



ACTA DE EVALUACIÓN DE LA TESIS DOCTORAL

Año académico 2019/20

DOCTORANDO: CRUZ PIRIS, LUIS DE LA  
D.N.I./PASAPORTE: \*\*\*\*3958R

PROGRAMA DE DOCTORADO: D445-TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES  
DPTO. COORDINADOR DEL PROGRAMA: TEORÍA DE LA SEÑAL Y COMUNICACIONES  
TITULACIÓN DE DOCTOR EN: DOCTOR/A POR LA UNIVERSIDAD DE ALCALÁ

En el día de hoy 14/11/19, reunido el tribunal de evaluación nombrado por la Comisión de Estudios Oficiales de Posgrado y Doctorado de la Universidad y constituido por los miembros que suscriben la presente Acta, el aspirante defendió su Tesis Doctoral, elaborada bajo la dirección de IVÁN MARSÁ MAESTRE // MIGUEL ÁNGEL LÓPEZ CARMONA.

Sobre el siguiente tema: CONTRIBUCIÓN A LAS ESTRATEGIAS DE OPTIMIZACIÓN MULTIOBJETIVO PARA LA COORDINACIÓN DE VEHÍCULOS EN INTERSECCIONES URBANAS

Finalizada la defensa y discusión de la tesis, el tribunal acordó otorgar la CALIFICACIÓN GLOBAL<sup>1</sup> de (no apto, aprobado, notable y sobresaliente): Sobresaliente

Alcalá de Henares, 14 de noviembre de 2019

EL PRESIDENTE

Fdo.: SASCHA OSSOWSKI

EL SECRETARIO

Fdo.: JUAN RAMÓN VELASCO PÉREZ

EL VOCAL

Fdo.: MARIO VEGA BARBAS

Con fecha 25 de noviembre de 2019 la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado, a la vista de los votos emitidos de manera anónima por el tribunal que ha juzgado la tesis, resuelve:

- Conceder la Mención de "Cum Laude"
- No conceder la Mención de "Cum Laude"

FIRMA DEL ALUMNO,

Fdo.: CRUZ PIRIS, LUIS DE LA

La Secretaria de la Comisión Delegada

<sup>1</sup> La calificación podrá ser "no apto" "aprobado" "notable" y "sobresaliente". El tribunal podrá otorgar la mención de "cum laude" si la calificación global es de sobresaliente y se emite en tal sentido el voto secreto positivo por unanimidad.

INCIDENCIAS / OBSERVACIONES:



Universidad  
de Alcalá

VICERRECTORADO DE INVESTIGACIÓN  
Y TRANSFERENCIA

En aplicación del art. 14.7 del RD. 99/2011 y el art. 14 del Reglamento de Elaboración, Autorización y Defensa de la Tesis Doctoral, la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado y Doctorado, en sesión pública de fecha 25 de noviembre, procedió al escrutinio de los votos emitidos por los miembros del tribunal de la tesis defendida por **CRUZ PIRIS, LUIS DE LA**, el día 14 de noviembre de 2019, titulada, **CONTRIBUCIÓN A LAS ESTRATEGIAS DE OPTIMIZACIÓN MULTIOBJETIVO PARA LA COORDINACIÓN DE VEHÍCULOS EN INTERSECCIONES URBANAS** para determinar, si a la misma, se le concede la mención "cum laude", arrojando como resultado el voto favorable de todos los miembros del tribunal.

Por lo tanto, la Comisión de Estudios Oficiales de Posgrado y Doctorado **resuelve otorgar** a dicha tesis la

**MENCIÓN "CUM LAUDE"**

EL VICERRECTOR DE INVESTIGACIÓN Y TRANSFERENCIA  
F. Javier de la Mata de la Mata  
Documento fechado y firmado digitalmente

**Copia por e-mail a:**

Doctorando: CRUZ PIRIS, LUIS DE LA

Secretario del Tribunal: JUAN RAMÓN VELASCO PÉREZ

Directores de Tesis: IVÁN MARSÁ MAESTRE // MIGUEL ÁNGEL LÓPEZ CARMONA

<b>Código Seguro De Verificación:</b>	1VuJmjJrMPYDfLaIsypG8Q==	<b>Estado</b>	<b>Fecha y hora</b>	
<b>Firmado Por</b>	Francisco Javier De La Mata De La Mata - Vicerrector de Investigación Y Transferencia	Firmado	26/11/2019 12:46:31	
<b>Observaciones</b>		<b>Página</b>	5/15	
<b>Uri De Verificación</b>	<a href="https://vfirma.uah.es/vfirma/code/1VuJmjJrMPYDfLaIsypG8Q==">https://vfirma.uah.es/vfirma/code/1VuJmjJrMPYDfLaIsypG8Q==</a>			



DILIGENCIA DE DEPÓSITO DE TESIS.

Comprobado que el expediente académico de D./D<sup>a</sup> \_\_\_\_\_  
reúne los requisitos exigidos para la presentación de la Tesis, de acuerdo a la normativa vigente, se  
procede, con fecha de hoy a registrar el depósito de la tesis en el Servicio de Estudios Oficiales de  
Posgrado, con número de páginas: \_\_\_\_\_.

Alcalá de Henares a \_\_\_\_ de \_\_\_\_\_ de 20\_\_\_\_



Pilar de la Vega Gallego  
Fdo. El Funcionario



Universidad  
de Alcalá

Programa de Doctorado en Tecnologías de la  
Información y las Comunicaciones

# CONTRIBUCIÓN A LAS ESTRATEGIAS DE OPTIMIZACIÓN MULTIOBJETIVO PARA LA COORDINACIÓN DE VEHÍCULOS EN INTERSECCIONES URBANAS

Tesis Doctoral presentada por  
**LUIS DE LA CRUZ PIRIS**

2019

C. del Bérquillo





Universidad  
de Alcalá

Programa de Doctorado en Tecnologías de la  
Información y las Comunicaciones

**CONTRIBUCIÓN A LAS ESTRATEGIAS DE  
OPTIMIZACIÓN MULTIOBJETIVO PARA LA  
COORDINACIÓN DE VEHÍCULOS EN  
INTERSECCIONES URBANAS**

Tesis Doctoral presentada por

**LUIS DE LA CRUZ PIRIS**

Directores:

DR. IVÁN MARSÁ MAESTRE  
DR. MIGUEL ÁNGEL LÓPEZ CARMONA

Alcalá de Henares, 2019





Dr. D. IVÁN MARSÁ MAESTRE y Dr. D. MIGUEL ANGEL LÓPEZ CARMONA, Profesores Titulares de Universidad del área de “Ingeniería Telemática”, adscrita al Departamento de Automática de la Universidad de Alcalá,

CERTIFICAN

Que la tesis “**Contribución a las estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas**”, presentada por D. Luis de la Cruz Piris, realizada en el Departamento de Automática bajo nuestra dirección y en la modalidad de compendio de artículos, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y lectura.

Alcalá de Henares, 16 de septiembre de 2019.



Fdo.: Dr. D. Iván Marsá Maestre      Fdo.: Dr. D. Miguel Ángel López Carmona



D. Luis de la Cruz Piris ha realizado en el Departamento de Automática y bajo la dirección del Dr. D. Iván Marsá Maestre y del Dr. D. Miguel Ángel López Carmona, la tesis doctoral titulada "**Contribución a las estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas**" en la modalidad de compendio de artículos, cumpliéndose todos los requisitos para la tramitación que conduce a su posterior lectura.

Alcalá de Henares, 16 de septiembre de 2019.

Coordinador del Programa de Doctorado



Fdo.: Dr. D. Sancho Salcedo Sanz



A continuación, se muestra el listado de las publicaciones han dado lugar a la Tesis titulada "**Contribución a las estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas**" realizado por D. Luis de la Cruz Piris, indicando brevemente su aportación en cada una de ellas.

1. **L. Cruz-Piris (LCP)**, D. Rivera (DRP), S. Fernandez (SFM), and I. Marsa-Maestre (IVM), "Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management", *Sensors*, vol. 18, no. 2, p. 435, Feb. 2018.  
**DOI:** <https://doi.org/10.3390/s18020435>  
**Tipo de publicación:** Artículo de revista    **Fecha de publicación:** 02/02/2018  
**JCR (2018):** Q1 (15/61)    **IF (2018):** 3.031  
**Aportación:** Realización de la propuesta inicial junto con IVM. Desarrollo del sistema multiagente (plataforma y lógica de los agentes) y conexión con el simulador de tráfico. Propuesta e implementación de la conversión de la red de tráfico a un grafo y del análisis de su centralidad. Diseño, implementación y ejecución de los experimentos para validar la propuesta junto con DRP. Escritura y revisión del resto del artículo junto con DRP, SFM y IVM.
2. **L. Cruz-Piris (LCP)**, M. A. Lopez-Carmona (MAL) and I. Marsa-Maestre (IVM), "Automated Optimization of Intersections Using a Genetic Algorithm", in *IEEE Access*, vol. 7, pp. 15452-15468, 2019.  
**DOI:** <https://doi.org/10.1109/ACCESS.2019.2895370>  
**Tipo de publicación:** Artículo de revista    **Fecha de publicación:** 25/01/2019  
**JCR (2018<sup>1</sup>):** Q1 (23/155)<sup>2</sup>    **IF (2018<sup>1</sup>):** 4.098  
**Aportación:** Realización de la propuesta inicial junto con MAL e IVM. Desarrollo del simulador de intersecciones y del sistema de codificación del modelo propuesto. Diseño de los experimentos junto con MAL e IVM. Ejecución de los experimentos y recolección de resultados. Escritura y revisión del resto del artículo junto con MAL e IVM.
3. **L. Cruz-Piris (LCP)**, I. Marsa-Maestre (IVM) and M. A. Lopez-Carmona (MAL), "A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem", in *IEEE Access*, vol. 7, pp. 111968-111981, 2019.  
**DOI:** <https://doi.org/10.1109/ACCESS.2019.2935041>  
**Tipo de publicación:** Artículo de revista    **Fecha de publicación:** 13/08/2019  
**JCR (2018<sup>1</sup>):** Q1 (23/155)<sup>2</sup>    **IF (2018<sup>1</sup>):** 4.098  
**Aportación:** Realización de la propuesta inicial junto con IVM y MAL. Implementación de las mejoras propuestas para el modelo de intersecciones. Propuesta y desarrollo de los operadores del algoritmo genético con cromosomas de longitud variable. Diseño de los experimentos junto con IVM y MAL. Ejecución de los experimentos y recolección de resultados. Escritura y revisión del resto del artículo junto con IVM y MAL.

<sup>1</sup> Último índice disponible a 16 de septiembre de 2019.

<sup>2</sup> Mejor categoría de la revista en el área de la Tesis.

Yo, D. Luis de la Cruz Piris, autor de la Tesis "Contribución a las estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas", certifico que las publicaciones tituladas "*Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management*", "*Automated Optimization of Intersections Using a Genetic Algorithm*" y "*A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem*", de las que además soy primer autor, no han sido utilizadas previamente por otros investigadores como aval de otras Tesis doctorales.

Alcalá de Henares, 16 de septiembre de 2019.



Fdo.: D. Luis de la Cruz Piris

INFORME DEL DIRECTOR/A DE TESIS SOBRE "ANÁLISIS DE COINCIDENCIAS"  
DE LA TESIS DOCTORAL A TRAVÉS DEL PROGRAMA TURNITIN

D. Iván Marsá Maestre y D. Miguel Ángel López Carmona, director/es de tesis, AUTORIZAN a D. Luis de la Cruz Piris, a defender la Tesis Doctoral que tiene por título "**Contribución a las estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas**".

El trabajo presentado ha sido analizado por la plataforma TURNITIN, arrojando un porcentaje de coincidencias del 8 %.

Justificación de la autorización:

*Tal y como exige la normativa vigente en relación con la defensa de Tesis Doctorales por compendio de artículos en el Programa de Doctorado de Tecnologías de la Información y las Comunicaciones de la Universidad de Alcalá, se incluyen en el libro las copias literales de los artículos que han dado lugar a la tesis en los intervalos de páginas 75 a 96, 99 a 115 y 119 a 132. Por este motivo, estos intervalos han sido excluidos del análisis de coincidencias.*

*Tras analizar el resumen de coincidencias del informe realizado con TURNITIN, todas ellas tienen un valor de "<1%" y se corresponde con el uso del título de los artículos publicados, sentencias introductorias de párrafos o de conexión entre dos frases, y aunque se ha filtrado excluyendo las citas y la bibliografía, ha marcado como coincidencia gran parte de los títulos y autores de la bibliografía.*

En Alcalá de Henares, a 16 de septiembre de 2019

Firmado: D. Iván Marsá Maestre D. Miguel Ángel López Carmona

Informada la Comisión Académica:

Fecha: 19-09-2019

Firmado: Presidente de la Comisión Académica del Programa





*A mi familia.*



# Resumen

Más del 72% de los ciudadanos europeos residen en núcleos urbanos. Algunas estimaciones cuantifican el posible coste asociado al tiempo perdido en atascos en la Unión Europea entre los años 2016 y 2026 en más de 200 mil millones de euros. Las situaciones de congestión son los problemas de tráfico urbano más habituales, pero no los únicos. El impacto ambiental que produce de forma directa el tráfico de vehículos, tanto por sus emisiones contaminantes como por el ruido y molestias que genera a los ciudadanos, ha provocado que en los últimos años se haya incrementado el número de iniciativas para paliar estos problemas.

Cada vehículo que circula por una red de carreteras puede ser considerado como un elemento independiente que, para alcanzar sus objetivos, utiliza los recursos de la red durante un periodo de tiempo concreto. Cuando dos o más vehículos necesitan utilizar un recurso durante el mismo instante temporal, siendo sus trayectorias incompatibles (el paso de uno de los vehículos produce la detención del otro), el uso de técnicas de coordinación se hace imprescindible. La evolución de las tecnologías relacionadas con los sistemas inteligentes de transporte (ITS, del inglés *Intelligent Transportation Systems*) tales como la mayor precisión de los sensores, sistemas de procesado más potentes y redes de comunicación inalámbricas que han reducido su latencia a niveles cercanos a los de las redes cableadas, han abierto un camino para aplicar nuevas técnicas de coordinación de vehículos sobre puntos clave de las redes de tráfico como pueden ser las intersecciones.

Esta tesis aborda el problema de la coordinación de vehículos a su paso por cruces de carreteras o intersecciones en un entorno urbano. El estudio de este problema comienza por analizar las posibles fuentes de información existentes en un escenario de tráfico urbano, teniendo en cuenta las tecnologías disponibles en la actualidad. En este ámbito se realiza la primera contribución, proponiendo una metodología basada en la centralidad de la red de carretera que permite obtener las mejores posiciones de la red donde ubicar sensores que proporcionen la información más relevante para modelar el tráfico. La hipótesis central de esta tesis es que es posible mejorar los procesos de optimización utilizados para la gestión actual de las intersecciones. Para conseguir este fin, el primer objetivo es definir un modelo capaz de etiquetar de

forma inequívoca todos los elementos que componen una intersección. Este modelo será el utilizado como base para las propuestas de optimización posteriores.

Desde el punto de vista de la coordinación de los vehículos, las contribuciones de esta tesis se clasifican dependiendo de si la coordinación de vehículos se realiza a través de elementos externos como los semáforos, o si por el contrario, esta coordinación se produce en un escenario de vehículos autónomos, donde cada uno de ellos es capaz de cruzar la intersección, sin detenerse, en condiciones seguras. En el primero de los casos, la propuesta realizada se centra en el desarrollo de un sistema multiagente, desplegado sobre un escenario simulado de tráfico real, capaz de gestionar las fases de los semáforos de cada intersección. Los agentes controladores de cada intersección tienen la capacidad de variar su comportamiento en base a los indicadores de tráfico que reciben. En el segundo de los casos, la investigación se centra en el comportamiento de los vehículos en el interior de las intersecciones, y los posibles métodos para la generación de patrones de llegada de vehículos que permitan su cruce, en condiciones seguras, sin que los vehículos se detengan y cumpliendo las preferencias de flujo del sistema. Con este fin, se ha diseñado en una primera iteración un proceso de optimización basado en un algoritmo genético con caminos fijos entre los puntos de entrada y salida de la intersección, centrado en obtener el mayor rendimiento posible para el caso de que todos los flujos de entrada sean iguales. Posteriormente, esta propuesta se ha extendido permitiendo todos los caminos posibles entre los puntos de entrada y de salida de la intersección, y generalizando además el objetivo de la optimización para lograr los flujos de entrada deseados, sin las restricciones de tener que contar con flujos de entrada simétricos. Debido al aumento exponencial de la complejidad del problema en esta última solución, se ha propuesto un algoritmo genético de longitud variable de cromosomas, adaptado a este problema.

Para validar las propuestas realizadas se han utilizado simuladores y escenarios de tráfico realistas, que permiten probar de forma intensiva cada una de ellas. Además del uso de simuladores comerciales, ha sido necesario implementar un simulador de intersecciones propio, el cual sea capaz de reproducir las peculiaridades del problema tratado. En cada caso, los resultados obtenidos utilizando las propuestas realizadas han sido comparados frente a otras soluciones ampliamente utilizadas. Los buenos resultados obtenidos con los métodos propuestos en esta tesis permiten confirmar las hipótesis planteadas en el inicio de la investigación.

**Palabras clave:** sistemas inteligentes de transporte, congestión del tráfico, intersección de tráfico, sistemas multiagente, redes de sensores, algoritmos genéticos

# Abstract

More than 72 % of Europeans live in urban areas. Some estimations quantify the possible cost related to the time lost in traffic jams in the European Union between 2016 and 2026 in more than 200 thousand million euros. Traffic jams are the most common urban traffic problem but not the only one. The environmental impact produced directly by vehicular traffic, both by pollutant emissions and by the noise and discomfort endured by citizens, has increased the number of initiatives to alleviate these problems in recent years.

Each vehicle that circulates on a road network can be considered as an independent element that, to achieve its objectives, uses the resources of the network for a specific period. When two or more vehicles need to use a resource during the same time, and their trajectories are incompatible (the passage of one of the vehicles causes the stopping of the other), then the use of coordination techniques is essential. The evolution of technologies related to Intelligent Transportation Systems such as improved sensor accuracy, more powerful processing systems and wireless communication networks that have reduced their latency to levels close to wired networks, has paved the way to apply new vehicle coordination techniques on key points of traffic networks such as intersections.

This thesis addresses the problem of the coordination of vehicles as they cross junctions or intersections in an urban environment. The study of this problem begins by analyzing the possible sources of information existing in an urban traffic scenario, considering the technologies currently available. In this area, our first contribution is the proposal of a methodology based on the study of the centrality of the road network. The goal of such methodology is to determine the best positions for sensors, so that they provide the most relevant data for traffic modeling. The central hypothesis of this thesis is that it is possible to improve the optimization processes used for the current management of intersections. To achieve this goal, the first objective is to define a model capable of labeling unequivocally all the elements that make up an intersection. This model is used later as the basis for subsequent optimization proposals.

From the point of view of the coordination of vehicles, the contributions of this thesis are classified depending on whether the coordination of vehicles is carried out through external elements such as traffic lights, or this coordination occurs in a scenario of autonomous vehicles, where each of them is able to cross the intersection, without stopping, in safe conditions. In the first case, the proposal focuses on the development of a multi-agent system, deployed on a simulated scenario based on real data, capable of managing the traffic light phases of each intersection. The controlling agents for each intersection can vary their behavior based on the traffic indicators they receive. In the second case, the research focuses on the behavior of vehicles inside the intersections, and the possible methods to generate vehicle arrival patterns that allow their crossing, in safe conditions, without the vehicles being hindered and complying with system flow preferences. To this end, in a first iteration, an optimization process based on a genetic algorithm with fixed paths between the entry and exit points of the intersection has been designed. The algorithm is focused on obtaining the highest possible yield if all entrance flows are equal. Subsequently, this proposal is extended allowing all possible paths between the entry and exit points of the intersection and generalizing the optimization goal to achieve the desired input flows, without the restriction of having symmetrical input flows. Due to the exponential increase in the complexity of the problem in this last scenario, a genetic algorithm with variable-length chromosomes is proposed, adapted for this problem.

Simulators and realistic traffic scenarios have been used to validate the proposals, which allow intensive testing of each of them. In addition to the use of commercial simulators, it has been necessary to implement our own intersection simulator, which can reproduce the peculiarities of the problem. In each case, the results obtained using the proposals have been compared with other widely used solutions. The good results obtained with the methods proposed in this thesis allow us to confirm the hypotheses raised at the beginning of the research.

**Keywords:** intelligent transportation systems, traffic congestion, road traffic intersection, multi-agent systems, sensor networks, genetic algorithms

# Agradecimientos

Quiero y debo comenzar por agradecer el trabajo y esfuerzo realizado por mis directores de tesis, Iván Marsá Maestre y Miguel Ángel López Carmona, durante el desarrollo de esta tesis. El camino recorrido con Iván es mayor ya que no se limita al desarrollo de tesis, sino que incluye muchas horas previas en proyectos de investigación y mi trabajo fin de máster. Por tanto, este agradecimiento debe extenderse más en el tiempo, porque sin todo lo anterior, no hubiera llegado aquí. Aunque el trabajo de investigación de una tesis doctoral puede llegar a parecer, en gran medida, un trabajo individual, sin los dos no hubiera sido posible.

Durante este camino son muchas las personas con las que trabajas y todas ellas, de alguna forma, te ayudan a recorrerlo. Debido al tiempo y a las penas y glorias compartidas, la primera persona que se me viene a la mente es Diego. También tengo que recordar en estas líneas a Enrique, al que acudí para preguntar sobre un proyecto fin de carrera relacionado con IPTV y terminé metido en el mundo del tráfico de vehículos. Sin la perspectiva del tiempo no podemos medir como nos influyen estas pequeñas decisiones. Tampoco puedo dejar de acordarme de los cohabitantes de la sala E24, Susel y Germán, compañeros en muchas batallas. También debo incluir en estos agradecimientos a Antonio, a Bernardo y a JuanRa, por la preocupación que siempre han tenido en ayudarme.

No puedo dejar de agradecer a los proyectos en los que he participado en los últimos años: Divide and NOT Conquer - Comportamientos emergentes en redes complejas egoístas (DANCE) TIN2014-61627-EXP (AEI/FEDER, UE), Gestión inteligente de la señalización para aumentar la eficiencia de las redes de tráfico de vehículos, CCG2018-EXP-041 (Universidad de Alcalá) y Dynamic Network Agreement: negociaciones estructurales en redes complejas (DNA) TIN2016-80622-P (AEI/FEDER, UE). Todos ellos me han aportado tanto una línea de investigación en la que apoyarme como, y no menos importante, la financiación necesaria para poder realizar esta tesis.

Por último, y saltándome la norma de colocar los agradecimientos por orden de importancia, el agradecimiento a mi familia.





# Índice general

Resumen	XIII
Abstract	XV
Agradecimientos	XVII
Índice de figuras	XXIII
Índice de tablas	XXV
I Motivación de la investigación y resumen amplio de la tesis	1
1 Introducción	3
1.1 Coordinación de vehículos en grandes ciudades . . . . .	3
1.2 Planteamiento del problema . . . . .	8
1.3 Objetivos y ámbito de las contribuciones . . . . .	9
1.3.1 Optimización de la ubicación de sensores según las caracterís- ticas de la red . . . . .	10
1.3.2 Modelado automático de intersecciones . . . . .	11
1.3.3 Mecanismos de coordinación de vehículos en intersecciones .	12
1.4 Estructura de la tesis . . . . .	13
2 Resumen amplio: Estrategias de optimización multiobjetivo para la coordi- nación de vehículos en intersecciones urbanas	17
2.1 Escenarios de tráfico urbano . . . . .	17
2.1.1 Análisis de las redes de tráfico como grafos . . . . .	18
2.1.1.1 Pseudo-Dual Graph . . . . .	20
2.1.1.2 Centralidad aplicada a escenarios de tráfico . . . . .	21
2.1.2 Propuesta de localización de sensores en una red de tráfico .	22
2.1.3 Experimentos y resultados . . . . .	24
2.2 Sistemas de gestión de intersecciones . . . . .	26
2.2.1 Sistemas multiagente en entornos de tráfico urbano . . . . .	27

2.2.2	Modelado de intersecciones . . . . .	28
2.2.2.1	Propuesta de modelado de una intersección basado en un TCA . . . . .	29
2.2.2.2	Modelado de intersecciones con formas irregulares .	31
2.3	Coordinación de vehículos a su paso por una intersección . . . . .	33
2.3.1	Gestión de los puntos de conflicto en una intersección urbana	34
2.3.2	Intersecciones señalizadas: gestión a través de semáforos . . .	37
2.3.2.1	Sistema multiagente para la gestión inteligente de semáforos . . . . .	38
2.3.2.2	Caso de uso del sistema multiagente propuesto . . .	39
2.3.2.3	Experimentos realizados para validar el sistema mul- tiagente propuesto . . . . .	41
2.3.3	Intersecciones sin señalar: circulación libre de vehículos . .	43
2.3.3.1	Conversión del modelo de intersección a un grafo dirigido . . . . .	44
2.3.3.2	Cálculo de caminos con mínimo puntos de conflicto	45
2.3.4	Optimización para maximizar la ocupación de la intersección	48
2.3.4.1	Codificación del problema . . . . .	49
2.3.4.2	Función de evaluación . . . . .	49
2.3.4.3	Población inicial, funciones de cruce y mutación, y configuración del GA . . . . .	51
2.3.4.4	Experimentos y resultados . . . . .	52
2.3.5	Optimización para ajustar flujos de llegada preestablecidos . .	55
2.3.5.1	Codificación del problema . . . . .	57
2.3.5.2	Función de evaluación . . . . .	58
2.3.5.3	Algoritmo genético con cromosomas de longitud va- riable . . . . .	59
2.3.5.4	Población inicial y funciones de selección, cruce y mutación . . . . .	60
2.3.5.5	Experimentos y resultados . . . . .	64
2.4	Relación de los artículos del compendio con la investigación realizada	68
II	Artículos publicados que dan lugar a la tesis	71
3	Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management	73
3.1	Introducción . . . . .	73
3.2	Relevancia de la publicación . . . . .	74
3.3	Artículo . . . . .	74
4	Automated Optimization of Intersections Using a Genetic Algorithm	97
4.1	Introducción . . . . .	97

4.2	Relevancia de la publicación . . . . .	98
4.3	Artículo . . . . .	98
5	A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem	<b>117</b>
5.1	Introducción . . . . .	117
5.2	Relevancia de la publicación . . . . .	118
5.3	Artículo . . . . .	118
III	Conclusiones	133
6	Conclusiones y líneas futuras	<b>135</b>
6.1	Discusión y conclusiones . . . . .	135
6.2	Difusión de las contribuciones de la tesis . . . . .	139
6.3	Líneas de investigación futuras . . . . .	141
IV	Apéndices	143
A	Recursos y herramientas para el estudio de escenarios de tráfico	<b>145</b>
A.1	Simulador de tráfico basados en el modelo de <i>car-following</i> . . . . .	147
A.1.1	Simuladores microscópicos comerciales . . . . .	147
A.1.2	Comunicación entre el sistema multiagente y el simulador de tráfico . . . . .	148
A.2	Simulador de intersecciones basado en <i>Cellular Automata</i> . . . . .	151
A.2.1	Estructuras de datos . . . . .	152
A.2.2	Métricas realizadas durante la simulación . . . . .	156
A.2.2.1	Ocupación media . . . . .	157
A.2.2.2	Medidas de error . . . . .	157
A.2.3	Visualización de resultados . . . . .	158
B	Escenarios de tráfico urbano y datos de entrada	<b>161</b>
B.1	Escenario de tráfico urbano realista: TAPASCologne . . . . .	161
B.1.1	Escenario de tráfico reducido basado en TAPASCologne . . . . .	163
B.1.2	Ajustes y calibración . . . . .	163
B.2	Generación de flujos de tráfico para el simulador de intersecciones . . . . .	166
B.2.1	Flujos de vehículos candidatos con soluciones conocidas . . . . .	167
B.2.2	Flujos de vehículos generados de forma aleatoria . . . . .	168
	Bibliografía	<b>173</b>
	Acrónimos	<b>181</b>



