tiempo de cómputo entre las soluciones generadas por el Algoritmo Genético implementado con las dos funciones de evaluación y el Algoritmo Genético que utiliza la función de evaluación con parámetros obtenidos directamente del Simulador SUMO. Como observamos en las comparaciones anteriores, los parámetros de evaluación son bastante similares, al ser las soluciones obtenidas mediante un algoritmo genético.

En la figura 38 observamos los tiempos de cómputo obtenidos en el cálculo de una generación para cada caso, en donde se aprecia la gran diferencia existente entre utilizar el simulador para evaluar cada una de las soluciones obtenidas y el tiempo requerido por las soluciones generadas por el algoritmo genético que en una primera instancia utiliza una función de evaluación rápida basada en los tiempos de espera y luego sobre una cantidad menor de individuos, evaluar haciendo uso del simulador.

Es necesario destacar que el proceso que realiza el simulador consta de una gran variedad de cálculos en cada timestep o intervalo de simulación, por lo que es comprensible la diferencia existente.

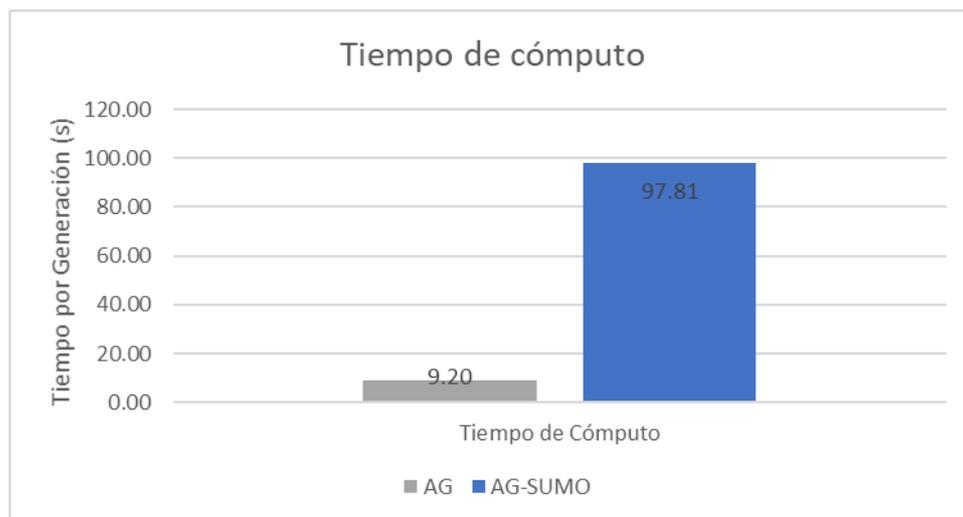


Figura 38: Tiempo de cómputo.

5. Conclusiones y Trabajos Futuros.

5.1. Conclusiones.

A través del desarrollo del presente trabajo con la aplicación de técnicas metaheurísticas, específicamente un algoritmo genético para la optimización de la duración de los ciclos de los semáforos podemos obtener las siguientes conclusiones:

- El costo computacional de los simuladores es una de las desventajas al momento de su aplicación, sobre todo en problemas de optimización como nuestro caso en donde la evaluación de miles de soluciones nos llevó a plantearnos una alternativa que es una función de evaluación menos estricta. Sin embargo, que presenta resultados aceptables en combinación con los datos obtenidos del simulador.
- La aplicación de un algoritmo genético nos permite utilizar funciones multi-objetivo que como en nuestro caso minimicen el conjunto de parámetros seleccionados. Además de que la representación elegida, permite adaptar las soluciones a cualquier entorno propuesto por las diferentes topologías de las calles de las ciudades.
- La función de fitness implementada se basa en los tiempos de espera de los semáforos, permitiendo mantener una relación con el objetivo de la optimización, que en nuestro caso es la duración de los ciclos de los semáforos. Sin embargo, si deseamos obtener valores de fitness más óptimos, lo adecuado es una combinación de las dos funciones de evaluación, lo que garantiza rapidez y eficiencia.
- El análisis de los parámetros que forman la función de evaluación nos demuestra que las soluciones generadas por nuestro algoritmo genético minimizan parámetros muy importantes como duración de tiempo de viaje, tiempos de espera, emisión de CO₂ y consumo de combustible.

5.2. Trabajos Futuros.

La investigación dentro de lo que se refiere a la optimización del tráfico es sumamente extensa, y el desarrollo del presente trabajo nos permite plantear las siguientes propuestas:



- Análisis de los modelos utilizados para la emisión de gases y consumo de combustible con el objetivo de obtener funciones de evaluación más precisas, que se adapten a entornos específicos como son, arquitectura de las ciudades y los demás factores que contribuyen al incremento de gases contaminantes.
- Hacer uso de las rutas generadas por el simulador SUMO y generar programas de optimización de la duración de los ciclos de los semáforos, y la generación de rutas que permitan que los vehículos vayan desde su origen a su destino de forma eficiente.
- Tomar en cuenta las rutas generadas por los vehículos autónomos y sincronizar con la optimización de los ciclos de los semáforos para generar programas de tráfico más eficientes.
- Aplicación de redes neuronales para la generación de programas de control de tráfico, que se adapten a las diferentes topologías de las ciudades, con base en las lecturas de los sensores de tráfico existentes.
- Uso de reportes históricos de tráfico para determinar patrones de comportamiento de conducción, que permitan establecer programas de optimización de tráfico que se ajusten directamente a las necesidades de cada ciudad.
- Aplicación de técnicas metaheurísticas para la optimización de la ubicación de señales de tráfico en las ciudades.

