



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño y aplicación de técnicas metaheurísticas para el control de tráfico

Trabajo Fin de Grado

Grado en Ingeniería Informática

Resumen

El incremento de la población a nivel mundial y su concentración en áreas urbanas causa múltiples problemas que van desde las pérdidas económicas hasta la disminución de la calidad de vida. Respecto del primero, es necesario reducir los tiempos de viaje de forma que se aumente la productividad y se reduzcan los costes sanitarios derivados de los tratamientos de los problemas causados por la contaminación, el estrés y los accidentes de tráfico. En cuanto al segundo, es importante continuar con la reducción de las emisiones de gases contaminantes y el ruido, principalmente, que ya se está llevando a cabo en las principales ciudades del mundo.

Para ello se propone una herramienta que permita la minimización de los tiempos de viaje y las emisiones, actuando sobre las fases semafóricas y los límites de velocidad de las calles. Dados un escenario y una cantidad de vehículos en circulación, mediante un algoritmo genético se optimizarán estos valores, tanto en fase de estudio de un proyecto futuro como en escenarios ya existentes, importando el mapa de la zona a estudiar. Se empleará el simulador de tráfico SUMO para evaluar las distintas soluciones y elegir la que mejor se adapte a las condiciones del escenario de nuestro interés.

Palabras clave: algoritmo genético, optimización, tráfico, semáforo, límite de velocidad, simulación, SUMO.

Abstract

The increase in the world's population and its concentration in urban areas causes multiple problems ranging from economic losses to a decrease in the quality of life. Regarding the first one, it is necessary to reduce travel times to increase productivity and reduce health costs derived from the treatment of problems caused by pollution, stress and traffic accidents. About the second one, it is important to continue with the reduction of polluting gas emissions and noise, mainly, which is already being carried out in the main cities of the world.

For all of this, a tool is proposed that allows the minimization of travel times and emissions, acting on the traffic light phases, and the street speed limits. Given a scenario and a number of vehicles in circulation, these values will be optimized by means of a genetic algorithm, both in the study phase of a future project and in already existing scenarios, importing the map of the area to be studied. The SUMO traffic simulator will be used to evaluate the different solutions and choose the one that best suits the conditions of the scenario of our interest.

Keywords: genetic algorithm, optimization, traffic, traffic lights, speed limit, simulation, SUMO.

Tabla de contenidos

1.	Introducción	2
1.1	Motivación	2
1.2	Justificación	5
1.3	Objetivos	6
1.4	Estructura	7
2.	Estado del arte	9
2.1	Simuladores de tráfico	9
2.2	<i>Simulation of Urban Mobility</i>	11
2.3	Trabajos previos	21
2.4	Trabajo actual	22
3.	Marco teórico	23
3.1	Semaforización	23
3.2	Gestión de la velocidad	26
3.3	Algoritmo genético	27
4.	Metodología	31
4.1	Definición	33
4.2	Escenarios	40
4.3	Generación de rutas	41
4.4	Archivos de configuración	41
4.5	Solución base	42
4.6	Funciones	42
4.7	Generación y evaluación de la población inicial	47
4.8	Evolución de la población	47
4.9	Gráfico de evolución	48
5.	Resultados	50
5.1	Hiperparámetros	51
5.2	Comparaciones	58

6.	Conclusiones	64
7.	Trabajos futuros.....	66
	7.1 Paradoja de <i>Braess</i>	66
	7.2 Paralelización	67
	7.3 Límites de velocidad	68
	7.4 Otras mejoras	68
8.	Referencias	70

Índice de figuras

Figura 01 – Evolución de la población mundial.....	2
Figura 02 – Porcentaje de la población en zonas urbanas	2
Figura 03 – Evolución de las emisiones de CO.....	3
Figura 04 – Evolución de las emisiones de NO _x	4
Figura 05 – Variación de la velocidad media con el límite de velocidad.....	5
Figura 06 – Definición xml de una calle	13
Figura 07 – Definición xml de un semáforo.....	14
Figura 08.a – Detalle de la fase 1 de un semáforo	14
Figura 08.b – Detalle de la fase 2 de un semáforo	15
Figura 08.c – Detalle de la fase 3 de un semáforo	15
Figura 08.d – Detalle de la fase 4 de un semáforo	16
Figura 09 – Definición xml de la configuración de la simulación	20
Figura 10 – Definición xml de las estadísticas.....	21
Figura 11 – N-VI a su paso por San Rafael (Segovia)	24
Figura 12 – Ciclo y fases de un semáforo	25
Figura 13 – Señalización específica	26
Figura 14 – Señalización variable	26
Figura 15 – Cromosoma codificando tiempos de fases de semáforos.....	28
Figura 16 – Cromosoma simple codificado en binario	28
Figura 17 – Población de individuos representados por cromosomas codificados en binario. ...	28
Figura 18 – Operación de cruce entre dos cromosomas codificados en binario	29
Figura 19 – Mutación aplicada a un cromosoma codificado en binario.....	30
Figura 20 – Flujo de ejecución	31
Figura 21 – Datos generales de la simulación.....	33
Figura 22 – Aviso: posible ralentización en la ejecución.....	34
Figura 23 – Aviso: parámetros de cálculo.....	34
Figura 24 – Aviso: parámetros de optimización	35
Figura 25 – Aviso: límites de velocidad.....	35
Figura 26.a – Red mallada.....	36
Figura 26.b – Red radial.....	36
Figura 26.c – Red aleatoria	37
Figura 26.d – Red importada de OpenStreetMap.....	37

Figura 27 – Seleccionar archivo de mapas.....	38
Figura 28 – Datos del algoritmo genético	38
Figura 29 – Aviso: suma de porcentajes de selección.....	39
Figura 30 – Definición xml de la configuración de la simulación	41
Figura 31 – Definición xml de las estadísticas.....	42
Figura 32 – Ejemplo de cruce	43
Figura 33 – Ejemplo de mutación	43
Figura 34 – Evolución de la población.....	47
Figura 35 – Gráfico de evolución de la puntuación	48
Figura 36 – Escenario para la obtención de hiperparámetros.....	50
Figura 37 – Puntuación según probabilidad de cruce, mutación y creación	53
Figura 38.a – Puntuación: población = 50 soluciones.....	55
Figura 38.b – Puntuación: población = 100 soluciones.....	55
Figura 38.c – Puntuación: población = 150 soluciones.....	55
Figura 38.d – Puntuación: población = 200 soluciones.....	55
Figura 38.e – Puntuación: población = 250 soluciones.....	55
Figura 38.f – Puntuación: población = 300 soluciones	55
Figura 38.g – Puntuación: población = 350 soluciones.....	56
Figura 38.h – Puntuación: población = 400 soluciones.....	56
Figura 39 – Puntuación según tamaño de población.....	56
Figura 40 – Soluciones generadas según tamaño de población	57
Figura 41 – Puntuación vs. Soluciones generadas	57
Figura 42 – Comparación con red mallada	60
Figura 43 – Comparación con red radial	61
Figura 44 – Comparación con mayor congestión.....	62
Figura 45 – Paradoja de Braess	66
Figura 46 – Multitarea en SUMO	67
Figura 47 – Multitasking	68

Índice de tablas

Tabla 01 – Licencias de los simuladores de tráfico.....	11
Tabla 02 – Codificación de los colores de los semáforos	13
Tabla 03 – Intensidad mínima horaria, durante más de ocho horas al día	23
Tabla 04 – Intensidad mínima horaria, para demoras	24
Tabla 05 – Archivos de la aplicación	46
Tabla 06– Evolución de la puntuación	49
Tabla 07.a – Puntuación según probabilidades I.....	52
Tabla 07.b – Puntuación según probabilidades II	52
Tabla 07.c – Puntuación según probabilidades III	52
Tabla 07.d – Puntuación según probabilidades IV.....	54
Tabla 08 – Estadísticas de la solución base I	58
Tabla 09 – Estadísticas de la solución optimizada I.....	59
Tabla 10 – Estadísticas de la solución base II.....	60
Tabla 11 – Estadísticas de la solución optimizada II	61
Tabla 12 – Estadísticas de la solución base III.....	62
Tabla 13 – Estadísticas de la solución optimizada III	62
Tabla 14 – Estadísticas de la solución base IV	63
Tabla 15 – Estadísticas de la solución optimizada IV.....	63

1. Introducción

1.1 Motivación

Tras la II Guerra Mundial, y más concretamente a partir de la década de 1960, la población mundial se estima que se ha incrementado en más de 2,5 veces, hasta alcanzar los casi 8.000 millones de personas.

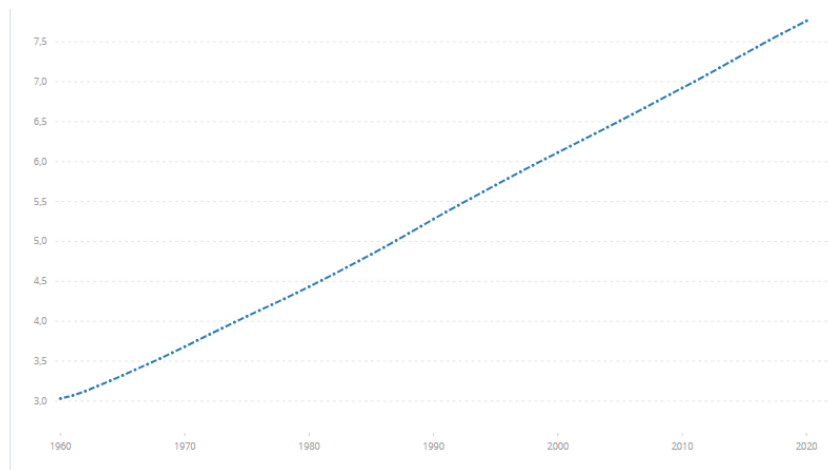


Figura 01 – Evolución de la población mundial

Fuente: Banco Mundial (1)

Este incremento en la población ha venido acompañado de un incremento en el porcentaje que vive en zonas urbanas.

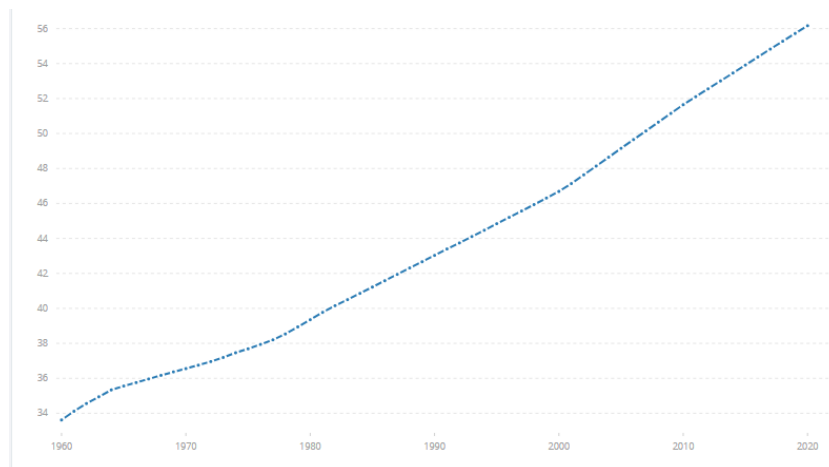


Figura 02 – Porcentaje de la población en zonas urbanas

Fuente: Banco Mundial (2)

La concatenación de estas dos circunstancias, junto con el aumento de vehículos a motor en circulación, conlleva un incremento en la cantidad de desplazamientos, concentrados en áreas

muy reducidas, que se realizan diariamente desde las zonas residenciales a los puestos de trabajo, en su mayoría. Esto, a su vez, tiene efectos negativos en la economía por la pérdida de productividad causada, por una parte, por la ralentización en los desplazamientos conforme aumenta la cantidad de población en las ciudades, y, por otra parte, por los efectos nocivos sobre la salud de las personas que viven en ellas, por las contaminaciones tanto del aire como acústica. A esto se añaden también los problemas de estrés, derivados de las congestiones de tráfico sufridas en los desplazamientos, a lo que se añade el coste sanitario para el tratamiento de los trastornos derivados. Otro coste, difícilmente cuantificable, pero no por ello menos importante, sería la pérdida de calidad de vida de la población de estas zonas respecto de la que vive fuera de las áreas urbanas.

Según el informe “2021 INRIX Global Traffic Scorecard” de la consultora INRIX (3) en el que se analiza el impacto económico de los atascos de tráfico, en donde Londres (148 horas), París (140 horas), Bruselas (134 horas), Moscú (108 horas) y Nueva York (102 horas) comprenden las cinco ciudades más congestionadas del mundo (con el tiempo en horas perdidas por habitante al año). Hay que tener en cuenta, según indica el propio estudio, que estos datos están afectados por la pandemia de SARS-CoV-2 y que, respecto del año 2019, son hasta un 37% inferiores en Estados Unidos, y un 36% menores en el caso del Reino Unido. Aun así, teniendo en cuenta estas reducciones, se estiman unas pérdidas económicas para el año 2021 de 53.000 millones de dólares y 8.000 millones de libras esterlinas, respectivamente. En España (4), la clasificación la encabezarían Barcelona (48 horas) y Madrid (39 horas), con un descenso del 39% y del 46% respecto al periodo inmediatamente anterior a la pandemia.

En cuanto a las emisiones de gases contaminantes, se ha hecho un esfuerzo importante en su control y reducción en las últimas décadas. Tomando como referencia la ciudad de Madrid (por ser la de mayor cantidad y densidad de población en España), se puede observar un importante descenso de estas emisiones en los últimos 30 años. Según el último informe “Inventario de emisiones de contaminantes a la atmósfera del Municipio de Madrid” elaborado por el Ayuntamiento (5), con datos hasta el año 2019, las emisiones que más se han reducido han sido: un 91% para el CO y un 53% para el NO_x.

A continuación, se observa cómo han evolucionado las emisiones de monóxido de carbono (CO):

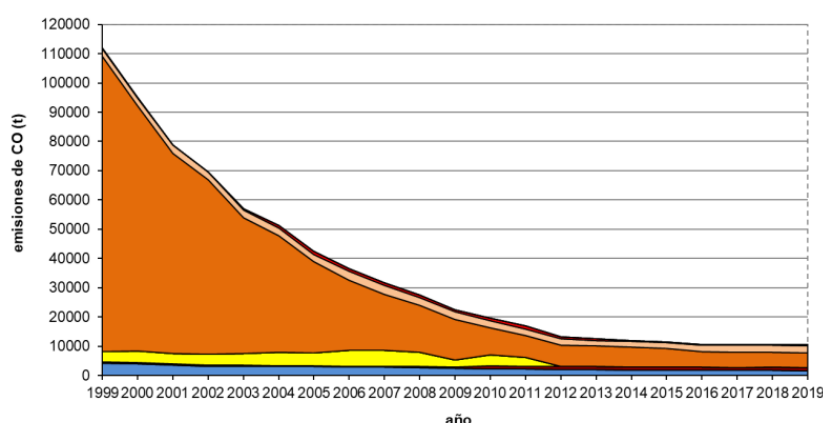


Figura 03 – Evolución de las emisiones de CO

Fuente: Excmo. Ayuntamiento de Madrid (5)

De igual forma, en la siguiente figura se muestra cómo se han reducido las emisiones de los óxidos de nitrógeno (NO_x) a lo largo de los últimos años:

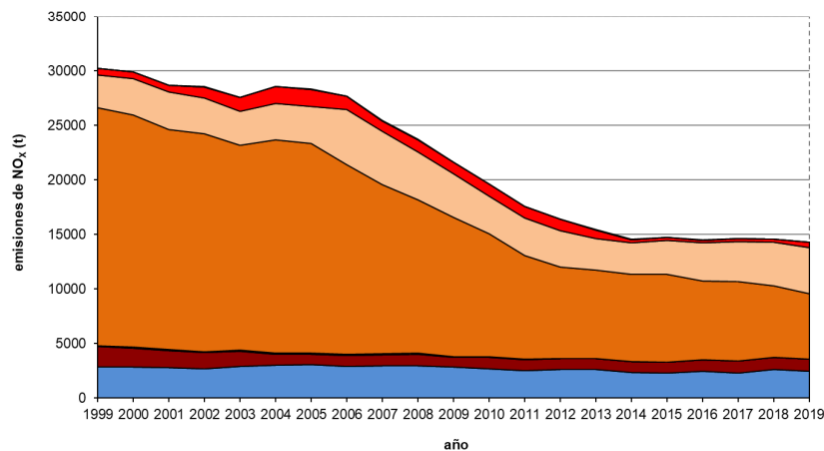


Figura 04 – Evolución de las emisiones de NO_x

Fuente: Excmo. Ayuntamiento de Madrid (5)

No obstante, aunque estos datos son el reflejo del aumento de la eficiencia en la generación de energía con motores de combustión interna, al cambio a combustibles menos contaminantes (como el cambio de carbón y derivados del petróleo a gas) y al aumento significativo en el número de vehículos eléctricos, no hay que olvidar que gran parte de la energía eléctrica proviene, en su gran mayoría, de la quema de gas natural que, por supuesto, también genera emisiones. Tampoco olvidar que los equipos de generación eléctrica sin emisiones también consumen una gran cantidad de esta energía en su fabricación y que no suele tenerse en cuenta, ya que las comparaciones únicamente se adscriben a la electricidad final producida. Por todo ello es necesario seguir trabajando en la reducción de las emisiones, sea cual fuere su origen.

Por último, reseñar un efecto interesante observado como resultado de diversas simulaciones realizadas sobre un conjunto de calles, como ejemplo sencillo. Se ha modelado una red mallada de 5 x 5 calles, con dos carriles por cada sentido de circulación en cada una de ellas, generando 36,80 km de viales de circulación y 25 intersecciones controladas por semáforos. Se han introducido un total de 2.000 vehículos, simulando una circulación de una hora de duración, con recorridos generados de manera aleatoria. Variando únicamente el límite de velocidad, en intervalos de 5 en 5 km/h, se ha obtenido que la velocidad media de todos los vehículos apenas varía, sobre todo en el rango entre los 15 y los 40 km/h.

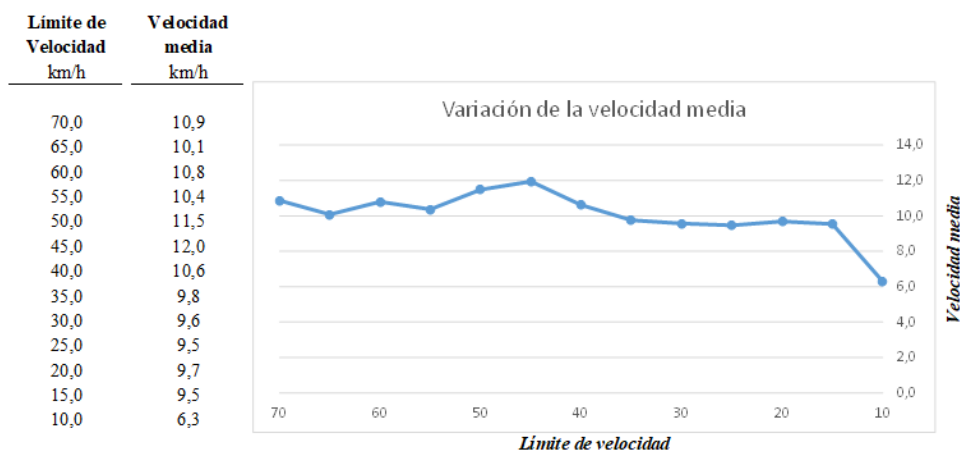


Figura 05 – Variación de la velocidad media con el límite de velocidad

Fuente: Elaboración propia

De estas simulaciones, respecto al límite de velocidad, podemos inferir que, si bien los consumos de combustible son muy dispares según el vehículo y la forma de conducción, por lo que son difícilmente cuantificables, sí existe una relación directa con la emisión de ruido y, por supuesto, con la gravedad de los accidentes que se puedan producir. Debido a esto se puede mejorar la calidad de vida en las ciudades interviniendo sobre este límite sin que el tiempo medio de los desplazamientos se vea afectado de manera significativa.

1.2 Justificación

Existe una tendencia clara a nivel global: el continuo incremento de la población, concentrándose esta en ciudades y conurbaciones. A esto se añade una demanda cada vez mayor de energía que no puede ser compensada, al menos hasta la fecha, por los avances tecnológicos en cuanto a reducir su necesidad y a hacer un uso más eficiente de la misma. Además, pese a la reducción en la emisión de gases contaminantes, todavía siguen perjudicando la salud de las personas que viven en estas áreas. Por tanto, teniendo en cuenta todo lo expuesto en la introducción, es necesario hacer un uso racional y óptimo de los recursos energéticos, ya que su disponibilidad, en un lugar y momento determinados, es finita.

En el presente proyecto se abordará la reducción de los tiempos en los desplazamientos. Dicho de otra manera: que, en un tiempo y un número total de vehículos en el área a estudiar dados, la distancia media recorrida en los trayectos de todos los vehículos sea la mayor posible, con una velocidad máxima (límite de velocidad) lo más baja posible. Para ello se realizarán dos aproximaciones para conseguir la reducción de estos tiempos:

- Se calcularán los tiempos de las fases de los semáforos.
- Se variarán los límites de velocidad de las calles.

1.3 Objetivos

El objetivo es reducir el impacto del tráfico en áreas urbanas. En concreto, los puntos de mejora serían los siguientes:

- Aumentar la distancia media recorrida por los vehículos en un tiempo dado.
- Reducir los tiempos efectivos de los viajes, entendiendo como tales las diferencias entre los lapsos totales empleados en los desplazamientos menos los tiempos perdidos, en esperas o atascos, por ejemplo.
- Reducir también estos tiempos perdidos en los desplazamientos, íntimamente relacionado con el punto anterior.
- Reducir los límites de velocidad de las calles.

El primer punto, como maximización, se evaluará como la proporción de la distancia media de los trayectos respecto a la media del escenario de partida. El segundo y tercer puntos se valorarán como el cociente entre los tiempos de viaje y perdidos obtenidos en la solución frente a los tiempos del escenario original. Por último, los límites de velocidad se valorarán según la suma de los límites de velocidad de las calles entre ambos escenarios, original y solución. Tanto el escenario, los recorridos, el tiempo a simular y el número de vehículos permanecerán invariantes para que los resultados sean comparables.

Como consecuencia de ello, al mejorar los tiempos de desplazamientos y ajustar los límites de velocidad, se mejorarán:

- Las emisiones de gases contaminantes.
- La contaminación acústica.
- La severidad de los accidentes de tráfico.
- La efectividad de los desplazamientos, mejorando las distancias recorridas y los tiempos empleados.

Aunque el simulador de tráfico que emplearemos y que se describe en el punto “2.2 *Simulation of Urban Mobility*” mide los dos primeros parámetros en función del tiempo en que se desplaza un vehículo (en miligramos de emisiones por segundo y en decibelios por segundo, respectivamente), no viéndose afectados por la velocidad de los vehículos en cada momento como sucede en la realidad. Por esto, no se tendrán en cuenta en la valoración (que se realizará como se indica en el párrafo anterior). Los detalles de estas puntuaciones se describirán pormenorizadamente en el apartado “4. Metodología”, donde se detalla la fórmula (función *fitness*) que puntuará cada solución.

Para intentar alcanzar todos estos objetivos, se programará una herramienta que permita múltiples configuraciones y posibilite al usuario evaluar la bondad de cada una de ellas:

- Por un lado, se permitirá crear escenarios, a partir de aplicaciones que proporciona el propio simulador, y que permitirá comparar diversas soluciones, sobre todo en fases de estudio. En estos casos se podrán variar el tamaño de la red, el número de carriles y la cantidad de vehículos, entre otros parámetros.