# Índice de figuras

2.1	Ejemplo de transformación de una red de carreteras a un grafo. . . . .	19
2.2	Resultados de los experimentos sobre la ubicación de cámaras en un escenario de tráfico. . . . .	26
2.3	División del espacio de la intersección en celdas rectangulares de igual tamaño. . . . .	29
2.4	Propuesta de modelado de intersecciones. . . . .	29
2.5	Ejemplo de aplicación del modelado a una intersección de 5 ramas. . .	32
2.6	Representación de dos trayectorias y el punto el punto de conflicto que generan. . . . .	35
2.7	Diagramas espacio/tiempo de los desplazamientos realizados durante el tiempo de simulación según origen y sistema de gestión. . . . .	36
2.8	Caso de uso del sistema multiagente propuesto para la gestión inteligente de semáforos. . . . .	40
2.9	Número de mensajes generados por cada sistema se gestión dinámica de las fases de los semáforos durante el tiempo de simulación. . . . .	42
2.10	Grafo dirigido obtenido a partir de la conversión del modelo de intersección de 4 brazos y 3 carriles por sentido y brazo visto en la figura 2.4. . . . .	45
2.11	Codificación del problema a través de la definición de carriles virtuales.	50
2.12	Resultados obtenidos de aplicar OCP al caso de uso de una intersección con 4 brazos y 3 carriles por brazo. . . . .	54
2.13	Representación de los posibles caminos entre los puntos de entrada del primer brazo hacia todas las salidas, siguiendo la restricción de caminos con un único cambio de trayectoria. . . . .	58
2.14	Proceso de cruce de dos individuos de una misma población utilizando el optimizador OCP-MP-VL . . . . .	62
2.15	Procesos de contracción y expansión de genes cuando los padres no tiene la misma longitud de patrón. . . . .	63
2.16	Relación entre las publicaciones que dan lugar a esta tesis y los principales apartados de la investigación realizada. . . . .	69

6.1	Relación del número de vehículos que mejora, iguala o empeora los tiempos de sus desplazamientos al utilizar el sistema de gestión de semáforos propuesto. . . . .	136
6.2	Media de los valores de evaluación (color azul) y tiempos de computación (color naranja) obtenidos en cada experimento. . . . .	137
A.1	Diagrama de bloques con los principales pasos del emulador del sistema multiagente y su interacción con el simulador de tráfico. . . . .	149
A.2	Principios básicos de un modelo basado de autómatas celulares . . . .	151
A.3	División del espacio de una intersección de 4 ramas, con 3 carriles de entrada y 3 de salida por cada rama. . . . .	153
A.4	Representación de los movimientos que realizan los vehículos y el cálculo de la ocupación de un paso de simulación. . . . .	157
A.5	Representación de la posición de cada vehículo según el instante de simulación. . . . .	159
B.1	Representación con SUMO-GUI del mapa de TAPASCologne . . . . .	162
B.2	Fragmento del mapa de Colonia recortado para la ver versión reducida del escenario. . . . .	164
B.3	Evolución del número de <i>teleports</i> registrados según se modifica el parámetro <i>scale</i> . . . . .	166
B.4	Flujos de tráfico (volumen y tipo de giro) con patrones de vehículos conocido. . . . .	167
B.5	Diagramas de caja donde se analizan los conjuntos de flujos de entrada generados aleatoriamente para los experimentos Bx. . . . .	170

# Índice de tablas

2.1	Número de nodos y enlaces del mismo escenario de tráfico según la estructura de datos utilizada . . . . .	21
2.2	Comparación de los resultados obtenidos por los sistemas de gestión “Actuated traffic lights” y el sistema multiagente propuesto frente a los semáforos con fases estáticas (porcentaje total de vehículos cuyas rutas duran menos, igual o más tiempo). . . . .	42
2.3	Resultados de la simulación basada en el número de vehículos que han finalizado su viaje durante el tiempos de simulación (1 hora). Se han realizado un total de nueve simulaciones combinando tanto sistemas de gestión de la intersección basados en prioridad (P), en semáforos (TL) y en patrones de cruce optimizados (OCP), como volúmenes de tráfico bajos, medios y altos. . . . .	55
2.4	Resultados obtenidos durante los experimentos basados en datos de entrada con solución conocida para los cuatro procesos de optimización definidos. . . . .	66
2.5	Resultados obtenidos durante los experimentos basado en datos de entrada aleatorios para los cuatro procesos de optimización definidos. . . . .	67
3.1	Datos asociados a la publicación: Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management. . . . .	74
4.1	Datos asociados a la publicación: Automated Optimization of Intersections Using a Genetic Algorithm. . . . .	98
5.1	Datos asociados a la publicación: A Variable-Length Chromosome Genetic Algorithm to Solve a Road Traffic Coordination Multipath Problem.118	





# Parte I

---

Motivación de la investigación y  
resumen amplio de la tesis



# Introducción

Al tratarse de una tesis realizada mediante la modalidad de compendio de artículos, según la normativa vigente del programa de doctorado en Tecnologías de la información y las comunicaciones de la Universidad de Alcalá, el primer capítulo debe contener un resumen amplio de la investigación realizada. Para facilitar la lectura de esta tesis, se ha optado por realizar una división de este resumen amplio en dos capítulos. En el primero de ellos, el capítulo actual, se expondrá la motivación y los objetivos de la investigación. Se comenzará por realizar una revisión breve de los conocimientos previos que permitan poner en contexto el planteamiento del problema estudiado. Después, se definirán los objetivos y sus correspondientes contribuciones al ámbito de estudio. Finalmente se mostrará el esquema de la tesis. El resumen amplio de la tesis y su relación con los artículos que la fundamentan será mostrado en el capítulo 2.

## 1.1 Coordinación de vehículos en grandes ciudades

Los problemas relacionados con el tráfico rodado en las grandes ciudades suponen cuantiosas pérdidas económicas tanto para los conductores como para la propia sociedad. En el informe “*INRIX Global Traffic Scorecard*” [1], la ciudad de Madrid aparece entre las 25 más congestionadas a nivel mundial, donde sus habitantes gastan de media 129 horas anuales en atascos.

Tanto las razones que inician este tipo de problemas como sus consecuencias son muy heterogéneas. Los atascos o las situaciones de congestión son el efecto más común de estos problemas, pero no el único. Al igual que en otros ámbitos, estas situaciones de congestión pueden abordarse una vez han sido detectadas o tratar de preverlas para evitar que ocurran.

Históricamente se han realizado grandes esfuerzos por encontrar mejores soluciones para los problemas del tráfico de vehículos en núcleos urbanos debido a sus consecuencias. Los avances tecnológicos han ido dando soporte en muchos casos a la evolución de estas soluciones. Durante los años ochenta y noventa se crearon los llamados sistemas avanzados de gestión de tráfico (en inglés, *Advanced Traffic*

*Management Systems* o ATMS) [2] definidos como el sistema nervioso central de cualquier escenario de transporte. Estos sistemas reciben y combinan información desde diferentes fuentes de datos y llevan a cabo la toma de decisiones apoyándose en la tecnología que utilizan. Estos sistemas se han unificado en los denominados sistemas inteligentes de transporte o ITS (siglas del inglés *Intelligent Transportation Systems*).

Es especialmente relevante para esta tesis el estudio hecho en [3] sobre las causas críticas de los accidentes de tráfico en Estados Unidos. Aunque estas causas se definen como el evento que se produce justo antes de un accidente, no determinan que sean el único desencadenante del mismo. Este informe establece que entorno al 95 % de estas causas críticas son producidas por los conductores. De este grupo, más el 70 % se deben a errores de percepción y en la toma de decisiones. Estos problemas son recurrentes en las zonas de tráfico donde se produce un cruce o unión de carreteras tales como intersecciones, rotondas o carriles de incorporación [4]. Los sistemas cooperativos pueden ayudar a mitigar este tipo concreto de errores de conducción en entornos de vehículos conectados y autónomos o CAVs (del inglés *Connected and Autonomous Vehicles*) [5].

El paradigma de los CAVs es posible gracias a los avances en el ámbito de las tecnologías de la información y las comunicaciones o TIC (en inglés *Information and Communications Technology* o ICT). Actualmente la mayor parte de los vehículos incorporan, en menor o mayor grado, sensores que proporcionan información detallada sobre ellos y su entorno. Además, las técnicas utilizadas para la comunicación entre vehículos, como pueden ser las redes VANETs (siglas del inglés *Vehicular Ad-Hoc Networks*), facilitan el intercambio de información. Cuando este intercambio se produce en tiempo real permiten la conexión y cooperación entre usuarios de carreteras, infraestructuras y centros de control, dando lugar a los ITS cooperativos (C-ITS) [6].

Teniendo en cuenta la casuística de estos problemas de tráfico como, por ejemplo, que constan de elementos distribuidos geográficamente, existen subsistemas en entornos dinámicos, y que cada subsistema necesita interactuar con los otros de forma flexible, los sistemas basados en agentes pueden adaptarse a su resolución [7]. Ya que los agentes pueden operar de forma colaborativa, sin la intervención directa de una persona, permiten desarrollar sistemas de gestión o de control del tráfico que funcionen de forma autónoma. Desde el punto de vista de los mecanismos de optimización, los sistemas multiagente o MAS (del inglés *Multi-Agent Systems*) se basan en la comunicación entre agentes para la consecución de uno o varios objetivos.

Para dar soporte a las aplicaciones y servicios de los C-ITS son fundamentales los sistemas basados en comunicación entre vehículos (V2V), entre vehículos e infraestructuras (V2I), o incluso, entre vehículos y personas (V2P). Uno de los grupos de trabajo del Instituto de Ingeniería Eléctrica y Electrónica (IEEE) agrupa varios protocolos de acceso inalámbrico para su estandarización en entornos de vehículos. WAVE (del inglés *Wireless Access in Vehicular Environments*) se centra en dos familias de protocolos, la IEEE 802.11p (capa física y capa MAC) e IEEE 1609 (resto de capas superiores) [8]. Por su parte, los sistemas basados en comunicaciones móviles promovidos por organismos como 3GPP (*3rd Generation Partnership Project*) también han sido utilizados en entornos V2X (*Vehicle-to-everything*) [9]. De estos últimos, la puesta en marcha de la tecnología 5G desde finales de 2018 ha supuesto un gran avance respecto a las soluciones de comunicaciones móviles anteriores (4G/LTE). Planificada para un despliegue progresivo hasta 2025 tanto a nivel geográfico como tecnológico, tiene como objetivo conseguir redes de comunicación con características como latencias inferiores a un milisegundo, aumento exponencial del número de dispositivos a los que se puede dar servicio por unidad de área y disponibilidad cercana al 100 %. Este tipo de prestaciones proporcionan el marco ideal para los sistemas de comunicación en un entorno vehicular.

Otro concepto que cabe mencionar cuando nos referimos a una red de vehículos conectados es el denominado Internet de los Vehículos o IoV (del inglés *Internet of Vehicles*). Actualmente no se puede hablar de una única arquitectura estándar para desarrollar este concepto debido a que hay diferentes interpretaciones del mismo. Aunque algunas investigaciones lo definen como una extensión de IoT (del inglés *Internet of Things*) donde se añaden nuevos servicios relacionados con los vehículos, también hay otra tendencia que consideran la arquitectura IoV como algo más complejo. En [10] realizan la siguiente clasificación, según la funcionalidad de cada una de las capas que componen la que para ellos sería una arquitectura IoV:

- Capa de control y percepción del entorno de la red de vehículos: Encargada de todo lo que tiene que ver con la interacción con los sistemas de recolección de información desde cada vehículo, desde el entorno o por parte de los propias personas.
- Capa de enlace y de transporte: Su función principal es la de proporcionar acceso a la red, así, como procesado, análisis y transmisión de la información.
- Capa de control de la coordinación del cálculo: Esta capa pone a disposición de las aplicaciones todas las capacidades las redes de vehículos, en términos de procesamiento puro y de procesamiento distribuido de datos o asignación de recursos.

- Capa de aplicación: Proporciona diversos tipos de servicios para la correcta coordinación entre los vehículos, el entorno y las personas.

Desde el punto de vista de los elementos de la red de carreteras sobre los cuales es posible mejorar la gestión, las intersecciones se revelan como especialmente interesantes, ya que son un recurso compartido que limita el número de vehículos que puedan utilizarla al mismo tiempo. En este tipo de recursos se hace imprescindible aplicar técnicas de planificación o coordinación entre vehículos. En [6] realizan un estudio sobre los sistemas de gestión cooperativa en intersecciones o CIMs (siglas de *Cooperative Intersection Management*) donde analizan algunos de los sistemas de modelado de intersecciones y de las metodologías de coordinación existentes. De forma específica se centra en los métodos que utilizan o no la señalización para realizar estas tareas de coordinación. Siguiendo este trabajo, a continuación se muestra una posible clasificación sobre aspectos importantes en la gestión de intersecciones:

- **Tipo de modelado:**

- Discretización del espacio y el tiempo: División en porciones o espacios discretos del tiempo y/o el espacio con el fin de facilitar la asignación de recursos a cada vehículo que interviene en el escenario de tráfico. Trata de reducir el problema a un ejercicio de planificación o de reparto del derecho de uso de cada fragmento.
- En base a trayectorias: Los caminos predefinidos que pueden seguir los vehículos forman un conjunto de trayectorias. La intersección de dos o más de estas trayectorias en el mismo instante temporal da lugar a un punto de conflicto. El objetivo en este caso es generar un plan de trayectorias libre de conflictos.

- **Metodologías de coordinación del tráfico:**

- CIM centralizado: Existe un elemento encargado de recoger la información de parte o de toda ella, de realizar la toma de decisiones y envío de órdenes al resto de elementos del sistema, para que cada uno de ellos realice los cambios oportunos en base a su planificación. Esta gestión centralizada puede estar dividida dependiendo del rango de acción de este elemento central (gestión de una intersección, de un servicio o de un área geográfica).
- CIM distribuido: Al contrario que el caso anterior, no existe una unidad central de gestión. Las decisiones son tomadas de forma individual por cada vehículo en base a la información que percibe o a la que ha recibido a través de la comunicación con otros vehículos. Un ejemplo claro de este

tipo de CIM es la aplicación de sistemas multiagente para la gestión de intersecciones.

■ **Tipos de gestión:**

- Intersecciones señalizadas: Utilizan sistemas de señalización variable para interactuar con los vehículos y favorecer su coordinación. El caso de uso más extendido es el la gestión dinámica de las fases de un conjunto de semáforos.
- Intersecciones sin señalar: En la intersección no se utiliza ningún tipo de señalización. Los conductores, o los propios vehículos en un entorno de vehículos autónomos, reciben las acciones que deben realizarse para poder pasar a través de una intersección. Un ejemplo en este ámbito es el de la formación de pelotones de vehículos que, de forma coordinada entre sí, permitan su cruce sin colisiones.

La planificación de los desplazamientos o la coordinación de los vehículos a su paso por estructuras de carreteras donde abundan los puntos de conflicto es uno de los retos más importantes en los sistemas de gestión del tráfico urbano. Este problema se ha abordado desde múltiples ámbitos en el estado del arte. Teniendo en cuenta la comunicaciones V2X que se generan en las VANETs, la propuesta de protocolos como RTAR (*Reliable Traffic Aware Routing*)[11] se centra en cómo reenviar los paquetes de información con vehículos que transiten por carreteras fiables con el objetivo de mejorar la toma de decisiones sobre posibles encaminamientos en las intersecciones.

Sobre las soluciones existentes en entornos de vehículos autónomos y cooperativos, en [12] realizan una revisión bastante profunda sobre los métodos de resolución de problemas de coordinación para agruparlos entre los métodos basados en reglas y los basados en procesos optimización. El primero de ellos consiste en definir un conjunto limitado de reglas que todos los vehículos que intervienen en el escenario deben cumplir. Las principales ventajas de este tipo de soluciones es su simplicidad a la hora distribuir la información, limitando el número de mensajes que se deben intercambiar. Por otro lado, en las soluciones basadas en procesos de optimización, los problemas son definidos de forma matemática y tratados con los métodos tradicionales de resolución, utilizando herramientas y algoritmos de control óptimo. En un entorno CAVs, el grado de inteligencia aplicada a la gestión de situaciones críticas se incrementa, teniendo la expectativa de conseguir sistemas de tráfico más eficientes, cómodos y virtualmente libres de accidentes [13]. Esta eficiencia se consigue, entre otros muchos aspectos, a través de imponer separaciones entre vehículos inferiores y teniendo reacciones más rápidas antes posibles incidentes [5]. Aplicando técnicas de control óptimo, Bichiou y Rakha enfocan su trabajo en [14] a como optimizar el



movimiento de los vehículos autónomos a través de intersecciones. Ellos plantean este caso como un problema de optimización sujeto a restricciones dinámicas como las generadas por las ecuaciones diferenciales que regulan el movimiento de los vehículos, y restricciones estáticas como las velocidades máximas alcanzables.

## 1.2 Planteamiento del problema

Los avances en las tecnologías y técnicas utilizadas han llevado a la resolución más eficiente de los problemas tradicionales de tráfico. En cuestiones como la toma de decisiones basada en la información del entorno en un escenario de tráfico, la proliferación de fuentes de datos ha provocado, en algunos casos, la idea de que la aplicación de técnicas basadas en el paradigma del Big Data permiten resolver cualquier clase de problema. Sin embargo, no se debe confundir la cantidad con la calidad de esos datos. Es necesario contar con técnicas que permitan establecer *a priori* un método de valoración de las fuentes de información disponibles, facilitando por un lado focalizar los recursos disponibles en los datos más valiosos, y por otro lado permitiendo realizar un despliegue más eficaz de los dispositivos de monitorización.

Actualmente, esta valoración sobre la información o sobre las posibles ubicaciones de sensores, se realiza principalmente a través del estudio de datos de tráfico históricos o recurriendo a ubicaciones cercanas a infraestructuras problemáticas. La primera de estas técnicas requiere el uso de sistemas de simulación basados en, o bien modelos de tráfico generados a partir de información histórica de la red, o de otras redes con una estructura compatible. Esta metodología plantea limitaciones temporales en cuanto a que, para nuevos desarrollos de carreteras, la información del tráfico real no se obtendrá hasta después de su construcción, así como limitaciones sobre la propia confianza en el análisis de los datos.

El análisis de la importancia de cada enlace de una red dada su posición y características dentro de ella, de forma aislada respecto a los flujos de tráfico, podría proporcionar indicadores independientes que faciliten tareas como la elección de datos para la toma de decisiones, la ubicación de sensores de forma más eficiente y la posible mejora de la planificación de carreteras de forma previa a su construcción. En teoría de grafos, el uso de medidas de centralidad de los nodos o los enlaces ha resultado muy útil en aplicaciones de diversa índole como la identificación de desviaciones y monitorización relacionadas con instituciones financieras [15] o la previsión de posibles nodos críticos en una red eléctrica [16].

A partir de escenarios basados en CAVs, los esfuerzos por mejorar las técnicas de coordinación de vehículos en intersecciones pueden ser abordados desde otros enfoques, debido a la eliminación progresiva de alguna de las barreras tecnológicas que existían. La estandarización de comunicaciones inalámbricas en entornos de vehículos (WAVE) y el uso de protocolos de comunicación dedicada de corto alcance (en inglés Dedicated Short-Range Communications o DSRC) permiten ya aplicaciones de seguridad activa basadas en comunicaciones V2X [17].

La generación de patrones de llegada a la intersección sincronizados (SIAP por sus siglas en inglés) propuesto en [18] se basa en la obtención de secuencias de vehículos, en unidades de tiempo preestablecidas, de tal forma que permitan su paso por una tipo de intersección concreta, sin detenerse y sin causar colisiones. Este tipo de solución que utiliza la formación de pelotones de vehículos, bajo ciertas condiciones, para mejorar el uso eficiente de los recursos o garantizar la coordinación de los vehículos no es única y puede verse en otros trabajos como [19] o [20].

Estas soluciones tienen importantes limitaciones en cuanto a la escalabilidad de la aplicación de determinadas metodologías a intersecciones diferentes a las utilizadas en cada estudio, y a obtener objetivos diferentes a los planteados. Por este motivo, partiendo de alguno de los conceptos planteados por anteriores trabajos, el primer problema que se abordará en este trabajo de tesis se centra en diseñar mecanismos que permitan generalizar la aplicación de técnicas conocidas a cualquier tipo de intersección. Tras obtener un método para el modelado de las intersecciones, el segundo de los retos es mejorar los procesos de optimización de coordinación de vehículos para, en base al modelo realizado, poder obtener el nivel de coordinación entre vehículos necesario para alcanzar los objetivos fijados en cada momento. Además, para garantizar la escalabilidad del sistema final, tanto el método de modelado como el optimizador deben poder ejecutarse como un proceso automatizado.

### 1.3 Objetivos y ámbito de las contribuciones

En este apartado se definirán los objetivos de esta tesis doctoral, poniendo en contexto la motivación de cada uno de ellos y, exponiendo en cada caso, las contribuciones que supondrían dentro de su ámbito. Todos los objetivos están pensados para lograr la misión central de esta tesis de contribuir a las técnicas de coordinación de vehículos cuando estos deben enfrentarse a la gestión de puntos de conflicto como los que se dan a su paso por cruces de carreteras o intersecciones.

### 1.3.1 Optimización de la ubicación de sensores según las características de la red

Los sensores, cuya función es percibir magnitudes físicas de su entorno y generar valores numéricos en base a ello, cuando trabajan en grupo de forma colaborativa son denominados redes de sensores (en inglés *Sensor Networks*). Además, cuando estas redes están desplegadas utilizando mecanismos de comunicación inalámbrica reciben el nombre de WSN (del inglés *Wireless Sensor Networks*). En el ámbito de la gestión de redes de carreteras, las WSN son ampliamente utilizadas para recoger información del escenario y utilizar ese conocimiento para la toma de decisiones. Trabajos como el recogido en [21] muestran las ventajas de utilizar este tipo de despliegues de sensores para alimentar a sistemas de gestión de tráfico, siendo en este caso utilizada para evitar las congestiones realizando acciones de control de flujo.

Desde el punto de vista de la optimización de las WSN, en [22] analizan el tipo de topologías existentes para este tipo de redes, exponiendo las ventajas e inconvenientes que presenta cada una de ellas. La mayor parte de los trabajos relacionados con la optimización se centran en el problema de la búsqueda de las mejores ubicaciones para los sensores en una red de tráfico. Las técnicas utilizadas en algunos de estos trabajos se basan en la aplicación de algoritmos de tipo evolutivo. Kulkarni y Venayagamoorthy en [23] realizan una investigación donde aplican PSO (del inglés *Particle Swarm Optimization*) para obtener el número y posición óptima de un conjunto de sensores en términos de cobertura, conectividad y eficiencia energética. Para conseguir este fin, necesitan realizar múltiples simulaciones con las posiciones candidatas en cada iteración del algoritmo.

Otro enfoque, en cuanto a la metodología para resolver el problema, es el uso de datos de tráfico históricos. Shan y Zhu proponen en [24] utilizar información procedente del GPS (del inglés *Global Positioning System*) de cerca de 8000 taxis y generar un grafo con las carreteras por las que han circulado. Los pesos de los enlaces se definen en base al número de veces que ha pasado cada vehículo por dicho enlace. Por último, aplican un algoritmo de tipo voraz que determine las ubicaciones idóneas. Con estas dos opciones, el despliegue de sensores siguiendo una topología conocida y el uso de información histórica, se engloban la mayor parte de las soluciones existentes. Siguiendo estas metodologías se genera la problemática de la dependencia de la ubicación de estos sensores de datos que deben ser preexistentes a su utilización o del grado de confianza que se tenga el uso de topologías genéricas.

*El objetivo de esta tesis en cuanto a la mejora en los métodos de ubicación de sensores en una red de tráfico de vehículos es desarrollar una propuesta basada en el estudio de medidas de la propia red, como puede ser la centralidad de cada uno de sus nodos o enlaces, para obtener una metodología de búsqueda de ubicaciones idóneas que dependa en menor grado de los datos históricos de la propia red.*

La consecución de este objetivo permitirá contar con una métrica que evalúe las ubicaciones elegidas para la colocación de sensores sin depender de factores externos a las características de la propia red de carreteras. Estos nuevos métodos podrán ser utilizados en combinación con las técnicas ya existentes, resultando especialmente útiles en los escenarios donde no se disponga de información previa de flujos de tráfico, o en los que aún disponiendo de ella, se tenga un grado de confianza bajo en la fiabilidad de estos.

### 1.3.2 Modelado automático de intersecciones

Los elementos que interactúan con una intersección requieren de un modelo de ella que represente sus características de forma esquemática y unívoca. Dependiendo de cada elemento, la información que debe almacenar el modelo puede variar desde el mapa realizado a partir de las coordenadas geográficas que forman la intersección, hasta el conjunto de señalizaciones que permite o restringe determinadas trayectorias en su interior. Siguiendo los aspectos señalados en [6], para el modelado de las intersecciones se deben tener en cuentas dos factores: el proceso de discretización en el espacio y el tiempo, y las trayectorias interiores. El primero de ellos se centra en el proceso seguido para pasar de un tiempo y espacio continuo a uno discreto. La mayor parte de trabajos que abordan este problema lo hacen desde el punto de vista de la codificación de las entradas y salidas de la intersección, planteando modelos que permitan resolver un problema concreto, pero sin entrar a definir una metodología para el reparto del espacio interior de la intersección. Por ejemplo, en [25] modelan las características de una intersección como una red de Petri, la cual es posteriormente utilizada para generar las políticas de control del sistema. Otro enfoque se encuentra en los estudios que únicamente modelan las trayectorias interiores que siguen los vehículos en su paso por la intersección. En un entorno de vehículos autónomos y conectados, en [26] se aborda este tema definiendo la metodología para la generación de posibles trayectorias y codificándolas a nivel de cada vehículo con programación no lineal mixta entera.

Existen propuestas como la realizada en [27] donde modelan tanto la características internas como externas de una intersección. En este caso, aplican un TCA (siglas del inglés Traffic Cellular Automata) [28] para reproducir los movimientos de los

vehículos. Presenta la limitación de que además de no definir todos sus elementos, únicamente realizan la propuesta para una intersección con formas geométricas regulares.

*El objetivo de esta tesis en relación con los métodos de modelado de intersecciones es realizar una propuesta de metodología de modelado que permita, a partir de características conocidas de la intersección como su forma, número de brazos, carriles por brazo y restricciones de giro, generar un modelo donde cada elemento esté etiquetado unívocamente utilizando un proceso automatizado.*

La posibilidad de contar con una metodología de modelado de intersección, capaz de ser aplicada de forma automática, que contenga tanto sus características físicas como las posibles restricciones en el comportamiento de los vehículos según la posición que ocupan, facilitará los procesos de optimización en los que se utilice. Estos procesos podrán centrarse en conseguir el objetivo para el que hayan sido diseñados, teniendo la seguridad de que la solución encontrada será extrapolable a la realidad. Es importante destacar que, si la metodología no requiere de intervención externa por parte de una persona, este proceso puede llegar a ser utilizado tanto para el modelado como para su posterior utilización por entidades autónomas como los agentes.

### 1.3.3 Mecanismos de coordinación de vehículos en intersecciones

Dentro de los escenarios de tráfico de vehículos, la coordinación de vehículos es un objetivo extensible a casi cualquier tipo de situación. Cuando dos o más vehículos tratan de utilizar un mismo recurso, entendiendo este recurso como un fragmento de la red de carreteras, en un mismo instante temporal, se produce una situación de contienda donde se debe decidir una secuencia ordenada de utilización de este recurso. Los posibles puntos de conflicto de una red de carreteras surgen en los lugares donde varias trayectorias coinciden [6].

Los mecanismos de coordinación pueden agruparse entre los sistemas que se sirven de la señalización de las carreteras para influir en el comportamiento de los vehículos, o los que interactúan directamente con los elementos que componen el escenario (vehículos, infraestructuras y persona) para llegar a acuerdos sobre el derecho de uso de los recursos.

Existen sistemas comerciales para controlar la señalización de puntos clave de la red como las intersecciones. SCATS (*Sydney Coordinated Adaptive Traffic System*) [29] y

SCOOT (*Split, Cycle and Offset Optimization Technique*) [30] son dos de los sistemas más extendidos que utilizan la metodología UTC (*Urban Traffic Control*) para variar las fases de los semáforos de los cruces, carriles de incorporación o accesos, en base a la información recibida de sensores ubicados en algunos puntos de la red. Desde el punto de vista teórico del problema, Hult *et al.* in [31] modela el caso de la coordinación de vehículos como un problema de control óptimo con restricciones en tiempo finito. Estas técnicas suelen ofrecer buenos resultados para un caso aislado concreto, pero presentan problemas de escalabilidad al aumentar la complejidad del escenario a analizar. Respecto a los métodos basados en sistemas no señalizados, en [32] se propone el uso de los llamados “*virtual traffic lights*”. Estos semáforos virtuales actúan de igual forma que los convencionales, pero se sirven de técnicas de comunicación entre vehículos para que sean ellos los que toman la decisión de qué vehículos o grupo de vehículos tienen prioridad al llegar a un cruce.