Bibliografía

- Eberhart, R., & Kennedy, J. (1995). A new Optimizer Using Particle Swarm Theory. *Proceedings of the Sixth International Symposium on*.
- Alba, E. F.-N. (2014). Optimising traffic lights with metaheuristics: Reduction of car emissions and consumption. *2014 International Joint Conference on Neural Networks (IJCNN)*.
- Badar, A., Umre, B. S., & Junghare, A. S. (2014). Study of Artificial Intelligence Optimization Techniques applied to Active Power Loss Minimization. *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*.
- Berzal, F. (26 de 07 de 2017). *Algoritmos Genéticos*. Obtenido de <http://elvex.ugr.es/decsai/iaio/slides/G2%20Genetic%20Algorithms.pdf>
- Castrovation. (08 de 2017). *Castrovation Blog*. Obtenido de <http://castrovation.blogspot.com.es/2016/01/algoritmo-genetico.html>
- De Schutter, B., & De Moor, B. (1998). Optimal Traffic Light Control for a Single Intersection. *European Journal of Control*.
- DGT. (10 de 08 de 2017). *Dirección General de Tráfico*. Obtenido de El Impacto Medioambiental del Tráfico: http://www.dgt.es/PEVI/documentos/catalogo_recurso/didacticos/did_adultas/impacto.pdf
- DLR, C. A. (10 de 08 de 2017). *Instituto de Sistemas de Transporte, SUMO (Simulator of Urban MObility)*. Obtenido de http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/
- Dorigo, M. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B*.
- García Nieto, J., & Alba, E. (2013). *Metaheurísticas para Smart Mobility: Reducción de Emisiones y Consumo de Carburantes en el Tráfico Urbano*.
- García-Nieto, J., Olivera, A. C., & Alba, E. (2013). Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 17*.
- Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*.
- Goldberg, D. E. (1985). *Genetic Algorithms in Search, Optimization and Machine*. Addison-Wesley.



- Holland, J. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.
- Jones, T. (2005). *AI Application Programming. Second Edition*. Charles River Media.
- Lawrence, D. (1985). Applying Adaptive Algorithms to Epistatic Domains. *International Joint Conference on Artificial Intelligence*.
- Lim, G., Kang, J., & Hong, Y. (2001). The optimization of traffic signal light using artificial intelligence. *Fuzzy Systems, 2001. The 10th IEEE International Conference on*.
- Mateos, A. (2004). *Algoritmos Evolutivos y Algoritmos Genéticos*. Madrid: Universidad Carlos III de Madrid.
- Mazek, P., Mazek, J., Frantik, P., Fujdiak, R., Aleksandr, O., Hosek, J., . . . Misurec, J. (2016). A Harmonized Perspective on Transportation Management in Smart Cities: The Novel IoT-Driven Environment for Road Traffic Modeling. *Sensors*.
- México, F. (10 de 08 de 2017). *Forbes México*. Obtenido de <https://www.forbes.com.mx/cdmx-mantiene-el-trono-de-la-ciudad-con-peor-trafico-del-mundo/>
- OpenStreetMap, F. F. (07 de 2017). *OpenStreetMap*.
- Rahbari, D. (2014). Help the Genetic Algorithm to Minimize the Urban Traffic on Intersections. *International Journal of Research in Computer Science*.
- Rodriguez-Piñero, P. (2003). Introducción a los Algoritmos Genéticos y sus Aplicaciones. *Universidad Rey Juan Carlos. Servicio de Publicaciones*.
- Sattari, M. R., Malakooti, H., Jalooli, A., & Noor, R. M. (2014). A Dynamic Vehicular Traffic Control Using Ant Colony and Traffic Light Optimization . *In Advances in Systems Science*.
- TomTom. (12 de 08 de 2017). *Full Ranking Traffic Index*. Obtenido de TomTom Traffic Index. Measuring Congestion Worldwide.
- Tráfico, D. D. (02 de 2015). *Dirección General de Tráfico*. Obtenido de <http://www.dgt.es/es/prensa/notas-de-prensa/2015/20150219-La-DGT-pone-en-marcha-nuevas-medidas-para-la-gestion-de-la-velocidad.shtml>
- Tubaishat, M., Shang, Y., & Shi, H. (2007). Adaptive traffic light control with wireless sensor networks. *In Consumer Communications and Networking Conference, 2007*.

Valencia, A. d. (07 de 2017). *Ayuntamiento de Valencia. Movilidad*. Obtenido de <http://www.ayto-valencia.es/ayuntamiento/trafico.nsf/fCategoriaVista?readForm&Vista=vCategoriaDescargas&Categoria=Sincat&titulo=Descargas&lang=1&nivel=7&expand=3&bdorigen=&idApoyo=>

Valencia, V. G. (2000). *Principios sobre semáforos*. Medellín: Universidad Nacional de Colombia.

Wiering, M. (2000). Multi-Agent Reinforcement Learning for Traffic Light Control. *In Machine Learning: Proceedings of the Seventeenth International Conference* .