De esta manera se podrá analizar el desempeño de distintas soluciones para unos datos dados. A modo de ejemplo, se podrá observar si se comporta mejor una red mallada o una red radial para

un volumen de tráfico determinado. Otro posible análisis sería determinar si es más conveniente la disposición de más calles, más próximas entre sí, con menos carriles cada una o, por el contrario, concentrar el tráfico en menor número de calles, con mayor distancia entre ellas y contando con más carriles de circulación.

- Por otra parte, se podrán realizar estas evaluaciones sobre escenarios reales, a partir de mapas importados. Sobre estos únicamente se podrán variar la cantidad de vehículos en circulación y el lapso a simular.

Partiendo de los datos del escenario relativos a las fases de los semáforos existentes y de los límites de velocidad en cada una de sus vías, se podrán determinar los mejores tiempos de las fases y/o los límites de velocidad, según los elementos de cálculo seleccionados, en función del volumen de tráfico.

- Se podrán realizar optimizaciones sencillas o multiobjetivo, modificando a su vez uno o varios parámetros. Así, se podrán emplear en los cálculos los límites de velocidad y/o las fases de los semáforos, para las optimizaciones de distancia media recorrida, tiempos de desplazamiento y/o límites de velocidad.

La aplicación que se desarrollará permitirá que el propio usuario introduzca los datos del problema de su interés, elija los parámetros con los que quiere trabajar y qué valores quiere optimizar. De esta manera, podrá realizar simulaciones para un tiempo, una cantidad de tráfico y un escenario de su elección. Además, decidirá si para los cálculos se emplean los tiempos de cada fase semafórica, los límites de velocidad de cada calle o ambos simultáneamente. Finalmente, podrá determinar qué desea optimizar, pudiendo seleccionar una o varias de las siguientes opciones: maximizar la distancia recorrida por los vehículos, minimizar los tiempos empleados en los recorridos, juntamente con la minimización de los tiempos perdidos en paradas, y minimizar los límites de velocidad del conjunto de calles del escenario.

La bondad o no de las soluciones se determinarán mediante la función *fitness* que se expondrá con detalle en el apartado “4. Metodología”. La fórmula que se empleará será asimilable a una valoración porcentual, que tomará como referencia una solución con los datos del escenario de partida y con una puntuación igual a 100, por lo que las puntuaciones de distintas soluciones, *grosso modo*, podrán compararse según la distancia a este valor tomado como base de comparación.

1.4 Estructura

El presente trabajo se articula en los siguientes apartados:

1. Introducción, apartado actual que ha comprendido la motivación, justificación y los objetivos que se pretenden alcanzar a su finalización.
2. Estado del arte, en dónde se analizará, en primer lugar, algunos simuladores existentes en el mercado actualmente, la descripción del simulador elegido (SUMO: *Simulation of Urban Mobility*) y finalmente se repasarán los trabajos realizados sobre este mismo tema, empleando este mismo simulador, y las novedades que se pretenden aportar con este proyecto.

3. Marco teórico, en el que se encuadra el fundamento del desarrollo del presente estudio. Se describirá de forma resumida, pero suficiente, cómo se realiza la gestión del tráfico en las ciudades en la actualidad y en qué consiste un algoritmo genético que se implementará como solución.
4. Metodología, dónde se justificarán las herramientas propias del simulador utilizadas y su configuración, así como el algoritmo genético desarrollado.
5. Resultados, que compararemos con los objetivos planteados en esta misma introducción.
6. Conclusiones.
7. Trabajos futuros que podrían realizarse y que puedan complementar a este.
8. Referencias, con el listado de fuentes consultadas para la elaboración de este trabajo.

2. Estado del arte

2.1 Simuladores de tráfico

Actualmente existen en el mercado distintos simuladores de tráfico:

AIMSUN: es una plataforma de modelado de resolución múltiple que simula la movilidad en redes de todos los tamaños, desde una sola intersección hasta una región completa. Al modelar el tráfico, los ciclistas, los peatones y el transporte público permite la planificación y la evaluación de sistemas de gestión de tráfico inteligente y vehículos autónomos (6).

Forma parte de *Yunex Traffic Group*, antigua división de ITS (*Intelligent Transportation Systems*, o sistemas de transporte inteligente) de *Siemens Mobility*. Es una importante empresa en el campo de los sistemas inteligentes de tráfico: control y gestión adaptativos del tráfico, la automatización de autopistas y túneles, soluciones inteligentes para el cobro de peajes a usuarios y la simulación predictiva, clave para el apoyo a la decisión para la gestión integrada de corredores y la optimización de señales.

INTEGRATION: el modelo fue concebido a mediados de la década de 1980 como un modelo integrado de simulación y asignación de tráfico. Lo que hizo que el modelo fuera único radica en que el enfoque del modelo utilizaba la misma lógica de flujo de tráfico para representar tanto la autopista como los enlaces señalizados, y que tanto la simulación como los componentes de asignación de tráfico también eran microscópicos, integrados y dinámicos (7).

El enfoque microscópico permite una flexibilidad en la representación de las variaciones de las condiciones del tráfico. Permite que la densidad de tráfico pueda variar de forma continua, mientras que la mayoría de los modelos en aquel momento consideraban condiciones de tráfico uniformes a lo largo del tiempo.

PARAMICS DISCOVERY: reproduce las condiciones del tráfico del mundo real, simulando el comportamiento detallado del vehículo individual en una red de carreteras definida por el usuario. Se puede utilizar para probar una amplia variedad de modificaciones: cantidad de tráfico, modificación o creación de nuevas intersecciones, cambios en la señalización, etc. (8).

Este simulador permite la variación de velocidad según la hora, emular la cesión de paso al tráfico que se aproxima con tráfico lento en calles estrechas y pasos de un solo carril. Se pueden analizar pormenorizadamente las rutas que utilizan los conductores para viajar a través de la red, para comprender más profundamente el modelo y cómo funciona, y en última instancia aplicar decisiones sobre la gestión del tráfico, señalización, etc. incluso teniendo en cuenta otros medios de transporte, como peatones, ciclistas y transporte público.

TRITONE: es un micro simulador de redes viales desarrollado por la Universidad de Calabria como una herramienta para la modelización del flujo de tráfico. La micro simulación se utiliza para representar de forma puntual, precisa y específica el tráfico y su evolución instantánea, teniendo en cuenta los aspectos geométricos de la infraestructura y el comportamiento real del conductor (9).

Se creó con el objetivo de convertirse en una herramienta de simulación no solo eficiente desde el punto de vista analítico, sino especialmente desde el punto de vista de la usabilidad, simplificando al máximo las operaciones que debe realizar el usuario. Las operaciones posibles, son totalmente accesibles a través de la interfaz gráfica. Devuelve los resultados en forma analítica, basada en intervalos de control definidos por el usuario, en forma gráfica 2D, y en forma tridimensional 3D.

SUMO: es un paquete de simulación de tráfico multimodal continuo, microscópico, altamente portátil y de código abierto diseñado para manejar grandes redes. Permite el modelado de sistemas de tráfico intermodal que incluyen vehículos de carretera, transporte público y peatones. Además, contiene una gran cantidad de herramientas de apoyo que manejan tareas como la búsqueda de rutas, la visualización, la importación de redes y el cálculo de emisiones. Se puede mejorar con modelos personalizados y proporciona varias *API* (*application programming interface*, o interfaz de programación de aplicaciones) para controlar la simulación de forma remota. Su detalle se presenta en el siguiente punto “2.2 *Simulation of Urban Mobility*” (10).

PTV VISSIM: permite planificar y optimizar la operación del tráfico adaptándose a las condiciones, permitiendo replicar condiciones locales de un lugar concreto para su análisis, estudio y/o diseño futuro. Permite modelar escenarios reales con un alto nivel de detalle y complejidad, incluso con gráficos en 3D, destacando sobre otros simuladores (11).

Algunas de sus funcionalidades son:

- Gestión de modelos con variedad de rutas: autovías con peajes, control de flujo del tráfico, vías de control de acceso, etc.
- Análisis de viabilidad de autovías, con opciones de rutas alternativas asignadas automáticamente.
- Indicaciones de carreteras con desvíos.
- Gestión de plazas de aparcamiento.
- Representación gráfica de los flujos de tráfico.

Es una herramienta de simulación que permite el análisis de flujo del tráfico, planificación de servicios, estrategias avanzadas y soluciones de transporte. Se basa en modelos de planificación de transporte como generación y distribución de viajes, selección del modo de transporte y selección de ruta (12).

En cuanto a los tipos de licencia de estas aplicaciones:

Simulador	Licencia
AIMSUN	Comercial
INTEGRATION	Comercial
PARAMICS DISCOVERY	Comercial
TRITONE	Libre
SUMO	Libre
PTV VISSIM	Comercial

Tabla 01 – Licencias de los simuladores de tráfico

Fuente: Elaboración propia

Teniendo en cuenta que, de todos estos simuladores, SUMO posee licencia GPL (*general public license*, o licencia pública general) que permite el acceso e incluso la modificación del propio código del programa, tiene un uso generalizado en labores de investigación, con cierta regularidad incorpora nuevas funcionalidades y herramientas, y existe extensa y detallada documentación de consulta, será la herramienta elegida para las simulaciones de este proyecto.

2.2 Simulation of Urban Mobility

Descripción de la aplicación

Simulation of Urban Mobility (en adelante: SUMO) es un simulador de tráfico multimodal continuo, microscópico de código abierto, desarrollado principalmente por el Centro Aeroespacial Alemán y está disponible gratuitamente desde 2001, siendo un proyecto de la Fundación Eclipse desde 2017. Las simulaciones de tráfico son muy importantes para la gestión del tráfico (13). Las nuevas estrategias de tráfico se pueden implementar y analizar en una simulación antes de que se utilicen con el tráfico real. El simulador de tráfico SUMO ofrece un paquete completo de herramientas de software para simular y analizar el tráfico rodado, utilizándose principalmente con fines de investigación tales como la previsión de tráfico, evaluación de semáforos o selección de rutas, entre otros.

SUMO se ha utilizado en varios proyectos para responder a una gran variedad de preguntas de investigación como, por ejemplo:

- Evaluar el desempeño de los semáforos, incluyendo la evaluación de algoritmos modernos y la valoración de planificaciones semanales.
- Se ha investigado la elección de rutas de vehículos, incluido el desarrollo de nuevos métodos, la evaluación de rutas ecológicas basadas en la emisión de contaminantes e investigaciones sobre las influencias en toda la red de la elección de rutas autónomas.
- Entrenamiento mediante inteligencia artificial de planes semaforicos.
- Simulación de los efectos del tráfico de vehículos autónomos.
- Simulación y validación de la función de conducción autónoma, en cooperación con otros simuladores.

- Simulación de tráfico ferroviario para la gestión de vehículos basado en inteligencia artificial.
- Seguridad vial y análisis de riesgos.
- Cálculo de emisiones (ruido y contaminantes).

El simulador maneja archivos con formato *xml*, que estructura la información mediante etiquetas. Para ello contiene una gran cantidad de herramientas para la creación y gestión de estos archivos, cuyo número se amplía continuamente, para los más diversos propósitos relacionados con la aplicación. De todas ellas se ha hecho uso de las siguientes:

- *netgenerate*: utilizada para crear escenarios genéricos, como puede ser una red mallada tipo *Manhattan*, una red radial tipo tela de araña o una red aleatoria.
- *netconvert*: para convertir mapas de *OpenStreetMap* a formato *xml* para SUMO.
- *randomTrips*: para la generación aleatoria de pares de puntos, partida y llegada, dentro del escenario.
- *duarouter*: que, de los puntos iniciales y finales generados anteriormente, crea las rutas que seguirán los vehículos. Puede indicarse qué algoritmo se empleará en esta creación de trayectos, como, por ejemplo: *Dijkstra* o *A**, entre otros.
- *generateContinuousRerouters*: que mantiene a los vehículos circulando durante un tiempo dado, siguiendo las rutas calculadas.

Además, contiene una interfaz denominada *TraCI*, abreviatura de “*traffic control interface*” (interfaz de control de tráfico), dando acceso a una simulación de tráfico en curso, permitiendo recuperar valores de objetos simulados y manipular su comportamiento.

Escenario

Para el desarrollo del presente trabajo se usarán archivos *.net.xml* con toda la información de los elementos de la red en la que se va a realizar la simulación. Este archivo contiene los datos relativos a la definición de las calles, sus intersecciones y la regulación mediante semáforos. Seguidamente se exponen de manera general estos elementos, detallando únicamente aquellos que sean de interés.

Los nodos representan los extremos de las calles, así como las intersecciones o cruces entre ellas. Cada uno viene definido, como mínimo, por su identificador único y las coordenadas que lo sitúan en el plano. Además, pueden añadirse otros parámetros como, por ejemplo, el tipo de regulación de la intersección: por prioridad, con semáforos, etc.

Las calles a su vez se definen por su identificador único y por los nodos de los extremos. También puede incluirse información relativa al número de carriles, la velocidad, la forma, el nivel de prioridad respecto a otras calles, entre otros.

```

<edge id="A0A1" from="A0" to="A1" priority="-1">
  <lane id="A0A1_0" index="0" speed="13.89" length="183.20" shape="4.80,6.40 4.80,189.60"/>
  <lane id="A0A1_1" index="1" speed="13.89" length="183.20" shape="1.60,6.40 1.60,189.60"/>
</edge>
<edge id="A0B0" from="A0" to="B0" priority="-1">
  <lane id="A0B0_0" index="0" speed="13.89" length="333.20" shape="6.40,-4.80 339.60,-4.80"/>
  <lane id="A0B0_1" index="1" speed="13.89" length="333.20" shape="6.40,-1.60 339.60,-1.60"/>
</edge>

```

Figura 06 – Definición xml de una calle

Fuente: Elaboración propia

Aunque puede especificarse un límite de velocidad distinta para cada carril de cada calle de forma individual, y teniendo en cuenta que esta aplicación no es muy común en la realidad, se calculará un mismo límite para todos los carriles de un mismo sentido en una misma calle, entendiendo como tal el tramo viario entre dos nodos o intersecciones. En base a las simulaciones referidas en el apartado “1.2 Motivación”, en las que se comparaba el efecto del límite de velocidad en la velocidad media, lleva a determinar que particularizar un límite para cada carril, con la restricción añadida por el Reglamento general de circulación (14), que especifica que el límite para los carriles situados más a la derecha debe ser menor o igual que el límite de los carriles situados hacia la izquierda, no es de interés y sólo añade complejidad al problema.

Los semáforos están definidos también por su identificador, y deberá coincidir con el mismo identificador de la intersección para asociar su control a este semáforo. Contiene datos relativos a la forma de funcionamiento, la pertenencia a un conjunto de semáforos y el retraso de su ciclo frente a un tiempo cero, si lo tuviera. Por cada fase, se indica la duración y el estado, es decir, el conjunto de colores de los semáforos en la intersección que conforman dicha fase.

La codificación de colores que se empleará será la siguiente:





Letra	Color	Descripción
r		Los vehículos deben detenerse.
y		Los vehículos deben decelerar si pueden detenerse con seguridad, o pasar la intersección en otro caso.
g		Sin prioridad. Los vehículos deben pasar la intersección si no interfieren con otros vehículos (con mayor prioridad).
G		Con prioridad. Los vehículos deben pasar la intersección.

Tabla 02 – Codificación de los colores de los semáforos

Fuente: Documentación de SUMO (15)

La codificación que realiza SUMO para un ciclo de semáforo en una intersección en la que confluyen cuatro calles con dos sentidos de circulación y dos carriles por cada sentido, es decir, en donde confluyen un total de 16 carriles, es la siguiente:

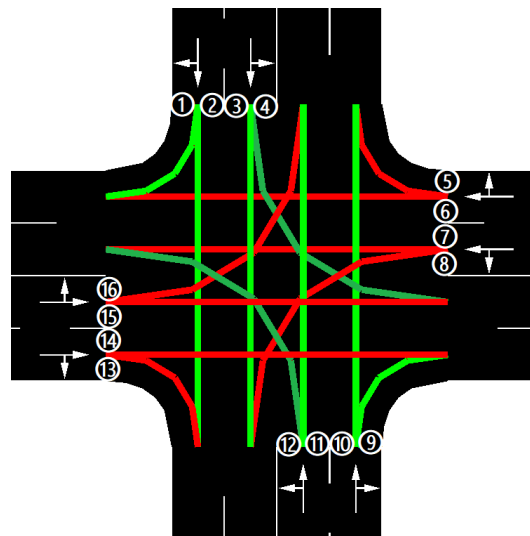
```
<tlLogic id="B1" type="static" programID="0" offset="0">
  <phase duration="42" state="GGGrrrrGGGrrrr"/>
  <phase duration="3" state="yyyyrrrrrrrrrr"/>
  <phase duration="42" state="rrrrGGGrrrrGGGg"/>
  <phase duration="3" state="rrrrrrrrrrrrrrrr"/>
</tlLogic>
```

Figura 07 – Definición xml de un semáforo

Fuente: Elaboración propia

Dónde los tiempos de las fases en ámbar permanecerán constantes e iguales a tres segundos, según establece SUMO por defecto, y para el resto se calculará su duración para mejorar la fluidez del tráfico. Por defecto, la suma total de las fases de un ciclo es de 90 segundos, aunque durante los cálculos se variará este total. La codificación del estado corresponde a la luz del semáforo en cada uno de los posibles movimientos dentro de la intersección.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
G	G	G	g	r	r	r	r	G	G	G	g	r	r	r	r

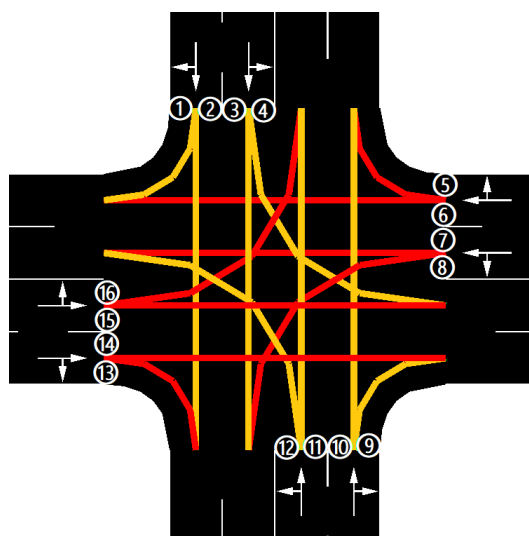


42 segundos

Figura 08.a – Detalle de la fase 1 de un semáforo

Fuente: Elaboración propia

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
y	y	y	y	r	r	r	r	y	y	y	y	r	r	r	r

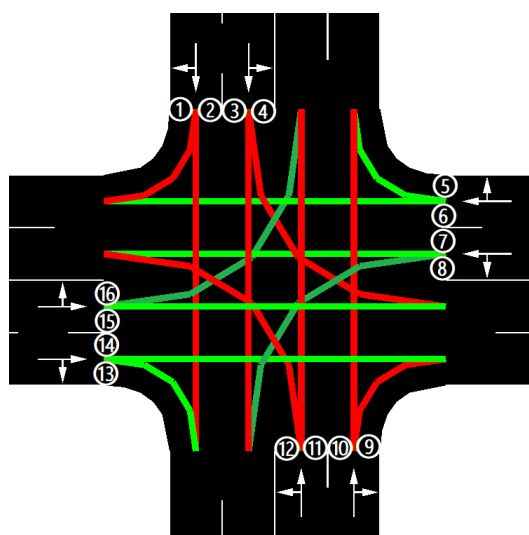


3 segundos

Figura 08.b – Detalle de la fase 2 de un semáforo

Fuente: Elaboración propia

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
r	r	r	r	G	G	G	g	r	r	r	r	G	G	G	g

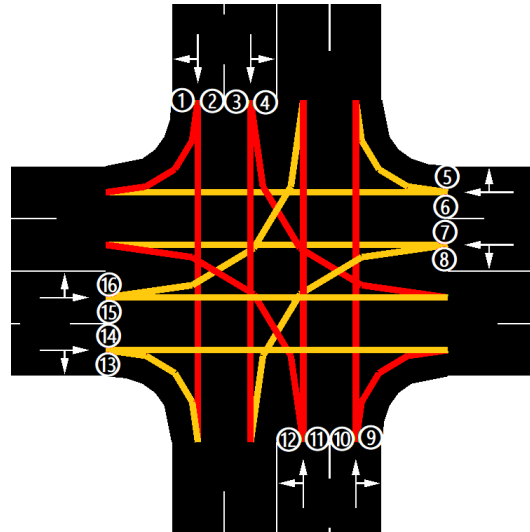


42 segundos

Figura 08.c – Detalle de la fase 3 de un semáforo

Fuente: Elaboración propia

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
r	r	r	r	y	y	y	y	r	r	r	r	y	y	y	y



3 segundos

Figura 08.d – Detalle de la fase 4 de un semáforo

Fuente: Elaboración propia

Para este ejemplo, con estas cuatro fases quedaría completado el ciclo.

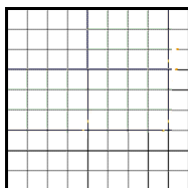
Herramientas

De todas las herramientas que contiene el simulador, a continuación, se describirán aquellas que se han empleado en este proyecto, detallando también el significado de los parámetros empleados en cada una de ellas:

<i>netgenerate</i>	Genera una red vial básica.
Parámetros generales	
<i>--default.lanenumber</i> <i><INT></i>	Indica el número de carriles a crear por cada sentido de circulación.
<i>--default.speed</i> <i><FLOAT></i>	Indica el límite de velocidad en todas las calles, en m/s.
<i>--no-turnarounds</i> <i>true</i>	Deshabilita la creación de cambios de sentido en las intersecciones.
<i>--no-turnarounds.tls</i> <i>true</i>	Deshabilita la creación de cambios de sentido en las intersecciones controladas por semáforos.
<i>--default-junction-type</i> <i>traffic_light</i>	Establece el tipo de intersecciones por defecto, controladas por semáforos.
<i>--tls.guess</i> <i>true</i>	La herramienta intentará crear intersecciones controladas por semáforos, siempre que sea posible.

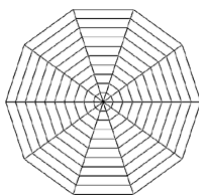
<code>--output-file</code> <code><FILE></code>	Ruta y nombre de archivo en el que se guardará el escenario creado.
---	---

Red mallada, tipo *Manhattan*



<code>--grid</code>	Indica el tipo de red a construir.
<code>--grid.x-number</code> <code><INT></code>	Establece el número de calles horizontales.
<code>--grid.x-length</code> <code><FLOAT></code>	Indica la distancia entre intersecciones de las calles horizontales.
<code>--grid.y-number</code> <code><INT></code>	Establece el número de calles verticales.
<code>--grid.y-length</code> <code><FLOAT></code>	Indica la distancia entre intersecciones de las calles verticales.

Red radial, tipo tela de araña



<code>--spider</code>	Indica el tipo de red a construir.
<code>--spider.arm-number</code> <code><INT></code>	Indica el número de calles radiales, con un extremo en el centro hasta el exterior del escenario.
<code>--spider.circle-number</code> <code><INT></code>	Indica el número de calles concéntricas.
<code>--spider.circle-number</code> <code><FLOAT></code>	Establece la distancia radial entre cada calle concéntrica.

Red aleatoria



Para escenarios aleatorios, el número de carriles a crear por defecto será el número máximo de carriles por cada sentido de circulación.