*El objetivo de esta tesis en cuanto a los mecanismos de coordinación de vehículos en intersecciones es proponer, tanto una metodología para codificar el problema de tal forma que pueda ser resuelto con procesos de optimización ampliamente utilizados, como la mejora de las técnicas utilizados para este propósito.*

La consecución de este objetivo permitirá tener una técnica fiable y eficiente que facilite la coordinación de los vehículos en intersecciones. La superación de este objetivo, junto con el descrito en el apartado 1.3.2, permitirá tener un sistema global de optimización que podrá llevarse a cabo de forma automática. Esto es algo fundamental para afrontar la tarea de optimizar una red de tráfico urbana, la cual suele tener una densidad de cruces o intersecciones muy elevada.

## 1.4 Estructura de la tesis

Esta tesis, con el fin de facilitar la presentación de las contribuciones realizadas, se ha estructurado en cuatro partes, siete capítulos y dos apéndices, tal y como se describe a continuación:

### **Parte I: MOTIVACIÓN DE LA INVESTIGACIÓN Y RESUMEN AMPLIO DE LA TESIS**

De acuerdo con la normativa vigente del programa de doctorado de Tecnologías de la Información y las Comunicaciones de la Universidad de Alcalá, la primera parte de esta tesis debe contener un resumen amplio de la investigación realizada. Previo a este resumen, se presentan de forma clara el ámbito de la tesis, sus objetivos y la estructura del presente libro.

**Capítulo 1, Introducción:** En él se expone la motivación y los objetivos de la investigación. Se comienza por mostrar una breve revisión de los trabajos previos que permiten poner en contexto el ámbito de la investigación y el planteamiento del problema a tratar. A continuación, se definen los objetivos de la tesis y las contribuciones derivadas de ellos en el ámbito de estudio. Finalmente se muestra el esquema de esta tesis.

**Capítulo 2, Estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas:** En el segundo capítulo se realiza un resumen amplio de la tesis donde se expone la investigación realizada de forma ordenada, introduciendo en cada apartado un breve estado del arte que permita sustentar las posteriores propuestas realizadas. De igual forma, se muestran los experimentos llevados a cabo y sus resultados. Al final de este capítulo se ha incluido un diagrama que, de forma esquemática, relaciona cada una de las contribuciones y trabajos realizados durante la tesis con los artículos incluidos en el compendio.

## **Parte II: ARTÍCULOS PUBLICADOS QUE DAN LUGAR A LA TESIS**

La segunda parte de este libro recopila los artículos que avalan la tesis doctoral. Para cada uno de estos artículos se ha incluido su versión impresa final en un capítulo propio. Al inicio de cada capítulo se realiza un resumen del artículo y incluye una tabla con valores y métricas relacionadas con la revista donde está publicado y sobre su relevancia.

**Capítulo 3, Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management:** El primero de los artículos comienza por realizar una propuesta para obtener las ubicaciones ideales de sensores en una red de tráfico. El segundo bloque del artículo está dedicado a la propuesta e implementación de un sistema multiagente para la gestión de las intersecciones a través de semáforos.

**Capítulo 4, Automated Optimization of Intersections Using a Genetic Algorithm:** En este capítulo se presenta el segundo de los artículos el cual propone un sistema de modelado automatizado de intersecciones que es utilizado como base para los procesos de optimización posteriores. El objetivo del optimizador propuesto es conseguir generar la cadencia de llegada de vehículos a una intersección que, permitiendo su cruce coordinado de forma segura, obtenga su máxima ocupación media.

**Capítulo 5, A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem:** El tercero de los artículos que avalan esta tesis doctoral extiende el modelo propuesto en el capítulo 4, aumentando la complejidad del problema a resolver, pero

consiguiendo a su vez un optimizador más versátil, capaz de obtener patrones de llegada a las intersecciones según las preferencias de cada uno de sus flujos de entrada.

### **Parte III: CONCLUSIONES**

En esta parte se remarcan las aportaciones del trabajo realizado y se realiza una discusión sobre los resultados obtenidos. Además, se introducen posibles futuras líneas de trabajo a partir de las contribuciones de esta tesis.

**Capítulo 6, Conclusiones y líneas futuras:** Se discuten los resultados obtenidos en cada uno de los capítulos anteriores y se detallan los objetivos alcanzados durante la investigación, examinando tanto las publicaciones principales como las secundarias fruto de esta tesis. A partir de lo anterior, se resaltan las contribuciones realizadas en cada apartado. Por último se enuncian posibles líneas de trabajo en las que puedan resultar útiles las contribuciones obtenidas a la consecución de esta tesis doctoral. Estas líneas de investigación pueden estar dirigidas a continuar con los trabajos realizados en esta tesis como en la aplicación de las técnicas desarrolladas en otros ámbitos de estudio.

### **Parte IV: APÉNDICES**

Durante el desarrollo de la tesis se han utilizado datos sobre tráfico de vehículos y se han implementado herramientas concretas para poder llevar a cabo la investigación. En esta última parte se recogen los desarrollos realizados y los datos utilizados con el fin de garantizar la reproducibilidad de los experimentos que validan las contribuciones de la tesis.

**Apéndice A, Recursos y herramientas para la investigación en un escenario de tráfico urbano:** Este apéndice se divide en dos partes. La primera de ellas está dedicada al funcionamiento interno del simulador de intersecciones desarrollado. En un segundo apartado se detalla el sistema multiagente emulado y el sistema de comunicación utilizado para conectarse con el simulador de tráfico microscópico SUMO.

**Apéndice B, Escenarios de tráfico urbanos y datos de entrada:** Este apéndice muestra en detalle los escenarios de tráfico utilizados, su origen, variaciones realizadas respecto al original y sus procesos de calibración. También se exponen los conjuntos de datos generados de forma específica para ser utilizados en los experimentos de la tesis que recrean diferentes condiciones de flujos de tráfico en nuestro simulador de intersecciones.





# Resumen amplio: Estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas

Este capítulo recoge un resumen amplio de la investigación llevada a cabo en esta tesis doctoral, mostrando las propuestas realizadas, sus desarrollos asociados, los resultados obtenidos y la relación de estos con los artículos incluidos en los capítulos 3, 4 y 5. Con el fin de facilitar la lectura y comprensión del proceso llevado a cabo durante esta investigación, en las primeras secciones se muestra el trabajo realizado dividido por temáticas. Esta separación coincide también con el orden cronológico de las contribuciones, permitiendo que los nuevos conceptos que se proponen sean mostrados antes de ser utilizados en sistemas posteriores más complejos. La última sección de este capítulo está dedicada a relacionar de forma esquemática las contribuciones de esta tesis, los objetivos alcanzados y las publicaciones realizadas.

## 2.1 Escenarios de tráfico urbano

La obtención de información es una de las principales tareas a las que se enfrentan los ingenieros y gestores de los centros de tráfico. Esta información es utilizada como apoyo para los sistemas de control de tráfico, la planificación del transporte, la evaluación del rendimiento de las acciones realizadas y, en general, para la toma de decisiones que llevan a cabo todos los procesos que se engloban en los sistemas inteligentes de transporte (ITS, del inglés *Intelligent Transportation Systems*) [33].

La evolución de las tecnologías en los últimos años, especialmente en lo que respecta a la conexión de casi cualquier tipo de dispositivo a Internet, ha permitido la generación masiva de datos en entornos de vehículos. Esta conectividad casi total ha dado lugar al concepto del denominado Internet de los Vehículos (*Internet of Vehicles*, IoV por sus siglas en inglés) [34]. Aunque la información que pueden llegar a proporcionar este tipo de sensores es muy importante, los sistemas tradicionales como las cámaras de tráfico, los detectores inductivos o los detectores basados en

láseres o microondas, siguen siendo un pilar fundamental en la toma de decisiones dada su fiabilidad y disponibilidad.

Una de las hipótesis de partida de esta tesis es que es posible mejorar los sistemas de recolección de datos y toma de decisiones en un entorno de tráfico urbano a través del estudio de las características intrínsecas de las redes de carreteras. Un escenario de tráfico podría definirse como los elementos y circunstancias que, a través de sus relaciones, caracterizan el comportamiento de un conjunto de vehículos en un espacio físico y temporal concreta. En la literatura actual, la resolución del denominado problema de ubicación de una red de sensores (NSLP, del inglés *Network Sensor Location Problem*) en un entorno de tráfico de vehículos es abordada principalmente utilizando dos métodos: la observación del tráfico y el uso de técnicas iterativas que evalúan su eficacia a través de simulaciones. Sobre el primero de los métodos, Castillo *et al.* proponen en [35] para este contexto un sistema centrado en la definición de flujos de tráfico basados en la observación. De forma similar, en [36] se propone realizar matrices de orígenes y destinos a partir de los datos históricos de desplazamientos en la zona de estudio. Ambas soluciones obtienen los puntos de la red con mayor número de rutas, siendo estas posiciones las elegidas para determinar las ubicaciones óptimas de los sensores. En relación con los métodos iterativos, el uso de algoritmos evolutivos está ampliamente extendido. Kulkarni y Venayagamoorthy [37] utilizan la optimización por enjambre de partículas (PSO, del inglés *Particle Swarm Optimization*) para obtener la ubicaciones idóneas de una red de sensores. Estos autores optimizan las ubicaciones teniendo en cuenta el rango de cobertura de cada sensor, así como cuestiones de conectividad y de eficiencia energética. Por último, una solución intermedia es la propuesta por Shan and Zhu en [24] donde a partir de las rutas obtenidas de los datos de GPS (sistema de posicionamiento global, del inglés *Global Positioning System*) de 8000 taxis, aplican un algoritmo voraz para obtener las posiciones idóneas.

### 2.1.1 Análisis de las redes de tráfico como grafos

Las redes de carreteras son la base de cualquier escenario de tráfico. Compuestas por caminos que conectan un origen a un destino, pueden tener diferentes características como el número de carriles, sentidos de circulación, variación de la velocidad máxima de circulación, etc. El punto donde dos o más carreteras se conectan da lugar a un cruce, el cual, puede implementarse a través de una intersección, rotonda, raqueta, carril de incorporación, o cualquier otra estructura diseñada con este fin. Este esquema de caminos o enlaces, y cruces o nodo, facilita el tratamiento de este tipo de redes como grafos. Aunque en un principio pudiera parecer que la semejanza entre una red de carreteras y un grafo permitiría un proceso de conversión directo,

las peculiaridades de las carreteras hacen necesario la elección de un tipo de grafo muy concreto y un proceso de adaptación.

Las ventajas de modelar una red de tráfico como un grafo son significativas, ya que a partir de ese momento es posible aplicar sobre esa red los algoritmos o técnicas de optimización ya conocidos. En un trabajo previo, analizamos esta problemática en [38]. Teniendo en cuenta las características de las redes de carreteras se determinó que el grafo utilizado debe ser **dirigido**, debido a que los sentidos de circulación de una carretera pueden hacer que haya un posible enlace desde el punto de inicio al de fin, pero no a la inversa. También debe ser un **multigrafo** o **pseudografo** ya que es común que entre dos puntos de la red haya más de un camino posible (por ejemplo, una carretera principal y una vía de servicio con mismos puntos de origen y destino).

Otro problema a tener en cuenta es la posibilidad de la presencia de giros prohibidos en una red. En la figura 2.1a se muestra un ejemplo básico de un cruce de carreteras. Siguiendo las restricciones marcadas por la señalización de las vías, un posible camino válido podría ser 'B → E → D'. Si se transforma esta red de forma directa a un multigrafo dirigido como el representado en la figura 2.1b, el camino 'C → E → D' podría ser válido pero, aunque esta la definición multigrafo dirigido permitiera este camino, se realizaría un giro prohibido desde C hasta D. Una posible solución a esta cuestión es la planteada en [39] donde se utiliza para este proceso un grafo línea dirigido (en inglés, *line digraph*) con una asignación de costes concreta, denotándolo como *pseudo-dual graph*. Un grafo línea dirigido de un grafo dirigido 'G' es otro grafo que representa las adyacencias entre las aristas de 'G', es decir, los enlaces o aristas de este nuevo grafo representan los giros permitidos del grafo original. En la figura 2.1c se puede observar el grafo obtenido con este segundo proceso de conversión.

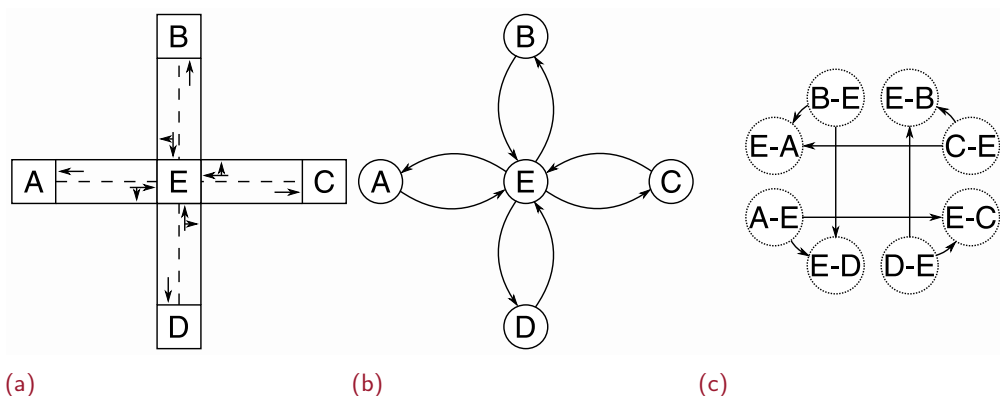


Figura 2.1.: Ejemplo de transformación de una red de carreteras a un grafo. (a) Cruce de carreteras original. (b) Conversión a multigrafo dirigido. (c) Conversión a *pseudo-dual graph*.

### 2.1.1.1. Pseudo-Dual Graph

Aplicando el concepto presentado en [39], el proceso a seguir para obtener el denominado *pseudo-dual graph* comenzaría por establecer los elementos que intervienen en esta conversión. La notación utilizada será la siguiente:

- $G$ : Grafo origen obtenido de la conversión directa de la red de carreteras.
- $N$ : Conjunto de nodos de  $G$ .
- $E$ : Conjunto de enlaces de  $G$ , siendo  $e_i$  cada uno de ellos.
- $f(e_i)$ : Nodo inicio del enlace  $e_i$ .
- $l(e_i)$ : Nodo fin del enlace  $e_i$ .
- $\delta^+(n)$ : Número de enlaces que terminan en un nodo.
- $\delta^-(n)$ : Número de enlaces que empiezan en un nodo.
- $W(r)$ : Coste del camino  $r$ .

A partir de esta notación el coste  $W$  de seguir la ruta  $r$  se define con la siguiente expresión:

$$W(r) = w(e_1) + \sum_{i=2}^k (w(e_{i-1}, e_i) + w(e_i))$$

donde  $w(e_{i-1}, e_i)$  representa el coste de atravesar el nodo desde el enlace anterior hasta  $e_i$  y  $w(e_i)$  es el coste del propio enlace. De esta forma, el valor mínimo de  $W$  sobre los posibles caminos  $r$  que conectan  $f(e_x)$  con  $l(e_y)$  nos proporcionará el camino con menor coste.

Teniendo en cuenta lo anterior, se definen las condiciones que se deben cumplir para que dado un grafo  $G(N, E)$  sea posible obtener un **pseudo-dual graph** denotado por  $D(N_D, E_D)$ :

1. Para cada enlace  $e_i \in G$  hay un nodo  $v_i \in D$  con  $v_i = d(e_i)$ . ( $d^{-1}(v_i) = e_i$ )
2. Por cada par de enlaces consecutivos  $(e_i, e_j) \in G$  hay un enlace  $\varepsilon \in D$  entre sus correspondiente nodos  $v_i = d(e_i)$ ,  $v_j = d(e_j)$ , tal que  $f(\varepsilon) = v_i$  y  $f(\varepsilon) = v_i$ .
3. Existen funciones de coste tal que  $w^\varepsilon : E_D \rightarrow \mathbb{R}^+$  y  $w^v : N_D \rightarrow \mathbb{R}^+$

Este tipo de grafo nos permite aplicar algoritmos de búsqueda de camino u obtener características del grafo con la seguridad de representar fielmente las restricciones de la red de carreteras original. El mayor de sus inconvenientes es el aumento en la complejidad del nuevo grafo. Aplicando esta conversión al mapa del escenario de TAPASCologne mostrado en el apéndice B.1 obtenemos los datos de la tabla 2.1.

Tipo de estructura de datos	Nodos	Enlaces
Red de carreteras	30 354	68 642
Multigrafo dirigido	30 354	68 642
Pseudo-Dual Graph	68 642	180 341

**Tabla 2.1.:** Número de nodos y enlaces del mismo escenario de tráfico según la estructura de datos utilizada

Cuando se realiza la conversión directa de la red de carreteras a un grafo dirigido, como era de esperar, el número de nodos y enlaces se mantiene. Por el contrario, al obtener el *pseudo-dual graph* el número de nodos es igual al número de enlaces de las dos estructuras anteriores y su número de enlaces aumenta hasta representar todos los posibles giros permitidos.

### 2.1.1.2. Centralidad aplicada a escenarios de tráfico

La centralidad de un grafo es la medida de la importancia que cada vértice tiene en el propio grafo. Aunque existen diversos tipos de medidas de centralidad en base a los parámetros o algoritmos utilizados para valorar un vértice (cercanía, grado, intermediación, Katz, etc.), en redes de carreteras urbanas es común utilizar la centralidad de intermediación o *betweenness* [40]. Esta centralidad basada en los caminos más cortos es una medida estándar de control utilizada ampliamente en estudios e implementada en muchos programas de análisis de redes [41]. Moura *et al.* plantean en [42] un sistema para posicionar torres de recepción en VANET (red ad-hoc vehicular, del inglés *Vehicular Ad-Hoc Network*) utilizando un algoritmo genético y realizando un preprocesamiento de la red aplicando la centralidad *betweenness*. Los autores obtienen hasta un aumento del 20% de la cobertura en escenarios concretos, con el mismo número de puntos de recepción. Otro estudio que aborda la validez de este tipo de medida de centralidad es el realizado en [43] donde en sus resultados muestran como los vehículos que pueden cursarse por un determinado punto de la red de carreteras depende de dos factores, el primero la capacidad de interconexión de ese punto, medido a través de su centralidad *betweenness*, y en segundo lugar la demanda de los propios trayectos.

En el siguiente apartado mostraremos la propuesta realizada que hace uso de esta medida de centralidad.

### 2.1.2 Propuesta de localización de sensores en una red de tráfico

Teniendo en cuenta los aspectos vistos anteriormente, es fundamental contar con información de los escenarios de tráfico para mejorar la toma de decisiones por los centros de control.

Nuestra contribución en este campo consiste en la propuesta de una metodología para elegir las ubicaciones idóneas de un conjunto limitado de sensores, que sean capaces de recoger la máxima cantidad de información posible del escenario de tráfico en cuestión. Esta información, unida al histórico de desplazamiento disponibles, podrá utilizarse para detectar alteraciones en el comportamiento de los flujos de tráfico, permitiendo a los sistemas gestores reaccionar de forma proactiva.

En concreto, la propuesta consiste en modelar la red de carreteras como un grafo dirigido que será transformado en una *pseudo-dual graph* y estudiada la centralidad de sus nodos. Debemos recordar que, en este tipo de grafos, los nodos se corresponden con los enlaces del grafo original, por lo que estaremos realizando una clasificación de los enlaces más centrales de la red. Esta información será combinada con los datos históricos de ocupación de los enlaces de la red, para obtener finalmente una lista de posibles ubicaciones.

En la ecuación 2.1 se muestra el cálculo de la centralidad *betweenness* para cada nodo, tal y como se propone en [44].

$$c_B(v_i) = \sum_{s,t \in V} \frac{\sigma(s,t|v_i)}{\sigma(s,t)}, \quad (2.1)$$

donde  $V$  es el conjunto de nodos del grafo dirigido,  $\sigma(s,t)$  es el número de caminos más cortos entre  $s$  (nodo origen) y  $t$  (nodo destino), y  $\sigma(s,t|v_i)$  es el número de esa rutas que pasan por un nodo  $v_i$  diferente de  $s$  y  $t$ .

A continuación, se van a enumerar las estructuras de datos que forman el método propuesto, definiendo cada una de ellas y detallando los pasos necesarios para su construcción:

1.  $L_c$ : Lista ordenada de enlaces según su valor de centralidad. Para su obtención se deben seguir los siguientes pasos:

a) Conversión de la red de tráfico en un grafo dirigido ( $G$ ):

$$G(V, E), \quad (2.2)$$

donde  $V$  y  $E$  son el conjunto de nodos y enlaces de la red.

b) Cálculo de la centralidad *betweenness* de  $G$ . Existen implementaciones para calcular directamente la centralidad de los enlaces de un grafo [45], pero en nuestro caso realizaremos la transformación de  $G$  en un *pseudo-dual graph* ( $G_{LG}$ ) siguiendo los pasos indicados en la sección 2.1.1.1.

$$G_{LG}(V_L, E_L), \quad (2.3)$$

$$V_L = E, \quad (2.4)$$

donde  $V_L$  es el conjunto  $E$  del grafo  $G$  y  $E_L$  es el conjunto compuesto por todos los posibles giros desde cada enlace de  $G$ . La medida de centralidad de  $G_{LG}$  es el conjunto  $S_c$ :

$$S_c = \bigcup_{i=0}^n \{(v_{L_i}, c_B(v_{L_i}))\} \quad (2.5)$$

Aplicando la ecuación (2.4) en la ecuación (2.5) obtenemos:

$$S_c = \bigcup_{i=0}^n \{(e_i, c_B(v_{L_i}))\}, \quad (2.6)$$

donde  $e_i$  es cada uno de los  $n$  elemento del conjunto  $E$  y  $c_B(v_{L_i})$  es la medida de centralidad de cada  $e_i$ .

c)  $L_c$  es una lista resultado de ordenar en orden decreciente el conjunto  $S_c$ . Los valores de centralidad de esta lista son normalizados tomando como valor máximo 1.

2.  $L_d$ : Lista de enlaces ordenados por su nivel histórico de ocupación. Esta lista es generada como se explica a continuación:

a) Adquisición de los valores de tráfico históricos. Estos valores pueden obtenerse desagregados o como un modelo de tráfico, el cual requiera de un proceso de simulación previo para la obtención de valores numéricos.  $S_d$  se define como:

$$S_d = \bigcup_{i=0}^n \{(e_i, v_{d_i})\}, \quad (2.7)$$



donde  $e_i$  es cada uno de los  $n$  elementos del conjunto  $E$  y  $v_{d_i}$  es el valor histórico de ocupación de cada  $e_i$ .

b)  $L_d$  es la lista ordenada de forma decreciente del conjunto de valores  $S_d$ . Esta lista de valores históricos de ocupación es normalizada en base a 1.

3.  $L_p$ : Lista obtenida de la combinación de las listas  $L_c$  y  $L_d$ . Esta combinación no tiene porqué ser proporcional, ya que se añade un factor  $\alpha$  cuyo valor dependerá de la calidad y confianza que tengamos sobre los datos históricos:

$$L_p = (1 - \alpha)L_c + \alpha L_d, \quad (2.8)$$

donde  $\alpha$  mide el grado de confianza en los datos históricos de ocupación y es un valor decimal entre 0 y 1 (1 indica el valor más próximo a la situación real). Además, tras formar la nueva lista  $L_p$ , esta es ordenada de forma decreciente.

Tras finalizar este proceso,  $L_p$  proporciona una lista ordenada de los enlaces más relevantes teniendo en cuenta su posible grado de ocupación.

### 2.1.3 Experimentos y resultados

Para validar la propuesta realizada sobre la ubicación de sensores basada en la centralidad de la red de tráfico, se ha llevado a cabo un conjunto de experimentos sobre el escenario de tráfico reducido de la ciudad alemana de Colonia [46] mostrado en el apéndice B.1.1. Este escenario es simulado utilizando SUMO (*Simulation of Urban MObility*)[47], un simulador de tráfico microscópico.

Se ha elegido como sensor de estudio las cámaras de tráfico para este experimento por ser un tipo ampliamente utilizado para la recolección de valores de ocupación de las vías. Para determinar el alcance de una cámara estándar es necesario fijar alguna de sus características como el tipo de lente de la cámara o la altura de instalación sobre la carretera. En [48] se muestra un estudio de tipos de sensores de tráfico donde se puede encontrar el análisis de las cámaras de tráfico. Utilizando los valores más comunes, la cámara tiene un campo de visión de 124 metros de profundidad y 15.7 metros de ancho.

El conjunto de pruebas consiste en la colocación de un número de cámaras sobre el escenario de tráfico mencionado anteriormente, analizando el conteo de vehículos que realizarán durante el tiempo de simulación, teniendo en cuenta su radio de acción. Para poder evaluar el rendimiento de la solución propuesta se han elegido un conjunto de métodos alternativos para la ubicación de los sensores. Para definir

el primero de ellos se ha seguido el trabajo mostrado en [49] donde se analizan dos sistemas: el método de ubicación en carreteras aleatorio o RRLM (siglas del inglés *Random Road Location Method*) y el método de ubicación en carreteras arteriales o ARLM (siglas del inglés *Arterial Road Location Method*). Mientras el primero de ellos ubica de forma completamente aleatoria las cámaras, el segundo elige los puntos según la capacidad de las carreteras. En un entorno urbano, las capacidades de las vías son cercanas entre ellas al no haber presencia de autovías. Teniendo en cuenta esto, se han definido los siguientes experimentos:

- **Valor máximo de referencia:** Todos los datos de la simulación son almacenados realizando el conteo de los vehículos que pasan por un determinado enlace. Teniendo en cuenta el radio de acción máximo de una cámara de tráfico, para los enlaces de longitud mayor a 120 metros, se ha optado por realizar una división de este en subenlaces, cada uno de ellos con longitud igual o inferior a 120 metros. A partir de esta información se genera una lista ordenada de forma decreciente de ocupación de todos los enlaces o subenlaces, posicionando las cámaras en los enlaces más transitados.
- **Ubicaciones aleatorias:** Siguiendo el proceso denominado RRLM, las ubicaciones de las cámaras son elegidas de forma aleatoria. Para cada conjunto de cámaras (entre 1 y 32), se realizan diez procesos independientes de simulación, con ubicaciones aleatorias en cada uno de ellos. Durante cada simulación se realiza el conteo de los vehículos que captan las cámaras y, al finalizar las diez simulaciones, se obtiene su valor medio.
- **Centralidad *betweenness*:** La ubicación de las cámaras se realiza en base a la lista  $L_c$  obtenida tras el cálculo de la centralidad del *pseudo-dual graph* obtenido a partir de la red original. La simulación se repite ubicando entre 1 y 32 cámaras.
- **Sistema propuesto:** La ubicación de las cámaras se realiza en base a la lista  $L_p$  con un factor  $\alpha = 0,5$ . Para la obtención de los datos históricos utilizados para generar la lista  $L_d$ , se realiza una simulación del escenario sin ubicar ninguna cámara. La simulación se repite ubicando entre 1 y 32 cámaras.

Tras llevar a cabo la totalidad de los experimentos, los resultados obtenidos se muestran en la figura 2.2, especificando el conteo de vehículos por parte de las cámaras para cada estrategia.

El objetivo de estos experimentos era validar la solución propuesta para la ubicación de sensores en una red de tráfico basada en el estudio de su centralidad y medir su posible ganancia respecto a otras. Para ellos se ha probado incluir entre 1 y 32 cámaras, simulando el número de vehículos que serían capaces de captar en su

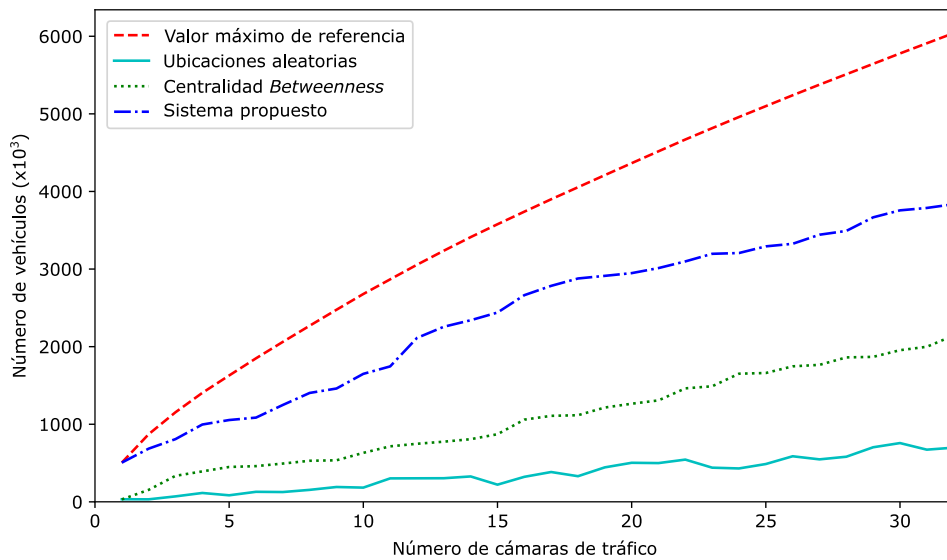


Figura 2.2.: Resultados de los experimentos sobre la ubicación de cámaras en un escenario de tráfico.

rango de visión cada una de ellas si son posicionadas en las ubicaciones obtenidas por cada método de posicionamiento. Según los resultados mostrados en la figura 2.2 podemos ver como los valores obtenidos basados en la medida de centralidad *betweenness* (línea verde punteada) de la red, proporciona mejores resultados que las técnicas basadas en posiciones aleatorias. El sistema propuesto (línea punto-rayado azul) ofrece valores muy próximos a los máximos posibles. En ambos casos cabe destacar la pendiente de las curvas, las cuales siguen una trayectoria similar al valor de referencia, lo que confirma la validez del uso de la medida de centralidad de la red y de sistema propuesto que la utiliza.