<code>--rand</code>	Indica el tipo de red a construir.
---------------------	------------------------------------

<code>--rand.iterations</code> <code><INT></code>	Indica el número de intentos de generación de intersecciones, cumpliendo las distancias máximas y mínimas entre ellas.
<code>--rand.min-distance</code> <code><FLOAT></code>	Establece la distancia mínima entre intersecciones.
<code>--rand.max-distance</code> <code><FLOAT></code>	Establece la distancia máxima entre intersecciones
<hr/>	
<code>netconvert</code>	Genera un escenario a partir de un mapa extraído de <i>OpenStreetMap</i> .
Parámetros generales	
<code>--osm-files</code> <code><FILE></code>	Ruta y nombre del archivo exportado de <i>OpenStreetMap</i> (.osm) para crear el escenario, obtenido de: https://www.openstreetmap.org/export <code><FILE></code> será la ubicación y nombre de archivo, por defecto: “./data/map.osm”
<code>--geometry.remove</code> <code>true</code>	Simplifica el escenario, reduciendo el tamaño del archivo resultante.
<code>--junctions.join</code> <code>true</code>	Une intersecciones próximas entre sí.
<code>--tls.guess-signals</code> <code>true</code>	Interpreta los cruces controlados por semáforos que rodean una intersección como parte de una misma.
<code>--tls.discard-simple</code> <code>true</code>	Descarta la creación de semáforos no presentes en el archivo importado.
<code>--tls.join</code> <code>true</code>	Intenta agrupar intersecciones controladas por semáforos.
<code>--tls.default-type</code> <code>actuated</code>	Genera ciclos semafóricos con intervalos de duración mínima y máxima, indicados.
<code>--tls.min-dur</code> <code><INT></code>	Mínima duración de una fase de ciclo semafórica. Por defecto se ha establecido en tres segundos, para coincidir con el tiempo de fases en ámbar.
<code>--tls.max-dur</code> <code><INT></code>	Máxima duración de una fase de ciclo semafórica.
<code>--no-turnarounds</code> <code>true</code>	Deshabilita la creación de cambios de sentido en las intersecciones.
<code>--no-turnarounds.tls</code> <code>true</code>	Deshabilita la creación de cambios de sentido en las intersecciones controladas por semáforos.
<code>--tls.guess</code> <code>true</code>	La herramienta intentará crear intersecciones controladas por semáforos, siempre que sea posible.
<code>--output-file</code> <code><FILE></code>	Ruta y nombre de archivo en el que se guardará el escenario creado.

Es importante señalar que, para el uso de estos mapas importados, el área a obtener debe estar convenientemente informada. La información faltante, errónea o no actualizada puede llevar a la simulación de escenarios cuyos resultados difieran de la realidad. La exportación de mapas a través de la web limita el número de nodos a un máximo de 5.000.

<i>randomTrips.py</i>	<i>Script</i> que genera, para un escenario dado, puntos de inicio y fin aleatorios para la creación de rutas a través de él.
-----------------------	---

Parámetros generales

<i>--net-file</i> <FILE>	Ruta y nombre del archivo que contiene los datos del escenario.
<i>--begin</i> <INT>	Instante de inicio de generación de rutas.
<i>--end</i> <INT>	Instante final de generación de rutas.
<i>--period</i> <FLOAT>	Tasa de llegada de vehículos por periodo.
<i>--flows</i> <INT>	Número de rutas a generar. Coincidirá con el número de vehículos a simular.
<i>--output-file</i> <FILE>	Ruta y nombre de archivo en el que se guardarán los puntos de inicio y final creados.

<i>duarouter</i>	Para un escenario dado y unos puntos de inicio y fin de recorrido, genera rutas aleatorias a través de él.
------------------	--

Parámetros generales

<i>--net-file</i> <FILE>	Ruta y nombre del archivo que contiene los datos del escenario.
<i>--route-files</i> <FILE>	Ruta y nombre del archivo que contiene los datos de las rutas.
<i>--begin</i> <INT>	Instante de inicio de generación de rutas.
<i>--end</i> <INT>	Instante final de generación de rutas. Coincidirá con el tiempo de finalización de la simulación.
<i>--route-steps</i> < INT >	Precarga de las rutas para la duración indicada a partir del instante actual. Coincidirá con el tiempo total de simulación para generar las rutas durante todo este intervalo.
<i>--routing-algorithm</i> <STRING>	Indica que algoritmo se empleará para calcular la ruta. Aunque existen más posibilidades, en este trabajo se permitirá elegir entre: <ul style="list-style-type: none"> • Dijkstra • A*
<i>--output-file</i> <FILE>	Ruta y nombre de archivo en el que se guardarán las rutas creadas.

generateContinuousRerouters.py Script que genera, para un escenario dado, rutas continuas a través de él, para un intervalo determinado.

Parámetros generales	
<code>--net-file</code> <code><FILE></code>	Ruta y nombre del archivo que contiene los datos del escenario.
<code>--end</code> <code><INT></code>	Tiempo, en segundos para el que es necesario crear las rutas continuas.
<code>--output-file</code> <code><FILE></code>	Ruta y nombre de archivo en el que se guardarán los puntos de inicio y final creados y el intervalo de tiempo a repetir cada una de forma continua.

Configuración de la simulación

La información de cada simulación se recoge en un archivo *sumocfg*, cuya información se gestiona mediante etiquetas, como los archivos *xml*. Esta información contiene por una parte los archivos de entrada, como el escenario, las rutas y la extensión de estas durante la duración del tiempo de simulación de forma continua; así como los datos relativos al lapso a simular, es decir, los instantes inicial y final.

```
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">
  <input>
    <net-file value="Simulacion.net.xml" />
    <route-files value="Simulacion.rou.xml" />
    <additional-files value="Simulacion.contRou.xml" />
  </input>
  <time>
    <begin value="0" />
    <end value="600" />
  </time>
</configuration>
```

Figura 09 – Definición xml de la configuración de la simulación

Fuente: Elaboración propia

No se añade la opción de evitar la *teleportación*, es decir, la eliminación de vehículos de la simulación si se encuentran detenidos por un tiempo elevado. Esto es debido a que, al crearse las rutas de manera aleatoria, y aunque en raras ocasiones se generan rutas que no son posibles, es conveniente eliminar estos vehículos cuando sigan una ruta que no sea realizable porque pueden impedir el paso a otros que sí sigan una ruta factible.

Estadísticas

Una vez realizada la simulación, SUMO guarda en un archivo *xml* los datos estadísticos de la simulación, agrupados según los distintos tipos de locomoción. Se obtienen a partir de un archivo intermedio que recoge la información individual de cada recorrido, e incluyen los de aquellos vehículos que no hayan completado sus respectivas rutas.

```

<statistics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/statistic_file.xsd">
  <vehicles loaded="250" inserted="250" running="249" waiting="0"/>
  <teleports total="0" jam="0" yield="0" wrongLane="0"/>
  <safety collisions="0" emergencyStops="0"/>
  <persons loaded="0" running="0" jammed="0"/>
  <vehicleTripStatistics
    routeLength="2776.40" speed="4.72" duration="592.79" waitingTime="369.73" timeLoss="535.69"
    departDelay="4.86" departDelayWaiting="0.00" totalTravelTime="148197.00" totalDepartDelay="1216.00"
  />
  <pedestrianStatistics number="0" routeLength="0.00" duration="0.00" timeLoss="0.00"/>
  <rideStatistics number="0"/>
  <transportStatistics number="0"/>
</statistics>

```

Figura 10 – Definición xml de las estadísticas

Fuente: Elaboración propia

Para la puntuación de cada simulación se emplearán los datos estadísticos relativos a los vehículos, y en concreto:

- *routeLength*, con la distancia media recorrida por los vehículos, en metros.
- *duration*, con el tiempo medio total real en el que el vehículo se encuentra en la simulación, es decir, el tiempo medio total del viaje.
- *timeLoss*, con la media del tiempo perdido durante el viaje. Incluye los tiempos de espera involuntarios, así como el tiempo que se circula a menos de 0,1 m/s.

A través de la línea de comandos al lanzar cada simulación, se informa del archivo intermedio con la información de los recorridos individuales, de la recopilación de datos de vehículos que no hayan completado la ruta calculada y también del archivo de salida con los datos estadísticos.

2.3 Trabajos previos

Existen múltiples y muy diversos estudios elaborados sobre las simulaciones de tráfico por ordenador. Cabe destacar tres temáticas que, recientemente, acaparan el mayor número de trabajos respecto de la gestión de tráfico:

- En primer lugar, existen estudios que se centran, por un lado, en simular los resultados de la aplicación de diferentes algoritmos relacionados con la inteligencia artificial como, por ejemplo, la programación de conjuntos de respuestas (*Answer Set Programming – ASP*) (16), una metodología de programación orientada a problemas de búsqueda combinatoria, donde el objetivo es encontrar una solución entre un número grande pero finito de posibilidades cuya idea surgió de la investigación sobre inteligencia artificial y lógica computacional: describe lo que se considera una solución al problema, pero no especifica un algoritmo para resolverlo. La búsqueda se realiza mediante sofisticados sistemas de software denominados solucionadores de conjuntos de respuestas.
- Otros estudios relacionados con el tráfico, la aplicación de inteligencia artificial y el empleo de SUMO, y que cada vez abundan más, son aquellos relacionados con la simulación de la conducción de vehículos autónomos, como la toma de decisiones para estos vehículos en entornos inesperados utilizando métodos eficientes de aprendizaje por refuerzo (17). Este tipo de aprendizaje permite que los vehículos tomen decisiones por sí mismos, siendo necesario determinar cuán buenas o no son estas decisiones, más en entornos altamente aleatorios o inesperados. Estas valoraciones se realizan en el simulador, permitiendo también el aprendizaje del agente (vehículo).

- Por último, existen estudios relacionados con la ingeniería civil evaluando, por ejemplo, el impacto de la colocación de semáforos en intersecciones que no disponen de ellos (18). El uso de simulaciones previas, además de abaratar costes, permite anticipar problemas no previstos, ya que la colocación de semáforos ralentiza los desplazamientos en las vías con prioridad en favor de vías secundarias que, de otra forma, quedarían colapsadas cuando la primera tenga elevados niveles de tráfico.

A fecha de redacción del presente, existen sendos trabajos previos directamente relacionados con este y realizados en la *Universitat Politècnica de València* como fin de grado (19) y fin de máster (20). En ellos se aplica un algoritmo genético para mejorar la gestión del tráfico actuando sobre la duración de los tiempos de fase de cada ciclo de cada semáforo. El primero, basa la optimización en el número de vehículos que alcanzan el destino respecto al total, midiendo también los consumos de combustible y tiempos de espera medios y máximos de los vehículos. El segundo, mide la bondad de la solución respecto a la distancia recorrida y tiempo empleado por cada vehículo, además de sus emisiones de CO₂, ajustado a la cantidad de vehículos que completan sus recorridos.

2.4 Trabajo actual

El presente proyecto pretende, como ya se ha comentado en el apartado “1.3 Objetivos”, y aplicando igualmente una solución mediante un algoritmo genético, no sólo actuar en las fases de los ciclos de los semáforos sino también en los límites de velocidad de las vías, de tal forma que se adapte al tráfico existente. De este modo se intentarán cumplir las finalidades planteadas al inicio: aumentar las distancias recorridas en un tiempo dado, reducir los tiempos en los desplazamientos, tanto tiempos de viaje como perdidos, y mejorar la calidad de vida, por la disminución de ruidos, y la seguridad vial, reduciendo la severidad potencial de los posibles accidentes, a través de la disminución de la velocidad de circulación en las calles.

Además, con el uso de herramientas que proporciona el propio simulador, se generarán cíclicamente las rutas creadas de manera aleatoria, de forma que se mantenga el nivel de congestión en el escenario durante la duración de la simulación. La finalidad de este propósito es conseguir adaptar los límites de velocidad y los ciclos semafóricos al tráfico, sobre todo en momentos de elevado volumen y congestión, adecuándose a las rutas efectivamente realizadas.

También se podrán realizar simulaciones en distintos escenarios genéricos, creados con herramientas incluidas en SUMO, que permitan comparar los resultados de diferentes opciones y como estas son más adaptables o no a cambios en los niveles de tráfico. Por ejemplo, se podrá estudiar, de forma previa a su construcción, si es más eficiente una red mallada o radial, o si es mejor construir vías más distantes y con más carriles de circulación o, por el contrario, que estas se encuentren más próximas, pero con menos viales en cada una.

3. Marco teórico

A continuación, se exponen los fundamentos teóricos del presente trabajo. Se parte del análisis de los sistemas de control de tráfico mediante semáforos en vías urbanas en la actualidad, analizando los criterios de necesidad, tiempos de las fases de cada ciclo, cómo se realiza su cálculo para, teóricamente, su mejor fluidez y que limitaciones tienen estos sistemas. Posteriormente se repasan los sistemas de gestión variable de la velocidad. Y finalmente, se describe en qué consiste un algoritmo genético y cómo este, de todas las posibles combinaciones de los valores de estos tiempos y velocidades, que crecen exponencialmente en función del número de semáforos, sus ciclos y el número de vías, es un algoritmo adecuado para obtener una solución óptima (o próxima a esta) sin necesidad de simular todas las posibles combinaciones.

3.1 Semaforización

En las intersecciones en las que, por la intensidad de tráfico en las calles que convergen en la misma no es posible regular el tráfico estableciendo una vía como principal y asignándole la prioridad frente al resto, es necesario realizar esta gestión mediante el uso de semáforos. Esto introduce demoras en los vehículos que circulan por la vía preferente, a cambio de agilizar los tránsitos por el resto de las vías y mejorar la seguridad en estos puntos.

Existen cuatro criterios fundamentales que aconsejan la instalación de semáforos:

- Intensidad mínima horaria, durante más de ocho horas al día: si se superan los valores de vehículos por hora de la tabla.

Número de carriles por acceso		Intensidad vía principal Ambos sentidos	Intensidad vía secundaria con mayor intensidad Un sentido
Vía principal	Vía secundaria		
1	1	500	150
≥ 2	1	600	150
≥ 2	≥ 2	600	200
1	≥ 2	500	200

Tabla 03 – Intensidad mínima horaria, durante más de ocho horas al día

Fuente: WikiVía – La Enciclopedia de la carretera (21)
Asociación española de la carretera

- Intensidad mínima horaria para demoras: si se superan las intensidades, la demora de los vehículos en vías secundarias, por la elevada intensidad en la vía principal, será elevada.

Número de carriles por acceso		Intensidad vía principal Ambos sentidos	Intensidad vía secundaria con mayor intensidad Un sentido
Vía principal	Vía secundaria		
1	1	750	75
≥ 2	1	900	75
≥ 2	≥ 2	900	100
1	≥ 2	750	100

Tabla 04 – Intensidad mínima horaria, para demoras

Fuente: WikiVía – La Enciclopedia de la carretera (21)
Asociación española de la carretera

- Tráfico peatonal: si durante más de ocho horas al día suceden, de forma simultánea, unas afluencias superiores a 150 peatones y 600 vehículos por hora.
- Accidentalidad: si suceden cinco o más accidentes en el transcurso de un año.

Como se puede observar en la figura 11, elevados niveles de tráfico en vías prioritarias, como es la N-VI, pueden provocar que se colapsen las vías que accedan a la principal, elevando los tiempos de demora, y, por tanto, los incrementos de tiempo de los desplazamientos. Es en estos casos cuando, según los datos referidos en las tablas 03 y 04 de intensidades de tráfico, se hace necesario el uso de semáforos para la dirección del tráfico.



Figura 11 – N-VI a su paso por San Rafael (Segovia)

Fuente: El Norte de Castilla (22)

Existen diversas formas de gestionar las fases de cada ciclo de los semáforos:

- Tiempos fijos: en la que el reparto es constante. Es la forma de regulación más sencilla, realizándose una programación para una estimación media de tráfico. Únicamente se puede modificar reprogramando el sistema, por lo que no puede adaptarse a la intensidad del tráfico ni reaccionar frente a congestiones.
- Tiempos variables por reloj: en dónde el reparto depende de las horas del día. Es una forma algo más compleja que la anterior. La variación de los tiempos de las fases se realiza en función de la hora del día y del día de la semana, realizado en función de una medición previa de las intensidades. Se adapta mejor a las condiciones de tráfico, pero no es capaz de reaccionar ante incrementos de intensidad no previstos.

- Tiempos variables por detectores descentralizados: aquella en la que el tiempo de fase y ciclo se realiza en función de la medición de aforos de tráfico en los accesos a la intersección. Es muy sensible frente al fallo de algún medidor de intensidad de tráfico.
- Regulación dinámica en tiempo real: realizado por un operador desde una sala de control de tráfico. La información procede de cámaras de televisión y de mediciones enviadas por las estaciones de aforo. Es el sistema más flexible y adaptable a las condiciones del tráfico, pero depende de la experiencia del operario.

Los semáforos cambian sus colores a lo largo del tiempo. Se denomina fase a cada una de las combinaciones de indicaciones que permiten uno o varios movimientos simultáneamente a través de la intersección, es decir, es el tiempo donde no se produce ningún cambio de color en los semáforos que hay en una intersección. Se denomina ciclo al tiempo necesario para que se dé una sucesión completa de indicaciones en los semáforos que se encuentran conectados a un mismo regulador. El ciclo es, por tanto, el tiempo total de verde más ámbar más rojo, que es común para los semáforos de una intersección. El reparto del ciclo es la distribución del ciclo entre los distintos ramales que confluyen en la intersección, es decir, la duración o el porcentaje de cada una de las fases que componen el ciclo de una intersección y es proporcional a la intensidad de tráfico en cada uno de estos viales.

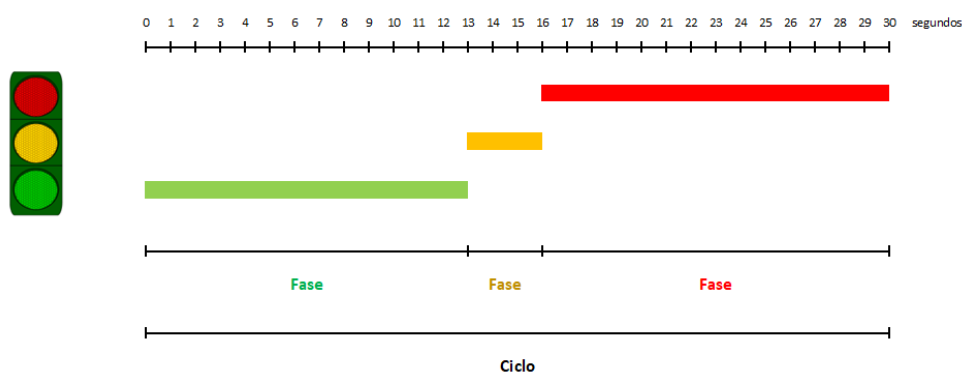


Figura 12 – Ciclo y fases de un semáforo

Fuente: Elaboración propia

Para el cálculo de la duración de cada fase se tienen en cuenta los siguientes tiempos:

- Tiempo de reacción: tiempo que transcurre desde que el color del semáforo se fija en verde y los vehículos comienzan a desplazarse. Se estima entre dos y tres segundos.
- Tiempo de ámbar: segundos en los que se realiza la transición de verde a rojo, necesario para que se detengan los vehículos que se encuentran a una distancia suficiente. Se encuentra fijado en tres o cuatro segundos, normalmente. Para el presente trabajo se fijará en tres segundos y será un valor constante, ya que cuatro segundos se emplea en vías con límites de velocidad iguales o superiores a 70 km/h.
- Tiempo de despeje: tiempo para que todos los vehículos que han accedido a la intersección la abandonen, liberando la zona para los siguientes movimientos. Correspondería con los instantes en que todos los semáforos se encuentran en rojo.
- Tiempos muertos: cantidad de tiempo en segundos para que se realice el despeje de la intersección comentado en el punto anterior.

Hay que tener en cuenta, si existen pasos de peatones, el tiempo necesario para cruzarlos, siendo normal emplear una velocidad para los viandantes de 1,50 m/s. Para la determinación del número de fases por cada intersección hay que realizar un estudio pormenorizado que comprenda, además de lo mencionado en este mismo apartado, los puntos conflictivos en la intersección, es decir, aquellos donde se cruzan o confluyen distintas trayectorias de vehículos. Como norma general y siempre que sea posible, estos puntos se evitarán dentro de una misma fase semafórica, por ejemplo: prohibiendo giros a la izquierda o permitiéndolo en otra fase distinta, si fuera necesario, para no perjudicar la seguridad vial. Sin menoscabo de todo lo expuesto, existirá el menor número posible de fases dentro de cada ciclo para minimizar los tiempos perdidos, que comprenden los tiempos de reacción y los tiempos muertos. Por tanto, el estudio de cada intersección debe alcanzar un compromiso entre todas estas variables.

3.2 Gestión de la velocidad

Se entiende como gestión de la velocidad a todas aquellas medidas destinadas a adecuar la rapidez de los movimientos de los vehículos a las condiciones imperantes. Los límites genéricos no se corresponden con la velocidad apropiada para todas las vías e incluso dentro de una misma vía, según el momento de la jornada (diurna o nocturna) o la estación meteorológica (invierno o verano).



Figura 13 – Señalización específica

Fuente: Carreteros.org (23)
Norma 8.1-IC – Señalización vertical

El uso de límites de velocidad variables en algunas circunstancias puede ayudar a mejorar los niveles de seguridad, los tiempos de recorrido y, por ello, las demoras de los usuarios. Por lo general, los límites de velocidad variables se emplean según criterios generales mediante señalización fija particularizando para la hora del día, la estación del año o las condiciones meteorológicas (lluvia).



Figura 14 – Señalización variable

Fuente: Carreteros.org (23)
Norma 8.1-IC – Señalización vertical

En algunos casos se han implementado límites dinámicos de velocidad que tienen en cuenta tanto las condiciones del tráfico real como las condiciones meteorológicas adversas, aunque su uso queda prácticamente limitado a fechas concretas de grandes desplazamientos, en vías de alta capacidad en entornos de grandes ciudades y destinos vacacionales, y fenómenos de climatología adversa.

3.3 Algoritmo genético

Un algoritmo genético pertenece a la familia de algoritmos de búsqueda inspirados en los principios de la evolución en la naturaleza. Imita el proceso de reproducción y selección natural, de tal forma que puede producir una solución de alta calidad para diversos problemas relacionados con la búsqueda, la optimización y el aprendizaje. Al mismo tiempo, su analogía con la evolución natural permite que los algoritmos genéticos superen algunos de los obstáculos que encuentran los algoritmos tradicionales de búsqueda y optimización, especialmente donde el espacio de búsqueda es tan grande que no es posible encontrar la solución óptima en un tiempo razonable.

Un algoritmo genético implementa una versión simplificada de la evolución de las especies que tiene lugar en la naturaleza. Los principios de la teoría de la evolución se pueden resumir en los siguientes:

- Variación: Los rasgos (atributos) de los especímenes individuales pertenecientes a una población pueden variar. Como resultado, los especímenes difieren entre sí hasta cierto punto; por ejemplo, en su comportamiento o apariencia.
- Herencia: algunos rasgos se transmiten constantemente de los especímenes a sus descendientes. Como resultado, la descendencia se parece más a sus padres que a especímenes no emparentados.
- Selección: las poblaciones suelen luchar por los recursos dentro de su entorno. Los especímenes que posean rasgos que se adapten mejor al medio ambiente tendrán más éxito para sobrevivir y también contribuirán con más descendencia a la próxima generación.

En otras palabras, la evolución mantiene una población de especímenes individuales que varían entre sí. Aquellos que están mejor adaptados a su entorno tienen más posibilidades de sobrevivir, reproducirse y transmitir sus rasgos a la siguiente generación. De esta manera, a medida que pasan las generaciones, las especies se adaptan más a su entorno y a los desafíos que se les presentan.