Estos resultados, junto con la propuesta realizada han sido publicados en el artículo [50] (secciones 2, 3.2, 5.2 y 6), recogido en el capítulo 3.

## 2.2 Sistemas de gestión de intersecciones

Las intersecciones son aquellos puntos de una red donde se produce el cruce de dos o más caminos y, por tanto los vehículos pueden variar su destino. Las intersecciones constituyen una región crítica debido a la coincidencia espacial de diferentes trayectos [12]. También podemos ver la intersección como un recurso compartido que diferentes vehículos pueden querer utilizar al mismo tiempo, siendo necesario planificar o coordinar su ocupación. La gestión cooperativa de intersección (CIM por sus siglas en inglés, *Cooperative Intersection Management*) [6] persigue evitar efectos negativos en estas regiones críticas tales como colisiones o atascos en alguna

de la entrada a la intersección provocados por la priorización ineficiente del paso de vehículos.

En esta investigación se ha abordado este problema comenzando por proponer dos elementos básico dentro de los CIMs, el diseño de una sistema de comunicación global entre los elementos que forman el escenario de tráfico (sección 2.2.1) y un método para modelar las intersecciones (sección 2.2.2) que pueda ser utilizado posteriormente en los procesos de optimización.

### 2.2.1 Sistemas multiagente en entornos de tráfico urbano

Una posible clasificación de las tecnologías basadas en agentes para entornos de tráfico es la que agrupa los trabajos en dos categorías: modelado y simulación, y gestión y control [51]. Los sistemas multiagentes son ampliamente utilizados para la coordinación del tráfico en intersecciones [6]. Esto se debe a que se tratan de problemas donde la distribución geográfica de los elementos del escenario de tráfico, la dinámica del entorno y la fuerte interacción entre estos elementos, hace idóneas las arquitectura distribuidas basadas en tecnologías de agentes [52].

Debemos empezar por definir los diferentes agentes que intervienen en el sistema propuesto. A continuación, se describirán los tres tipos diseñados para realizar tareas concretas en un escenario de tráfico, donde la gestión de las intersecciones es el centro del problema a optimizar.

#### 1. “TLAgent”: Agente de gestión de semáforos

- **Descripción:** Este tipo de agente es el encargado de realizar los cambios de fase de las luces de los semáforos. Se comunica de forma directa con otros agentes de su mismo tipo para asegurar la correcta sincronización de las base de tiempo que utilizan para realizar los cambios.
- **Ubicación:** En cada semáforo de cada intersección de la red.

#### 2. “TJamAgent”: Agente detector de atascos

- **Descripción:** Este tipo de agentes recibe la información directa de los sensores de tráfico integrados en él o en su entorno cercano. Es el encargado de notificar las variaciones en el estado del tráfico (por ejemplo, aumentos de los flujos de vehículos o situaciones de atasco).
- **Ubicación:** Desplegados a lo largo del escenario de tráfico.

#### 3. “IntersectionAgent”: Agente controlador de intersecciones

- **Descripción:** Este tipo de agentes toma las decisiones sobre posibles cambios de fases en los semáforos que regula una intersección. Se comunica entre otros agentes de su mismo tipo para realizar tareas de coordinación, recibe información de los “TJamAgent” sobre el estado del tráfico de las carreteras de entrada o salida de su intersección, y ordena realizar un cambio de fase a los “TLAgent” de su intersección.
- **Ubicación:** En cada intersección que gestione el sistema multiagente.

Para poder probar y evaluar las posibles propuestas de comportamiento de estos agentes, es necesario contar con el soporte que permita reproducir un escenario de tráfico, donde el sistema multiagente esté integrado. La comunicación entre el sistema multiagente emulado y el simulador de tráfico se aborda en el apéndice A.1.2.

### 2.2.2 Modelado de intersecciones

Tanto para los procesos de optimización como para el funcionamiento autónomo de un agente, es necesario contar con un modelo de intersección que identifique cada uno de sus elementos de forma unívoca. Si este modelo, además, cuenta con información adicional como las trayectorias posibles de un vehículo cuando se encuentra en un punto concreto de la intersección, puede ser utilizado como base para su movimiento autónomo.

A la hora de diseñar un método para el modelado se debe plantear qué sistema puede ser más útil para identificar puntos claves de la intersección y cómo establecer las posibles trayectorias que guiarán a los vehículos a su paso por ella. Si nos fijamos en los métodos utilizados por los simuladores microscópicos, los cuales modelan cada vehículo como una entidad independiente, los denominados autómatas celulares de tráfico (TCA -*Traffic Cellular Automata*- por sus siglas en inglés) son un modelo muy eficiente para simular flujos bajo estas condiciones [53]. Existen diferentes soluciones para realizar modelos de tráfico utilizando TCAs. Respecto a la división del espacio en celdas, es posible realizar esta división utilizando formas triangulares, hexagonales o rectangulares, siendo estas últimas las más utilizadas. En la figura 2.3 se muestra un ejemplo de división del espacio que ocupa la intersección siguiendo un reparto rectangular. El tamaño de cada una de estas celdas se debe hacer teniendo en cuenta, entre otros aspectos, la velocidad de paso de los vehículos, los posibles puntos de conflicto y los radios de giro, tal y como se aborda en el capítulo 7 de [54], en la sección 1 de [55] o en la sección 3 de [56].

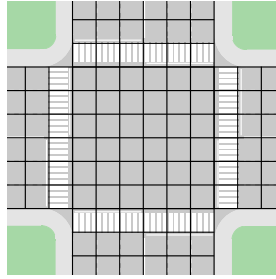


Figura 2.3.: División del espacio de la intersección en celdas rectangulares de igual tamaño.

La intersección mostrada en la figura 2.3 cuenta con cuatro ramas y tres carriles de entrada y de salida por cada rama. La elección de esta intersección para su utilización como caso de uso, la cual se utilizará de forma frecuente a la largo de la investigación como ejemplo sobre el que aplicar los conceptos definidos, se debe a que es un tipo frecuentemente utilizada en el estado del arte [54], [56]-[59].

### 2.2.2.1. Propuesta de modelado de una intersección basado en un TCA

Una vez se ha realizado la división del espacio de la intersección en celdas, se procede a proponer una metodología y nomenclatura para etiquetar cada una de estas celdas. El objetivo de este método es proporcionar un sistema de etiquetado automático de las intersecciones.

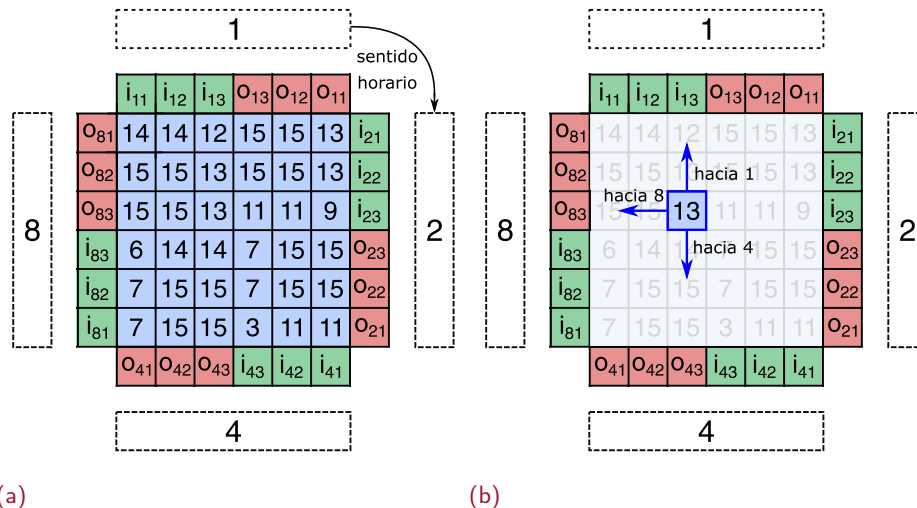


Figura 2.4.: Propuesta de modelado de intersecciones. (a) Etiquetado de cada una de las celdas en las que se divide la intersección. (b) Movimientos posibles de un vehículo que se encuentra en la celda resaltada.

El resultado de la propuesta aplicada sobre el caso de uso se muestra en la figura 2.4a. Para obtener este modelo, los pasos a seguir son los siguientes:

- **Identificación de los puntos de entrada y salida de la intersección:** Cada brazo de la intersección puede ser un posible punto de entrada, punto de salida o punto de entrada y salida de vehículos. Estos puntos son los canalizadores de los flujos de tráfico que atraviesan la intersección y, por tanto, son la base del modelo propuesto. El resto de los elementos de la intersección se etiquetarán en relación a estos puntos.

Se comenzará el etiquetado por el brazo más cercano al norte geográfico de la intersección. Si hubiera dos brazos equidistantes respecto de este punto, el brazo seleccionado será el que se encuentra a su derecha. Los siguientes brazos se etiquetarán siguiendo el sentido horario. En los siguientes pasos se procederá a la asignación de valores a los elementos interiores de la intersección, los cuales identificarán las posibles opciones de giro de los vehículos. Debido a esto, el valor utilizado para etiquetar cada brazo debe permitir agruparse y descomponerse para obtener los posibles giros. Por este motivo se han utilizado valores en exponenciales con base 2 tal y como se muestra en la ecuación 2.9.

$$Ta(j) = 2^j \quad (2.9)$$

donde  $Ta$  es la etiqueta de un brazo y  $j$  es un valor entero creciente entre 0 y  $n$ , donde 0 se asigna al primer brazo y  $n$  es asignado al último brazo. De acuerdo a esto, para el caso de uso tendremos que  $Ta([0, 1, 2, 3]) = [1, 2, 4, 8]$ , tal y como se muestra en la figura 2.4a.

- **Etiquetado de los puntos de entrada y salida:** El modelo propuesto incluye tanto las celdas interiores de la intersección como los puntos de entrada y salida de la misma. Es decir, las celdas utilizadas por los vehículos para entrar o abandonar la intersección. Estas celdas exteriores son etiquetadas de derecha a izquierda según el sentido de la marcha de la vía, utilizando número enteros y empezando por el valor uno. La notación de las etiquetas de estas celdas es:

$$\begin{aligned} \text{inputs} &: i_{ab} \\ \text{outputs} &: o_{ab} \end{aligned} \quad (2.10)$$

donde  $a$  es el valor de un brazo concreto ( $Ta$ ) y  $b$  es el carril donde la celda está ubicada. En la figura 2.4a se muestra este etiquetado utilizando los colores verde y rojo para las celdas de entrada y salida respectivamente.

- **Valor de giro de las celdas interiores:** Cuando un vehículo accede al espacio que ocupa una celda, uno de los parámetros más importantes a tener en cuenta en relación a su comportamiento son los siguiente movimientos permitidos. Es-

ta tarea se facilita si cada celda tiene un valor capaz de indicar esta información tal y como se define en la ecuación 2.11.

$$Tv = \sum_{k \in PossibleTurns(CurrentCell)} Ta(k) \quad (2.11)$$

donde  $PossibleTurns(CurrentCell)$  es el conjunto de giros desde la celda actual hacia cada posible brazo de la intersección. Por ejemplo, observando la figura 2.4a, si un vehículo puede moverse hacia la izquierda, el valor de  $Tv$  contendrá la etiqueta de ese brazo ( $Ta = 8$ ). Si también puede moverse hacia abajo,  $Ta = 4$  será sumado al valor anterior. De esta forma, en la figura 2.4b se muestra el caso de una celda con  $Tv = 13$ . Descomponiendo este valor en potencias de dos tenemos que  $Tv = Ta(0) + Ta(2) + Ta(3) = 1 + 4 + 8 = 13$ . Esto significa que un vehículo situado en esa celda podría continuar su trayecto hacia la rama superior, izquierda o inferior de la intersección. Estas opciones de giro son las que indica la intersección, siendo posible que los vehículos tengan restricciones propias según su posición anterior.

Siguiendo los pasos anteriores, el modelo de intersección se ha completado. Este proceso puede ser llevado a cabo de forma automatizada a partir de la información de forma, situación y reglas de circulación de las intersecciones.

### 2.2.2.2. Modelado de intersecciones con formas irregulares

Aunque el caso de uso utilizado en la sección 2.2.2.1 muestra una intersección con forma rectangular, permitiendo una división exacta del espacio, el método propuesto es aplicable a cualquier otro tipo de intersección.

Las diferentes características que puede tener una intersección como pueden ser su número de brazos, de carriles por brazo, las restricciones de giro y su forma, puede producir que el reparto del área interna de la intersección no sea viable utilizando directamente una cuadrícula homogénea como en el ejemplo visto en la figura 2.4. En el campo de los TCA, además del modelo en el cual una celda tiene el tamaño suficiente para contener a un vehículo y su perímetro de seguridad, existen otros sistemas en los cuales un vehículo ocupa más de una celda. En [56] se estudian modelos de autómatas celulares que reducen el tamaño de celda hasta adaptarse al área que desean simular. En estos modelos un vehículo puede ocupar más de una celda (por ejemplo  $1 \times 2$ ,  $2 \times 6$ , etc.).

La figura 2.5 muestra una intersección real con 5 brazos o ramas. Cada uno de estos brazos tiene un número de carriles de entrada y salida que varían entre uno y tres.



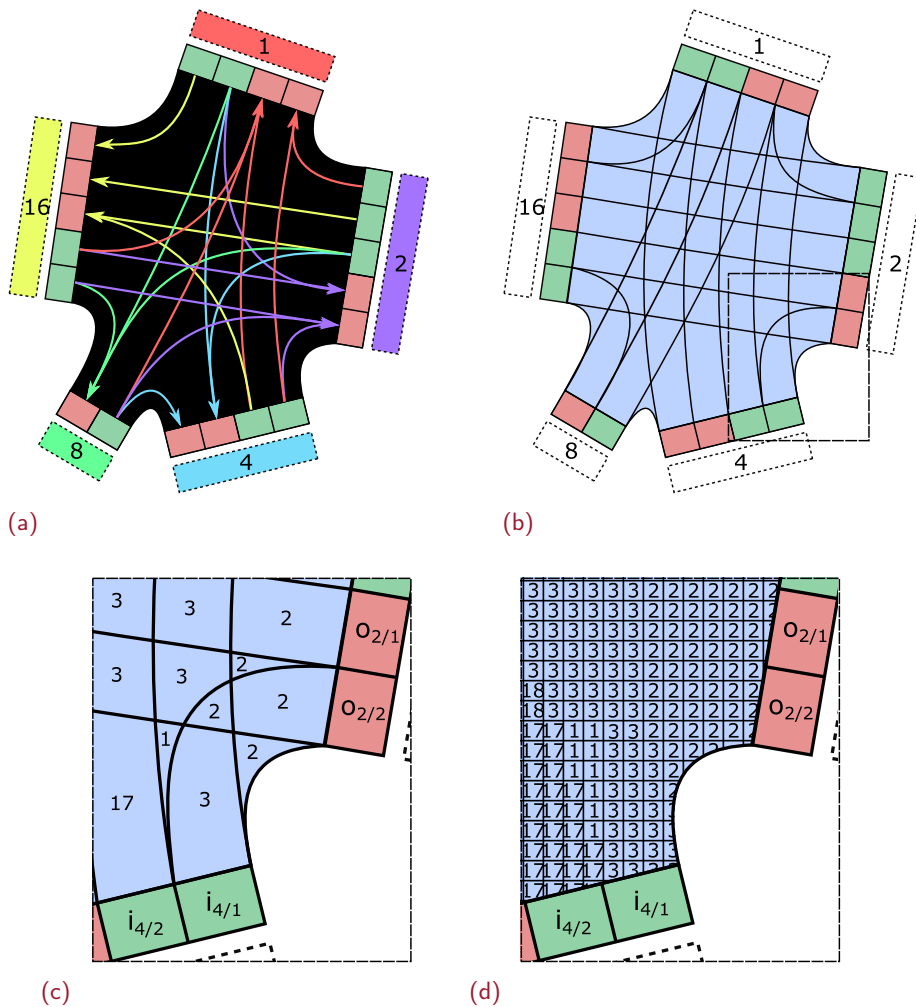


Figura 2.5.: Ejemplo de aplicación del modelado a una intersección de 5 ramas. (a) Representación de las posibles trayectorias entre los puntos de entrada y los puntos de salida. (b) Formación de carriles virtuales en el interior de la intersección a partir de las posibles trayectorias. (c) Etiquetado de cada celda de los carriles virtuales. (d) Generalización del etiquetado a un esquema con celdas de igual tamaño.

Por cada carril de entrada a la intersección hay tantas posibilidades de giro como permita la señalización de esta. A partir de esta información de los giros posibles y de la forma de la intersección, se generan las trayectorias que puede darse entre todos los puntos de entrada y sus posibles puntos de salida (figura 2.5a). El modelado de la intersección sigue los mismos pasos explicados en la sección 2.2.2.1. Se comienza etiquetando cada brazo de la intersección, empezando por el situado más próximo al norte geográfico, siguiendo el sentido horario y utilizando valores potencia de dos. En este caso, para la intersección de cinco ramas, los valores utilizados serán 1, 2, 4, 8 y 16. Dada la forma particular de esta intersección, no es posible realizar una división del espacio siguiendo una cuadrícula de forma directa. Por este motivo es necesario definir un conjunto de carriles virtuales basados en las posibles trayectorias (figura 2.5b). Como se puede observar, en este punto se tiene una división del espacio

realizada por la superposición de los carriles virtuales, que de forma análoga a los visto en la sección 2.2.2.1, pueden ser etiquetados dependiendo de los posibles puntos de destino alcanzables desde cada división.

El tercer paso de este modelado es el etiquetado de cada una de estas divisiones. En la figura 2.5c se muestra un fragmento de la intersección ampliada, donde a cada división se le ha asignado una etiqueta en base a los posibles destinos alcanzables por un vehículo situado sobre ella. Empezando por el punto  $i_{4/1}$ , la primera división tiene el valor 3, lo cual nos indica que un vehículo situado en esta porción de intersección podría continuar recto hacia  $o_{1/x}$  o girar a la derecha hacia  $o_{2/x}$ .

Por último, esta asignación de giros es transferida de forma directa a un modelo TCA cuyo tamaño de celda, mucho menor, permita adaptarse a las formas los carriles virtuales y de la intersección. Para el ejemplo de la figura 2.5d se ha optado por un tamaño de celda de  $1.5 \times 1.5$  metros, lo que supondrá que los vehículos deban ocupar  $2 \times 3$  celdas. En el caso de que una celda del autómata celular ocupe más de una región de la división anterior, se le asignará el valor de la región que coincida en mayor superficie con ella. Si dos regiones ocupan la misma área de la celda, el valor asignado será el que coincida en mayor grado con sus celda vecinas.

Los movimientos de los vehículos siguen la misma dinámica descrita para el modelo genérico de TCA teniendo en cuenta que, en este caso en lugar de ocupar una única celda, ocuparán  $2 \times 3$  celdas.

## 2.3 Coordinación de vehículos a su paso por una intersección

Aunque las intersecciones constituyen una pequeña parte de las redes de carreteras, casi la cuarta parte de los accidentes de tráfico ocurren en ellas [60]. De forma general las intersecciones pueden ser divididas entre las señalizadas y las no señalizadas. Además, según el método utilizado para su gestión, pueden también distinguirse entre sistemas centralizados o distribuidos [6].

Los problemas de coordinación de vehículos en cruces de carreteras pueden ser abordados desde diferentes puntos de vista. El primero de ellos es el basado en generar otro tipo de conexiones entre carreteras, diferentes a las intersecciones, que gestionen las prioridades de paso de forma diferente. El caso más representativo es la sustitución de las intersecciones tradicionales por rotondas [61], las cuales reducen el número de puntos donde pueden producirse potencialmente colisiones. Existen otros tipos de infraestructuras cuyo objetivo similar, reducir los puntos

críticos y repartir de forma ecuaníme el derecho de paso. Los denominados en inglés “Median U-Turn” consisten en intersecciones que prohíben los giros a la izquierda. Cuando un vehículo quiere tomar ese sentido debe girar antes de su llegada a la intersección a la derecha y realizar un cambio de sentido. De esta forma se evitan las trayectorias que más conflicto pueden generar, que son los giros a la izquierda, a cambio de los vehículos que quieran tomar ese sentido deban recorrer mayor distancia [62]. El segundo de los tipos es el basado en estrategias tradicionales de control de flujos utilizando señales de tráfico. Cualquier intersección, incluso en ausencia de señalización, conlleva una regulación de la preferencia de paso. En el caso de la legislación española, salvo excepciones, en una intersección sin señalizar se debe ceder el paso a los vehículos que se aproximen por la derecha. A través de la colocación de señales este derecho de paso puede ser alterado de forma estática (señales de ceda el paso o stop) o de forma dinámica (semáforos). Por último, un nuevo paradigma ha surgido como tercera opción debido a la evolución tecnológica de las últimas décadas: la coordinación de vehículos autónomos. En este escenario los vehículos pueden comunicarse entre ellos y percibir el entorno, haciendo viable el uso de un recurso compartido como es una intersección.

### 2.3.1 Gestión de los puntos de conflicto en una intersección urbana

Tal y como se ha mencionado anteriormente en la sección 2.2.2, cuando dos o más trayectorias (camino entre un punto de entrada y un punto de salida de la intersección) coinciden en el mismo lugar, se genera un posible punto de conflicto. Estos puntos son candidatos a provocar colisiones, que sucederán cuando esa coincidencia espacial sea también temporal. Siguiendo el caso de uso de intersección mostrado en la figura 2.3, podemos ver un posible punto de conflicto entre dos trayectorias en la figura 2.6. Tanto la trayectoria “A” (línea verde) como la “B” (línea azul) realizan el mismo giro a la izquierda desde sus posiciones de inicio. Los giros a la izquierda son los movimientos que más problemas generan en la intersección de las regiones donde el sentido de circulación es por el carril derecho.

Para entender el comportamiento de los vehículos ante diferentes sistemas de gestión de las intersecciones, se plantea un experimento consistente en la simulación de una intersección de ejemplo con rutas que sigan la trayectoria “A” y “B”. Estas trayectorias garantizan una situación donde se deba gestionar al menos un punto de conflicto. Entre los métodos de coordinación abordados en esta investigación se encuentran los basados en prioridades con señalización fija, señalización variable y sistemas de libre circulación donde la cadencia de llegada de vehículos es ajustada para evitar colisiones.

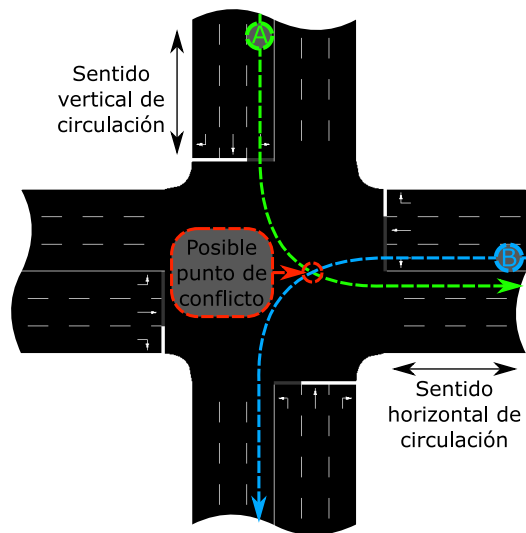


Figura 2.6.: Representación de dos trayectorias y el punto el punto de conflicto que generan.

La base de este experimento es la generación de un escenario de tráfico que reproduce la intersección y la rutas mostradas en la figura 2.6. Variando el sistema de gestión de la intersección, se recogerán la posición de cada vehículo en cada paso de simulación. En este caso, un paso de simulación corresponde con un periodo de un segundo en tiempo real. Los tres sistemas de gestión de la intersección simulados son:

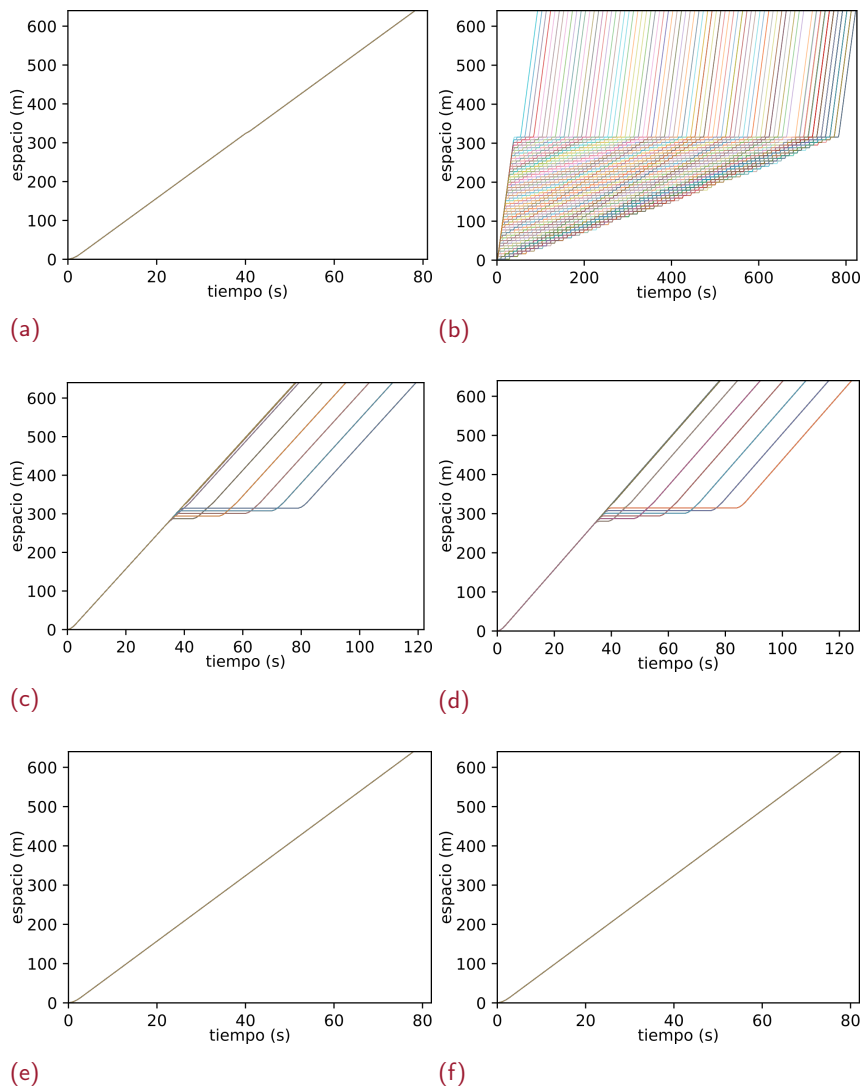
- **Prioridad:** La intersección es gestionada a través del uso de señales de ceda el paso. Estas señales se han colocado en los carriles de entrada a la intersección en el sentido horizontal de la marcha.
- **Semáforos:** El paso de vehículos a la intersección es controlado por semáforos colocados sobre los carriles de acceso a la intersección. Estos semáforos actúan de forma conjunta para cada rama en ciclos de 33 segundos. Esta configuración es la que ha generado por defecto SUMO teniendo en cuenta las características de la intersección.
- **Libre circulación:** Los vehículos son insertados en la simulación fijando la velocidad a la que deben llegar a la intersección y siguiendo ciclos alternos entre los que siguen el trayecto “A” y el “B”. La simulación se considera correcta si no se ha producido ninguna colisión al finalizar la misma.

Para todos los experimentos, el tiempo real de simulación es de 3600 segundos y el número de vehículos insertados para cada trayecto es de 360 (720 en total). Al mantener la misma tasa de llegada de vehículos en cada sentido, los resultados mostrarán el caso más extremo posible.

En la figura 2.7 aparecen los diagramas espacio/tiempo de los trayectos que han realizado todos los vehículos durante cada simulación, separados por el tipo de gestión

utilizada en cada caso (prioridad, semáforos o propuesta) y según la trayectoria que han seguido (origen en “A” o en “B”). Debido a que todas las rutas son iguales, para facilitar la comparación entre los desplazamientos realizados sin tener en cuenta el paso de simulación en el que hayan comenzado la marcha, el instante temporal cero representa el inicio de la ruta.

El planteamiento de estos experimentos y sus resultados están recogido en la publicación [63].



**Figura 2.7.:** Diagramas espacio/tiempo de los desplazamientos realizados durante el tiempo de simulación según origen y sistema de gestión. (a) Trayecto “A”: Gestión por cedas el paso. (b) Trayecto “B”: Gestión por cedas el paso. (c) Trayecto “A”: Gestión por semáforos. (d) Trayecto “B”: Gestión por semáforos. (e) Trayecto “A”: Gestión por ajuste de los instantes de llegada. (f) Trayecto “B”: Gestión por ajuste de los instantes de llegada.

Analizando los resultados obtenidos, en el caso de la gestión de la intersección con señales de ceda el paso, el trayecto de los vehículos con origen en “A” es prioritario

respecto a los que parten del punto “B”. En la figura 2.7a se aprecia como todos los vehículos siguen la misma curva en el diagrama espacio/tiempo, realizando el trayecto completo en 81 segundos sin detenerse. Por el contrario, los vehículos que tratan de acceder a la intersección por el sentido perpendicular de la circulación deben ceder el paso. Desde el inicio de la simulación comienzan a acumular retrasos, los cuales se van incrementando por la formación de colas al llegar nuevos vehículos. Para los trayectos con origen en el punto “B” (figura 2.7b), el tiempo medio del viaje es de 613.54 segundos, es decir, más de un 750 % superior al tiempo que tardan los vehículos que acceden a la intersección en el otro sentido. Teniendo en cuenta este incremento podría considerarse que, de forma práctica, una cadencia constante de vehículos en el sentido vertical de circulación crea inanición en el sentido horizontal.

Respecto a los resultados de la segunda simulación (intersección gestionada por semáforos), presentan equidad en el reparto del uso de la intersección en ambos sentidos de la marcha. En las figuras 2.7c y 2.7d se puede observar como los vehículos sólo se detienen en el tramo más cercano a la intersección (25-40 metros previos), reanudando la marcha en el siguiente ciclo en verde del semáforo que le corresponda. No se detectan colas de vehículos que requieran más de un ciclo para atravesar la intersección.

Por último, ajustando la velocidad y la cadencia de llegada de vehículos a la intersección para evitar colisiones en los puntos de conflicto, en las figuras 2.7e y 2.7f se muestra como este sistema permite el paso de vehículos en ambos sentidos de la marcha, sin que se produzcan reducciones de velocidad a su paso por la intersección. Los valores numéricos son los mejores obtenidos de las tres simulaciones, siendo los más próximos al valor ideal.

Este conjunto de experimentos permite establecer los problemas de los diferentes tipos de sistemas de gestión en intersecciones.

### 2.3.2 Intersecciones señalizadas: gestión a través de semáforos

Como se ha podido comprobar con los resultados de los experimentos mostrados en las figuras 2.7c y 2.7d, el uso de semáforos permite el reparto ecuánime del uso de la intersección. Cuando los flujos de tráfico no son constantes y homogéneos, la programación estática de los semáforos puede resultar ineficaz. En las siguientes secciones se mostrará la propuesta realizada en este contexto y su aplicación a un escenario de tráfico realista con una complejidad elevada.

### 2.3.2.1. Sistema multiagente para la gestión inteligente de semáforos

A partir de la arquitectura definida en la sección 2.2.1, se ha propuesto un comportamiento específico de los agentes que componen el sistema con el doble objetivo de reducir la duración de los trayectos y minimizar el número de mensajes utilizado en el propio sistema multiagente.

Comenzaremos por definir el comportamiento de cada agente que interviene en el sistema:

- **“TLAgent” (Agente de gestión de semáforos):** En su estado inicial, el agente realiza los cambios en las fases de los semáforos (rojo, amarillo, verde) siguiendo patrones estáticos predefinidos. Cuando uno de estos agentes recibe un mensaje solicitándole un cambio de los ciclos de alguna de las fases, el agente verifica si el nuevo valor está dentro del intervalo máximo y mínimo permitido, en caso afirmativo actualiza sus registros y aplica los nuevos tiempos a partir del siguiente cambio de fase.
- **“TJamAgent” (Agente detector de atascos):** Estos agentes están repartidos a lo largo del escenario de tráfico. Pueden ubicarse en vehículos con dispositivos inteligentes con capacidad para registrar su velocidad actual y posición. También pueden ser otro tipo de sensores con ubicación fija en la red tráfico tales como cámaras, detectores inductivos, etc. Cada agente cuenta con un conjunto de reglas específico para determinar si, en el área de influencia de su sensor, se está produciendo un atasco. En el caso de los vehículos, este umbral se puede determinar por ir a velocidad inferior a la de la vía por la que circula. En otro tipo de sensores, estos umbrales pueden determinarse por las mediciones de sus valores históricos. Cuando un agente detecta una situación de congestión, envía un mensaje a los agentes de tipo “IntersectionAgent” de su área.
- **“IntersectionAgent” (Agente controlador de intersecciones):** Situado en el entorno de la intersección a la gestionan, pueden enviar mensajes de consulta a los “TLAgent” de su intersección para conocer las fases que se están aplicando en cada momento. Con esta información y los valores de posibles atascos recibida desde los “TJamAgent” de su área, realizan el proceso de toma de decisiones para generar las posibles nuevas planificaciones de las fases de la intersección. Tras finalizar el cálculo de las nuevas fases, comunica a cada “TLAgent” los cambios que debe realizar. Además de este flujo de comunicación interna entre los agentes de una misma intersección, los “IntersectionAgent” adyacentes colaboran de la siguiente forma:

- **Modo colaborativo:** Los agentes controladores de cada intersección aceptan sugerencias de limitar o favorecer sus flujos de paso de los “IntersectionAgent” de intersecciones cercanas. Estas sugerencias se combinan con los datos sobre puntos de congestión que recibe desde los “TJamAgent” para realizar posibles cambios en la planificación de las fases de sus semáforos.
- **Modo aislado:** Cuando en la zona que gestiona uno de estos agentes se producen situaciones de congestión graves, es decir, que las medidas que puede tomar el agente no consiguen mejorar las esperas de los vehículos, el agente controlador pasaba a un estado aislado. En este modo de trabajo omite cualquier tipo de solicitud de colaboración de otros agentes de su mismo tipo. Además, centra sus esfuerzos de planificación de los flujos en determinar que “IntersectionAgent” adyacentes podrían contribuir a una mejora en la situación de su zona, iniciando en envío de mensajes de solicitud de colaboración a ellos. La planificación de las fases de sus semáforos responderá únicamente a los valores recibidos por los “TJamAgent” de su zona y a las posibles aceptaciones de colaboración que reciba. Un “IntersectionAgent” puede trabajar en este modo aislado un tiempo limitado.

### 2.3.2.2. Caso de uso del sistema multiagente propuesto

En esta sección se muestra un caso de uso sobre el que aplicar el comportamiento de los agentes descritos en la sección 2.3.2.1. Se han definido dos zonas de tráfico alrededor de las dos intersecciones a estudio. Para simplificar la explicación, los vehículos no realizan cambios de sentido, por lo que las dos intersecciones comparten flujos de tráfico en la carretera horizontal de la figura 2.8.

La evolución de la llegada de vehículos, tanto en las carreteras horizontales como verticales, hace que el comportamiento de los agentes varíe entre las siguientes tres situaciones:

1. **Estado inicial** (figura 2.8a): Ambos agentes controladores inician su funcionamiento en modo colaborativo, es decir, aceptando sugerencias externas para variar su comportamiento.

Los vehículos que circulan en sentido horizontal, de derecha a izquierda, comienzan a reducir su velocidad en la zona “A” debido al aumento de la intensidad del tráfico. Los “TJamAgent” de alguno de estos vehículos notifican de



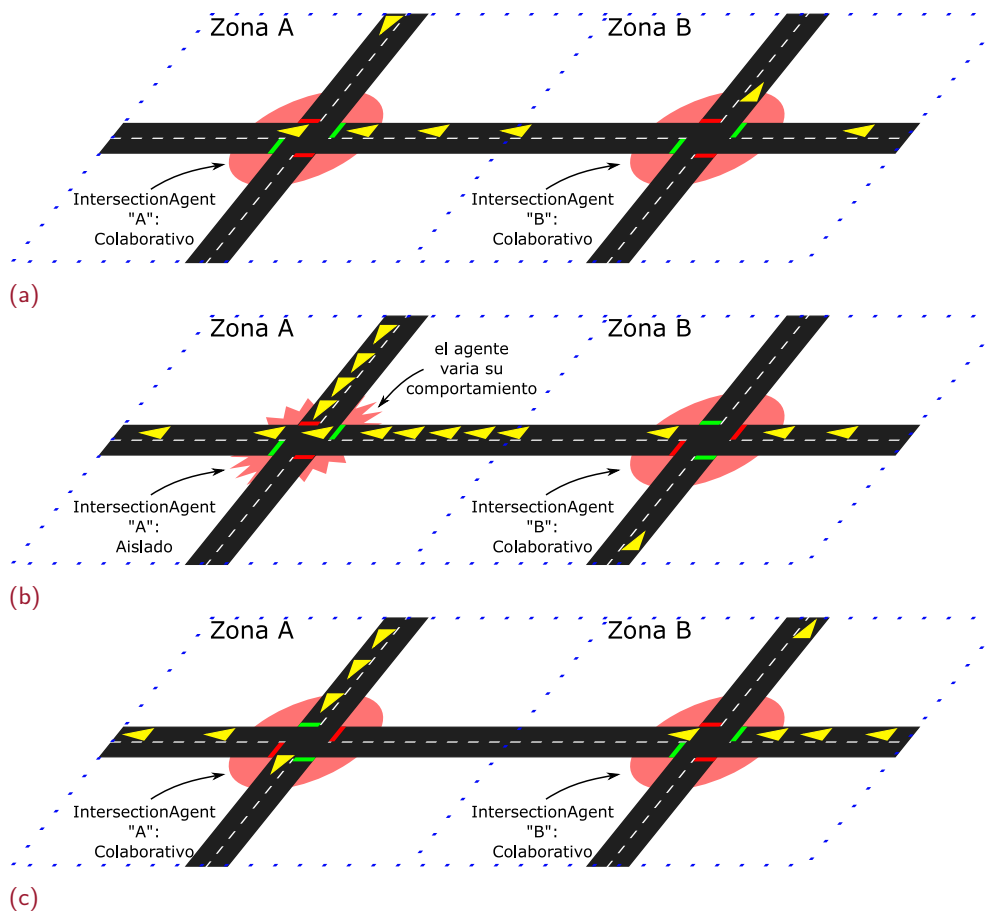


Figura 2.8.: Caso de uso del sistema multiagente propuesto para la gestión inteligente de semáforos. (a) Sin congestión, ambos “IntersectionAgents” trabajan de forma colaborativa. (b) Situación de congestión en la Zona A que su agente no es capaz de mitigar, cambio a modo aislado. (c) Tras solucionar el problema de congestión en sentido horizontal, “IntersectionAgents A” vuelve a modo colaborativo.

esta situación al “IntersectionAgent A”. Este agente prioriza en su planificación el tráfico en ese sentido, proponiendo nuevas fases a los “TLAgents” de su intersección para aumentar el tiempo de fase en verde en el sentido horizontal de la marcha.

2. **Congestión en zona “A”** (figura 2.8b): el agente controlador de la zona “A” comienza a recibir mensajes de “TJamAgents” situado en ambas direcciones de los flujos. El agente determina con la información recibida que no es capaz por sí sólo de desbloquear la situación y pasa a trabajar en modo aislado. Envía un mensaje de solicitud de colaboración al “IntersectionAgent” de la zona “B” para que limite el flujo de vehículos en sentido horizontal. Al mismo tiempo genera una nueva planificación de las fases de sus semáforos para repartir el tiempo de paso según el grado de congestión de los dos sentidos de la marcha.

3. **Flujo normal zona “A”** (figura 2.8c): Tras limitar el tráfico en sentido horizontal el agente de la zona “B”, el de la zona “A” ha conseguido descongestionar ese sentido. Una vez que los indicadores de atascos que recibe “IntersectionAgent A” le confirman la mejoría, vuelve a un modo de funcionamiento colaborativo.

### 2.3.2.3. Experimentos realizados para validar el sistema multiagente propuesto

Los experimentos llevados a cabo se han realizado sobre el escenario de tráfico descrito en el apéndice B.1.1. Además, se ha implementado un sistema multiagente emulado programado en Python el cual se comunica con el simulador de tráfico SUMO con el proceso que se explica en el apéndice A.1.2.

Para evaluar los resultados obtenidos con la propuesta, siguiendo las referencias existentes en la bibliografía, se ha optado por simular los siguientes tres posibles sistemas de gestión de los semáforos:

- **Semáforos con fases estáticas:** El escenario de tráfico de TAPASCologne incluye una programación fija de los semáforos de su red. La programación de estas fases no varía durante el tiempo de simulación, por tanto no es necesario el despliegue de sensores por la red.
- **Actuated traffic lights:** Para este de gestión se debe colocar un detector en cada carril de entrada a una intersección, a una distancia concreta que permita el cambio de fase a tiempo. Cada vez que de uno de estos sensores detecta el paso de un vehículo, envía un mensaje al semáforo correspondiente a ese sentido y carril para que favorezca su paso.
- **Sistema multiagente propuesto:** Se ha integrado el sistema multiagente emulado con SUMO para poder recopilar datos de tráfico y variar las fases de los semáforos en tiempo de simulación. Se han utilizado dos tipos de sensores para recoger información. El primero de ellos son cámaras de tráfico ubicadas siguiendo la metodología propuesta en la sección 2.1.2. Además, se ha considerado que todos los vehículos de la simulación cuentan con un dispositivo inteligente que integra un “TJamAgent” y un sensor GPS. En cada paso de simulación se comprueba las velocidades actuales de los vehículos y cuando esta es menor al 50 % de la permitida por la vía, los “TJamAgent” generan un mensaje de posible congestión. En el caso de las cámaras, el aviso se produce cuando la ocupación de las carreteras que vigilan supera el 60 %. Con esta

información, los “IntersectionAgent” siguen el comportamiento explicado en la sección 2.3.2.1.

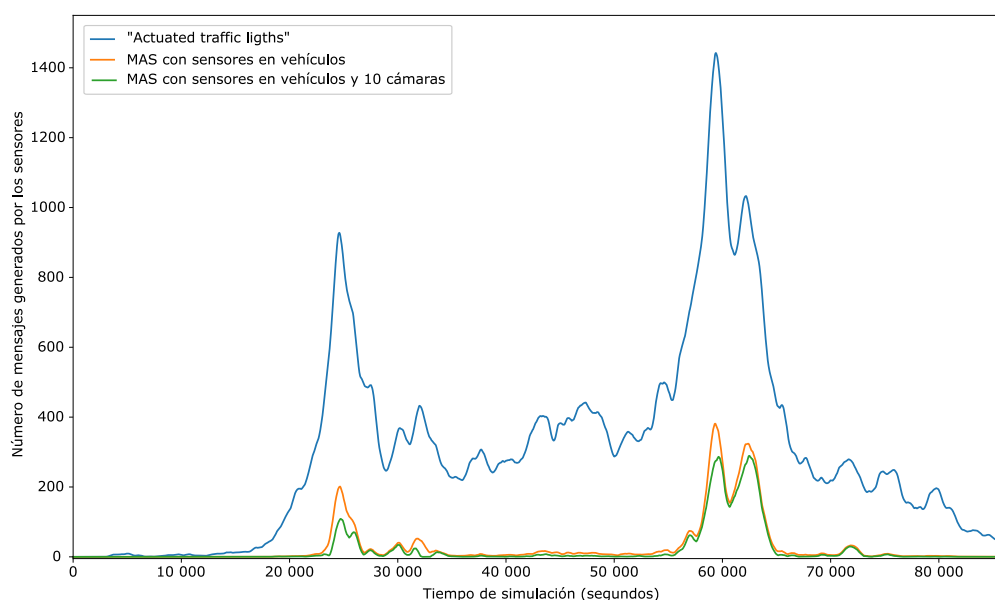
Durante la ejecución de las tres simulaciones se han recopilado la información relativa a la duración del desplazamiento de los vehículos y el número de mensajes generados por los diferentes sistemas gestores.

Duración de los viajes	Menor	Igual	Mayor
“Actuated traffic lighths”	58,70	28,41	12,89
Sistema multiagente propuesto	60,52	27,08	12,41

**Tabla 2.2.:** Comparación de los resultados obtenidos por los sistemas de gestión “Actuated traffic lighths” y el sistema multiagente propuesto frente a los semáforos con fases estáticas (porcentaje total de vehículos cuyas rutas duran menos, igual o más tiempo).

La tabla 2.2 muestra los resultados obtenidos en relación a la duración de los trayectos realizados durante la simulación. Como se puede observar, los dos sistemas que utilizan una adaptación dinámica de las fases de los semáforos consiguen reducir en casi un 60 % los tiempos de desplazamiento.

En relación a los mensajes generados por cada sistema, la figura 2.9 muestra la evolución del número de mensajes generados en cada instante de simulación (0 a 84 000 segundos) para cada sistema de gestión de semáforos.



**Figura 2.9.:** Número de mensajes generados por cada sistema de gestión dinámica de las fases de los semáforos durante el tiempo de simulación.

En las curvas que aparecen en la figura se pueden diferenciar claramente los tramos horarios en los que se producen mayor concentración de vehículos (sobre los

instantes de simulación 25 000 y 60 000, que corresponde a las 7 de la mañana y las cuatro y media de la tarde). Como se puede apreciar, el sistema basado en “Actuated traffic lights” genera mensajes durante todo el periodo de simulación, con independencia de que haya situaciones de congestión o no. Sin embargo, el sistema multiagente concentra el envío de mensaje únicamente en las horas cercanas a las de máxima afluencia de vehículos, que son los periodos donde se produce congestión. Si además añadimos cámaras de tráfico situadas en los puntos estratégicos que hemos propuesto anteriormente, suponiendo que los agentes que se encuentran en los vehículos dejan de emitir mensajes de aviso cuando entran una zona con cobertura de estos sensores, podemos ver como el número de mensajes necesario se reducen aún más.

A partir de estos resultados podemos concluir la validez del sistema multiagente propuesto, tanto en términos de mejora de rendimiento, como de reducción del número de mensajes necesario para llevar a cabo la toma de decisiones. Esta reducción es importante ya que repercute directamente tener una red de comunicación menos saturada y, además en disminuir el coste energético que conlleva el envío de esos mensajes extra.

### 2.3.3 Intersecciones sin señalizar: circulación libre de vehículos

En los últimos años, las mejoras tecnológicas en el campo de los vehículos autónomos y conectados (CAVs por sus siglas en inglés, *Connected and Autonomous Vehicles*) han creado nuevas oportunidades para diseñar y desarrollar sistemas, a través de este nuevo paradigma, para resolver problemas comunes en escenarios de tráfico de vehículos.

El grado de autonomía en la conducción de un vehículo puede medirse por niveles utilizando la escala definida por SAE (del inglés, *Society of Automotive Engineers*) [64]. Esta escala va desde vehículos a motor sin ningún tipo de ayuda (nivel 0) hasta la automatización completa de la conducción (nivel 5). Teniendo en cuenta el precio de la tecnología necesaria para aumentar el grado de automatización de un vehículo, en [65] predicen que en el año 2030 el 98 % de los vehículos tendrán algún tipo de conectividad.

Esta previsión de capacidad de comunicación de los vehículos durante los próximos años, abre nuevos caminos para la resolución de problemas de coordinación de vehículos clásicos. Uno de estos problemas recurrentes son los giros a la izquierda en las intersecciones ya que reducen de forma significativa la capacidad de la intersección, tal y como hemos visto en la sección 2.3.1. Algunas propuestas como

la realizada por Xuan *et al.* en [66] consisten en agrupar a los vehículos por lotes dependiendo del giro que vayan a realizar en la siguiente intersección de su ruta. Para ello definen un área previa a la intersección donde los vehículos deben reordenarse según la planificación del sistema. De esta forma, al llegar a una intersección gestionada por semáforos, las fases de estos pueden estar adaptadas a los lotes de vehículos que llegan.

Desde el punto de vista de las intersecciones sin señalización en sistemas de gestión distribuidos, una de las técnicas que aplica muchas de las ventajas de los vehículos conectados es la denominada semáforos virtuales o “*virtual traffic lights*” (VTL) [32]. En estos sistemas, los vehículos siguen el mismo principio de funcionamiento de los semáforos tradicionales, pero el lugar de reaccionar a un señal luminosa, los vehículos deciden de forma colaborativa las prioridades a través de las comunicaciones que realizan entre ellos.

Varios aspectos de la conducción cooperativa en intersecciones son abordados en la tesis doctoral de Azimi [27]. Especialmente interesante es el capítulo dedicado a la sincronización de vehículos en intersecciones donde define lo que denomina “*Ballroom Intersection Protocol*” (BRIP) con el objetivo de gestionar de forma segura y eficiente el paso de vehículos autónomos a través de intersecciones. Esta técnica utilizan las ventajas ofrecidas por las comunicaciones V2X y los sistemas de geoposicionamiento para crear un conjunto de patrones de vehículos de llegada sincronizados o SIAP (del inglés, “*Synchronized Intersection Arrival Patterns*”). Este trabajo es la base de algunas de las propuestas que se realizarán en las secciones siguientes.

Un primer paso imprescindible para la coordinación de vehículos en las intersecciones es la obtención de los posibles caminos que estos vehículos pueden seguir en su interior. En la sección 2.2.2 se mostró la propuesta de modelado automático de cualquier tipo de intersección. A partir de ese modelo, en los siguientes apartados se explicará el proceso de conversión de ese modelo a un grafo dirigido que pueda ser utilizado para calcular los caminos posibles. También se verán las propuestas relacionadas con la obtención de caminos que minimice los puntos de conflicto de sus trayectorias.

### **2.3.3.1. Conversión del modelo de intersección a un grafo dirigido**

Para llevar a cabo el cálculo de posibles caminos, debemos comenzar por tener un grafo que represente tanto los puntos de entrada y de salida de la intersección, como

los posibles puntos intermedios de paso. El proceso de conversión realizado a partir de modelo basado en un autómata celular propuesto tiene los siguientes pasos:

1. Cada celda del modelo original es transformada a un nodo del nuevo grafo.
2. Los valores de cada celda del modelo son recuperados para obtener movimientos permitidas en cada caso. Para ellos se debe:
  - Descomponer el valor en potencias de dos.
  - Para cada dirección obtenida de la descomposición del valor de la celda, se genera un nuevo enlace dirigido desde el nodo actual hasta el nodo que represente la celda adyacente en la dirección del valor de giro.

Al finalizar el proceso, tendremos un grafo dirigido formado por tantos nodos como celdas tenga el modelo de intersección original y, tantos enlaces dirigidos como posibilidades de giro o movimientos haya en el modelo. El grafo correspondiente al caso de uso utilizado con anterioridad (figura 2.3) puede verse en la figura 2.10.

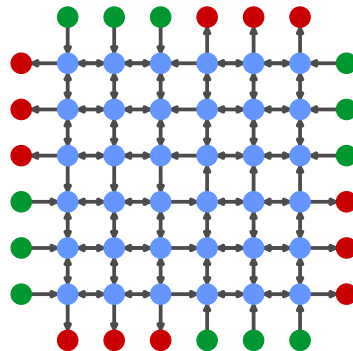


Figura 2.10.: Grafo dirigido obtenido a partir de la conversión del modelo de intersección de 4 brazos y 3 carriles por sentido y brazo visto en la figura 2.4.

### 2.3.3.2. Cálculo de caminos con mínimo puntos de conflicto

A partir del grafo definido en la sección anterior sería posible calcular todos los posibles caminos que existen entre cada nodo de entrada a la intersección (identificados en el modelo original) y todos los posibles nodos de salida. Aunque este proceso se utilizará en alguna de las propuestas realizadas, generar un conjunto reducido de caminos que garanticen la conectividad entre todas las entradas y salidas de la intersección, pero reduciendo al mínimo los puntos de conflicto posibles, puede facilitar posteriormente la convergencia de los algoritmos de optimización propuestos.

Es habitual que en una intersección se deba poder tomar desde un punto de entrada cualquier punto de salida, a excepción de volver por la misma rama por la que

se ha accedido a ella. También pueden existir restricciones de giro fijadas por la regulación específica del tráfico en la intersección. Aunque debido a la propia naturaleza de los cruces de carreteras es inviable conseguir caminos para realizar todas estas conexiones sin que haya zonas de conflicto, utilizando el siguiente algoritmo propuesto se conseguirá un conjunto de caminos que garanticen el menor número de puntos de conflicto posible.

Se comenzará por la definición de los elementos utilizados durante el proceso. El conjunto de los posibles brazos o ramas de la intersección ( $D$ ) se muestra en la ecuación 2.12.

$$D = \{Ta(j) : j \text{ es un entero; y } 0 \leq j \leq n\}, \quad (2.12)$$

donde  $j$  identificad a cada rama, variando entre 0 (primera rama) y  $n$  (última rama), siempre siguiente el sentido horario.

El conjunto de carriles de entrada y salida en un brazo concreto de la intersección  $Ta(j)$  se denota con  $L_{Ta(j)_{in}}$  y  $L_{Ta(j)_{out}}$  respectivamente (ecuaciones 2.13 y 2.14).

$$L_{Ta(j)_{in}} = \{i_{Ta(j)b} : b \text{ es un entero; y } 0 \leq b \leq m\}, \quad (2.13)$$

donde  $b$  es un valor entero entre 1 (carril más a la derecha del brazo) y  $m$  (carril más a la izquierda del brazo).

$$L_{Ta(j)_{out}} = \{o_{Ta(j)b} : b \text{ es un entero; y } 0 \leq b \leq p\}, \quad (2.14)$$

donde  $b$  es un valor entero entre 1 (carril más a la derecha del brazo) y  $p$  (carril más a la izquierda del brazo).

Tras definir los conjuntos de datos elementos que se van a utilizar, los pasos que componen el algoritmo de búsqueda de caminos con menos puntos de conflicto son los siguientes:

1. Se selecciona un elemento  $Ta(x) \in D$ . Este elemento es fijado como punto de entrada inicial de la intersección. El conjunto  $I$  está formado por un único elemento tal que:

$$I = \{Ta(x)\} \quad (2.15)$$

2.  $O$  es el conjunto de posibles brazos de salida, formado como la diferencia entre todas las posibles ramas de la intersección ( $D$ ) y el brazo elegido como brazo de entrada ( $I$ ):

$$O = D - I \quad (2.16)$$

3. Definimos  $r$  como el tamaño del conjunto  $L_{Ta(x)_{in}}$ . Se calculan todas las permutaciones de tamaño  $r$  del conjunto de posibles brazos de salida ( $O$ ). Estas permutaciones se donatan por el conjunto  $\mathfrak{S}(O)$  el cual está compuesto por todas las tuplas de longitud  $r$  que contiene las posibles combinaciones de los elementos sin repetición.
4. Se calcula el producto cartesiano de cada conjunto de carriles en cada brazo de salida ( $L_{Ta(y)_{out}}$ , donde  $y \in O$ )

$$L_{product} = \prod_{y \in O} L_{Ta(y)_{out}}$$

5. El producto cartesiano de cada conjunto  $\mathfrak{S}(O)$  y  $L_{product}$  es obtenido y proporciona una lista ordenada de todas las posibles combinaciones de los brazos de salida y carril del brazo de entrada:  $C = \mathfrak{S}(O) \times L_{product}$ .
6. Cada elemento  $c \in C$  está formado por la combinación de un brazo de salida  $Ta(y)$  y un carril de entrada concreto de un  $Ta(y)$ . Para esta pareja de valores se calculan todos los posibles caminos más cortos entre un punto fijo de entrada  $Ta(x)$  (nodo origen) y cada  $c$  (nodo destino), obteniendo un nuevo conjunto  $P_c$ . Debido a que el tamaño de una tupla en  $\mathfrak{S}(O)$  es igual al tamaño de  $L_{Ta(x)_{in}}$ , la elección de un carril de entrada concreto en  $L_{Ta(x)_{in}}$  como nodo origen se determina por la posición que ocupa en el conjunto (por ejemplo, si  $L_{Ta(x)_{in}} = \{1, 2, 3\}$  y  $O = \{2, 4, 8\}$ , para un elemento concreto  $(4, 8, 2) \in \mathfrak{S}(O)$ , los carriles de entrada 1, 2 y 3 tienen como brazos de destino las ramas 4, 8 y 2 respectivamente).
7. Para cada conjunto  $P_c$  se rellena una matriz con los posibles caminos calculados, obteniendo los puntos de conflicto en las posiciones de la matriz que se encuentre más de un valor. El número de posibles puntos de conflictos es utilizado como el coste de esa combinación. La tupla formada por  $P_c$  y este coste es añadida a la lista *paths*.
8. La lista *paths* es ordenada de menor a mayor valor de coste. Los elementos que se sitúan en la primera posición de la lista es aquel conjunto de posibles caminos que genera menos puntos de conflicto posible.
9. Si la topología de intersección no es simétrica (presenta diferentes combinaciones entre las entradas y las salidas dependiendo de la rama elegida), los pasos anteriores deben repetirse para el resto de elemento de  $D$ .