Un algoritmo genético busca la solución óptima para un problema dado, manteniendo una población de soluciones candidatas, llamadas individuos, para ese problema. Estas soluciones candidatas se evalúan iterativamente y se utilizan para crear una nueva generación de soluciones. Aquellas que son mejores para resolver este problema tienen más posibilidades de ser seleccionadas y transmitir sus cualidades a la próxima generación de soluciones candidatas, mejorando sucesivamente para resolver el problema en cuestión.

Los diversos componentes de un algoritmo genético que permiten esta analogía para la evolución darwiniana son:

Genotipo: imitando el concepto de cromosoma, cada individuo (solución) está representado por una colección de genes. En nuestro caso, esta codificación corresponderá a los segundos de las fases semafóricas y/o a los límites de velocidad de las calles. Por ejemplo, para el primer caso, obviando las fases en ámbar:

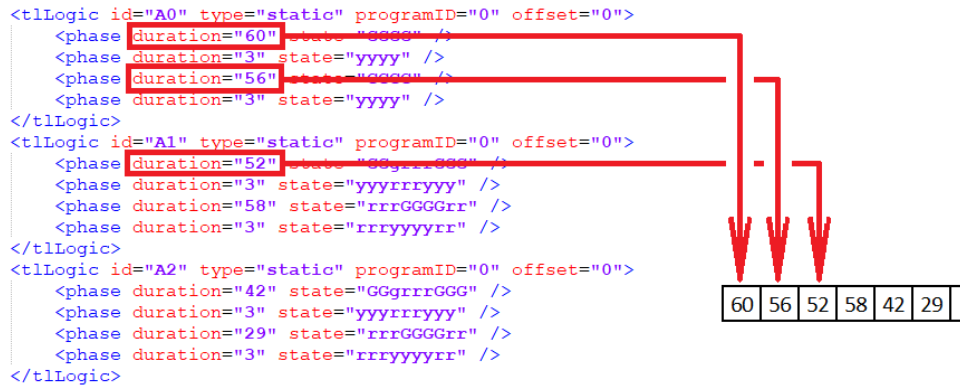


Figura 15 – Cromosoma codificando tiempos de fases de semáforos

Fuente: Elaboración propia

En lo sucesivo, para esa exposición teórica y por simplicidad, cada gen lo expresaremos como una cadena binaria, donde cada bit representa un solo gen:

010111010

Figura 16 – Cromosoma simple codificado en binario

Fuente: Hands-on genetic algorithms with Python (24)

Población: colección de soluciones candidatas para el problema. Dado que cada individuo está representado por algún cromosoma, esta población de individuos puede verse como una colección de tales cromosomas:

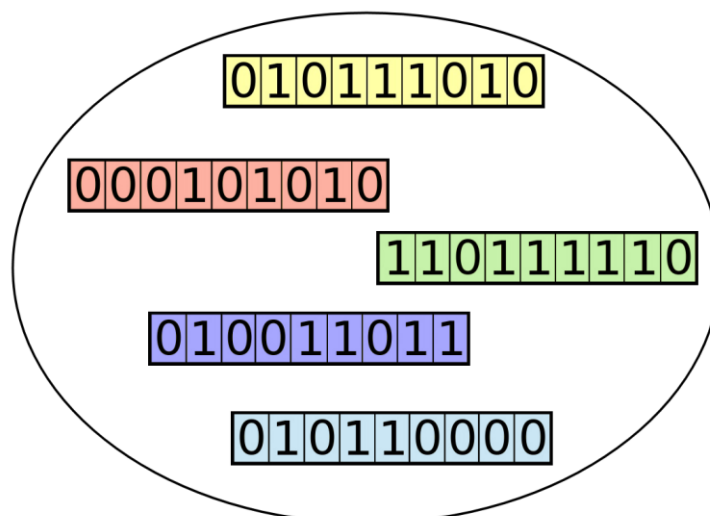


Figura 17 – Población de individuos representados por cromosomas codificados en binario.

Fuente: Hands-on genetic algorithms with Python (24)

La población representa continuamente a la generación actual y evoluciona con el tiempo cuándo la generación actual es reemplazada por una nueva. La generación de la población inicial se realiza, generalmente, de manera aleatoria según los criterios o restricciones que deben cumplir las soluciones al problema.

Función *fitness* (función de aptitud o función objetivo): para evaluar a los individuos, en cada iteración del algoritmo. Esta es la función que buscamos optimizar o el problema que intentamos resolver. Los individuos que logran una mejor puntuación representan mejores soluciones y es más probable que sean elegidas para reproducirse y estar representadas en la próxima generación. Aún se pueden elegir individuos con valores bajos de *fitness*, pero con menor probabilidad. De esta forma, su material genético no queda completamente excluido. Con el tiempo, la calidad de las soluciones mejora.

Selección: Después de calcular la puntuación de cada individuo de la población, se utiliza un proceso de selección para determinar qué individuos de la población podrán reproducirse y crear la descendencia que formará la próxima generación. Existen diferentes formas de ponderar esta valoración:

- Selección por rueda de ruleta: A cada individuo de la población se le asigna una parte proporcional a su *fitness* en una ruleta, tal que la suma de todos los porcentajes sea la unidad.
- Selección por torneo: Se establecen grupos de individuos, seleccionando el mejor de cada uno. Se realiza de manera sucesiva hasta seleccionar el número de padres deseados.
- Selección por rango: la población se ordena de acuerdo con sus valores de función *fitness*. La posición de cada individuo en la lista ordenada representa su rango, ponderado a su vez aplicando otra función que lo puntúe. Este método será el empleado en este trabajo.

Cruce o recombinación: Para crear un par de nuevos individuos, generalmente se eligen dos padres de la generación actual, y partes de sus cromosomas se intercambian (cruzan) para crear dos nuevos cromosomas que representan a la descendencia.

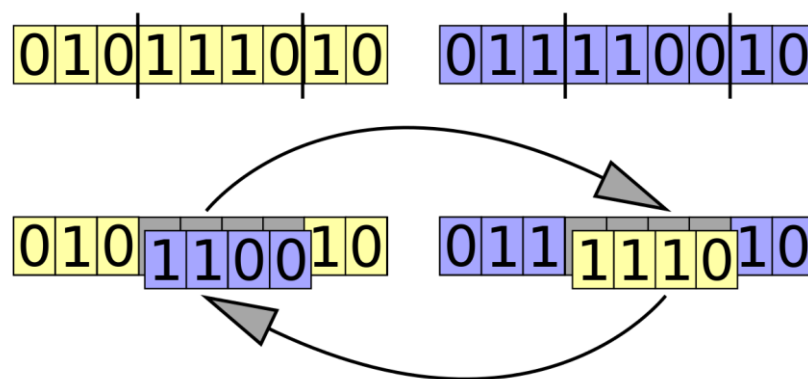


Figura 18 – Operación de cruce entre dos cromosomas codificados en binario

Fuente: Hands-on genetic algorithms with Python (24)

Cada cromosoma es cortado en n puntos y las proporciones de cada cromosoma se recombinan, realizándose normalmente entre uno y tres puntos de corte. La figura 18 es un ejemplo para $n = 2$.

Otra forma es realizar un cruce uniforme, es decir, la combinación aleatoria de los genes de cada progenitor para formar un nuevo hijo.

Mutación: actualización periódica y aleatoria de la población, introduciendo nuevos patrones en los cromosomas, fomentando la búsqueda en áreas desconocidas del espacio de solución. Puede manifestarse como un cambio aleatorio en uno o varios genes. Las mutaciones se implementan como cambios aleatorios en uno o más valores cromosómicos; por ejemplo, cambiando un bit en una cadena binaria:

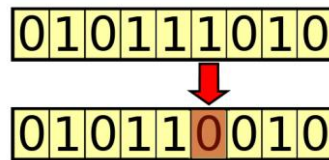


Figura 19 – Mutación aplicada a un cromosoma codificado en binario

Fuente: Hands-on genetic algorithms with Python (24)

También puede realizarse seleccionando un bit y moviéndolo a una posición arbitraria, o tomando dos e intercambiando sus posiciones.

Reemplazo de la población: Establece el criterio para decidir qué individuos se mantienen tras cada iteración y cuales son descartados. Estos criterios van desde el reemplazo completo en cada generación, o de sólo un porcentaje del total. También puede realizarse, cada cierto número de iteraciones, la eliminación de todos los individuos excepto aquel con mejor *fitness*, o seleccionando un individuo para grupos de individuos con la misma puntuación, y completando el resto de los miembros de la población generándolos de forma aleatoria.

Criterio de finalización: Existen distintas condiciones de parada del algoritmo, como, por ejemplo:

- Número fijo de generaciones o soluciones creadas.
- Tras un tiempo de cómputo fijo.
- Considerando medidas de diversidad en la población (convergencia).
- Cuando se alcance el óptimo (si es conocido), o un determinado valor de *fitness*.
- Tras varias generaciones sin ninguna mejora (convergencia).

4. Metodología

En el presente apartado se abordará la implementación de la solución, describiendo y justificando los detalles de esta. Se ha elegido *Jupyter Notebook* como espacio para desarrollarla. Dentro de este cuaderno, el flujo de ejecución es el siguiente:

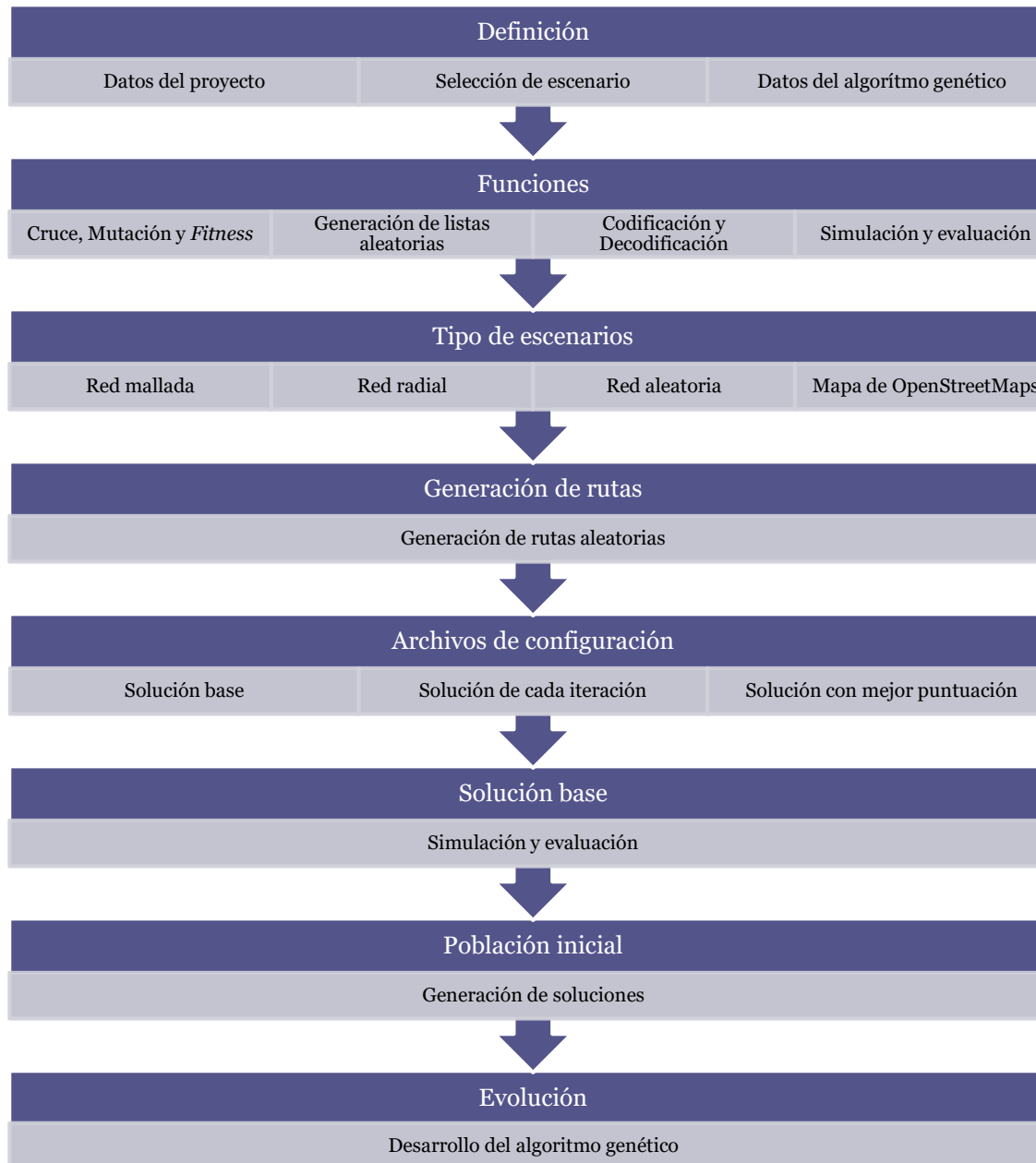


Figura 20 – Flujo de ejecución

Fuente: Elaboración propia

Project Jupyter, al que pertenece *Jupyter Notebook*, es un proyecto de código abierto sin ánimo de lucro, nacido del Proyecto *IPython* en 2014 a medida que evolucionaba para admitir la ciencia de datos interactiva y la computación científica en todos los lenguajes de programación. *Jupyter*

es un software completamente de código abierto y de uso gratuito. Se desarrolla abiertamente en el gestor de repositorios de *software GitHub*, a través del consenso de la comunidad (25).

Se ha elegido *Jupyter Notebook* por ser un entorno interactivo, basado en la web y en el que, por tanto, puede incluir el código del programa acompañado de texto, imágenes, gráficos, etc. que facilitan su comprensión. Estas características hacen que su uso se extienda cada vez más en entornos académicos y científicos.

Como se ha descrito a lo largo de este trabajo, la aplicación desarrollada aborda la simulación de tráfico en SUMO. En primera instancia, el usuario introducirá tanto el tiempo como el número de vehículos que formarán parte de la simulación. Para cada uno de estos vehículos se generarán rutas de forma aleatoria dentro del escenario, y serán repetidas cíclicamente para mantener el nivel de congestión de tráfico deseado.

También se elegirán los elementos que formarán parte de los cálculos, es decir, tiempo de fases semafóricas y/o límites de velocidad, y qué valores se desean optimizar. La optimización puede comprender uno o varios de los siguientes parámetros:

- Maximizar las distancias medias recorridas por los vehículos.
- Minimizar los tiempos de los desplazamientos, tanto la duración efectiva de los viajes como los tiempos perdidos en ellos.
- Minimizar los límites de velocidad en las calles.

Después, seleccionará el escenario objeto de estudio, pudiendo elegir entre diferentes opciones: escenarios generados de forma automática u obtenidos de mapas reales. Mediante un algoritmo genético, y con los datos elegidos para los cálculos, se realizarán las optimizaciones seleccionadas.

Para valorar la bondad de las soluciones que se obtengan en el proceso del programa, se puntuarán respecto a una solución base. Esta solución será aquella obtenida de la simulación con los datos del escenario de partida, y tendrá una puntuación de 100. Las puntuaciones de las distintas soluciones serán asimilables a la variación porcentual respecto de esta solución de partida.

Finalmente, se seleccionarán los parámetros del propio algoritmo genético. Se introducirá la población inicial, que será un conjunto de soluciones generadas aleatoriamente. Se hará evolucionar a esta población de diferentes formas: los individuos de esta población se mezclarán entre sí, mediante cruce, se modificarán puntualmente, por mutación, o se generarán soluciones nuevas de forma aleatoria, como la población inicial. La elección de una u otra forma de evolución será definida por el usuario, mediante asignación de probabilidades de elección de cada método.

El objetivo es encontrar una solución óptima, o lo más próxima a esta, sin necesidad de simular todas las posibles combinaciones. Para ello se mantendrá un conjunto de soluciones más prometedoras, con mayor puntuación, del tamaño de la población inicial que haya introducido el usuario. Para que este objetivo comprenda todo el espacio de búsqueda, no sólo se cruzan las soluciones de la población, si no que se introducen valores nuevos (mutación), e incluso nuevas soluciones.

A continuación, se describirá pormenorizadamente el desarrollo de la aplicación, describiendo todas estas fases de manera detallada. Para una mejor comprensión, el orden seguido es el normal

de los cálculos, aunque no coincida con el momento en que aparecen en el cuaderno, por la distinta necesidad de definición de ciertos elementos del código.

La ejecución del programa se inicia desde el menú: *Run > Run All Cells*, o pulsando sobre: » en la parte superior del cuaderno.

4.1 Definición

Datos del proyecto

A través de una sencilla interfaz gráfica se introducirán los datos generales relativos al escenario a simular y evaluar. En primer lugar, los datos relativos a la simulación:

- Título, que será usado para identificar el proyecto de simulación en curso y para nombrar todos los archivos que se generen.
- Tiempo a simular, indica el lapso de tiempo simulado del que se analizarán y obtendrán datos estadísticos de los recorridos. Influye directamente en la duración de la ejecución del programa.
- Número de vehículos en el escenario, para los que se generarán rutas aleatorias durante la duración completa del tiempo a simular indicado. La variación de este valor afecta al nivel de congestión del tráfico en el escenario. Un valor elevado de vehículos ralentizará la ejecución del programa, ya que SUMO es un simulador microscópico, es decir, se simula cada vehículo de forma individual, calculando las interacciones de cada uno de ellos con el resto de los elementos en el escenario.
- Algoritmo para el cálculo de rutas, pudiendo seleccionar uno entre *Dijkstra* o *A**. Será el usado para la creación de cada ruta individualizada para cada uno de los vehículos.

DATOS GENERALES DE LA SIMULACIÓN

Título para el proyecto:

Tiempo a simular (en minutos):

Nº de vehículos en el escenario:

Algoritmo para el cálculo de rutas: ☐ Dijkstra ☒ A* (A estrella)

Incluir en los cálculos: ☒ Límites de velocidad ☒ Fases semafóricas

Optimización: ☒ MAX: Distancia recorrida ☒ MIN: Tiempos de viaje ☒ MIN: Límite de velocidad

Figura 21 – Datos generales de la simulación

Fuente: Elaboración propia

Para valores de tiempo a simular superiores a 60 minutos, y cantidad de vehículos mayor a 2.500, se informará de la ralentización posible en la ejecución del programa, ya que cada una de estas simulaciones son las que más tiempo de ejecución consumen.

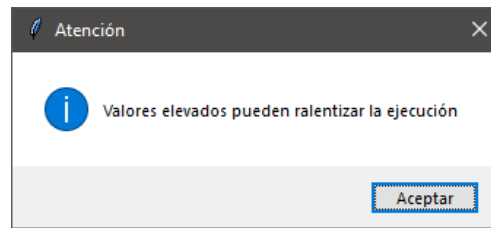


Figura 22 – Aviso: posible ralentización en la ejecución

Fuente: Elaboración propia

Además, se podrán seleccionar los datos del escenario que serán utilizados en los cálculos. Habrá que escoger al menos uno entre:

- Límites de velocidad, que se calcularán para todos los carriles de una misma calle, entendiendo como calle la vía con un mismo sentido de circulación entre dos intersecciones.
- Fases semafóricas, es decir, los tiempos, de forma individual, de todas las fases de las intersecciones, exceptuando las fases en ámbar.

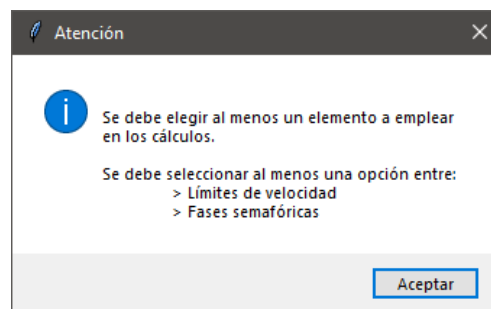


Figura 23 – Aviso: parámetros de cálculo

Fuente: Elaboración propia

De forma análoga, se podrán elegir qué parámetros se quieren optimizar, marcando al menos uno entre:

- Distancia recorrida, que se medirá como la distancia promedio de los desplazamientos de los vehículos durante el tiempo simulado. El objetivo es maximizar esta distancia.
- Tiempos de viaje, obtenidos como la diferencia entre el tiempo total en que los vehículos realizan los recorridos menos los tiempos perdidos, bien en atascos o en paradas en semáforos, y también estos mismos tiempos perdidos. El objetivo es minimizar ambos tiempos.
- Límites de velocidad, como la suma de los límites de velocidad de las calles del escenario. Obviamente, es necesario incluir en los elementos de cálculo a estos mismos límites de velocidad. Si no se incluyen en los elementos de cálculo, esta opción se deshabilitará, avisando de esta circunstancia al usuario.

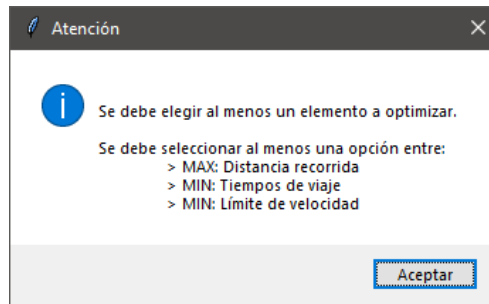


Figura 24 – Aviso: parámetros de optimización

Fuente: Elaboración propia

La optimización de los límites de velocidad de cada calle se realiza como la suma de todas ellas. No se tiene en cuenta la longitud de cada una de estas calles, para realizar una ponderación respecto a esta distancia, evitando ralentizar la ejecución añadiendo más complejidad. Además, hay que tener en cuenta que seleccionar únicamente la optimización de los límites de velocidad, tenderá a que estos converjan al valor mínimo, esto es, a 10 km/h.

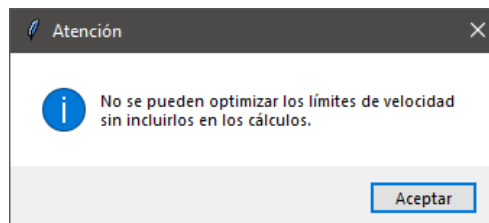
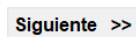


Figura 25 – Aviso: límites de velocidad

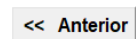
Fuente: Elaboración propia

Una vez completados los campos, se pasará a la elección del escenario pulsando sobre:



Selección de escenario

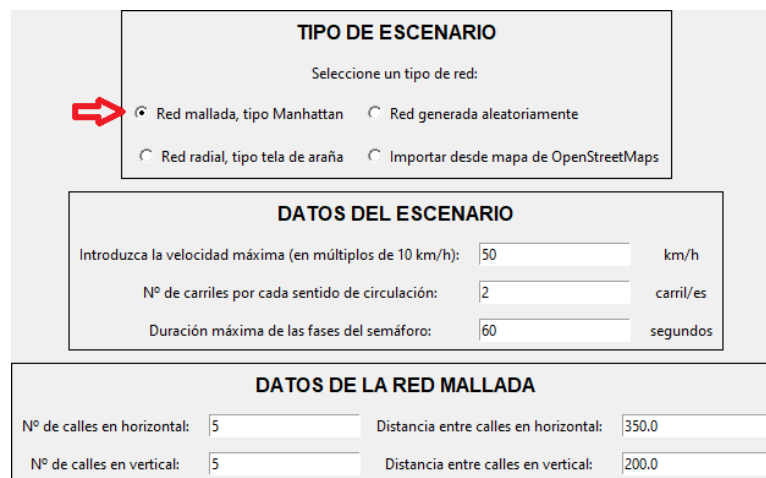
Se podrá retroceder al menú de datos del proyecto pulsando sobre:



Como se ha descrito en la parte de herramientas del simulador, en el apartado “2.2 Simulation of Urban Mobility”, será posible la selección de diferentes escenarios. Estos podrán ser generados automáticamente por la aplicación, en función de los parámetros que se introduzcan, o a partir de un mapa importado desde *OpenStreetMap*.