Este algoritmo y sus correspondientes pruebas de validación están incluidos en el artículo [67] que está recogido en el capítulo 4.

### 2.3.4 Optimización para maximizar la ocupación de la intersección

Uno de los principales objetivos de esta tesis, definido en la sección anterior, es el de mejorar los mecanismos de coordinación de vehículos en intersecciones. El método expuesto en la sección 2.3.3 basado en la creación de patrones de vehículos de llegada sincronizados que utiliza el protocolo BRIP es muy interesante, pero el trabajo de Azimi [27] tiene una limitación muy importante, los patrones los obtiene a través de un estudio manual de las intersecciones. Este hecho limita enormemente la escalabilidad de su método, no pudiendo garantizar una solución válida para cualquier forma de intersección.

Los Algoritmos Genéticos (GA) se utilizan para resolver problemas complejos en áreas como la ingeniería. Suelen llegar a soluciones válidas cuando el espacio de búsqueda es extenso. Nuestra propuesta consiste en diseñar un algoritmo genético capaz de obtener los patrones de llegada óptimos de forma automatizada. Como base de la codificación del problema se utilizará el sistema de modelado automático de intersecciones visto en la sección 2.2.2. Esta combinación permitirá realizar el proceso completo de obtención de patrones sin necesidad de intervención externa.

El objetivo buscado en este caso será el mismo que en el del trabajo de Azimi para tener unos resultados de referencia para valorar la propuesta realizada. Este objetivo es obtener la máxima ocupación posible de la intersección. Además, se utilizarán las mismas condiciones iniciales de su estudio, donde utiliza tasas de llegada de vehículos iguales por cada rama de la intersección y los lugares de entrada a la intersección determinan el camino que tomarán los vehículos.

Los patrones de llegada a una celda  $i_{ab}$  determinan la cadencia de llegada de los vehículos. Es decir, los patrones definen los intervalos temporales de llegada a cada punto de entrada de una intersección. La longitud del patrón es una variable del problema que puede ir desde dos celdas virtuales hasta infinitas celdas. Dependiendo de las características de la intersección es posible que el sistema alcance un estado estable para una longitud de patrón determinada. Este estado se produce cuando un mismo valor de ocupación se repite para valores de patrones que sean múltiplos entre sí.

Por último, es necesario contar con un simulador de tráfico para intersecciones que compruebe los patrones candidatos para la intersección de estudio. El desarrollo de

esta herramienta se muestra en el apéndice A.2 y métricas o valores que recogerá este simulador vienen marcados por la función de evaluación utilizada por el GA y expuesta en la sección 2.3.4.2.

#### 2.3.4.1. Codificación del problema

Teniendo en cuenta las restricciones del problema, el modelo de intersección realizado (figura 2.11a) y el funcionamiento de las funciones de cruce y mutación que se implementarán posteriormente, se han definido el proceso de optimización denominado “OCP” (del inglés, *Optimized Crossing Patterns*), basado en un GA. Para la codificación de cada individuo de este algoritmo se han tenido en cuenta las siguientes consideraciones:

- El patrón será una secuencia de bits (valor binario 0 o 1).
- El valor 0 representa la ausencia de un vehículo en una celda virtual concreta, al igual que el valor 1 representa la presencia de un vehículo. Por ejemplo, un patrón con la secuencia “1 0 1” se corresponde con una cadencia de llegada a un punto concreto de la intersección de “vehículo - hueco - vehículo”.
- Las longitudes de los patrones para cada  $i_{ab}$  de una intersección son iguales.
- Durante el periodo de simulación de cada individuo, el patrón es extendido tantas veces como sea necesario hasta completar todo el periodo de simulación.

Un esquema de la codificación del problema propuesta se muestra en la figura 2.11b. Una intersección se compone de  $n$  brazos. A su vez, cada carril es numerado de 1 a  $m$  de derecha a izquierda en el sentido de la marcha. Por último, cada carril tiene el mismo número de celdas virtuales, obteniendo por tanto patrones de longitud fija. Siguiendo el orden de definición de cada elemento, estas cadenas de bits se colocan de forma secuencial hasta formar el individuo, tal y como se muestra en la parte inferior de la figura 2.11b. Si la longitud de un patrón es  $l_p$ , el tamaño de la secuencia codificada será  $l_{ind} = (l_p \cdot m) \cdot (n + 1)$  donde cada 0 o 1 de la secuencia representa la presencia o ausencia de un vehículo en una determinado posición de un carril virtual de un carril y brazo concreto.

#### 2.3.4.2. Función de evaluación

La función de evaluación diseñada debe tener en cuenta el objetivo de maximizar el uso de la intersección. Debido a las peculiaridades de los escenarios de tráfico

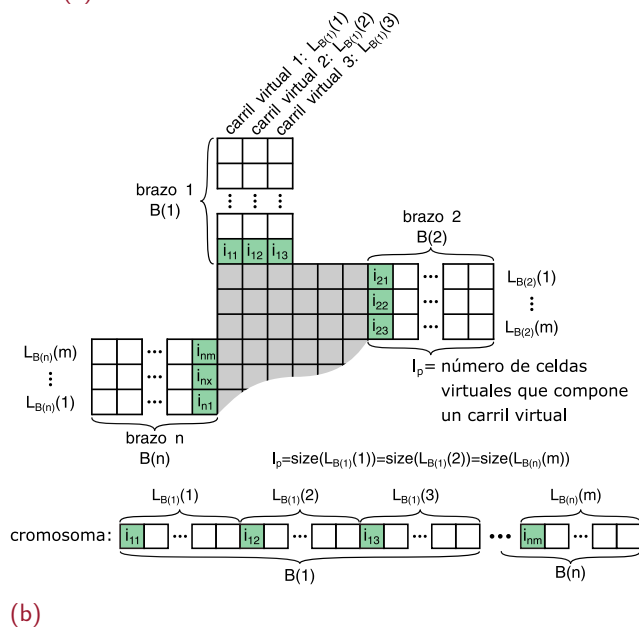
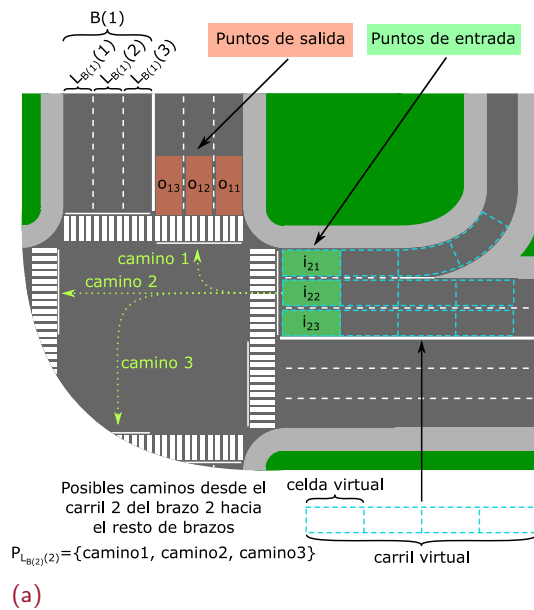


Figura 2.11.: Codificación del problema a través de la definición de carriles virtuales. (a) Elementos de la intersección utilizados para la codificación. (b) Formación del cromosoma utilizado en el Algoritmo Genético.

de vehículos, la ocupación no puede ser medida aplicando de forma directa una fórmula o ecuación, ya que requiere del uso de un simulador de tráfico.

El simulador desarrollado y explicado en el apéndice A.2, toma como argumento de entrada el modelo de una intersección concreta, los posibles caminos (optimizados previamente) entre los puntos de entrada y de salida, y la cadena de bits correspondiente a cada individuo de la población. La primera acción que realiza el simulador es descomponer esa cadena de bits dependiendo del número de ramas y carriles por rama que tenga el modelo de intersección utilizado. De esta forma

obtiene los patrones que debe probar en cada entrada de la intersección. Estos patrones son extendidos a través de su duplicación hasta completar el tiempo total de simulación. Este tiempo total de simulación depende del tamaño de la intersección y del patrón, siendo igual para todos los individuos de una misma población. Cada paso de simulación representa el tiempo que emplea un vehículo en pasar de una celda a otra del modelo de autómatas celulares de la intersección. Debido a que todos los vehículos tienen la misma velocidad en este escenario, en cada paso de simulación se actualiza la posición de todos ellos.

La ocupación de la intersección (número de celdas ocupadas por vehículos) puede variar a lo largo de la simulación según los patrones utilizados. Por este motivo la métrica utilizada es el valor medio de las ocupaciones obtenidas en cada paso de simulación, descartando los primeros valores hasta que la intersección se ha rellenado con un patrón completo. En la ecuación 2.17 se muestra este cálculo.

$$\bar{o} = \frac{\sum_{i=t_{stable}}^{i=t_{end}} o(i)}{t_{end} - t_{stable} + 1} \quad (2.17)$$

donde  $t_{stable}$  es el paso de simulación donde la simulación ha alcanzado un estado estable,  $t_{end}$  es el último paso de simulación y  $o(i)$  es la ocupación en cada instante de simulación.

El valor de ocupación total ( $o(t)$ ) es medido en base a la capacidad máxima de la ocupación ( $I_s$ ) que depende del número de celdas que compone la rejilla interior de la intersección. Cuando dos o más vehículos coinciden en la misma celda, durante el mismo instante de simulación se produce una colisión. El número total de colisiones se denota por  $C$ . Utilizando estos valores, el valor final de la función de evaluación se obtiene de la siguiente forma:

$$Fv = \frac{(1 - \frac{\bar{o}}{I_s}) + \frac{C}{T}}{2} \quad (2.18)$$

$Fv$  es un valor entre 0 y 1. Para el máximo valor medio de ocupación sin colisiones la función devuelve un valor  $Fv = 0$ .

### 2.3.4.3. Población inicial, funciones de cruce y mutación, y configuración del GA

Para definir el comportamiento del algoritmo genético es necesario desarrollar las funciones que intervienen en cada parte del proceso. A continuación, se explicará la propuesta para cada una de ellas:

- **Generación de la población inicial:** Dada una longitud de patrón  $l_p$  determinada, se genera una cadena de bits aleatoria de longitud  $l_{ind}$ . El número de individuos que forma una población es un factor importante ya que un número pequeño de individuos proporcionaría poca diversidad en la población, mientras un número excesivamente grande puede generar problemas de convergencias hacia una solución válida. En nuestra propuesta el número de individuos dependerá del tamaño del patrón.
- **Función de cruce:** La función de cruce elige de forma pseudoaleatoria (se realiza una clasificación según los valores de evaluación obtenidos en la generación anterior y se barajan) dos individuos de la población  $p_1$  y  $p_2$ . Los genes de estos individuos son mezclados para obtener un nuevo individuo  $c_1$ . Para realizar el cruce se genera un vector aleatorio ( $cv$ ) con valores binarios de la misma longitud de los padres. En las posiciones de  $cv$  que tienen el valor 1,  $c_1$  toma el valor de esa misma posición de  $p_1$ . Si el valor del elemento de  $cv$  es 0,  $c_1$  toma el valor de la misma posición de  $p_2$ . A continuación se muestra un ejemplo de la aplicación de este operador:

$$\begin{aligned}
 p_1 &= [A \ B \ C \ D \ E \ F \ G] \\
 p_2 &= [a \ b \ c \ d \ e \ f \ g] \\
 cv &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1] \\
 c_1 &= [A \ B \ c \ d \ E \ f \ G]
 \end{aligned}$$

- **Función de mutación:** Este proceso se aplica de forma aleatorio a cada individuo de la población. Los genes mutados cambian sus valores por el contrario que tiene en ese momento. La selección de los genes que varían se realiza utilizando una función de distribución normal con media 0 y desviación estándar dependiente del tamaño del individuo y de las diferencias de los valores de evaluación de la generación anterior.

En cada generación, una vez obtenidos los valores de evaluación de cada individuo, se seleccionan un porcentaje de los mejores de ellos que pasan directamente a la siguiente generación (élite). El resto de los individuos de la siguiente generación son obtenidos a partir de la función de cruce explicada anteriormente.

#### 2.3.4.4. Experimentos y resultados

Para validar esta solución propuesta se han diseñado dos conjuntos de experimentos diferentes. El primero de ellos consiste en optimizar el mismo caso de uso mostrado en el trabajo de Azimi [27] y compararlo con el obtenido por nuestra propuesta

utilizando el simulador de intersecciones desarrollado. El segundo conjunto de experimentos se basa en crear una tasa de llegada de vehículos que cumpla con el patrón obtenido en el experimento anterior y reproducir el escenario de tráfico con un simulador microscópico convencional, en este caso SUMO. En estos experimentos de tiempo continuo, se comparará el sistema propuesto con otros sistemas de gestión de intersección como son el uso de señales de ceda el paso y el uso de semáforos. Mientras el primer experimento nos permite evaluar nuestra solución frente a un máximo teórico, el segundo conjunto nos proporciona información sobre el rendimiento de esta solución en comparación a otros sistemas de gestión.

Para la definición del experimento, teniendo en cuenta las dimensiones de la intersección del caso de uso y los tamaños de los posibles caminos, se ha optado por probar con longitudes de patrón de entre 2 y 14 celdas. De igual forma, para evaluar la variabilidad del GA, cada proceso de optimización se repetirá 100 veces. Los parámetros de configuración GA para este experimento serán:

- Tamaño de patrón: 2 a 14 unidades.
- Tamaño de la población: 100 individuos.
- Número máximo de generaciones: 150.
- Valor de tolerancia:  $10^{-12}$ .
- Tamaño de la élite: 5 % del tamaño de la población.

El experimento se ha realizado utilizando MATLAB R2018a y la “*Global Optimization Toolbox*” para implementar el GA. También se ha utilizado “*Parallel Computing Toolbox*” durante la ejecución del algoritmo utilizando un máximo de 4 hilos. El equipo de pruebas contaba con un procesador Intel i7-7700, 32 GB de memoria RAM (DDR4 2400 MHz) y Ubuntu 18.04 como sistema operativo.

Los resultados obtenidos tras la ejecución de OCP 100 veces para cada longitud de patrón se muestran en la figura 2.12. Como valor de referencia se han simulado los patrones propuestos por el trabajo de Azimi, cuya longitud es de 5 unidades y valor  $P = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$ . Esto significa que la cadencia de llegada de vehículos por el carril derecho (giran a la derecha) es  $L_{right} = [1 \ 1 \ 1 \ 1 \ 1]$ , la del carril central (continúan recto) es  $L_{middle} = [0 \ 1 \ 1 \ 0 \ 0]$  y la del carril izquierdo (giran a la izquierda) es  $L_{left} = [1 \ 0 \ 0 \ 0 \ 0]$ . Con este patrón el valor medio de ocupación obtenida  $\bar{o} = 53,33\%$ , siendo el valor utilizado de referencia marcado con línea verde discontinua en la figura 2.12a.

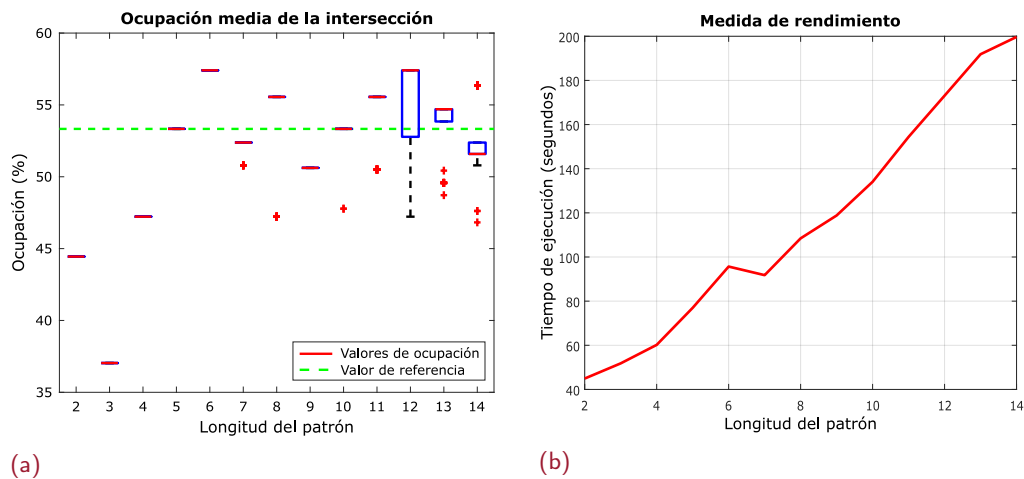


Figura 2.12.: Resultados obtenidos de aplicar OCP al caso de uso de una intersección con 4 brazos y 3 carriles por brazo. (a) Diagrama de cajas que muestra los valores medios de ocupación (color rojo) para cada longitud de patrón probada. (b) Medida del tiempo de ejecución del algoritmo para cada longitud de patrón hasta converger en una solución.

Utilizando OCP, el valor medio máximo se da para una longitud de patrón de 6 unidades, situándose en  $\bar{o}_{OCP} = 57,407\%$ , lo que supone una mejora de casi el 8 % frente al supuesto máximo teórico anterior. En cuanto a rendimiento, es posible observar en la gráfica de la figura 2.12b que al aumentar la longitud de patrón, el tiempo de convergencia aumenta de forma proporcional. También cabe destacar como es posible observar, a partir de los valores de ocupación medio obtenidos, como para valores múltiples entre si tales como 5 y 10 o 6 y 12, se obtienen los mismos valores medios de ocupación. Esto confirma la teoría de que, según el tamaño de la intersección, a partir de una determinada longitud los patrones y sus resultados se repiten.

En cuanto al conjuntos de experimentos realizados en tiempo continuo con SUMO se ha definido una intersección con el mismo caso de uso, pero para este escenario es necesario generar una longitud de carriles de entrada que soporte el número de vehículos que van a ser insertados, y que permita que estos adquieran la velocidad marcada para su entrada a la intersección. Con este objetivo, los carriles de entrada a la intersección miden 300 metros, y la velocidad máxima que pueden alcanzar los vehículos en ellos es de 13.89 m/s (50 km/h). Se han creado tres niveles de tráfico diferentes: bajo, medio y alto. Esto permitirá observar el comportamiento de cada tipo de sistema de gestión con diferentes cargas de trabajo. Los sistemas de gestión utilizados serán tres. El primero de ellos el basado en prioridades (P) a través de la colación de señales de ceda el paso en las entradas a la intersección. El segundo de ellos es el basado en semáforos (TL) con ciclos fijos de 42 segundos en verde. Por último, se utilizará OCP utilizando el patrón que mejor valor medio de ocupación ha obtenido en los experimentos anteriores.

Tipo de giro	De*	A*	Bajo			Medio			Alto		
			P	TL	OCP	P	TL	OCP	P	TL	OCP
A derecha	1	8	884	887	872	1181	1180	1255	1419	1255	1396
	2	1	887	886	880	1182	1181	1242	1418	1242	1408
	4	2	886	887	872	1182	1180	1250	1417	1250	1396
	8	4	843	887	872	1180	1179	1250	1417	1250	1396
Continuar recto	1	4	355	440	437	474	586	804	708	804	873
	2	8	269	442	437	168	590	798	52	798	873
	4	1	354	440	438	473	586	796	710	796	877
	8	2	255	444	437	174	857	800	55	800	873
A izquierdas	1	2	178	147	146	237	196	189	354	189	291
	2	4	148	146	146	64	194	180	23	180	291
	4	8	177	147	146	236	196	179	354	179	291
	8	1	122	146	147	65	194	183	22	183	293

\* Los valores 1,2,4, y 8 se corresponden con las entradas superior, derecha, inferior e izquierda de la intersección, respectivamente.

Nota: Todos los valores para P, TL y OCP indican el número de vehículos que han completado su viaje durante la duración de la simulación.

**Tabla 2.3.:** Resultados de la simulación basada en el número de vehículos que han finalizado su viaje durante el tiempo de simulación (1 hora). Se han realizado un total de nueve simulaciones combinando tanto sistemas de gestión de la intersección basados en prioridad (P), en semáforos (TL) y en patrones de cruce optimizados (OCP), como volúmenes de tráfico bajos, medios y altos.

Los resultados obtenidos tras la ejecución de todas las simulaciones de muestran en la tabla 2.3. Observando estos valores podemos ver como los giros a la derecha presentan valores constantes en para todos los sistemas de gestión y para todas las cargas de tráfico. Esto se debe a que el camino utilizado no tiene puntos de conflicto. Por el contrario, el giro a la izquierda es el más complicado de las tres opciones de giro y es donde se pueden observar mejor los efectos de cada sistema de gestión. Las intersecciones gestionadas por cedas el paso, para cargas de tráfico medias y alta, presentan desigualdades importantes según el sentido de la marcha. El sistema basado en semáforos reparte por igual el tiempo en ambos sentidos, pero como los vehículos deben detenerse en tiempos alternos, obtiene valores inferiores a los conseguidos por OCP.

Este sistema de optimización denominado OCP y experimentos realizados para su validación están incluidos en el artículo [67] que está recogido en el capítulo 4.

### 2.3.5 Optimización para ajustar flujos de llegada preestablecidos

El objetivo del problema tratado en el sección anterior es maximizar la utilización de la intersección bajo una condiciones muy concretas. En esa solución se fijaban los flujos de llegada a la intersección para que fueran iguales en todas sus entradas y se



limitaban los movimientos de los vehículos dependiendo de su punto de entrada a la intersección.

Aunque este planteamiento permitió verificar la validez del proceso de modelado de la intersección propuesto, obtuvo mejores resultados que los estudios anteriores, y proporciona un método de obtención de patrones de llegada automatizado, presenta algunas carencias a la hora de generalizar el problema. Por este motivo, en esta sección se verá una nueva propuesta que extiende el proceso de optimización OCP, añadiendo la posibilidad de que cada vehículo tenga libertad de movimiento dentro de la intersección, es decir, pueda ir desde cualquier punto de entrada de la intersección hacia cualquier brazo de salida. A este nuevo proceso se le denominará OCP-MP (del inglés, *Optimized Crossing Patterns with Multiple Paths*).

El modelo de intersección es el mostrado en la sección 2.2.2 y utilizado en OCP pero el espacio de búsqueda del algoritmo se extiende exponencialmente al aumentar el número de caminos permitidos. Por este motivo es necesario diseñar nuevas técnicas para generar la población inicial y, nuevas funciones de cruce y mutación que tengan en cuenta las peculiaridades de esta nueva propuesta.

Los elementos utilizados durante del desarrollo del sistema OCP-MP son análogos a los utilizados en OCP, aunque se ha refinado su nomenclatura. En esta nueva versión se define el conjunto de los brazos que forman la intersección como  $B(x)$  (ecuación 2.19) y, para cada brazo, el conjunto de carriles que lo forma como  $L_{B(x)}(y)$  (ecuación 2.20).

$$B(x) \mid x = \{1, 2, \dots, n\}, \quad (2.19)$$

$$L_{B(x)}(y) \mid y = \{1, 2, \dots, m\}, \quad (2.20)$$

donde  $x$  es un conjunto de números enteros entre 1 y  $n$ , donde  $n$  indica el número total de brazos de la intersección. De igual forma,  $y$  es el conjunto de números enteros entre 1 y  $m$  que identifica cada carril de entrada del brazo  $B(x)$ . Sólo se deben identificar los carriles de entrada ya que son los únicos que intervienen en la formulación del problema.

El nuevo sistema de optimización OCP-MP está publicado en el artículo [68] que está recogido en el capítulo 5.

### 2.3.5.1. Codificación del problema

A diferencia de OCP, donde cada posición de una celda virtual únicamente representaba la presencia o ausencia de un vehículo (el camino venía marcado por el punto de acceso a la intersección), en OCP-MP es posible utilizar diferentes caminos. Por este motivo el valor de la codificación de cada gen de un individuo no puede ser un valor binario. En la ecuación 2.21 se muestra este nuevo conjunto de posibles valores.

$$P_{L_{B(x)}(y)} = \{0, 1, \dots, r\}, \quad (2.21)$$

donde  $P_{L_{B(x)}(y)}$  es el conjunto de los posibles caminos desde un punto de entrada concreto del brazo  $B(x)$  y el carril  $L_{B(x)}(y)$  hasta cualquier brazo de salida de la intersección. Estos posibles caminos son etiquetados utilizando un número entero entre 0 y  $r$ , donde 0 representa la ausencia de vehículo.

El tamaño de  $P$  puede variar dependiendo del número total de brazos de la intersección, del número de carriles de cada brazo y de las posibles restricciones de giro en cada caso. La metodología utilizada para obtener los posibles caminos con OCP se describe en la sección 2.3.3.2 y se basa en obtener el mínimo número de caminos que generen la menor cantidad de puntos de conflicto. Este conjunto de caminos presenta limitaciones en cuanto al número de vehículos que desde un brazo en concreto puede tomar una opción de giro concreta. Por ejemplo, aunque todos los vehículos que accedan a la intersección por un brazo determinado quisieran girar hacia otro brazo, sólo podrían utilizar el punto de entrada que proporciona ese destino. Por el contrario, como ya se ha mencionado, en OCP-MP se propone que cada vehículo tenga libertad de elección de caminos aplicando una única restricción: que los vehículos realicen sólo un cambio de trayectoria en el interior de la intersección. Esta restricción es aplicada porque cualquier cambio en la trayectoria de un vehículo se traduce como un ajuste o reducción de su velocidad. Limitando el mismo número de cambios de trayectos por igual, todos los caminos dentro de la intersección pueden ser tomados con la misma velocidad.

Siguiendo el caso de uso utilizado durante los casos anteriores de una intersección de cuatro brazos y tres carriles de salida y entrada por brazo, los caminos  $P$  obtenidos para una rama concreta se muestran en la figura 2.13.

Como se puede observar, con esta metodología, desde un mismo punto de entrada a la intersección es posible llegar a cualquiera de las otras ramas. Cumpliendo la restricción de un único cambio de trayectoria, para algunos de los giros es posible que se den varios caminos (caso del giro a la derecha o a la izquierda de la figura 2.13), mientras que para otros brazos tengamos únicamente un camino posible. De esta forma,  $P_{L_{B(1)}(1,2,3)} = \{0, 1, 2, 3, 4, 5, 6, 7\}$ .

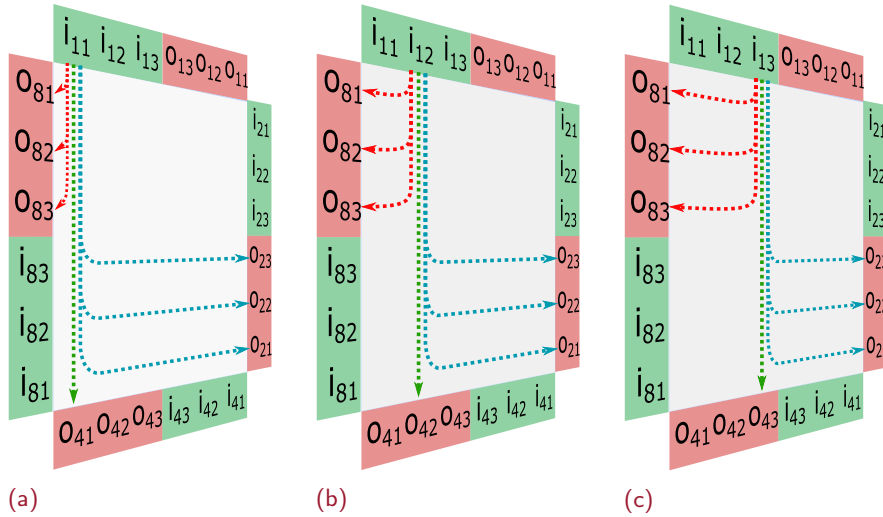


Figura 2.13.: Representación de los posibles caminos entre los puntos de entrada del primer brazo hacia todas las salidas, siguiendo la restricción de caminos con un único cambio de trayectoria. (a) Punto origen  $i_{11}$ . (b) Punto origen  $i_{12}$ . (c) Punto origen  $i_{13}$ .