En general, para las redes generadas automáticamente habrá que introducir el límite máximo de velocidad en el escenario, que será múltiplo de 10 km/h, el número de carriles (por calle y sentido de circulación) y la duración máxima de una fase semafórica, en segundos. En el caso de no seleccionar como elemento de cálculo los límites de velocidad, no se indicará el primero, y en el caso de no marcar las fases de los semáforos, no se introducirá el último.

Para una red mallada, tipo *Manhattan*, será necesario indicar la cantidad de calles en horizontal, en vertical, y las distancias para cada tipo, es decir, distancias entre calles horizontales y distancias entre calles verticales.



TIPO DE ESCENARIO

Seleccione un tipo de red:

☒ Red mallada, tipo Manhattan ☐ Red generada aleatoriamente

☐ Red radial, tipo tela de araña ☐ Importar desde mapa de OpenStreetMaps

DATOS DEL ESCENARIO

Introduzca la velocidad máxima (en múltiplos de 10 km/h): km/h

Nº de carriles por cada sentido de circulación: carril/es

Duración máxima de las fases del semáforo: segundos

DATOS DE LA RED MALLADA

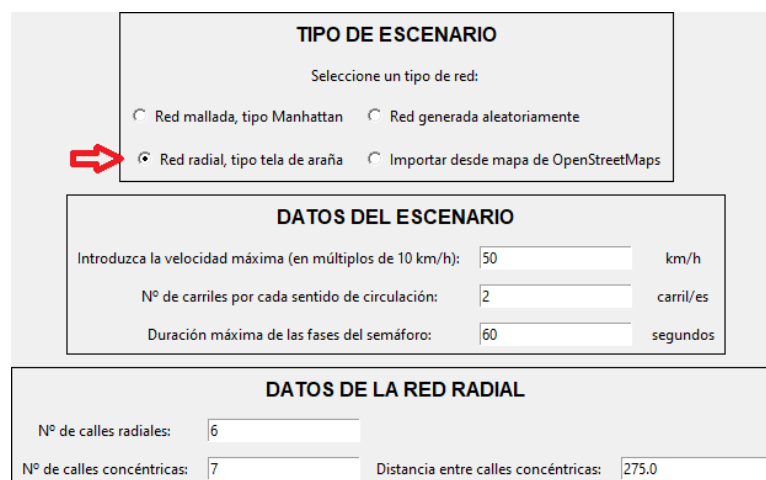
Nº de calles en horizontal: Distancia entre calles en horizontal:

Nº de calles en vertical: Distancia entre calles en vertical:

Figura 26.a – Red mallada

Fuente: Elaboración propia

Si se selecciona una red de tipo radial, será necesario indicar el número de calles que parten del centro del escenario, cuantos círculos concéntricos se crearán y la distancia entre ellos.



TIPO DE ESCENARIO

Seleccione un tipo de red:

☐ Red mallada, tipo Manhattan ☐ Red generada aleatoriamente

☒ Red radial, tipo tela de araña ☐ Importar desde mapa de OpenStreetMaps

DATOS DEL ESCENARIO

Introduzca la velocidad máxima (en múltiplos de 10 km/h): km/h

Nº de carriles por cada sentido de circulación: carril/es

Duración máxima de las fases del semáforo: segundos

DATOS DE LA RED RADIAL

Nº de calles radiales:

Nº de calles concéntricas: Distancia entre calles concéntricas:

Figura 26.b – Red radial

Fuente: Elaboración propia

Si se selecciona la generación de una red aleatoria, se indicarán los datos referentes a los intentos de la herramienta para crear intersecciones que cumplan las distancias mínimas y máximas entre ellas, que también deben especificarse. Por lo general, se suelen crear tantas intersecciones como se indiquen, aunque es frecuente que estos sean meros cambios de dirección en lugar de cruces controlados por semáforos.

TIPO DE ESCENARIO

Selecione un tipo de red:

☐ Red mallada, tipo Manhattan
☒ Red generada aleatoriamente
☐ Red radial, tipo tela de araña
☐ Importar desde mapa de OpenStreetMaps

DATOS DEL ESCENARIO

Introduzca la velocidad máxima (en múltiplos de 10 km/h):

50

km/h

Nº de carriles por cada sentido de circulación:

2

carril/es

Duración máxima de las fases del semáforo:

60

segundos

DATOS DE LA RED ALEATORIA

Nº de intersecciones:

175

Distancia mínima entre intersecciones:

350.0

Distancia máxima entre intersecciones:

500.0

Figura 26.c – Red aleatoria

Fuente: Elaboración propia

Para una red aleatoria, el número de carriles por cada sentido de circulación será el máximo, pudiendo existir calles con menor número del consignado.

Por último, si se va a simular un escenario real, se importará un mapa de *OpenStreetMap*, que deberá contener los datos completos de la red.

TIPO DE ESCENARIO

Selecione un tipo de red:

☐ Red mallada, tipo Manhattan
☐ Red generada aleatoriamente
☐ Red radial, tipo tela de araña
☒ Importar desde mapa de OpenStreetMaps

OPEN STREET MAP

Seleccionar archivo de mapa de OpenStreetMap...

Archivo:

./data/map.osm

Figura 26.d – Red importada de OpenStreetMap

Fuente: Elaboración propia

Pulsando el botón:

Seleccionar archivo de mapa de OpenStreetMap...

Aparecerá un explorador de archivos para poder seleccionar aquel que contiene el escenario a simular. Bajo este botón se mostrará la ruta del archivo que se haya seleccionado.

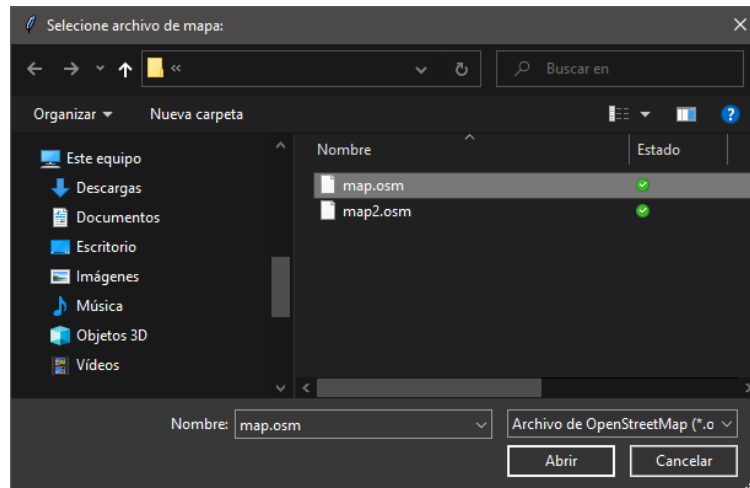


Figura 27 – Seleccionar archivo de mapas

Fuente: Elaboración propia

Elegido el escenario, se pasará a definir los datos del algoritmo pulsando:

Siguiente >>

Hiperparámetros del algoritmo genético

Se podrá volver al menú anterior para modificar los datos introducidos pulsando en:

<< Anterior

En cuanto a los datos del algoritmo genético, habrá que indicar el tamaño de la población, las iteraciones a realizar, la forma de selección de individuos, así como la probabilidad de cruce, mutación o creación de individuos nuevos.

DATOS DEL ALGORITMO GENÉTICO	
Introduzca el tamaño de la población:	150
Introduzca el número de iteraciones:	1000
Se efectuarán, al menos, un 20% del valor de iteraciones desde la obtención del mejor individuo	
Indique el modo de selección de los individuos:	<input type="radio"/> Equiprobable <input checked="" type="radio"/> Ponderada
En cada iteración:	
% de probabilidad de creación individuos a través de CRUCE:	60.0
Puntos de cruce:	2
% de probabilidad de creación individuos a través de MUTACIÓN:	30.0
Nº de genes a mutar:	10
% de probabilidad de creación de un NUEVO individuo de forma aleatoria:	10.0
Suma total (%):	100.0

Figura 28 – Datos del algoritmo genético

Fuente: Elaboración propia

La población será la cantidad de soluciones, ordenadas por puntuación, de mayor a menor, que se mantienen a lo largo de la ejecución. No incluirá la solución base, con la que se compararán los resultados y se empleará en la función *fitness*.

El número de iteraciones indica la cantidad de individuos, además de la población inicial, que se generarán. Además, siempre se crearán, al menos, dos veces el tamaño de la población desde la obtención del mejor individuo. Es decir, para una población de 200 soluciones, si se establecen 1.000 iteraciones y el mejor individuo es el que ha aparecido en la iteración 943, se crearán individuos hasta la iteración 1.343, ya que:

$$943 + 2 * 200 = 943 + 400 = 1.343$$

Para la selección, por rango, de un individuo i en una población de tamaño n , para realizar el cruce o la mutación, se puede elegir una de las siguientes dos opciones:

- Selección equiprobable, en donde todos los individuos tienen la misma probabilidad de ser seleccionados.

$$p = \frac{1}{n}$$

- Selección ponderada, donde la probabilidad de selección del individuo en la posición i es inversamente proporcional a la posición que ocupan en la población (ordenada por puntuación).

$$p_i = \frac{1}{i} * \frac{1}{\sum_{j=1}^n \frac{1}{j}}$$

Se establecerá la probabilidad con que se crearán nuevos individuos a partir de la población, a través del cruce entre dos de ellos, la mutación de uno o la creación de un nuevo individuo. Habrá que indicar la cantidad de puntos de cruce, para el primer caso, y el número de genes a mutar, para el segundo, estando este limitado por código al 25% de los genes, si se especificara un valor mayor, para preservar la información del individuo de partida y que no sea más parecido a un nuevo individuo creado aleatoriamente. La suma de los tres porcentajes tiene que ser igual al 100%, mostrándose un mensaje de advertencia en caso contrario.

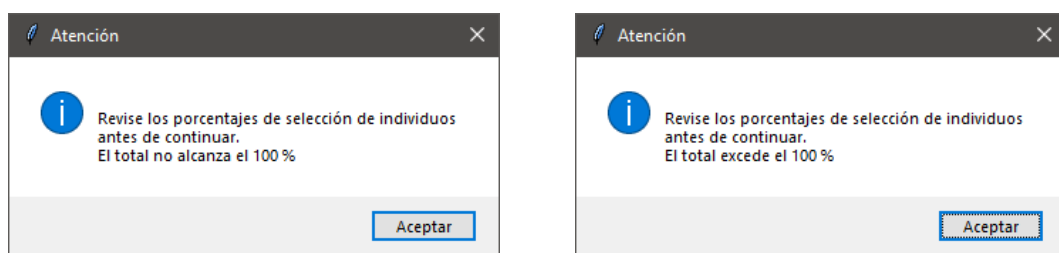


Figura 29 – Aviso: suma de porcentajes de selección

Fuente: Elaboración propia

Se ha tenido en cuenta que de cada cruce se obtienen dos nuevos individuos, que serán simulados, evaluados e incluidos en la población, por lo que el algoritmo, internamente, reduce este porcentaje a la mitad y, junto con el resto, se calculan las probabilidades de selección para que el total sea igual al 100%, es decir, se normalizan las probabilidades. De esta manera, aunque en cada iteración se elija al azar un método distinto de creación de soluciones, garantizamos que la generación de individuos se aproxime a la elegida por el usuario.

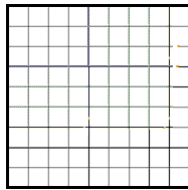
Una vez completados todos los datos, se prosigue la ejecución pulsando sobre:

Continuar...

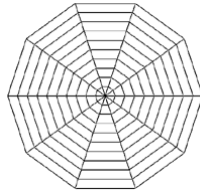
4.2 Escenarios

Según la elección del usuario se lanzará el subproceso correspondiente para la generación del escenario automático, si se seleccionó una de las tres primeras opciones. Este subproceso es equivalente a introducir el nombre de la herramienta *netgenerate* y sus parámetros por la línea de comandos. Los posibles escenarios automáticos son:

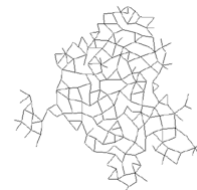
Red mallada, tipo *Manhattan*



Red radial, tipo tela de araña



Red aleatoria



Para la cuarta opción, al importar un mapa de *OpenStreetMap*, el subproceso que se iniciará será *netconvert*, que transformará los datos importados en un escenario para el simulador SUMO. Los límites de velocidad y las fases semafóricas deberán estar perfectamente descritas en el mapa a importar para que la simulación sea extrapolable a la realidad. Si la exportación se realiza a través de la web <https://www.openstreetmap.org/export>, el número de nodos, es decir, el número de puntos extremos de cada calle está limitado a 5.000.

Mapa importado de *OpenStreetMap*



Ambas herramientas y los parámetros empleados y sus significados se han descrito con más detalle en el apartado “2.2 *Simulation of Urban Mobility*”. Corresponden, además de la configuración propia de la herramienta, a aquellos datos introducidos por el usuario a través de la interfaz gráfica.

4.3 Generación de rutas

Una vez definidos escenario, número de vehículos y tiempo de simulación, ya se han recopilado los datos necesarios para la generación de rutas en el escenario. Inicialmente se generan el doble de pares de puntos que número de vehículos para que se disperse el tráfico a lo largo del escenario. Estos pares de puntos representan los puntos iniciales y finales de cada trayecto.

Con estos pares de puntos, y siguiendo el algoritmo elegido, bien sea *Dijkstra* o bien *A** (A estrella), se calculan las rutas, calle a calle, a seguir por los vehículos entre ambos puntos. De manera muy ocasional, se pueden construir rutas que no sean posibles. En estos casos, tras un tiempo determinado por el simulador, el vehículo se elimina del escenario para no impedir el paso a otros que tengan en común ese punto de la ruta.

Finalmente, estas rutas se extrapolan a la duración del lapso a simular, de tal manera que se mantenga el número de vehículos constante en el escenario, descontando aquellos que hayan sido eliminados del escenario (*teleportados*), como se ha indicado en el párrafo anterior. También, por esta razón, el número de rutas que se generan aleatoriamente son el doble del número de vehículos, para evitar que, si una de estas rutas es incorrecta, se repita durante el tiempo simulado, arrojando unos datos estadísticos que se alejarían de la realidad.

4.4 Archivos de configuración

Se creará un archivo de configuración para cada tipo de solución a simular:

- Solución base, con los datos del escenario original, el número de vehículos y la duración especificados. Indica al simulador qué archivo contiene el escenario, las trayectorias a recorrer y las rutas continuas, creadas a partir de las anteriores para el lapso a simular.
- Solución de la población a simular, con los datos decodificados en el escenario que se va a evaluar (*test*) en cada iteración, y las trayectorias y rutas continuas igual al caso anterior. Para el escenario se mantiene un único archivo que será sobre escrito en cada iteración para minimizar el uso de espacio en disco.
- Mejor solución (*best*), que guardará los datos de la mejor solución alcanzada en un instante dado, para poder reproducir, si así se desea, la simulación con la interfaz gráfica de SUMO. Los datos del escenario a los que corresponde esta solución mejor puntuada también sobre escribe a las anteriores, manteniendo únicamente esta última versión.

```
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/sumoConfiguration.xsd">
  <input>
    <net-file value="Simulacion.net.xml" />
    <route-files value="Simulacion.rou.xml" />
    <additional-files value="Simulacion.contRou.xml" />
  </input>
  <time>
    <begin value="0" />
    <end value="600" />
  </time>
</configuration>
```

Figura 30 – Definición xml de la configuración de la simulación

Fuente: Elaboración propia

4.5 Solución base

De los datos del escenario se obtiene un individuo cuyo identificador será el número cero. En función de los elementos de cálculo seleccionados, constará de un listado con el límite de velocidad de cada calle y/o las fases semafóricas de cada intersección. Las cantidades de uno y otro se guardarán para generar posteriormente los miembros iniciales de la población con el tamaño adecuado.

```
<statistics xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://sumo.dlr.de/xsd/statistic_file.xsd">
  <vehicles loaded="250" inserted="250" running="249" waiting="0"/>
  <teleports total="0" jam="0" yield="0" wrongLane="0"/>
  <safety collisions="0" emergencyStops="0"/>
  <persons loaded="0" running="0" jammed="0"/>
  <vehicleTripStatistics
    routeLength="2776.40" speed="4.72" duration="592.79" waitingTime="369.73" timeLoss="535.69"
    departDelay="4.86" departDelayWaiting="0.00" totalTravelTime="148197.00" totalDepartDelay="1216.00"
  />
  <pedestrianStatistics number="0" routeLength="0.00" duration="0.00" timeLoss="0.00"/>
  <rideStatistics number="0"/>
  <transportStatistics number="0"/>
</statistics>
```

Figura 31 – Definición xml de las estadísticas

Fuente: Elaboración propia

Con el archivo de configuración comentado en el primer punto del apartado anterior, se genera la simulación para obtener los datos estadísticos, que serán empleados en la función *fitness*, y que son la referencia de la bondad o no de las diferentes soluciones que se van a valorar a lo largo de la ejecución.

De los datos estadísticos tomaremos los siguientes:

- *routeLength*, con la distancia media recorrida por los vehículos, en metros.
- *duration*, con el tiempo medio total real en el que el vehículo se encuentra en la simulación, es decir, el tiempo medio total del viaje, en minutos.
- *timeLoss*, con la media del tiempo perdido durante el viaje. Incluye los tiempos de espera involuntarios, así como el tiempo que se circula a menos de 0,1 m/s.

4.6 Funciones

A continuación, se describen las funciones de las que se hará uso a lo largo del programa, incluidas las ya mencionadas en “3.3 Algoritmo genético”.

Cruce

A partir de dos individuos, y según los datos seleccionados para incluir en los cálculos, límites de velocidad y/o fases semafóricas, se realizará un cruce de ambos según el número de puntos que se indiquen. La disposición de estos puntos de cruce será aleatoria y diferente en cada iteración. A partir de este cruce, se obtienen dos nuevos individuos, obtenidos de la mezcla de los dos primeros, que serán incluidos en la población, simulados y evaluados.

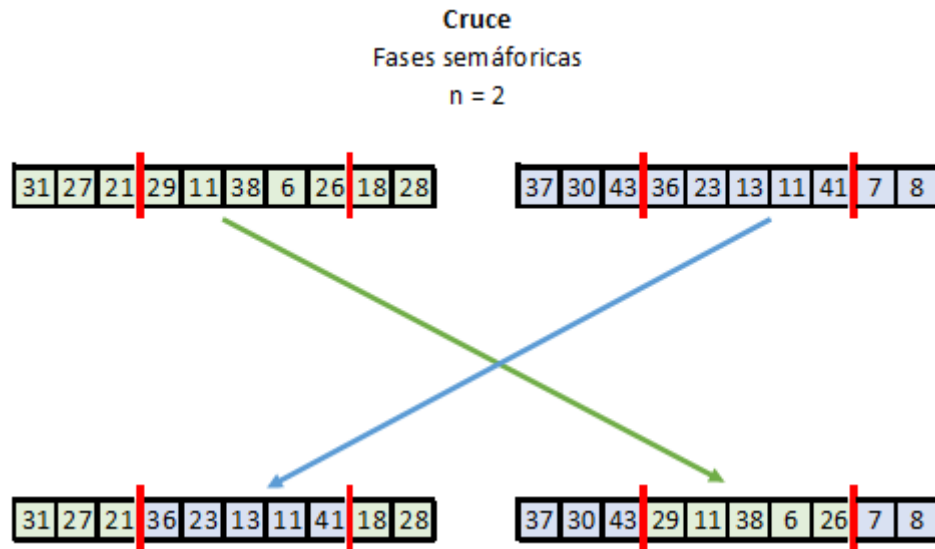


Figura 32 – Ejemplo de cruce

Fuente: Elaboración propia

Como ejemplo, la figura 32 representa el cruce de la parte correspondiente a las fases semafóricas de dos individuos, con dos puntos de cruce. De esta combinación surgen dos nuevos individuos a simular, evaluar e introducir en la población. Se mantendrán en la misma aquellas cuyas puntuaciones se encuentren entre las i -ésimas mejores, siendo i el tamaño de la población.

Mutación

A partir de un individuo, y también respecto de los datos a incluir en los cálculos, se modificará una cantidad de genes igual a la que se indique, sin superar en ningún caso el 25% de estos, para evitar que se pierda excesiva información del individuo de partida. Los genes a mutar, es decir, la posición de estos, también se determinará aleatoriamente y variará de una iteración a otra.

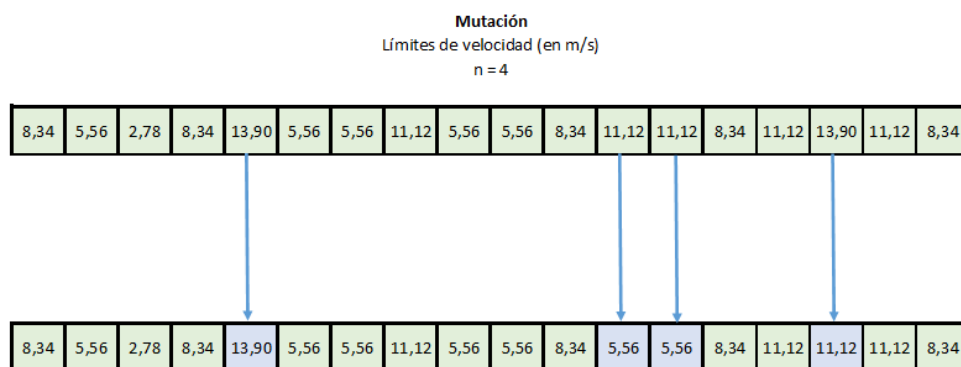


Figura 33 – Ejemplo de mutación

Fuente: Elaboración propia

En la figura 33 se muestra la mutación de cuatro genes relativos a los límites de velocidad, de cada calle del escenario, de una solución. Estos cuatro genes se han elegido al azar y sus nuevos valores también se han determinado de manera aleatoria, dados un rango de posibles valores, mínimo y máximo, y una multiplicidad. Para este ejemplo en concreto, los valores se situarán entre 10 y 50 km/h, y serán múltiplos de 10 km/h. Se convierten en m/s, ya que es la unidad de medida de velocidad que utiliza SUMO.

Fitness

Atendiendo a los objetivos planteados en la introducción, la función *fitness* evaluará las mejoras de los parámetros a optimizar, comparados con los datos obtenidos en una solución base, realizada con los datos originales del escenario, cuya puntuación será igual a 100 puntos sea cuales sean los parámetros elegidos para optimizar. De esta forma, valores superiores a este indicarán una mejora, asimilable a un incremento porcentual, respecto a los datos de partida, y un empeoramiento en caso contrario.

Para la ocasión en el que se seleccionen los tres parámetros disponibles a optimizar, es decir, el problema a resolver es multiobjetivo, la ponderación de las distancias recorridas será un 70%, frente a un 10% de los tiempos de viaje y un 20% de los límites de velocidad.