Una vez definido estos nuevos caminos y variado el valor de cada gen del individuo, el resto de la codificación es igual a la mostrada en la figura 2.11b.

### 2.3.5.2. Función de evaluación

El objetivo de este proceso de optimización es obtener los patrones de llegada de vehículos que más se aproximen a unos flujos de llegada preestablecidos. Denominaremos  $Q$  al conjunto de tasas de llegada de vehículos preestablecidas.

$$Q = [q_1(n), \dots, q_1(3), q_1(2), \dots, q_n(1)], \quad (2.22)$$

donde  $Q(i)$  es el elemento  $i$  del vector de valores  $Q$  (por ejemplo,  $Q(1) = q_1(n)$ ). Cada una de estas proporciones se define en base a la máxima capacidad del brazo de la intersección al que hace referencia y, por tanto, debe ser un valor menor o igual a 1. Asumiendo que no es habitual que un vehículo pueda salir y entrar por el mismo brazo de la intersección, el número de elemento de  $Q$  será  $n \cdot (n - 1)$ . El simulador utilizado por la función de evaluación es el descrito en el apéndice A.2 e implementa las siguientes métricas:

- **Colisiones medias** ( $C_{avg}$ ): En cada paso de simulación  $i$ , se almacena el número de celdas ocupadas por más de un vehículo. Cuando el tiempo de simulación  $l_{sim}$  finaliza, se calcula la media todos los valores de colisión parciales obtenidos (ecuación A.1).

- **Suma de errores absolutos ( $SAE$ ):** Al finalizar la simulación, se comparan los flujos preestablecidos definidos en  $Q$  con los obtenidos al final el proceso de optimización  $q_{opt}$ . La suma del valor absoluto de la diferencia de estos valores proporciona el valor final de  $SAE$  (ecuación A.2).
- **Error medio absoluto normalizado por flujos ( $NMAE_{flows}$ ):** Este parámetro mide la diferencia entre  $Q$  y  $Q_{opt}$ , evaluando cada subflujo de forma independiente (ecuación A.3).

Con esta nuevas métricas, la función de evaluación se define tal y como se muestra en la ecuación 2.23.

$$f = \alpha \cdot C_{avg} + \beta \cdot SAE + \gamma \cdot NMAE_{flows}, \quad (2.23)$$

donde  $\alpha$ ,  $\beta$  y  $\gamma$  son los pesos asociados a cada una de las métricas. Cada uno de estos pesos puede ser utilizado para propósitos diferentes. El valor de  $\alpha$  es utilizado para obtener una solución sin colisiones. Los valores  $\beta$  y  $\gamma$  permiten variar los objetivos de la optimización entre buscar un patrón que minimice la diferencia entre las tasas de flujo buscadas y las obtenidas, en valor absoluto o en valor medio.

### 2.3.5.3. Algoritmo genético con cromosomas de longitud variable

Otras de las limitaciones del sistema OCP era que debía de ejecutarse de forma independiente para cada longitud de patrón que quisiera probarse. Como se puede observar en los resultados de la figura 2.12b, el tiempo de ejecución aumenta progresivamente al aumentar la longitud del patrón. Además del inconveniente del tiempo de ejecución, OCP forzaba a tener que elegir el tamaño de patrón candidato, por lo que podría quedarse algún valor sin probar.

Para resolver este problema se ha propuesto el uso de un algoritmo genético con individuos con longitud variable. De esta forma sería posible que en una misma población hubiera individuos que representaran a diferentes tamaños de patrón, y la solución final obtenida nos proporcionaría al mismo tiempo el mejor valor de evaluación y la longitud de patrón que lo genera. A este sistema de optimización le denominaremos OCP-MP-VL (*Optimized Crossing Patterns with Multiple Paths and Variable Length*).

Revisando algunos de los trabajos disponibles en el estado del arte, en [69] se realizar una revisión de las soluciones que utilizan genomas de longitud variable y los efectos de diferentes tipos de codificación en la convergencia de los algoritmos. Un ejemplo de un caso de uso centrado en la organización de tareas aparece en

el estudio [70]. Centrado en los escenarios de tráfico, en [71] Qiongbing and Lixin aplican un GA con cromosomas de longitud variable para la búsqueda de caminos óptimos. Cada gen representa un nodo de la red de carreteras, por lo que la sucesión ordenada de genes que forman cada individuo representa ruta candidata. Los métodos anteriormente citados pueden resumirse en tres conceptos: métodos de selección óptima para los puntos de cruce (con cromosomas de diferente tamaño, qué puntos de cada individuo hacen referencia a la misma información), procesos de cruce (como generar un nuevo individuo y corregir el genotipo tras el proceso) y cómo añadir o eliminar valores de genes independientes (por ejemplo, para añadir una nueva tarea o un nuevo nodo intermedio).

La mayor parte de los trabajos relacionados con GA con cromosomas de longitud variable están centrados en aplicaciones donde el valor de cada gen tiene entidad propia y cada individuo representa una sucesión de estos elementos. En el caso de OCP-MP-VL no se cumple este principio, por lo que es necesario realizar funciones para generar la población inicial o realizar el cruce y la mutación, teniendo en cuenta las características concretas de nuestra propuesta de codificación.

#### 2.3.5.4. Población inicial y funciones de selección, cruce y mutación

Antes de definir la función de generación de la población inicial para OCP-MP y OCP-MP-VL, debemos tener en cuenta algunas restricciones de nuestro escenario:

- La longitud de cada individuo depende tanto del número de brazos  $B(x)$  como del número de carriles de cada brazo  $L_{B(x)}$ , y también del número de celdas virtuales de cada carril (longitud del patrón)  $l_p$ .
- El valor de cada gen debe estar contenido en cada conjunto  $P_{L_{B(x)}(y)}$ .

Considerando estos puntos, los pasos a seguir para el proceso de **generación de la población inicial** propuesto son:

1. La función recibe como parámetros el número de brazos y de carriles de la intersección, las longitudes ( $l_p$ ) mínimas y máximas del patrón buscado, el número de individuos de longitud máxima de patrón a generar y el conjunto de flujos de llegada (con sus opciones de giro), que serán objetivo del proceso de optimización.
2. El tamaño de la población depende del intervalo entre el tamaño mínimo y máximo de patrón, y del número de individuos con tamaño mayor de patrón. El número total de individuos se obtiene de la siguiente forma:

- Se definen  $min(l_p)$  como la longitud mínima de patrón,  $max(l_p)$  como la longitud máxima de patrón y  $max(subP)$  como el número de individuos con  $max(l_p)$ .

- El conjunto  $L$  contiene los tamaños de patrón que el algoritmo tendrá en cuenta en su búsqueda:

$$L = [min(l_p), min(l_p) + 1, \dots, max(l_p)]$$

donde, por tanto, el tamaño de  $L$  es  $max(l_p) - min(l_p) + 1$ .

- El tamaño para el conjunto de individuos con tamaño  $min(l_p)$  es el valor entero de  $max(subP)/3$ . Este valor ha sido obtenido de forma empírica para repartir el tamaño de la longitud total en subpoblaciones.

- El vector que contiene los tamaños de cada subpoblación es:

$$subP = [max(subP)/3, (max(subP)/3) + inc, \dots]$$

$$\text{donde } inc = \frac{max(subP) - \frac{max(subP)}{3}}{max(l_p) - min(l_p)}$$

3. Teniendo en cuenta los flujos esperados, se genera un vector formado por las probabilidades de destino denominado  $P_{L_{B(x)}(y)}$ .
4. Los genes de cada individuo son generados a partir de los valores posibles definidos en  $P_{L_{B(x)}}$  siguiendo las probabilidades de  $P_{L_{B(x)}(y)}$ . Se generan tantos individuos de cada tamaño como se indican en  $subP$ .

El **operador de selección** utilizado es el denominado *selectionremainder* incluido en biblioteca *global optimization* de Matlab R2018b. Este operador está basado en la obtención de un vector de expectativas a partir de las puntuaciones de evaluación de la generación actual. Los padres son elegidos de forma determinista, basada en el valor de la parte entera de este valor de expectativa. Al finalizar el proceso, si todavía quedan padres que seleccionar, la función utiliza la parte decimal para elegirlo de forma estocástica [72].

Tras analizar los posibles opciones de cruce estudiados en [73], el método que mejor se adapta a las características de este tipo de individuos es la metodología de cruce gen a gen. En nuestro caso, para garantizar el cumplimiento de todas las restricciones que contiene el modelado, el cruce se realizará patrón a patrón tal y como aparece en la figura 2.14. Recordemos que cada patrón lo forman un conjunto de genes consecutivos de un individuo. En el caso de OCP-MP, al tener la misma longitud de patrón todos los individuos, el cruce puede realizarse de forma directa. En el caso de que alguna de las ramas de la intersección no tuvieran las mismas opciones de giro (valores de  $P_{L_{B(x)}(y)}$  diferentes) tras el cruce, debería realizar un proceso para arreglar posibles individuos inválidos.

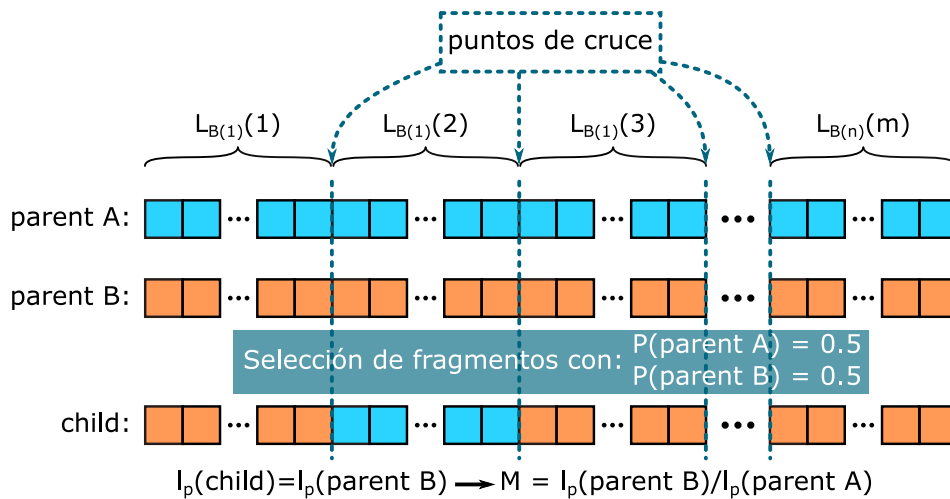


Figura 2.14.: Proceso de cruce de dos individuos de una misma población utilizando el optimizador OCP-MP-VL

Para OCP-MP-VL el proceso de cruce debe enfrentarse al problema de cómo intercambiar dos porciones de genes en individuos de diferente longitud. Para ello<sup>o</sup> es necesario definir los posibles casos que se pueden dar y el funcionamiento de los operadores de compresión y extensión utilizados.

El proceso de cruce en OCP-MP-VL se muestra en la figura 2.15 y seguirá los siguientes pasos:

1. Una vez seleccionados dos individuos de la población (*parentA* y *parentB*) se dividen en el mismo número de fragmentos, dependiendo del número de brazos y carriles de la intersección.
2. Se crea un nuevo hijo del tamaño de *parentB* con los genes vacíos. La relación (*M*) entre los tamaños de los dos padres se obtiene de la siguiente forma:

$$M = \frac{l_p(\text{parentB})}{l_p(\text{parentA})} \quad (2.24)$$

3. Cada uno de los fragmentos de individuos creados en el paso 1 es evaluado de forma simultánea para (*parentA* y *parentB*). Se elige uno de los padres con una probabilidad de 0,5. Si el seleccionado es *parentB*, se copia el fragmento correspondiente (conjunto de genes consecutivos que representa el patrón de entrada de un carril concreto de la intersección) al hijo, ocupando las mismas posiciones ya que *parentB* y el hijo tiene el mismo tamaño. Si el elegido es *parentA* pueden darse diferentes casos:

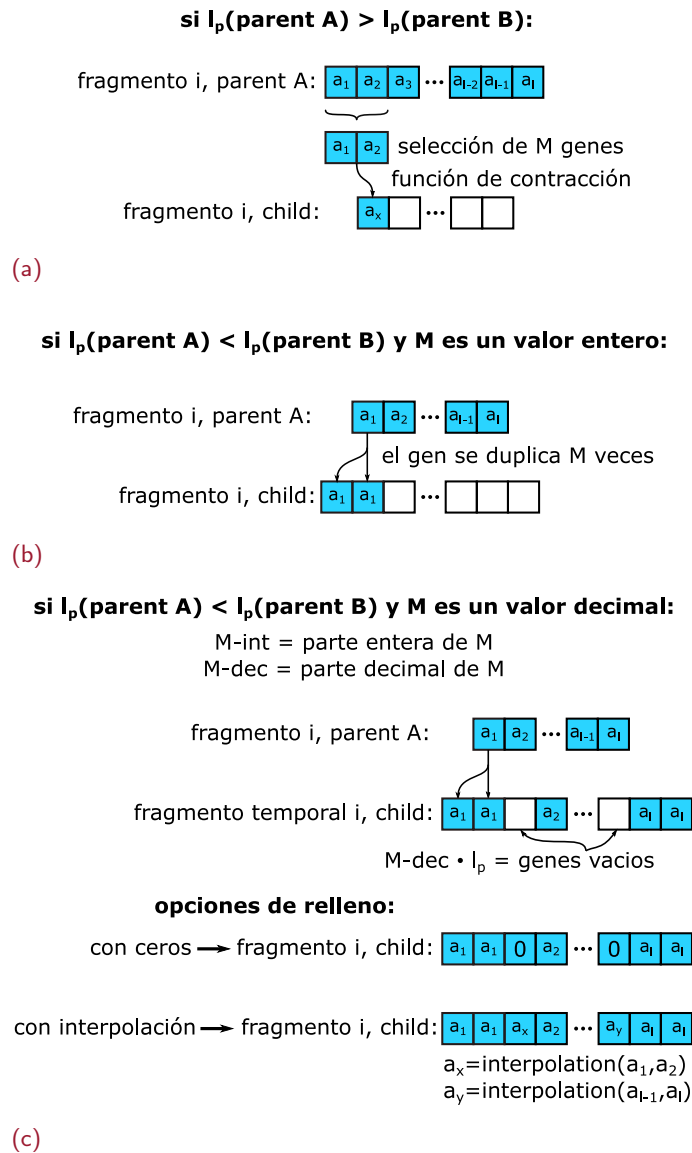


Figura 2.15.: Procesos de contracción y expansión de genes cuando los padres no tiene la misma longitud de patrón. (a) Proceso de expansión de genes. (b) Proceso de contracción de genes cuando M es un valor entero. (c) Proceso de contracción de genes cuando M es un valor decimal.

- a) Si  $l_p(\text{parentA}) = l_p(\text{parentB})$ , los genes correspondientes a ese subindividuo de *parentA* se copian directamente en esa misma ubicación del hijo (figura 2.15a).
  - b) Si  $l_p(\text{parentA}) > l_p(\text{parentB})$ , se produce un proceso de contracción de ese conjunto de genes, tal y como se muestra en la figura 2.15b.
  - c) Por último, si  $l_p(\text{parentA}) < l_p(\text{parentB})$  se debe realizar el proceso de expansión mostrado en la figura 2.15c.
4. El proceso finaliza cuando se han evaluado todos los subindividuos de los padres.



Por último, el **proceso de mutación** recibe como parámetro los genes de un individuo concreto, la tasa de mutación que indica la proporción de genes que pueden ser mutados y los posibles valores a los que puede variar un gen. Este último parámetro garantiza la validez de un individuo una vez finaliza el proceso.

Este operador decide, con una probabilidad igual a la tasa de mutación, si un gen debe ser modificado. En caso afirmativo, elige aleatoriamente un nuevo valor entre los disponibles según su posición en el individuo. Recordemos que su posición le encuadra dentro de un patrón en un brazo/carril específico, con una opciones de giro ( $P_{L_{B(x)}(y)}$ ) concreta.

Para un número configurable de generaciones iniciales, el operador de mutación no es utilizado para favorecer la convergencia más rápida a una solución subóptima.

### 2.3.5.5. Experimentos y resultados

A diferencia del método OCP, cuya función de evaluación perseguía obtener los patrones de cruce que obtuvieran la máxima ocupación posible de una intersección, con OCP-MP y OCP-MP-VL el objetivo es mucho más ambicioso al buscar los patrones que obtengan los flujos de tráfico más cercanos a los deseados. En este caso es necesario definir flujos de tráfico candidatos para poder realizar los experimentos. Teniendo en cuenta esto, se han diseñado dos tipos de experimentos para validar los optimizadores propuestos. El primero de ellos está basado en flujos de tráfico de los cuales se conoce una combinación de patrones válida (apéndice B.2.1). Para el segundo conjunto de experimentos se ha optado por generar 100 flujos de tráfico de forma aleatoria, siguiendo diferentes distribuciones que permitan probar casos extremos (apéndice B.2.2). Además de para OCP-MP y OCP-MP-VL, estos conjuntos de datos se probarán con los dos optimizadores cuyo funcionamiento ya es conocido: el primero basado en un algoritmo de Hill Climber (HC) [74], [75] y el segundo será OCP.

La configuración para lanzar los optimizadores OCP-MP y OCP-MP-VL se muestra a continuación:

- **Función de evaluación:** Aplicando la ecuación 2.18 se ha optado por utilizar un objetivo global que minimice la diferencia entre  $Q$  y  $Q_{opt}$ , estableciendo los parámetros  $\alpha = 0,5$ ,  $\beta = 0,5$  y  $\gamma = 0$ .

- **Población inicial:** A partir de los resultados obtenidos para este tipo de intersección en otros experimentos se ha definido  $min(l_p) = 3$  y  $max(l_p) = 13$ , lo que genera una población con 462 individuos.
- **Operador de mutación:** El operador se activa tras la décima generación y la tasa de mutación se ha definido con el valor 0,05.

Para el primer conjunto de experimentos (datos de entrada con solución conocida), mostrado en el apéndice B.2.1, se han definido los siguientes vectores de flujos  $Q$ :

- **Experimento A1:** Todos los vehículos quieren girar a la derecha cuando llegan a la intersección.
- **Experimento A2:** Evitando giros a la izquierda, un tercio de los vehículos quieren girar a la derecha y dos tercios quieren continuar recto.
- **Experimento A3:** El tercero combina las tres opciones de giro y es la solución que propone Azimi en [27] para este tipo de intersecciones.

El segundo conjunto de datos (generados de forma pseudoaleatoria) es analizado en profundidad en el apéndice B.2.2. La distribución de los flujos elegida se ha generado utilizando las siguientes indicaciones:

- **Experimento B1:** Se realiza un reparto de la capacidad máxima de cada rama de forma aleatorio. Se comienza por obtener un valor de hasta el 0,6 de su capacidad (siendo 1 el valor máximo de capacidad). El segundo valor se obtiene como un valor aleatorio entre 0 y la capacidad restante una vez se ha asignado el primer valor. El tercer y último valor, se calcula aleatoriamente entre 0 y la capacidad restante que se le ha asignado el primer y al segundo valor. Los tres valores calculados son asignados de forma aleatoria a las posiciones del vector que representa cada opción de giro ( $q_{right}(i)$ ,  $q_{go}(i)$  y  $q_{left}(i)$ ). El proceso se repite hasta completar todas las ramas de la intersección.
- **Experimento B2:** Se generan flujos calculando valores aleatorios entre 0 y 0,7, 0,2 y 0,1. Los valores obtenidos son barajados y asignados a cada opción de giro.
- **Experimento B3:** El último experimento es el más extremo. En este caso se asigna un valor de giro entre 0 y 1 a un sentido de giro.

Siguiendo estas distribuciones definidas anteriormente, para B1 se obtendrá un conjunto de datos con el objetivo de repartir la capacidad máxima de cada rama de la intersección, de mayor a menor tasa, para cada opción de giro. Para B2, el tráfico se concentra mayoritariamente en una de las opciones de giro, pero pudiendo haber flujos las otras dos opciones de giro. Por último, en B3, todo el posible tráfico generado para cada rama de entrada a la intersección cuenta con la misma preferencia de giro.

Los resultados numéricos obtenidos tras finalizar todos los experimentos se muestran en la tabla 2.4 para A1, A2 y A3, y en la tabla 2.5 para B1, B2 y B3. En este segundo caso, los resultados aparecen agrupados por la media y desviación estándar obtenida a partir de repetir el mismo proceso de optimización por los 100 vectores de entradas que componen cada uno de los experimentos.

Valor	Optimizador	Tipo de relleno	Experimento A1	Experimento A2	Experimento A3
Puntuación de evaluación	HC	-	1,3387	0,6356	0,1223
	OCP	-	0,1197	0,0438	0
	OCP-MP	-	0	0	0,0417
	OCP-MP-VL	Ceros	0,0694	0,0556	0
	OCP-MP-VL	Interpolación	0	0	0,0417
Tiempo de cálculo (s)	HC	-	466,5469	285,9582	234,9061
	OCP	-	2763,1273	3148,6648	2812,8252
	OCP-MP	-	1841,2997	1459,9598	1737,867
	OCP-MP-VL	Ceros	615,7362	310,6485	1585,902
	OCP-MP-VL	Interpolación	306,1409	118,2788	283,3829
Suma de errores absolutos	HC	-	0	0,0972	0,0586
	OCP	-	0,2393	0,0876	0
	OCP-MP	-	0	0	0,0833
	OCP-MP-VL	Ceros	0,1389	0,1111	0
	OCP-MP-VL	Interpolación	0	0	0,0833
Colisiones	HC	-	83	27	8
	OCP	-	0	0	0
	OCP-MP	-	0	0	0
	OCP-MP-VL	Ceros	0	0	0
	OCP-MP-VL	Interpolación	0	0	0

**Tabla 2.4.:** Resultados obtenidos durante los experimentos basados en datos de entrada con solución conocida para los cuatro procesos de optimización definidos.

La definición de los datos de entrada de los experimentos “Ax” se ha llevado a cabo utilizando vectores de flujos cuyos patrones ideales eran conocidos. Por este motivo, el resultado esperado del valor de su puntuación de evaluación es cero, es decir, obtener la solución ideal. Como se muestra en la tabla 2.4, únicamente los procesos de optimización que utilizan OCP-MP y OCP-MP-VL alcanzan este resultado.

Concretamente llegan a  $f = 0$  en la función de evaluación utilizando la función de cruce basada en la interpolación de valores para la función de extensión de genes para A1 y A2, y con la función de extensión que añade ceros en A3. Esta variación se debe a las propias características de los flujos de tráfico buscados ya que cuando se pretenden obtener flujos que en su totalidad ocupen la capacidad total de una rama (suma de los valores que hacen referencia a alguna de ramas del vector  $Q$  sea igual a 1), el método de añadir ceros al expandir genes crea huecos que disminuyen la puntuación, por lo que la interpolación funciona mejor. Sin embargo, cuando el optimizador se enfrenta a un problema donde no se utiliza la totalidad de la capacidad de alguna de las ramas, la expansión con ceros ofrece mejores resultados.

En relación con los tiempos de cálculo requeridos, OCP-MP-VL consigue valores hasta doce veces menores en algunos casos, mantiene o mejora los resultados numéricos obtenidos.

Valor	Optimizador	Experimento B1	Experimento B2	Experimento B3
Puntuación de evaluación	HC	1,169 ± 0,209	0,6278 ± 0,1572	0,5433 ± 0,2456
	OCP	0,0642 ± 0,0057	0,1186 ± 0,0138	0,1746 ± 0,0282
	OCP-MP	0,0649 ± 0,0031	0,1126 ± 0,0390	0,1536 ± 0,0867
	OCP-MP-VL	0,0683 ± 0,008	0,1072 ± 0,013	0,1312 ± 0,0199
Tiempo de cálculo (s)	HC	375,2 ± 7,402	320,6 ± 23,83	326,8 ± 44,83
	OCP	3022 ± 135,5	2992 ± 358	3104 ± 295,1
	OCP-MP	1426,57 ± 134,8	1208,89 ± 114,39	1064,5 ± 175,49
	OCP-MP-VL	496,3 ± 207,01	886,6 ± 433,7	389,1 ± 301,8
Suma de errores absolutos	HC	0,1628 ± 0,0376	0,2133 ± 0,0257	0,273 ± 0,039
	OCP	0,1283 ± 0,0113	0,2371 ± 0,0275	0,3491 ± 0,0565
	OCP-MP	0,1298 ± 0,0061	0,2253 ± 0,078	0,3072 ± 0,1734
	OCP-MP-VL	0,1366 ± 0,016	0,2144 ± 0,0259	0,2625 ± 0,0397
Colisiones	HC	70,14 ± 26,51	38,95 ± 12,86	29,58 ± 22,9
	OCP	0	0	0
	OCP-MP	0	0	0
	OCP-MP-VL	0	0	0

Tabla 2.5.: Resultados obtenidos durante los experimentos basado en datos de entrada aleatorios para los cuatro procesos de optimización definidos.

Respecto a los resultados obtenidos para los experimentos “Bx” mostrados en la tabla 2.5, los datos numéricos obtenidos proporcionan conclusiones similares al conjunto de experimentos anterior. En este caso, dado el carácter aleatorio de los datos de entrada, no era de esperar poder conseguir puntuaciones de evaluación iguales a cero ya que es muy poco probable encontrar de forma aleatoria una combinación de flujos y preferencias de giro compatibles entre sí. Los algoritmos de búsqueda local como HC presentan problemas importantes para espacios de búsqueda tan amplios

como los abordados en este caso, obteniendo las peores puntuaciones de todos. OCP-MP y OCP-MP-VL vuelven a conseguir puntuaciones mejores que OCP, obteniéndolas en tiempos de cálculo mucho menores (entre 2,5 y 8 veces inferiores). Los motivos de la reducción de estos tiempos es debida a dos aspectos: la convergencia más rápida hacia una solución válida al permitir caminos múltiples en OCP-MP, cuando los flujos de entrada son complejos; y la posibilidad de conseguir el tamaño del patrón ideal en un único proceso de optimización (OCP-MP-VL).

Todo el desarrollo de esta variación del proceso de optimización denominada OCP-MP-VL, y los experimentos que permiten evaluarlo respecto a todo lo visto anteriormente se ha publicado en [68]. Este artículo está recogido en el capítulo 5.

## 2.4 Relación de los artículos del compendio con la investigación realizada

En este capítulo se ha realizado un resumen amplio de la investigación realizada durante el desarrollo de esta tesis doctoral. Como ya se ha mencionado en la introducción, esta tesis se presenta para su defensa siguiendo la modalidad de compendio de artículos (incluidos en los capítulos 3, 4 y 5).

La figura 2.16 muestra de forma esquemática la relación de la investigación con estos artículos. Las tres primeras columnas se refieren a cada uno de los artículos publicados y en su encabezado aparece el título de la publicación y la revista. En la cuarta columna se han incluido las secciones principales del resumen amplio, manteniendo la numeración utilizada en este capítulo. Para reflejar qué partes de la investigación están incluidas en cada artículo, se han utilizados círculos de tres tamaños que indican si ese punto de la investigación tiene una relación baja, media o alta con la publicación en cuestión.