De esta manera, las distancias recorridas se valorarán:

$$D = \frac{d_{media_{SOLUCIÓN}}}{d_{media_{BASE}}}$$

Y los tiempos efectivos de viaje y tiempos perdidos como:

$$T = 0,5 * \frac{t_{efectivo_{BASE}}}{t_{efectivo_{SOLUCIÓN}}} + 0,5 * \frac{t_{perdido_{BASE}}}{t_{perdido_{SOLUCIÓN}}}$$

En dónde:

$$t_{efectivo} = t_{total} - t_{perdido}$$

Por último, los límites de velocidad se puntuarán:

$$V = \frac{\sum \lim_{vel_{BASE}}}{\sum \lim_{vel_{SOLUCIÓN}}}$$

Por tanto, con los pesos indicados para cada sumando:

$$fitness = 100 * \left(\frac{\alpha}{\alpha + \beta + \gamma} * D + \frac{\beta}{\alpha + \beta + \gamma} * T + \frac{\gamma}{\alpha + \beta + \gamma} * V \right)$$

$$fitness = \frac{100}{\alpha + \beta + \gamma} * (\alpha * D + \beta * T + \gamma * V)$$

Siendo los coeficientes: $\alpha = 7,0$, si se selecciona la optimización de distancias, $\beta = 1,0$, si se selecciona optimizar los tiempos, y $\gamma = 2,0$ para la optimización de los límites de velocidad. En caso de no ser marcados, el coeficiente respectivo (α , β , γ) será igual a cero. De esta forma se

mantiene la puntuación de la solución base siempre igual a 100 puntos, como nivel de comparación.

Como se puede observar en el desglose de las fórmulas, según estas sean de maximización o minimización, los valores de la solución base y de la solución a evaluar intercambian posiciones entre numerador y denominador. En la maximización, la solución a evaluar se sitúa en el numerador para que, a mayor número, mayor puntuación. De forma inversa, para la minimización, los valores de la solución se sitúan en el denominador para que, a menor cantidad, mayor puntuación.

Muy importante a destacar es que, seleccionar únicamente la optimización de límites de velocidad hagan que el algoritmo converja estableciendo para todas las calles el límite mínimo de velocidad, es decir, todos los límites de velocidad tenderán a 10 km/h.

Generación de listas aleatorias

Para la creación de individuos nuevos, se crearán tantas listas de valores aleatorios como elementos seleccionados para incluir en los cálculos. Así, se crearán listas de cantidad especificada, entre unos valores máximos y mínimos y múltiplos de un valor dado.

- Para los límites de velocidad, existirán tantos valores como calles en el escenario, con un mínimo de 10 km/h y el máximo que se introduzca a través de la interfaz gráfica. Los distintos valores del listado serán múltiplos de 10 km/h.
- En cuanto a las fases de los semáforos, habrá tantos elementos en la lista como el total de fases distintas de ámbar en todas las intersecciones del escenario. El valor mínimo será de 3 segundos (como el valor constante de las fases en ámbar) y el valor máximo será el introducido por el usuario. Los valores serán múltiplos de un segundo.

La longitud de cada una de las listas se obtiene al computar la solución base, ya que los tamaños de estas no variarán a lo largo de la ejecución, aunque sí lo harán sus valores.

Codificación

Dado un número que identifique a la solución y un escenario en formato *xml*, se codificará dicho escenario como un individuo que constará de los siguientes atributos: identificador, puntuación tras la simulación según la función *fitness* (por defecto será igual a $-\infty$), y un listado por cada uno de los elementos de cálculo: límites de velocidad y/o fases semafóricas. Para la creación de estos listados, y que la correspondencia sea unívoca, se ordenarán según el orden de aparición de la etiqueta dentro del archivo *xml* del escenario, debido a que este orden permanece constante durante toda la ejecución.

Decodificación

Es el proceso inverso a la codificación mediante el cual, a partir de un individuo, se trasladan sus datos a un escenario en formato *xml* para poder ser simulado por SUMO. Análogamente a como sucede con la codificación, para mantener la relación unívoca entre los elementos del listado y

los del escenario, cada límite de velocidad en el listado corresponderá con la calle ubicada en la misma posición de la etiqueta en el archivo *.xml* del escenario, como también sucede para las fases de los semáforos del listado.

Simulación y evaluación

A partir de un archivo de configuración de la simulación, con las referencias a los archivos con la información de escenario y rutas, se procede a la simulación y a la obtención de los datos estadísticos. Con ellos, se puntuará con la función *fitness* y será añadido al conjunto de la población.

Además, se mantendrán sendos listados: uno donde se añaden sucesivamente las mayores puntuaciones obtenidas hasta ese instante y el identificador, que coincide con el ordinal de creación de este, al que corresponde la puntuación. Esto servirá para ver gráficamente la evolución de las mejores puntuaciones según se desarrolla el algoritmo.

Durante el proceso de simulación y evaluación de los escenarios, se hará uso de los siguientes archivos, según su extensión:

Extensión	Contenido
<i>.net.xml</i>	Escenario original, generado automáticamente o desde un mapa importado de <i>OpenStreetMaps</i> .
<i>.test.net.xml</i>	Escenario de la solución cuya simulación y evaluación está en curso.
<i>.BEST.net.xml</i>	Escenario de la mejor solución alcanzada hasta el momento.
<i>.sumocfg</i>	Configuración para lanzar la simulación del escenario original.
<i>.test.sumocfg</i>	Configuración para lanzar la simulación del escenario con la simulación y evaluación en curso.
<i>.BEST.sumocfg</i>	Configuración para lanzar la mejor solución alcanzada.
<i>.trips.xml</i>	Información de los puntos de origen y destino de las rutas, y del instante en que se iniciarán.
<i>.tripInfo.xml</i>	Información detallada de los trayectos de cada vehículo para la obtención de estadísticas del escenario original.
<i>.test.tripInfo.xml</i>	Información detallada de los trayectos de cada vehículo para la obtención de estadísticas del escenario de la simulación y puntuación en curso.
<i>.tripStats.xml</i>	Estadísticas del escenario original.
<i>.test.tripStats.xml</i>	Estadísticas del escenario en curso.
<i>.BEST.tripStats.xml</i>	Estadísticas de la mejor solución.
<i>.rou.xml</i>	Información de las rutas, calle a calle, entre los puntos inicial y final.
<i>.rou.alt.xml</i>	Archivo intermedio para generar rutas continuas.
<i>.contRou.xml</i>	Información de las rutas continuas en el lapso de la simulación.

Tabla 05 – Archivos de la aplicación

Fuente: Elaboración propia

4.7 Generación y evaluación de la población inicial

Se crearán tantas soluciones como haya indicado el usuario, codificadas. El identificador corresponderá con el orden de generación de la solución, y empezará por el número uno, ya que el cero se ha reservado para la solución base. Cada una de estas soluciones se simulará y evaluará, añadiéndose a la población. Una vez completada la población, esta será ordenada, de mayor a menor valoración.

Estas soluciones serán el punto de partida para el algoritmo genético. Se les aplicarán las funciones de cruce y mutación, además de la evaluación: simulación y puntuación. Todas estas funciones comentadas en el punto anterior.

4.8 Evolución de la población

Se procederá a evolucionar, de forma cíclica, la población inicial previamente creada, siguiendo un ciclo como el siguiente:

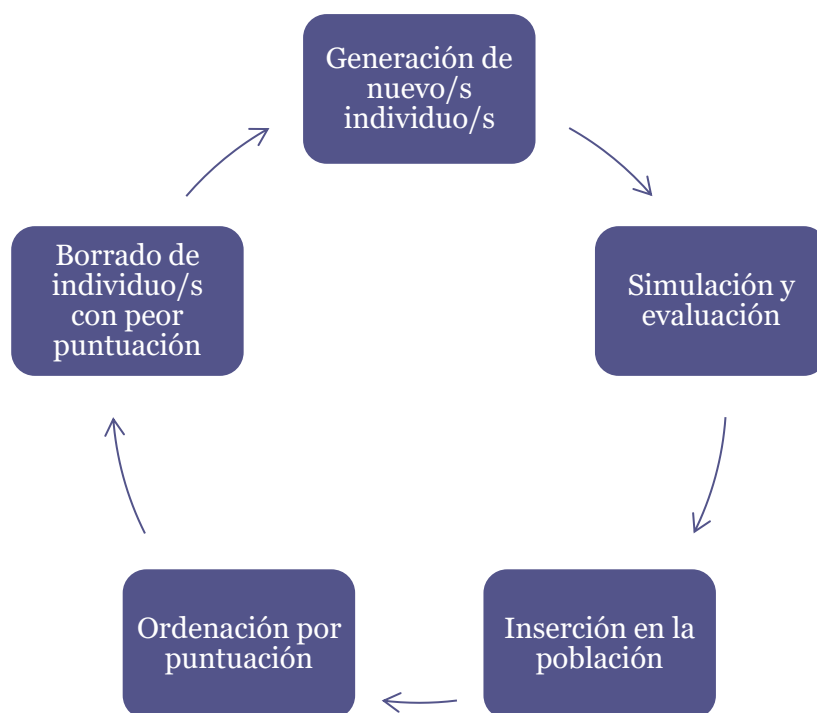


Figura 34 – Evolución de la población

Fuente: Elaboración propia

En primer lugar, se determinará, al azar, la forma de obtener nuevos individuos: por cruce, mutación o creación de nuevos, con las probabilidades indicadas por el usuario a través de la interfaz gráfica. Para la generación de un nuevo individuo, se realizará el mismo proceso que para la creación de un individuo de la población inicial, y el objeto de su inclusión es explorar todo el espacio de búsqueda para evitar alcanzar óptimos locales en vez del óptimo global. En el caso de

la mutación, se seleccionará un individuo de forma aleatoria, equiprobable o ponderada, y se mutarán sus genes, obteniéndose una nueva solución. Si la obtención de soluciones es mediante cruce, se seleccionarán dos individuos como se ha descrito para el caso de la mutación, obteniendo dos nuevas soluciones, con los genes intercambiados de los dos individuos seleccionados y con tantos fragmentos de cada uno de ellos como se haya indicado a través de la interfaz gráfica.

Para todos los casos, las soluciones obtenidas se simularán y evaluarán, pasando a formar parte de la población. Esta se ordenará y se limitará al número de soluciones que se haya establecido, eliminando aquellas con peor puntuación, repitiendo el proceso con aquellas soluciones, en principio, más prometedoras por tener mejor puntuación.

Este proceso se realizará tantas veces como iteraciones haya indicado el usuario. Además, desde la iteración en la que se haya obtenido la mejor solución, se realizarán, al menos, iteraciones de cantidad igual a dos veces el tamaño de la población, para permitir que puedan generar, mediante cruce o mutación, soluciones mejores a partir de esta.

4.9 Gráfico de evolución

Finalmente se obtiene, además del archivo *xml* del escenario que mejor puntuación ha obtenido y sus datos estadísticos, un gráfico con la evolución de la ejecución, es decir, cómo ha mejorado la puntuación máxima en el conjunto de la población respecto al número de soluciones generadas. Como, por ejemplo:

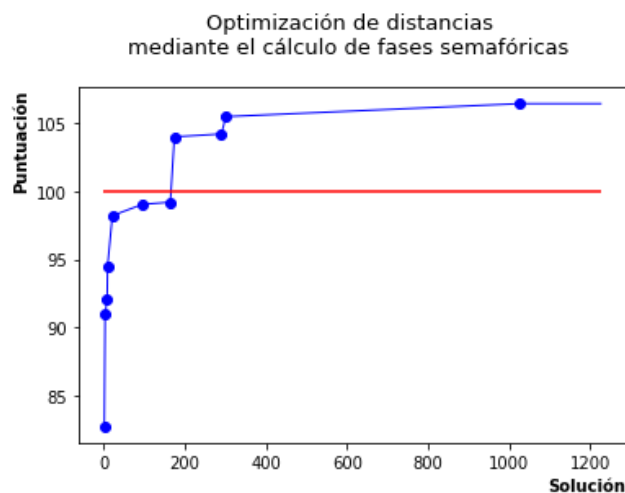


Figura 35 – Gráfico de evolución de la puntuación

Fuente: Elaboración propia

En el gráfico se pueden observar las mejoras sucesivas en la valoración, marcada con puntos en azul, y unidos por una línea para observar mejor su progreso. La línea horizontal en rojo muestra la solución base, usada como referencia.

En este ejemplo, con una población de 150 individuos y 1.000 iteraciones, los puntos entre 1 y 150 son los obtenidos aleatoriamente al crear la población inicial, los puntos entre 151 y 1.150 son los creados durante la evolución de la población. Además, se da el caso de que se ha obtenido

un individuo con máxima puntuación en la iteración 876 a la que, sumando la población inicial, obtenemos un identificador igual a 1.026. Como $1.000 < 876 + 2 * 150$, se incrementarán las iteraciones en $2 * 150 = 300$ desde esta, es decir, hasta la iteración 1.176. Se habrán creado, por tanto, 1.326 soluciones, al sumar a las iteraciones, la cantidad de la población inicial.

El gráfico corresponde a los datos siguientes:

id.	Puntuación	id.	Puntuación
1	82,738 puntos	174	103,999 puntos
3	90,970 puntos	290	104,209 puntos
8	92,034 puntos	301	105,481 puntos
9	94,472 puntos	1.126	106,434 puntos
21	98,214 puntos	1.376	106,434 puntos
95	99,041 puntos		
164	99,203 puntos		

Tabla 06– Evolución de la puntuación

Fuente: Elaboración propia

5. Resultados

Este apartado desarrolla cómo se debería proceder para evaluar un escenario a través de un sencillo ejemplo. Este procedimiento es extensible a cualesquiera otros ejemplos que se deseen analizar. Constará de dos partes bien diferenciadas:

- En primer lugar, se realizarán una serie de pruebas para determinar cuáles son los mejores valores de los hiperparámetros del algoritmo genético, ya que estos son distintos para cada escenario en cuestión. Para ello, se mantendrán fijos el número de vehículos y el tiempo a simular, para que los resultados sean comparables entre sí.
- Posteriormente, se realizarán distintas comparaciones de resultados respecto del anterior escenario: optimizando únicamente las distancias medias recorridas, empleando otros escenarios asimilables y con este mismo escenario, pero con mayor congestión de tráfico. Se realizará una discusión comparativa de sus desempeños.

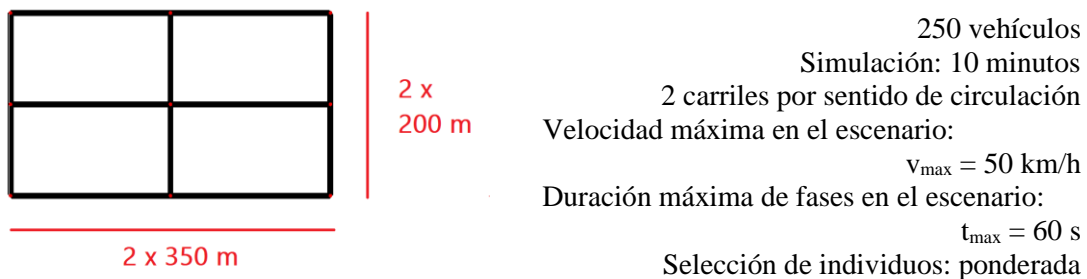


Figura 36 – Escenario para la obtención de hiperparámetros

Fuente: Elaboración propia

Se empleará un ejemplo sencillo, el cual consta de una cuadrícula de 3 x 3 calles, que conforman un total de 5 intersecciones y 24 calles. Este escenario consta de unos 6,60 km de calles, aproximadamente, con el doble de esta longitud de viales (carriles de circulación).

Según se refiere como recomendación en la bibliografía (24), el número de puntos de cruce se establecerá en dos. El número de iteraciones, inicialmente se fija en un valor elevado (3.000), de tal forma que pueda ajustarse posteriormente a la baja hasta aquel en que, o bien no se producen mejoras en la puntuación o estas son mínimas. El tamaño de la población también se establecerá en un valor elevado (500 soluciones), de forma que se creen al menos el doble de este número de soluciones desde la aparición de la mejor puntuación, por ser este el criterio de finalización del programa. Este valor también se ajustará posteriormente, de forma que se utilicen los recursos necesarios.

Se emplearán en los cálculos todos los datos disponibles: tanto los tiempos de fases semafóricas como límites de velocidad de las calles. Se optimizarán también todos aquellos parámetros susceptibles de mejora, es decir, las distancias medias recorridas, los tiempos de viaje efectivos junto con los tiempos perdidos, y los límites de velocidad de las calles.

5.1 Hiperparámetros

Se realizarán diferentes ejecuciones, modificando únicamente el parámetro que se quiere establecer, manteniendo el resto constantes.

Los hiperparámetros a fijar son:

- Porcentaje de individuos obtenidos mediante cruce de dos soluciones.
- Porcentaje de individuos obtenidos mediante mutación de una solución.
- Porcentaje de individuos nuevos, obtenidos aleatoriamente.
- Número de iteraciones.
- Tamaño de la población.

Probabilidades de mutación, cruce y generación de individuos nuevos.

La bibliografía sugiere probabilidades de cruce altas y de mutación bajas, aunque hay disparidad en la determinación de estos valores. La probabilidad de cruce oscila entre 40 – 50% (26) y el 60% (27). Para el caso de la probabilidad de mutación, la variación es mucho mayor, entre el 0,1% (27) y el 20% (26).

Por el contrario, sí que existe mayor consenso entre distintos autores en cuanto a la cantidad de puntos de cruce y de mutaciones. Se fijará, por tanto, el número de puntos de cruce en dos y el número de mutaciones en una, para mantener una gran cantidad de información de las soluciones de partida y que sea transmitida a las siguientes soluciones a evaluar.

No se hace referencia a la probabilidad de generación de individuos nuevos que, como se ha referido en este trabajo, se incluye para posibilitar la exploración de todo el espacio de búsqueda, intentando evitar alcanzar óptimos locales y no globales. Por ello, y debido a que la probabilidad de generación de una solución aleatoria y que además sea prometedora es baja, más cuanto más evolucione la población, analizaremos valores menores al 20%. Obviamente, el usuario podrá modificar cualesquiera de estos valores.

Se realizan diversas ejecuciones variando los porcentajes de probabilidad de cruce, mutación y generación de nuevos individuos, y obteniendo sus puntuaciones. El objetivo es determinar qué valores de probabilidades arrojan mayor puntuación. Se toman valores de probabilidad distantes a las recomendaciones que aparecen en la bibliografía para mostrar que, efectivamente, estos producen puntuaciones no óptimas. Así, la probabilidad de cruce se situará en valores entre 30 y 100%, variando de 10 en 10%. A su vez, la probabilidad de mutación toma valores entre 0 y 50%, con la misma variación. La generación de nuevos individuos (creación) variará entre 0, 10 y 20%. La suma de todas las probabilidades en cada ejecución es igual al 100%.

De esta manera obtenemos las siguientes puntuaciones:

Cruce	Probabilidad		Puntuación
	Mutación	Creación	
100 %	0 %	0 %	113,127 puntos
90 %	10 %	0 %	117,810 puntos
80 %	20 %	0 %	119,261 puntos
70 %	30 %	0 %	112,122 puntos
60 %	40 %	0 %	114,220 puntos
50 %	50 %	0 %	110,531 puntos

Tabla 07.a – Puntuación según probabilidades I

Fuente: Elaboración propia

Cruce	Probabilidad		Puntuación
	Mutación	Creación	
90 %	0 %	10 %	112,080 puntos
80 %	10 %	10 %	115,496 puntos
70 %	20 %	10 %	117,156 puntos
60 %	30 %	10 %	113,244 puntos
50 %	40 %	10 %	113,397 puntos
40 %	50 %	10 %	110,168 puntos

Tabla 07.b – Puntuación según probabilidades II

Fuente: Elaboración propia

Cruce	Probabilidad		Puntuación
	Mutación	Creación	
80 %	0 %	20 %	113,162 puntos
70 %	10 %	20 %	114,467 puntos
60 %	20 %	20 %	111,464 puntos
50 %	30 %	20 %	112,079 puntos
40 %	40 %	20 %	111,331 puntos
30 %	50 %	20 %	109,132 puntos

Tabla 07.c – Puntuación según probabilidades III

Fuente: Elaboración propia

Representando gráficamente estos datos para su mejor interpretación:

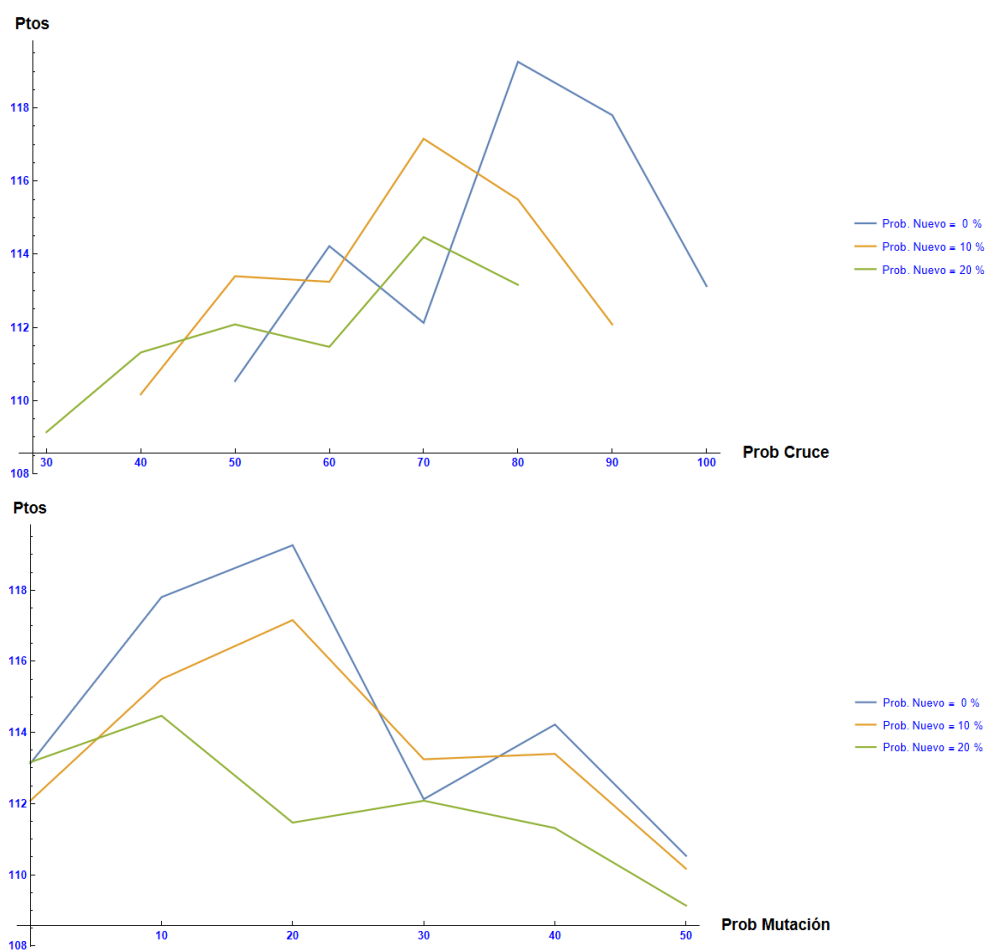


Figura 37 – Puntuación según probabilidad de cruce, mutación y creación

Fuente: Elaboración propia

A grandes rasgos, podemos determinar que las mayores puntuaciones se consiguen con mayor probabilidad de cruce, menor probabilidad de mutación y sin insertar en la población nuevos individuos creados de forma aleatoria. La mayor puntuación obtenida corresponde a unos valores iguales a:

- Probabilidad de cruce: 80%
- Probabilidad de mutación: 20%
- Probabilidad de creación: 0%

Además, existe otro punto con una puntuación próxima al máximo, con probabilidades de cruce de 70%, mutación de 20% y creación de 10%. Para confirmar que entre ambos no existe un óptimo mejor, realizamos una prueba adicional para comprobarlo, interpolando los valores de probabilidad que varían entre ambas valoraciones, marcados en rojo en la siguiente tabla:

Cruce	Probabilidad		Puntuación
	Mutación	Creación	
80 %	20 %	0 %	119,261 puntos
75 %	20 %	5 %	115,294 puntos
70 %	20 %	10 %	117,156 puntos

Tabla 07.d – Puntuación según probabilidades IV

Fuente: Elaboración propia

Como no se ha obtenido una mejora en la puntuación, los valores que se fijarán para las probabilidades de generación de soluciones son: 80% para el cruce, el 20% para la mutación y no se incluirán nuevas soluciones aleatorias en la población.

Para el ejemplo en estudio, podemos extraer las siguientes conclusiones generales:

- La probabilidad de cruce tiene un máximo de puntuación en el 80%, como máximo global, y en el 70%, como máximo local, decreciendo para valores mayores al primero y menores al segundo.
- De forma similar, para unas probabilidades de mutación en torno al 20% arroja las mayores puntuaciones.
- Finalmente, la probabilidad de creación de nuevas soluciones tiene el mejor desempeño para un valor igual al 0%, reduciéndose, generalmente, para valores superiores.

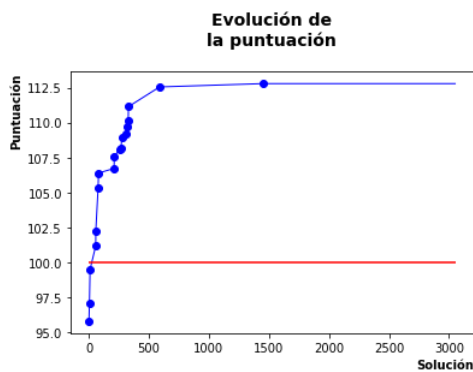
Ha sido necesario realizar al menos 3.000 iteraciones en cada experimento para la obtención de estos valores, con una población de 500 soluciones. Ambos números se ajustan a continuación.

Tamaño de la población. Número de iteraciones

Una vez fijados los valores de selección de individuos para el escenario en estudio, se realizan ejecuciones con distintos tamaños de población. De aquí podremos obtener tanto el tamaño adecuado de la población como el número de iteraciones a efectuar, atendiendo al momento en que se haya obtenido la máxima puntuación.

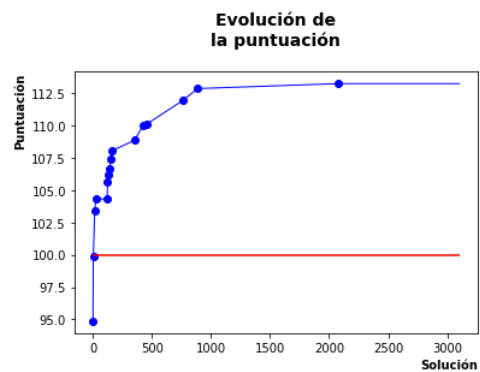
Si fuera necesario, habrá que sopesar si, dado el caso, es conveniente incrementar las iteraciones para mejorar la evaluación de la solución. Hay que tener en cuenta que se realiza un paso de ejecución en el simulador por cada segundo de lapso a simular, y que puede conllevar entre 2 y 7 milisegundos de tiempo de ejecución cada uno de estos pasos. En ocasiones, incrementar el número de iteraciones eleva el tiempo total de ejecución del programa para mejoras despreciables en la puntuación del escenario en estudio.

Los tamaños de población que se han evaluado son: 50, 100, 150, 200, 250, 300, 350 y 400 soluciones, para analizar los resultados obtenidos, obteniendo las gráficas de evolución de puntuación siguientes:



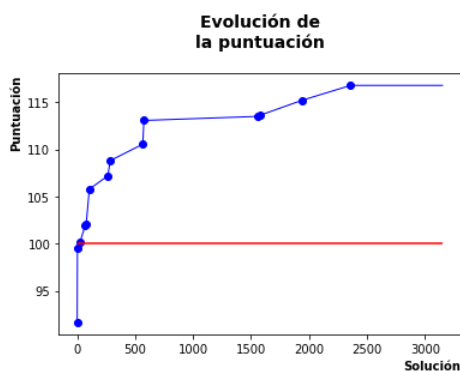
112,789 puntos
Id: 1.449

Figura 38.a – Puntuación: población = 50 soluciones



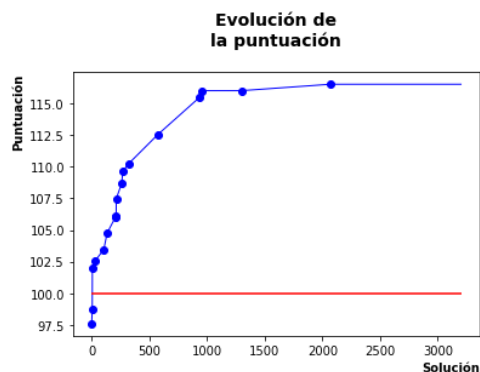
113,222 puntos
Id: 2.080

Figura 38.b – Puntuación: población = 100 soluciones



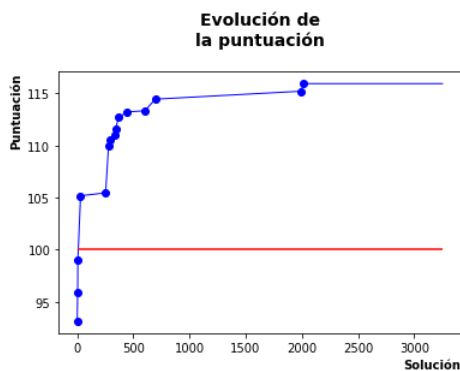
116,782 puntos
Id: 2.359

Figura 38.c – Puntuación: población = 150 soluciones



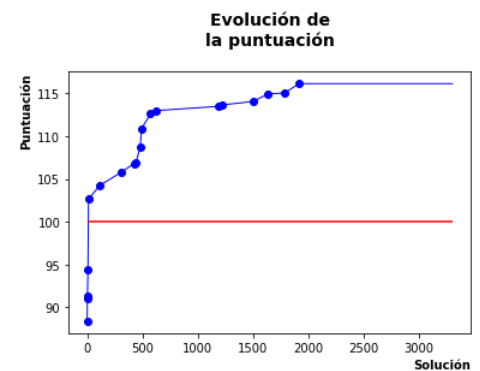
116,510 puntos
Id: 2.069

Figura 38.d – Puntuación: población = 200 soluciones



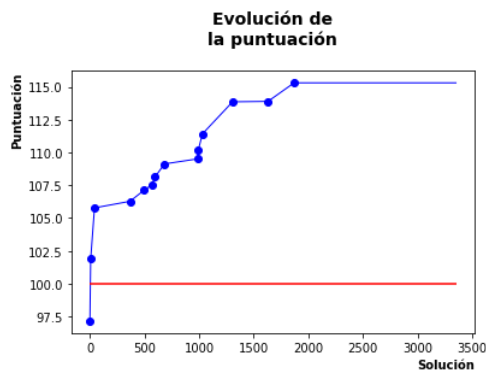
115,936 puntos
Id: 2.018

Figura 38.e – Puntuación: población = 250 soluciones



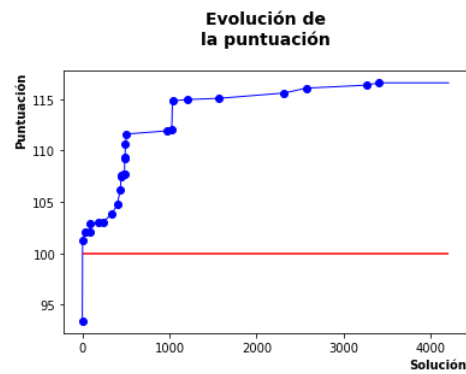
116,129 puntos
Id: 1.915

Figura 38.f – Puntuación: población = 300 soluciones



115,302 puntos
Id: 1.870

Figura 38.g – Puntuación: población = 350 soluciones



116,581 puntos
Id: 3.282

Figura 38.h – Puntuación: población = 400 soluciones

Fuente: Elaboración propia

A partir de la población inicial generada de manera aleatoria, se obtienen los datos de evolución en la puntuación en cada iteración. De las gráficas, anteriores no sólo extraemos el mejor tamaño de la población para el escenario. Al haber fijado un número de iteraciones deliberadamente elevado, podemos también fijar este parámetro. Debe existir un compromiso entre la puntuación obtenida y los recursos empleados.

A partir de un tamaño de población mayor o igual a 150 soluciones, la puntuación obtenida se sitúa en torno a los 116 puntos. Las mayores han sido 116,782, 116,581 y 116,510 puntos, para tamaños de 150, 400 y 200 individuos, respectivamente.

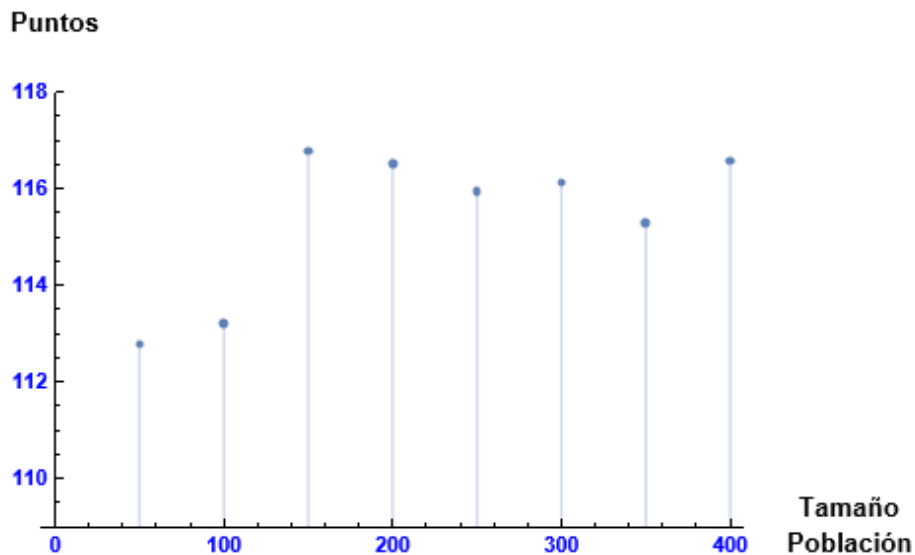


Figura 39 – Puntuación según tamaño de población

Fuente: Elaboración propia

A su vez, el número de soluciones generadas, correspondiente al número que identifica la mejor solución más dos veces el número de soluciones en la población inicial, para alcanzar esta valoración es aproximadamente 2.500, por lo que tomaremos este valor como número de

iteraciones, al que, añadiendo el tamaño inicial de la población, será el número total de individuos a generar.

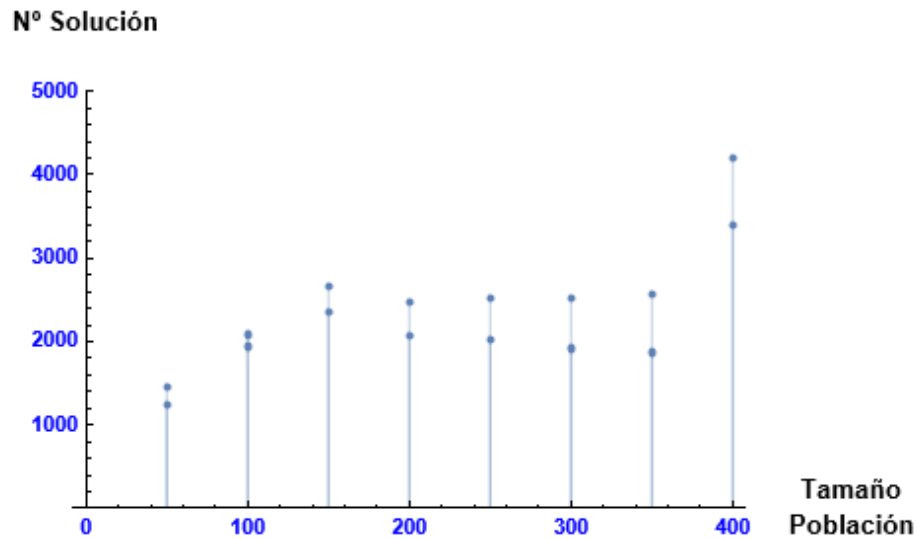


Figura 40 – Soluciones generadas según tamaño de población

Fuente: Elaboración propia

En la figura 40 se muestra, por cada tamaño de población, el número de soluciones creadas hasta alcanzar la puntuación máxima y, además, este mismo número incrementado en dos veces el tamaño de la propia población, establecida como condición de parada del programa. De esta manera se pueden comparar el número total de soluciones que se crean en cada ejecución.

Por último, analizando la ratio entre la valoración obtenida frente al número de soluciones creadas, para aquellos tamaños de población con mayores puntuaciones (150, 200 y 400), es ligeramente superior para un tamaño de población inicial de 200 soluciones.

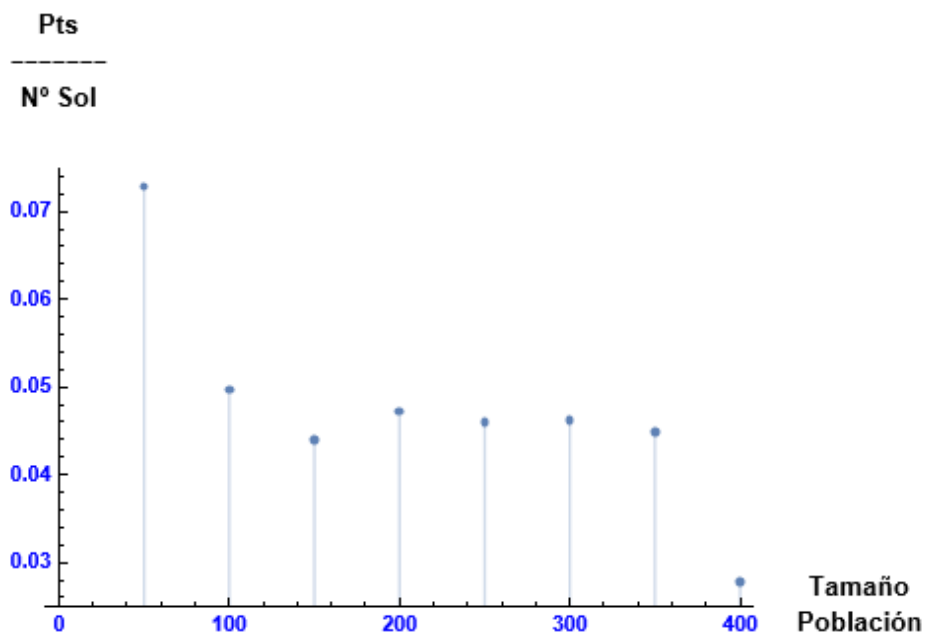


Figura 41 – Puntuación vs. Soluciones generadas

Fuente: Elaboración propia

Por tanto, y teniendo en cuenta todos los datos aquí expuestos, referidos al número de soluciones y su valoración, y que el programa no es determinista, es decir, que los resultados varían entre distintas ejecuciones, fijaremos el tamaño de la población inicial en 200 soluciones, y se realizará una evolución de esta población a través de 2.500 iteraciones.

5.2 Comparaciones

A continuación, analizaremos las mejoras del problema planteado respecto de la solución inicial, y se analizarán también los resultados de otras dos posibles alternativas. Como las alternativas a estudiar serán escenarios equivalentes al primero, y para no alargar en exceso el presente trabajo, se tomarán los mismos valores de los hiperparámetros que para el escenario de partida.

Para estas alternativas, sería necesario volver a estudiarlos, ya que, como se ha indicado, son diferentes para cada escenario. Esto es debido a que, para diferentes escenarios el genotipo es distinto. Dicho de otra manera, los datos del escenario que se incluyen en los cálculos que realiza el algoritmo genético no son iguales, y podría existir una mejor combinación de probabilidades de cruce, mutación y creación, así como el tamaño de la población y el número de iteraciones que sería necesario realizar en cada caso. Es obvio que, a distinto número de parámetros, distinta cantidad de posibles combinaciones, de aquí que, posiblemente, los valores de los hiperparámetros sean distintos.

Por otra parte, los valores obtenidos son acordes a las recomendaciones dadas en la bibliografía (26; 27). Por ello se tomarán los datos ya obtenidos de los hiperparámetros, aunque un estudio pormenorizado probablemente, por las razones expuestas en el párrafo precedente, podría alcanzar una optimización mejor.

Mejora de la solución base

Para el escenario de partida, con los límites de velocidad y fases de semáforos establecidos por defecto por el simulador, se obtienen los siguientes datos estadísticos, que emplearemos en la función *fitness*:

Distancia media recorrida	2.147,85 m
Tiempo total de viaje	533,73 s
Tiempo perdido	375,35 s

Tabla 08 – Estadísticas de la solución base I

Fuente: Elaboración propia

Una vez optimizados los tiempos de las fases semafóricas y las velocidades de las calles, los datos que resultan son:

Distancia media recorrida	2.343,16 m
Tiempo total de viaje	547,31 s
Tiempo perdido	271,29 s

Tabla 09 – Estadísticas de la solución optimizada I

Fuente: Elaboración propia

- Velocidad media de los desplazamientos

Se han aumentado la distancia recorrida y el tiempo total de viaje, y se han reducido el tiempo perdido en el trayecto. Así, si comparamos las velocidades medias en ambas situaciones, base y optimizada, resulta:

$$\overline{v_{base}} = \frac{\frac{2.147,85 \text{ m}}{1.000 \text{ m/km}}}{\frac{533,73 \text{ s}}{3.600 \text{ s/h}}} = 14,487 \text{ km/h}$$

$$\overline{v_{opt}} = \frac{\frac{2.343,16 \text{ m}}{1.000 \text{ m/km}}}{\frac{547,31 \text{ s}}{3.600 \text{ s/h}}} = 15,412 \text{ km/h}$$

Por tanto, hemos mejorado los desplazamientos, compensando el aumento en el tiempo total con una mayor distancia recorrida.

- Velocidad media en movimiento

Si repetimos esta misma operación, con los tiempos efectivos de viaje, es decir, los tiempos en que realmente se están desplazando los vehículos, que son los tiempos totales menos los tiempos perdidos, obtenemos:

$$\overline{v_{base}}' = \frac{\frac{2.147,85 \text{ m}}{1.000 \text{ m/km}}}{\frac{(533,73 - 375,35) \text{ s}}{3.600 \text{ s/h}}} = 48,821 \text{ km/h}$$

$$\overline{v_{opt}}' = \frac{\frac{2.343,16 \text{ m}}{1.000 \text{ m/km}}}{\frac{(547,31 - 271,29) \text{ s}}{3.600 \text{ s/h}}} = 30,561 \text{ km/h}$$

Por lo que hemos mejorado también tanto la emisión de ruidos como la seguridad al reducir la velocidad real de los desplazamientos.

Optimización de distancias

Si seleccionamos únicamente la optimización de las distancias recorridas, modificando tanto los tiempos de fases de los semáforos como los límites de velocidad, resulta una distancia media recorrida de 2.354,75 m. Tan sólo es un 0,5% mejor que la optimización del punto anterior. Se ha obtenido un tiempo total de 543,30 s, con un tiempo perdido de 311,51 s. Más próximos a la solución base, obviamente debido a que no se ha incluido en la función para su optimización. Aunque la mejora en la distancia recorrida no es importante respecto de la optimización del punto anterior, al menos sí garantizamos que se realice, en vez de primar a otras optimizaciones, como los tiempos o los límites de velocidad.

Comparación con red mallada

Seguidamente, repetiremos la experimentación con una red de la misma tipología, con el doble del número de calles tanto en horizontal como en vertical, pero con un único carril de circulación (en lugar de dos). No variaremos los hiperparámetros, por ser un escenario similar y por no repetir el proceso del apartado anterior “5.1 Hiperparámetros” que no aporta nada nuevo a este trabajo, aunque, si se tratase de un estudio real de tráfico, y no una mera exposición de características del programa, sería imprescindible hacerlo.

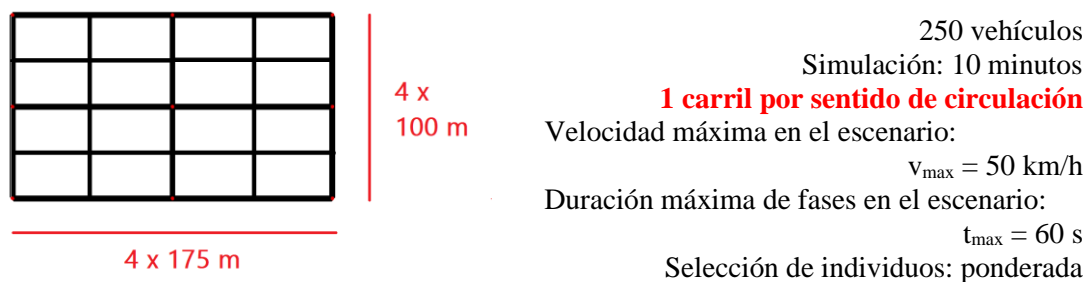


Figura 42 – Comparación con red mallada

Fuente: Elaboración propia

La solución base para este escenario arroja los siguientes resultados:

Distancia media recorrida	2.219,53 m
Tiempo total de viaje	587,37 s
Tiempo perdido	420,76 s

Tabla 10 – Estadísticas de la solución base II

Fuente: Elaboración propia

Y una vez optimizados:

Distancia media recorrida	2.097,07 m
Tiempo total de viaje	588,38 s
Tiempo perdido	312,66 s

Tabla 11 – Estadísticas de la solución optimizada II

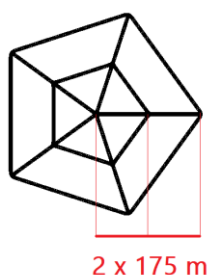
Fuente: Elaboración propia

Para este caso, la mejora radica en la disminución de los tiempos perdidos en los desplazamientos, aunque los tiempos totales sean mayores y se haya reducido ligeramente la distancia media recorrida. Esto es debido a que se han aumentado de forma importante el número de intersecciones, pasando de cinco en el escenario original a veintiuna en este, aunque sean más sencillas, al confluir únicamente un carril por cada calle y sentido de circulación, en lugar de dos.

En este punto, sería interesante analizar este escenario optimizando únicamente las distancias recorridas, y comparar sus datos estadísticos. No se abordará en este trabajo ya que una nueva comparación de datos tampoco se considera relevante. Por tanto, a la vista de los resultados obtenidos, podemos concluir que es preferible el escenario de partida al analizado en este apartado.

Comparación con red radial

Otra posible comparación sería cambiando la tipología de la red. Respecto del escenario original, con una red radial, con 5 vías radiales y dos vías concéntricas que distan entre sí 175 m, obtenemos una red con la misma longitud de viales y ocupando un área similar. Como en el caso anterior, mantendremos los hiperparámetros.



250 vehículos
 Simulación: 10 minutos
 2 carriles por sentido de circulación
 Velocidad máxima en el escenario:
 $v_{\max} = 50 \text{ km/h}$
 Duración máxima de fases en el escenario:
 $t_{\max} = 60 \text{ s}$
 Selección de individuos: ponderada

Figura 43 – Comparación con red radial

Fuente: Elaboración propia

De forma análoga a los puntos previos, la solución base para este escenario arroja los siguientes resultados:

Distancia media recorrida	1.767,62 m
Tiempo total de viaje	578,44 s
Tiempo perdido	447,14 s

Tabla 12 – Estadísticas de la solución base III

Fuente: Elaboración propia

Y una vez optimizados:

Distancia media recorrida	1.564,66 m
Tiempo total de viaje	579,21 s
Tiempo perdido	336,13 s

Tabla 13 – Estadísticas de la solución optimizada III

Fuente: Elaboración propia

Al igual que en el punto anterior, la mejora radica en la disminución de casi un 25% en los tiempos perdidos. Las distancias medias recorridas, una vez realizada la optimización, son un 33% inferiores al escenario de partida. Por tanto, podemos concluir que esta tipología es completamente descartable.

Aumento de la congestión

Como hemos visto, el escenario de partida sería el que mejores resultados obtiene. Por ello, en último lugar, estudiaremos el comportamiento del tráfico en este escenario, pero aumentando el número de vehículos en circulación al doble, es decir, hasta 500 vehículos.

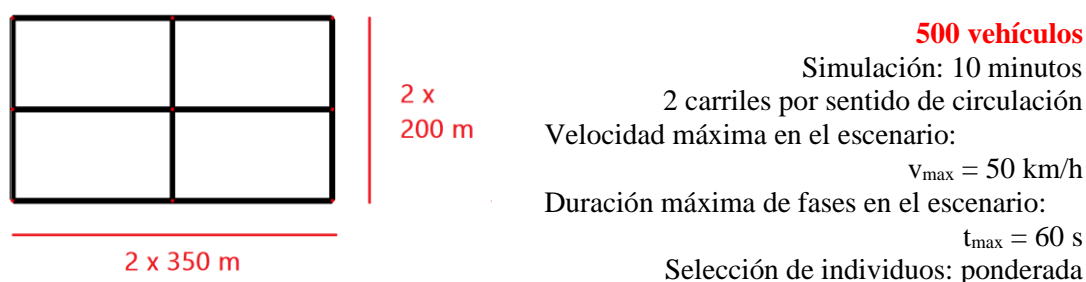


Figura 44 – Comparación con mayor congestión

Fuente: Elaboración propia

Obteniendo los siguientes datos:

Distancia media recorrida	1.349,86 m
Tiempo total de viaje	488,63 s
Tiempo perdido	388,57 s

Tabla 14 – Estadísticas de la solución base IV

Fuente: Elaboración propia

Distancia media recorrida	1.275,90 m
Tiempo total de viaje	499,81 s
Tiempo perdido	310,96 s

Tabla 15 – Estadísticas de la solución optimizada IV

Fuente: Elaboración propia

Se observa claramente que el incremento en el nivel de tráfico al doble repercute en una distancia recorrida aproximadamente la mitad que en el escenario de inicio (exactamente el 45,5%). Como en las comparaciones previas, la mejora principal es la reducción de los tiempos en paradas involuntarias.

6. Conclusiones

Los objetivos planteados al inicio del presente proyecto, para reducir el impacto del tráfico en áreas urbanas, eran:

- Aumentar la distancia media recorrida por los vehículos en un tiempo dado.

Tal como se ha expuesto en el apartado “5.2 Comparaciones”, el aumento de esta distancia se consigue al seleccionar a esta como único parámetro a optimizar. Si, por el contrario, se pretende una optimización multi objetivo, no existe garantía de que se logre.

Seleccionando todos los posibles parámetros a optimizar en la función *fitness*, la distancia media recorrida tiene una ponderación del 70%. Con esto se garantiza que, aunque no se consiga mejorar su valor en favor de los otros parámetros (tiempos de recorrido y límites de velocidad), la penalización por su disminución sea importante y las reducciones sean mínimas.

Por tanto, aunque no siempre se consiga un aumento en la distancia media recorrida por los vehículos, si se reduce, esta reducción no es significativa. Además, si no se alcanza este objetivo es por la mejora en los tiempos de desplazamiento, y en concreto, en la disminución de los tiempos perdidos como se indica a continuación.

- Reducir los tiempos efectivos de los viajes, entendiendo como tales las diferencias entre los lapsos totales empleados en los desplazamientos menos los tiempos perdidos, en esperas o atascos, por ejemplo.

Los tiempos efectivos de los viajes no se han reducido como se había previsto, al contrario, han aumentado. Esto ha sido debido a que los vehículos se encuentran circulando durante el lapso previsto a simular, tal como se ha configurado: el tiempo total, aproximadamente, es este lapso, menos el tiempo que emplea el simulador en introducir el vehículo en la simulación; además, se reducen considerablemente los tiempos perdidos. Por tanto, como los tiempos efectivos de viaje son la diferencia entre ambos, se han aumentado en lugar de disminuir.

Este aumento de tiempo de viaje, a igualdad del resto de parámetros, se refleja en la disminución de la velocidad media de los vehículos mientras se desplazan. De forma indirecta se mejora la seguridad vial al desplazarse los vehículos a una velocidad menor, en lugar de aumentarla mientras se pierde tiempo en paradas involuntarias.

- Reducir también estos tiempos perdidos en los desplazamientos.

Como se ha indicado en el punto anterior, esta mejora, siempre que se seleccione, es la que mejor resultado obtiene, es decir, en la que mayor reducción se consigue y, como se ha explicado, reduce la velocidad media, aumentando la seguridad vial.

Un efecto que produce la reducción de estos tiempos perdidos, además de lo ya indicado, es la percepción por parte de los usuarios de estar en movimiento. Es decir, se mitiga la sensación de pérdida de tiempo al encontrarse detenido por la congestión, incluso aunque el tiempo total del viaje sea el mismo. Al ser esta una percepción subjetiva, no se ha incluido su valoración en el presente proyecto.

- Reducir los límites de velocidad de las calles.

Como se ha descrito a lo largo de esta memoria, esta optimización por si sola únicamente conlleva a que los límites de velocidad converjan al mínimo, establecido en 10 km/h. Como parámetro complementario al resto de optimizaciones, mejora la seguridad vial, si no en todas las vías, si al menos en aquellas en que sea posible su disminución, lo cual se consigue en las vías menos transitadas o en escenarios con alto nivel de congestión.

También, se ha establecido la posibilidad de comparar los resultados de un escenario de partida con otros. Estas comparaciones pueden ir desde la variación de los tiempos de simulación, nivel de congestión (número de vehículos), a obtener resultados de escenarios asimilables. Por tanto, podemos optimizar la circulación en un área ya existente o elegir una solución que obtenga un mejor desempeño futuro, en el caso de estudio.

Finalmente, como recapitulación, se ha conseguido crear una aplicación que mejora la calidad de vida en las ciudades haciendo una mejor gestión del tráfico. Aunque no se haya alcanzado el objetivo de incrementar la distancia media recorrida, a no ser que se elija esta como única optimización, sí que se ha disminuido el tiempo perdido en los viajes. De esta forma, al estar en movimiento durante mayor tiempo para una misma distancia, se reduce la velocidad media y, por tanto, se aumenta la seguridad vial. Además, se puede comparar los resultados obtenidos entre distintas configuraciones y escenarios.

7. Trabajos futuros

En el presente apartado se abordarán aquellos trabajos que podrían llevarse a cabo en un futuro y que podrían complementar al presente. También se incluyen aquellos que se intentaron incluir en este y que, por las razones que se explicarán en el punto correspondiente, no fueron factibles.

7.1 Paradoja de Braess

“La paradoja de *Braess* es la observación de que añadir una o más carreteras a una red de carreteras puede acabar dificultando el flujo de tráfico general a través de ella. (...) Se ha sugerido que, en teoría, la mejora de una red podría lograrse eliminando ciertas partes de la misma.” (28)

Con un sencillo ejemplo se demuestra que, al añadir más vías a una red de tráfico, se puede llegar a perjudicar el tránsito a través de la misma, debido al “efecto llamada” que se produce en los usuarios que anteriormente utilizaban otras rutas alternativas y, debido al aumento de capacidad realizado, pasan a transitar por esta vía.

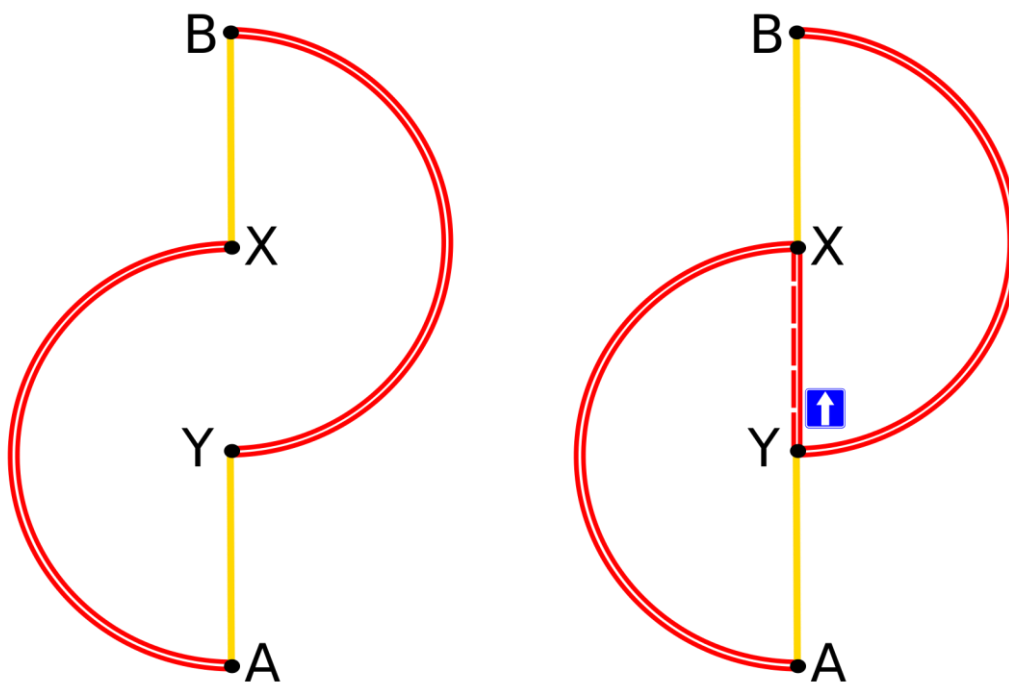


Figura 45 – Paradoja de Braess

Fuente: Wikipedia.org (28)

Como se muestra en la figura 45, los recorridos entre X e Y tienen, en un primer momento, dos posibles alternativas: YAX e YBX. Tras la mejora, construyendo una vía entre X e Y, todo el tráfico que antes se distribuía entre dos rutas, se concentra ahora en una única carretera, pudiendo llegar a colapsarla.

La trasposición de esta paradoja al presente proyecto se basaba en la eliminación de uno o varios carriles de algunas vías para impedir la congestión en algunas calles, es decir, para intentar dispersar el tráfico en toda la red. Además de pretender homogeneizar la densidad de tráfico, los objetivos eran:

- En estudios previos: evitar emplear recursos en la construcción de estos carriles, o destinarlos a otros usuarios, como carriles bici o aceras para peatones.
- En viales ya construidos: cerrarlos al tráfico para reducir su desgaste, con el consecuente ahorro en su mantenimiento, o cambiar sus usos a otros alternativos, como los que se acaban de mencionar, o emplearlos, por ejemplo, como zona de aparcamiento.

Para poder llevar a cabo esta implementación era necesario, para cada carril que se suprimiera, editar los nodos extremos. Por un lado, era necesario redirigir todo el tráfico entrante a otros carriles. Por otro lado, para el tráfico saliente habilitar los posibles giros de este carril a otros, en los carriles adyacentes. No considerar estos dos aspectos conlleva al bloqueo de rutas por aquellos vehículos que pretendieran acceder a un carril inhabilitado, siendo necesaria la eliminación de estos vehículos tras un tiempo determinado por el propio simulador (*teleportación*), para permitir el paso de otros vehículos que pudieran interferir con los primeros.

Se desestimó incluir esta funcionalidad por la complejidad y la ralentización en la ejecución que provocaría. Era necesario editar dos intersecciones por cada carril suprimido en el escenario, y por cada iteración. Además, una comprobación añadida, para calles con múltiples carriles, es que habría que cancelar, bien los carriles situados junto al borde derecho, o junto al izquierdo, pero no los centrales para no desdoblar calles.

7.2 Paralelización

Debido al coste computacional de cada simulación, se ha intentado paralelizarlas mediante la ejecución de varios hilos (*threads*) a la vez, para reducir el tiempo de ejecución del programa. Para evitar condiciones de carrera, se protegió la manipulación en la población mediante la directiva correspondiente (*lock*), al añadir, ordenar y, principalmente, eliminar las soluciones menos prometedoras.

El simulador arroja un error, ya que la librería denominada *TraCI* (*traffic control interface*, o interfaz de control de tráfico) sólo permite una única simulación en curso, en un instante dado. Sí permite ejecutar varias en paralelo:

```
traci.start(["sumo", "-c", "sim1.sumocfg"], label="sim1")
traci.start(["sumo", "-c", "sim2.sumocfg"], label="sim2")
traci.switch("sim1")
traci.simulationStep() # run 1 step for sim1
traci.switch("sim2")
traci.simulationStep() # run 1 step for sim2
traci.switch("sim1")
traci.close()
traci.switch("sim2")
traci.close()
```

Figura 46 – Multitarea en SUMO

Fuente: Documentación de SUMO (29)

Pero es semejante al procesamiento multitarea (*multitasking*) con un único procesador, en el que se alternan ambas ejecuciones (*switch*), diferenciándose mediante la asignación de distintas etiquetas, pero sin ser concurrentes:

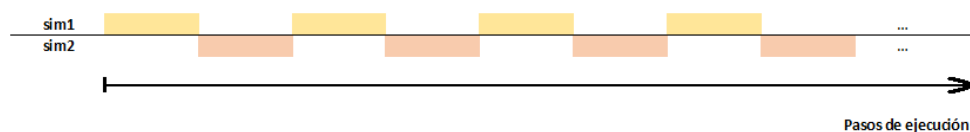


Figura 47 – Multitasking

Fuente: Elaboración propia

No se ha implementado porque, además de que la ejecución es única (no en paralelo), el coste de los cambios de contexto de cada ejecución no mejora el tiempo de cómputo. En el futuro, si el desarrollo del simulador lo permite, será una mejora a incorporar, ya que reduciría enormemente este tiempo.

7.3 Límites de velocidad

En cuanto al tratamiento de los límites de velocidad que se han incluido como mejora en este trabajo frente a otros proyectos previos, razón por la que referimos posibles mejoras en este punto de manera específica, se podrían añadir dos mejoras: la primera de ellas referida al detalle de estos límites, y la segunda a la evaluación en la función *fitness*.

Se podría hacer un cálculo de los límites de velocidad, particularizando por cada uno de los carriles del escenario. Esto supone añadir una comprobación extra, de tal manera que el límite de velocidad de un carril debe ser mayor o igual al límite del carril situado inmediatamente a su derecha (14). Esta mejora reforzaría la reducción en los niveles de ruido ya que se reduciría la velocidad de los vehículos más pesados, que son los que circulan más a la derecha, y además son los que mayor ruido emiten y, potencialmente, los accidentes son más graves.

La función *fitness* podría añadir una medida ponderada de los límites de velocidad respecto de la longitud de la calle en la que se aplican. De este modo, tendrá más peso en la valoración la reducción en calles de mayor longitud ya que, *a priori*, habrá un mayor número de habitantes que se beneficiarán de esta medida.

7.4 Otras mejoras

Finalmente, se citarán otras posibles mejoras que se podrían desarrollar para complementar el presente trabajo, algunas ya comentadas en trabajos previos (19) (20), haciendo uso de opciones del propio simulador empleado. Como SUMO es un simulador multimodal, se podrán añadir otros modos de transporte como peatones, bicicletas e, incluso, distintos transportes públicos, como taxis, autobuses, tranvías, transportes fluviales, entre otros. Además de regular las fases de los

semáforos que gestionan estos desplazamientos, se podría extender al estudio pormenorizado del dimensionamiento del transporte público: cantidad, frecuencias, etc.

Por otra parte, se podría mejorar el análisis de las emisiones. Como se ha comentado en “1.3 Objetivos”, la razón por la que éstas no se han tenido en cuenta en la valoración de las soluciones es que sus mediciones son función del vehículo y del tiempo de desplazamiento, es decir, miligramos de emisiones por segundo, para el caso de los gases, y en decibelios por segundo, para la contaminación acústica. En la realidad, existen unas emisiones mínimas, cuando el vehículo está en reposo, y se incrementan a medida que aumenta la velocidad. Lo que se propone para el futuro es implementar este comportamiento real en el simulador, analizando si la relación entre emisiones y velocidad es o no lineal, para acabar obteniendo como resultado datos fidedignos.

En cuanto a la semaforización, se podría incluir como parámetro la compensación de cada semáforo en los cálculos. Esta compensación o decalaje (*offset*) es la diferencia en segundos entre los inicios de ciclos. Esta diferencia es función de la distancia entre dos semáforos consecutivos y del límite de velocidad de la calle entre estas dos intersecciones. Esta mejora permitirá una mayor homogeneización en los tiempos de fases semaforicas, mejorando también la coordinación de las distintas luces entre semáforos de cruces adyacentes.

Una última mejora haría referencia a la finalización del programa. Actualmente se ha fijado en un número de iteraciones para la evolución de la población. Podría ampliarse para que el usuario elija a su conveniencia que finalización desea:

- Tras un tiempo de cómputo fijo.
- Considerando medidas de diversidad en la población (convergencia).
- Cuando se alcance un determinado valor de *fitness*.
- Tras varias generaciones sin ninguna mejora (convergencia).

8. Referencias

1. **Banco Mundial.** Población. *Total*. [En línea] Grupo Banco Mundial, 2022. [Citado el: 2 de junio de 2022.] <https://datos.bancomundial.org/indicador/SP.POP.TOTL>.
2. —. Población urbana. *% del total*. [En línea] Grupo Banco Mundial, 2022. [Citado el: 2 de junio de 2022.] <https://datos.bancomundial.org/indicador/SP.URB.TOTL.IN.ZS>.
3. **INRIX, Inc.** *2021 INRIX Global Traffic Scorecard*. Kirkland, WA 98033, USA : s.n., 2021.
4. —. *2021 INRIX Global Traffic Scorecard. Interactive Ranking & City Dashboards*. [En línea] 2022. [Citado el: 3 de junio de 2022.] <https://inrix.com/scorecard/>.
5. **Excmo. Ayuntamiento de Madrid.** *Inventario de emisiones contaminantes a la atmósfera en el municipio de Madrid 2019*. Madrid : Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid, 2021.
6. **Aimsun, SLU.** Acerca de Aimsun. [En línea] [Citado el: 14 de junio de 2022.] <https://www.aimsun.com/es/about-aimsun/>.
7. **Van Aerde, M., y otros.** *INTEGRATION: An Overview of Traffic Simulation Features*. Kingston, ON K7L 3N6, Canadá : Department of Civil Engineering, 1996.
8. **SYSTRA, Ltd.** Paramics Microsimulation. *3D traffic simulation*. [En línea] 2019. [Citado el: 15 de junio de 2022.] <https://www.paramics.co.uk/en/about-us/article/paramics-microsimulation>.
9. **Università della Calabria.** TIS - Software per Trasporti Innovativi. *Tritone*. [En línea] 2017. [Citado el: 14 de junio de 2022.] <http://tis.unical.it/index.php/development/tritone/>.
10. **Eclipse Foundation - openMobility Working Group.** SUMO. *Simulation of Urban MObility*. [En línea] 2022. [Citado el: 22 de febrero de 2022.] <https://www.eclipse.org/sumo/>.
11. **PTV, GmbH.** Soluciones de software para la movilidad. [En línea] 2022. [Citado el: 14 de junio de 2022.] <https://company.ptvgroup.com/es/productos>.
12. **Bedoya Reyes, A.** Estudio de entornos de simulación en redes de vehículos. *Anexo A*. Zaragoza, España : Departamento de Informática e Ingeniería de Sistemas, 2013.
13. **Eclipse Foundation - openMobility Working Group.** SUMO. *About Eclipse SUMO*. [En línea] 2022. [Citado el: 15 de junio de 2022.] <https://www.eclipse.org/sumo/about/>.

14. **Ministerios del Interior, de Defensa, de Fomento y de Ciencia y Tecnología.** *Reglamento General de Circulación*. Real Decreto 1428/2003, de 21 de noviembre, «BOE» núm. 306, de 23/12/2003 : Ministerio de la Presidencia, 2003.
15. **Eclipse Foundation - openMobility Working Group.** SUMO. *SUMO User Documentation*. [En línea] 2022. [Citado el: 22 de febrero de 2022.] <https://sumo.dlr.de/docs/>.
16. **Cardellini, M.** An ASP Framework for Efficient Urban Traffic Optimization. Génova, Italia : Politecnico di Torino*, 2022.
17. **Kim, M.-S., Eoh, G. y Park, T.-H.** Decision Making for Self-Driving Vehicles in Unexpected Environments Using Efficient Reinforcement Learning Methods. Cheongju, 28644, Korea : Department of Intelligent Robotics, 2022.
18. **Samal, S.R., y otros.** Analysis of traffic congestion impacts of urban road network under indian condition. Indore, Madhya Pradesh, India : Departament of Civil Engineering, 2020.
19. **Hernández Yáñez, M.** Diseño y aplicación de técnicas metaheurísticas para control de tráfico. *TFG*. Valencia, España : Escola Tècnica Superior d'Enginyeria Informàtica - Universitat Politècnica de València, 2021.
20. **Valdiviezo Condolo, M.F.** Diseño y aplicación de técnicas metaheurísticas para control de tráfico. *TFM*. Valencia, España : Escola Tècnica Superior d'Enginyeria Informàtica - Universitat Politècnica de València, 2017.
21. **WikiVía - La enciclopedia de la carretera.** Asociación española de la carretera. *Criterios para la instalación de semáforos*. [En línea] 2009. [Citado el: 9 de junio de 2022.] http://www.wikivia.org/wikivia/index.php?title=Criterios_para_la_instalación_de_semaforos.
22. **El Norte de Castilla.** Noticia. *Más de 2,5 kilómetros de retenciones hacia Madrid en la N-VI a su paso por San Rafael*. [En línea] 2022. [Citado el: 5 de julio de 2022.] <https://www.elnortedecastilla.es/segovia/kilometros-retenciones-madrid-20220703212836-nt.html>.
23. **Carreteros.org.** Norma 8.1-IC Señalización vertical. *Instrucción de carreteras*. [En línea] 2014. [Citado el: 3 de julio de 2022.] http://carreteros.org/normativa/s_vertical/8_1ic_2014/apartados/7.htm.
24. **Wirsansky, E.** *Hands-on genetic algorithms with Python*. Birmingham, B3 2PB, United Kingdom : Packt Publishing Ltd., 2020. ISBN 978-1-83855-774-4.
25. **The Jupyter / IPython Project.** Project Jupyter. [En línea] [Citado el: 18 de julio de 2022.] <https://jupyter.org/>.
26. **Joshi, P.** *Artificial intelligence with Python*. Birmingham, B3 2PB, United Kingdom : Packt Publishing, Ltd., 2017. ISBN 978-1-78646-439-2.

27. **Palma, J.T. y Marin, R.** *Inteligencia artificial: técnicas, métodos y aplicaciones*. Aravaca, Madrid, España : McGraw-Hill - Interamericana de España, SAU, 2008. ISBN: 978-84-481-5618-3.

28. **Wikipedia.org.** Paradoja de Braess. [En línea] 2022. [Citado el: 31 de enero de 2022.] https://es.wikipedia.org/wiki/Paradoja_de_Braess.

29. **Eclipse Foundation - openMobility Working Group.** SUMO. *Interfacing TraCI from Python*. [En línea] 2022. [Citado el: 8 de junio de 2022.] https://sumo.dlr.de/docs/TraCI/Interfacing_TraCI_from_Python.html.