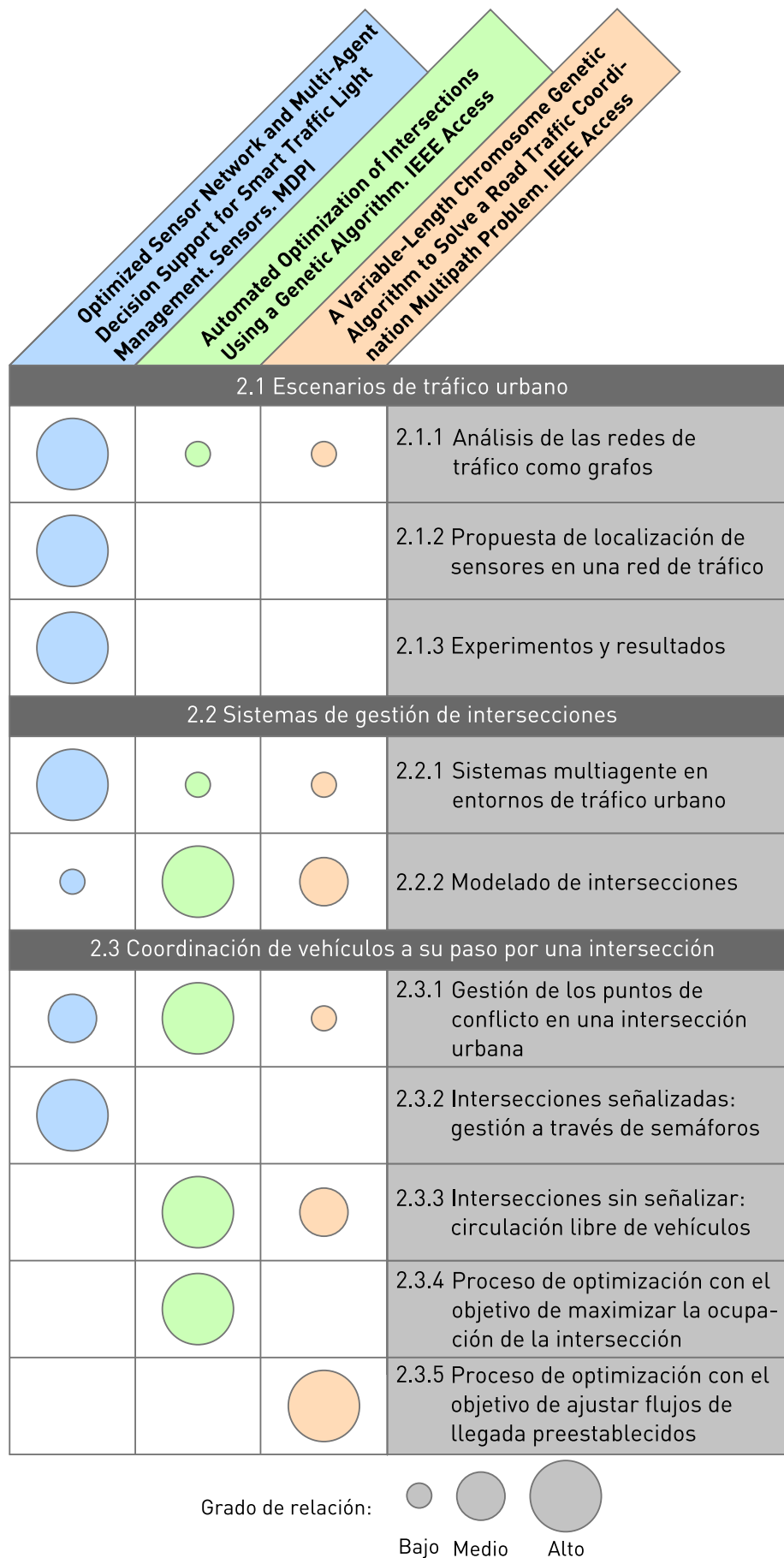


Figura 2.16.: Relación entre las publicaciones que dan lugar a esta tesis y los principales apartados de la investigación realizada.



# Parte II

---

Artículos publicados que dan  
lugar a la tesis





# Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management

En este capítulo se muestra el artículo publicado “Red optimizada de sensores y apoyo multiagente a la toma de decisiones para la gestión de semáforos inteligentes”. Tras un breve resumen de éste, se muestran los datos sobre la publicación y su relevancia. Por último, se ha incluido una copia de la versión publicada del artículo.

## 3.1 Introducción

Uno de los mayores retos en las sociedades modernas es disminuir los problemas relacionados con el tráfico rodado. Las redes de sensores en entornos de tráfico de vehículos han contribuido a mejorar el proceso de toma de decisiones de los sistemas inteligentes de transporte.

Aunque en las últimas décadas la popularización de dispositivos con acceso a Internet ha ocasionado la posibilidad de la generación masiva de datos en casi cualquier entorno, la instalación de sensores especialmente diseñados para escenarios de tráfico sigue siendo necesaria. La instalación de estos sensores suelen conllevar un alto coste económico debido a la infraestructura que necesitan. Nuestra propuesta consiste en utilizar la medida de centralidad de la red de carreteras para obtener las ubicaciones idóneas de instalación de estos sensores. Tras desplegar los sensores en un escenario simulado siguiendo esta propuesta, se ha diseñado un sistema multiagente compuesto por tres tipos de agentes: agentes de gestión de semáforos, agentes para la detección de atascos y agentes controladores de cada intersección. El objetivo final de este sistema de múltiples agentes es mejorar la duración del viaje de los vehículos en la red.

Para validar nuestra solución, hemos desarrollado los elementos necesarios para modelar los sensores y los agentes en un entorno de simulación. Hemos llevado a cabo experimentos utilizando el simulador SUMO (*Simulation of Urban MObility*) y el escenario de tráfico de la ciudad de Colonia denominado “TAPASCologne”. Los resultados obtenidos muestran que nuestra propuesta permite reducir la red de

sensores sin dejar de obtener información relevante para tener una visión global del entorno. Finalmente, con respecto al sistema multiagente, hemos llevado a cabo experimentos que demuestran que nuestra propuesta puede mejorar a otras soluciones existentes, como los sistemas convencionales de gestión de semáforos (estáticos o dinámicos) en términos de reducción de la duración del viaje del vehículo y de reducción en el número de mensajes necesario para llevar a cabo la toma de decisiones.

## 3.2 Relevancia de la publicación

En la Tabla 3.1 se muestra la información relativa al tipo de publicación, revista, identificador y los datos más importantes sobre la relevancia de la publicación.

Título:	Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management		
Autores:	Luis Cruz-Piris, Diego Rivera, Susel Fernandez, e Ivan Marsa-Maestre		
Estado:	Publicado		
Fecha de publicación:	2/02/2018		
DOI:	<a href="https://doi.org/10.3390/s18020435">https://doi.org/10.3390/s18020435</a>		
Revista:	Nombre:	Sensors	
	Editorial:	MDPI	
	País:	Suiza	
Relevancia	Revista:	Índice de impacto	3.031
		JCR (2018):	Q1 (15/61)
	Artículo:	Lecturas desde su publicación <sup>1</sup> :	1773
		Citas <sup>2</sup> :	7

<sup>1</sup> Fuente: MDPI, Fecha de consulta: 15/09/2019





<sup>2</sup> Fuente: Google Scholar, Fecha de consulta: 15/09/2019

**Tabla 3.1.:** Datos asociados a la publicación: Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management.

## 3.3 Artículo

Article

# Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management

Luis Cruz-Piris \* , Diego Rivera , Susel Fernandez  and Ivan Marsa-Maestre 

Departamento de Automática, Escuela Politécnica Superior, Universidad de Alcalá, 28805 Alcalá de Henares, Madrid, Spain; diego.rivera@uah.es (D.R.); susel.fernandez@uah.es (S.F.); ivan.marsa@uah.es (I.M.-M.)

\* Correspondence: luis.cruz@uah.es; Tel.: +34-918-856-644

Received: 15 December 2017; Accepted: 31 January 2018; Published: 2 February 2018

**Abstract:** One of the biggest challenges in modern societies is to solve vehicular traffic problems. Sensor networks in traffic environments have contributed to improving the decision-making process of Intelligent Transportation Systems. However, one of the limiting factors for the effectiveness of these systems is in the deployment of sensors to provide accurate information about the traffic. Our proposal is using the centrality measurement of a graph as a base to locate the best locations for sensor installation in a traffic network. After integrating these sensors in a simulation scenario, we define a Multi-Agent Systems composed of three types of agents: traffic light management agents, traffic jam detection agents, and agents that control the traffic lights at an intersection. The ultimate goal of these Multi-Agent Systems is to improve the trip duration for vehicles in the network. To validate our solution, we have developed the needed elements for modelling the sensors and agents in the simulation environment. We have carried out experiments using the Simulation of Urban MObility (SUMO) traffic simulator and the Travel and Activity PATterns Simulation (TAPAS) Cologne traffic scenario. The obtained results show that our proposal allows to reduce the sensor network while still obtaining relevant information to have a global view of the environment. Finally, regarding the Multi-Agent Systems, we have carried out experiments that show that our proposal is able to improve other existing solutions such as conventional traffic light management systems (static or dynamic) in terms of reduction of vehicle trip duration and reduction of the message exchange overhead in the sensor network.

**Keywords:** sensor networks; optimized sensor deployment; multi-agents system; intelligent transportation system; smart cities; traffic simulations; traffic light management

## 1. Introduction

Intelligent Transportation Systems (ITS) is a concept that includes all the technological solutions and research proposals which, through an intelligent systems approach, aim to solve and improve the issues related with traffic management. The applications of ITS are very diverse, as this kind of system encompasses all of the elements in the road (from the infrastructure to vehicles and users).

This paper is an extended version of the work presented in JITEL 2017 [1], where we proposed the use of a Multi-Agent System (MAS) composed of multi-behavioral agents to optimize the management of traffic lights. This work led us to focus our research on obtaining an optimal set of sensor locations in the network used to provide data to the MAS. The results of this work contribute to this specific goal.

The sensors distributed across the traffic network generate the knowledge base in which the ITS relies on for the decision-making process. Traffic sensor technologies have noticeably improved in the last years, but, although there are some sensors not dependent on the roadway infrastructure (for instance, the Global Positioning System (GPS) installed in Smartphones), many of them still require an installation process and maintenance, which is traduced at a high cost. This cost limits the

number of sensors that can be deployed in these environments. Therefore, the correct selection of sensor location in a traffic sensor network is crucial for guaranteeing the efficiency and functionality of the network.

The intelligent management of traffic lights is a common method used to intervene in the events produced in these scenarios. One of the biggest challenges in the traffic light management systems is to be able to process all the information that is produced in their environment and make effective decisions using it. MAS can provide important advantages in solving this kind of challenge, as they offer the possibility of dividing the problem into smaller sub-problems, and therefore improve scalability and efficiency of the system. In Section 2, we show a selection of works in sensor locations in sensor networks, studies on the application of graph theory for traffic networks, and ITS based on MAS.

Our proposal can be divided into two main phases as shown in Figure 1. The first one, planification phase, is focused on defining a methodology to obtain a list of possible sensor locations ordered by their relevance. This list will be based on a study of the centrality (indicator of the importance of each node in a graph) for the graph determined by the traffic network and historical data of traffic demand (this phase is detailed in Section 3). Once the network is optimized by placing sensors in the more relevant locations, we define an MAS to manage the phases of the traffic lights in the network (management phase). In our proposed MAS, we define agents for intersection traffic light management and coordination between intersections (which is detailed in Section 4).

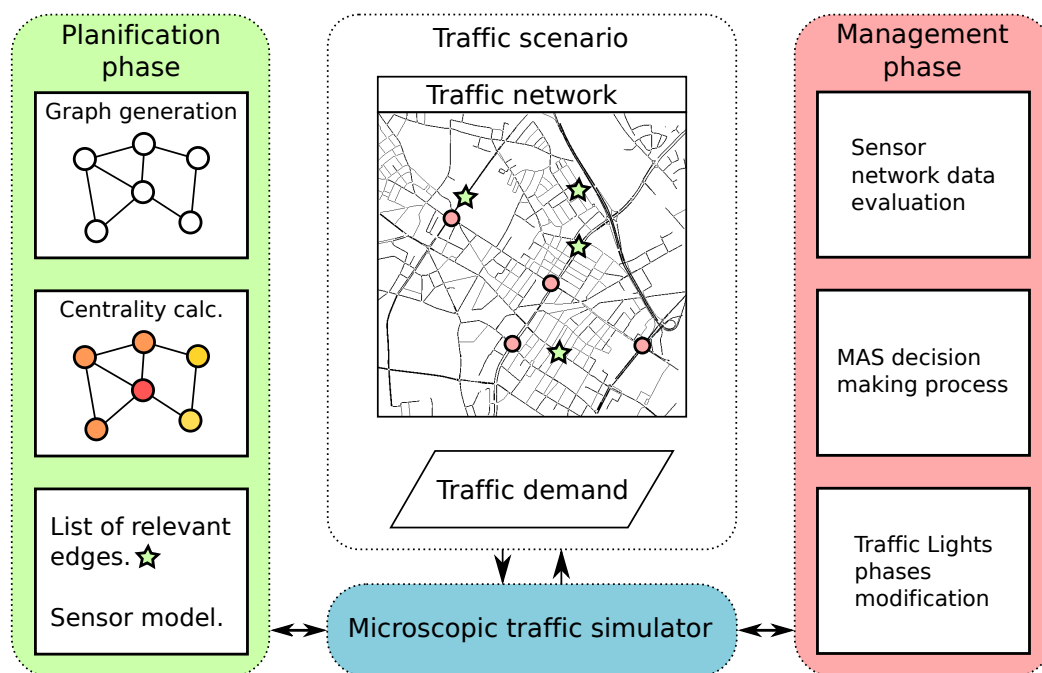


Figure 1. Overview of the proposed system.

Most of the current proposals (as shown in Section 2) for the improvement of sensor locations in sensor networks are based on using information from the network itself (for instance, historical data from traffic flows). One of the novelties of our method when compared with others is that we use indicators that are only based on the network topology (once it is transformed into a graph) for our proposed algorithm. These indicators allow us to obtain near-optimal locations, which can also be improved by using real traffic data.

The validation of the proposed solution is performed using a microscopic traffic simulator and a scenario based on real-life data from the city of Cologne in Germany (Section 5). Then, we discuss the obtained results both from the planification phase and the management phase of the proposal (Section 6).

## 2. Related Work

There are many factors involved in the correct circulation of vehicles. In fact, problems related to vehicle traffic are widely studied, as it is one of the most relevant challenges in modern societies. ITS are used with goals as diverse as improving toll collection [2,3], data collection from highways [4,5] and vehicles [6], emergency vehicles preemption [7], improving the transit signal priority [8] or other traffic management systems [9]. Using the data from the traffic sensor networks, ITS can offer solutions to solve many traffic related problems [10] such as traffic jams in certain zones or time slots.

In this section, we will comment the related work in sensor networks used in traffic scenarios, specifically those focused on sensor location, intelligent traffic light management, and the use of MAS in distributed traffic problems.

There are some works specifically focused on using sensor networks for road traffic management. In the paper in [11], the authors characterize and discuss probable topologies of these networks. The aim of the work is to provide network models that can be used to evaluate protocols and algorithms using realistic scenarios in place of generic random graphs. They deploy such networks over 52 city maps extracted from OpenStreetMap and characterize the resulting graphs, with focus on the connectivity aspects (degree distribution and connected components). The use of the sensor networks for measuring traffic volume, waiting time, vehicle density and other factors, which can be used as parameters for an algorithm to adapt the traffic light phases length is also approached in [12]. There is also a proposal for using sensor networks to feed an algorithm for flow control and congestion avoidance in traffic networks [13]. They improve the average waiting time and queue length both in isolated intersections and multiple intersections.

In [14], the authors present an ontology-based architecture to improve the driving environment through a traffic sensor network. The system performs different tasks in an automated way such as the traffic light management that includes the adjustment of the duration and intensity of the light, considering the traffic flow and the weather conditions. Another contribution in this topic can be found in [15], where they use fuzzy logic controllers to manage dynamically the traffic light phases, using also as input the information of a wireless sensor network.

In the literature, there is a large number of works related with the optimal location of sensors in Wireless Sensor Networks (WSNs). Many of these works are focused specifically on finding the best location of the sensors used in vehicular traffic networks. Some of these proposals are based on the use of Evolutionary Algorithms (EA) to obtain the optimal sensor locations. Kulkarni and Venayagamoorthy [16] show a study on the use of Particle Swarm Optimization (PSO) in WSN to optimize the number and location of sensors in terms of coverage, connectivity and energy efficiency. This kind of optimization technique is popular due to its simplicity, high quality of solutions, fast convergence, and insignificant computational burden. The drawbacks of using PSO are that they require large amounts of memory and that they might not be suitable for real-time applications. Based on using historical vehicular traffic data, Shan and Zhu [17] propose a methodology to locate camera sensors to estimate traffic in real time. As a data source, they use real GPS data from more than 8000 taxis. They transform the problem in a graph theory problem and then they propose a method based on a greedy algorithm to obtain an optimal result. This proposal obtains good results when compared with random sensor positioning or a positioning based on the most important roads in the network. The experiments lack validation through a simulation system, which could bring the solution closer to its real-life implementation.

Another approach to this problem is the one defined as the observability problem [18], which is based on determining the number of observed traffic flows and installing the required instruments to make it possible. The problem is classified in four sub-categories: link flow observability, origin-destination (O-D) flow observability, route flow observability and general case flow observability. Focusing on the first sub-category, which consists of determining which subset of link flows can be calculated in terms of another subset of links flows, there is a previous study by Hu et al. [19], where they work on the Network Sensor Location Problem (NSLP), which addresses the same challenges. NSLP has been used to determine the minimum number of sensor locations or the optimal locations for a given number of sensors, in order to estimate the O-D demand from historical data.

The methods we have described are focused on obtaining traffic flows in a network using information from a limited subset of sensors. Although the results are good, the complexity of the systems is high, and most of them require an a priori knowledge of the historical traffic flows (for instance, an estimation of O-D matrices). Moreover, the location of sensors that would benefit the construction of O-D matrices do not necessarily coincide with the areas in the network with a high traffic density. These areas are more interesting for monitoring to detect real-time traffic problems such as traffic jams.

Regarding the traffic light management, it is possible to find many works where EA are used to investigate the potential of these types of algorithms for the optimization of traffic light controllers. For instance, in [20], they use a Genetic Algorithm (GA), one of the most popular algorithms of this category. They conclude that, for solving these types of problems, it is useful to use evolution strategies. In [21], authors show the use of an iterative optimization algorithm (specifically a PSO algorithm) to find successful cycle programs of traffic lights. They validate their proposal using the Simulation of Urban MObility (SUMO) microscopic traffic simulator, obtaining an improvement in terms of total trip times and number of vehicles that arrive at their destination in a predefined simulation time.

Ref. [22] proposes a real-time adaptive lighting scheme, which detects the presence of vehicles and pedestrians and dynamically adjusts their brightness to the optimal level. This improves the energy efficiency of street lighting and its usefulness. The proposed scheme is simulated using an environment modelling a road network, users, and a networked communication system and considers a real streetlight topology from a residential area.

In ITS, the use of MAS is also common, given that the geographically distributed nature of the traffic elements, the dynamic environment, and the strong interaction between these elements can be easily approached using a distributed architecture such agent-based technologies [23]. For instance, in [24], there is a thorough review on agent-based technology for traffic, grouping the works in two main categories: modelling and simulation, and control and management. The paper in [25] also reviews works that use agent technologies to improve traffic related problems.

Some other works use the information generated by sensor networks to feed MAS focused on the traffic light management. In [26], the authors combine the information from sensors with an MAS to provide an adaptive management.

The work presented in [27] defines the generic elements that should compose the MAS for urban signal control, while Ref. [28] proposes a hierarchical multi-agent architecture that includes the internal description of the system. In both works, they demonstrate the viability of using MAS applied to traffic light management.

The ACTAM (Adaptive and Cooperative Traffic light Agent Model) system [29] aims to reduce the traffic congestion in urban roads by using a complete agent-based system that synchronizes and improves the efficiency of traffic management in cities. Unfortunately, it has only been validated in a small traffic network with about 30 intersections, which is far from a real-life scenario. Furthermore, the results are only compared with fixed-time signal control, and are not compared with other available management systems like actuated control traffic.

In Ref. [30], CARTESIUS, another architecture based on agents is proposed. The goal of this architecture is to provide a cooperative system to allow the collaboration between systems managed by different authorities to solve congestion problems. Analogously, Ref. [31] proposes an MAS based on mobile agents for traffic management and control, and, in [32], the authors propose using negotiation techniques between agents to optimize vehicular routes, based on the contents of a traffic data matrix.

### 3. Design and Optimization of a Sensor Network for ITS

In Section 2, we have shown the main current sensor architectures applied to ITS. In this section, we first show a classification of the sensors used in these environments and we define how we model them in a microscopic traffic simulator. Then, we propose a methodology for the optimal location of the sensors across the network. We aim to reduce as much as possible the number of messages exchanged by them, while maintaining enough information for a complete view of the scenario.

#### 3.1. ITS Sensor Classification and Simulation Models

ITS can improve by using information offered by different types of sensors. The sensors in charge of measuring physical parameters such as temperature, humidity, wind velocity, luminosity, etc. can aid the intelligent systems to be more efficient. In this section, we focus on those sensors that can obtain information from the traffic network itself, considering the network and the vehicles using it.

A first classification can be made between sensors that offer direct traffic information: it is possible to differentiate between those that require the installation of equipment in the vehicles (e.g., GPS) and those that can be considered autonomous [33]. In addition, these autonomous sensors can be classified in those that are embedded in the pavement of the roadway and those that are embedded above or alongside the roadway [34].

In Table 1, we are showing a selection of these sensors.

**Table 1.** Intelligent Transportation Systems' sensor classification.

Sensors equipped in vehicles	Global navigation Satellite System (GNSS)	GPS and Galileo are the most well-known systems of this type. They provide information about the position and current velocity of the vehicles.
Autonomous sensors	In-roadway sensors	Senses the presence of a conductive metal object by inducing currents in the object, which reduce the loop inductance. Magnetic sensors: Similar to the inductive-loop detectors, are able to sense the presence of vehicles using the perturbation they cause in the magnetic field.
	Over-roadway sensors	Microwave or laser radar sensors: Detect vehicles by transmitting microwave or laser signals and receiving the echoes from them.
		Infrared sensors: Detect vehicles by receiving the energy emitted by roadways and vehicles or energy reflected from them. Cameras: It is possible to detect vehicles by the processing of the images taken by one or more video cameras.

Modern traffic simulators feature the possibility of using some of these sensors. It is important that these sensors behave as closely as possible to their real-life counterparts. The data output of SUMO can offer information about what has happened in each simulation from the vehicle or edge point of view. Therefore, it is possible to model the sensor behavior using these features. This information can be analyzed once the simulation has ended or during the simulation time. The communication between both modules is performed using the Traffic Control Interface (TraCI) tool [35], included in the SUMO package. This tool provides a Transmission Control Protocol (TCP) based client/server architecture that allows to control and modify the SUMO simulations through an external application.



In Figure 2, we show the communications between the sensor modelling process and the traffic simulator.

The main steps followed during system execution are:

1. System initialization: The application reads the data from the scenario, initializes the sensor configuration, including the definition of its features, and launches a subprocess that starts the SUMO simulator.
2. Loop until end of the simulation: There is a parameter in the application configuration that states the duration in seconds of each simulation step. The sensor simulation module will perform the following tasks including requests to the simulation module (via TraCI):
  - (a) Requests one step simulation and then waits for the end of the step,
  - (b) Requests the current parameters of each vehicle that was active in that step of simulation and the edge information,
  - (c) Processes the information, and models it for the specific sensors,
  - (d) Returns the values generated for the sensor,
3. End of the simulation.

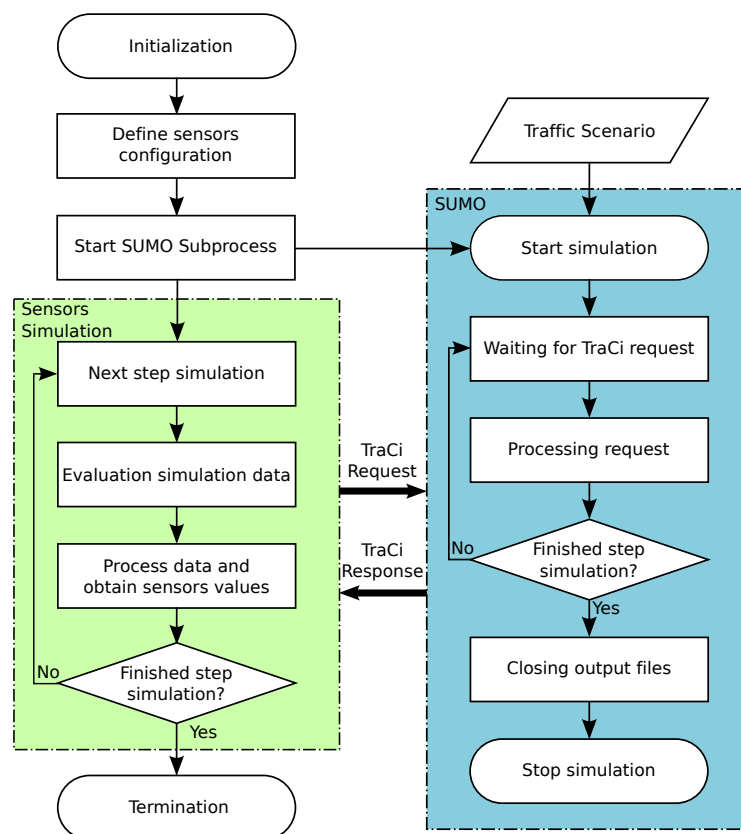


Figure 2. Block diagram of the sensor modelling and simulation platform.

Depending on the specific traffic sensor, the modelling process can be complex. For instance, GPS sensor modelling can be performed by obtaining the relative position and instant velocity value from the simulation data. This value can be modified by applying a correction factor to add the possible error introduced by GPS real systems accuracy (e.g., in GPS-enabled smartphones the typical accuracy is within a 4.9 m radius under open sky [36]).

Other sensors, such as video cameras, need a previous study to define its characteristics. To carry out this study, we need the camera installation height, its focal length, and its inclination angle, among other parameters. Typical values for a camera installed in a roadway could be 8 mm of focal length and 12.2 m of height. Using these values, we could obtain a minimum distance value of 15.4 m and a maximum of 139.4 m, and values from 15.7 m to 112 m in width [33]. According to this, we can model the simulated camera by obtaining the number of vehicles that are located within the range of vision of the camera starting from the camera position (in the example above, a maximum of 124 m from the camera, and with a width of 15.7 m). As with GPS simulated sensors, a correction factor can be added to model the possible camera errors.

### 3.2. Sensor Location Proposal

The vehicle embedded sensors depend on the technology available in each vehicle and the driver's preferences. Although the price of the sensor is lowering for some of them, it is still a high cost, and this is especially true for the infrastructure needed for some of them. This means that it is an essential issue to position the sensors wisely across the network.

In a traffic scenario, we must be able to obtain a global view of what is happening at each time. Depending on the type of sensor, we can get data that is related (e.g., traffic density data in some place across the network and the weather conditions of that same place).

In a traffic network, there are certain spots that will be revealed as especially relevant, as they will be important in a later analysis that will feed the ITS and help in the decision-making process. Detecting those spots will allow us to position in them autonomous sensors. These sensors will offer accurate and constant information about their surroundings, without depending on the information from vehicle embedded sensors, which will lead to a global improvement in the data quality.

These relevant spots can be calculated using historical data of the vehicular traffic flows, as we can select those where the highest number of coincident traffic flows is shown. Unfortunately, the complexity of obtaining a reliable traffic model makes necessary to design methodologies to obtain the spots using different parameters.

Our proposal is abstracting the traffic network as a directed graph, and then study its centrality. Through this study, it is possible to obtain a ranking of the most relevant edges. Then, we can combine this ranking with a ranking obtained from traffic historical data, and then obtain a list of spots where the sensors would provide the most useful information.

The volume of vehicles in an edge depends on the traffic demand of the edge and on the ability of the edge to be a connector based on its location in the road, that is, in its betweenness centrality. There are studies that establish a relationship between the traffic flows and the betweenness centrality values of a graph [37]. This centrality of an edge is "proportional to the number of shortest paths between all pairs of nodes passing through it" [38] and can be measured by averaging over each pair of nodes and following the shortest path to the destination. Consequently, we have chosen the betweenness centrality to obtain the ranking of relevant edges in the network. In Equation (1), we show the betweenness centrality calculation for one node, as proposed in [39]:

$$c_B(v_i) = \sum_{s,t \in V} \frac{\sigma(s,t|v_i)}{\sigma(s,t)}, \quad (1)$$

where  $V$  the set of nodes from a directed graph,  $\sigma(s,t)$  is the number of shortest paths between  $s$  (source node) and  $t$  (target node), and  $\sigma(s,t|v_i)$  is the number of those paths passing through a node  $v_i$  different from  $s$  and  $t$ .

The methodology we propose is based on calculating the following data structures:

1.  $L_c$ : List of edges ordered by their centrality value. Obtained following the next steps:
  - (a) Transformation of the traffic network in a directed graph ( $G$ ):

$$G(V, E), \quad (2)$$

where  $V$  is the set of network nodes and  $E$  is the set of network edges.

- (b) Calculation of the betweenness centrality of  $G$ . There are implementations that allow to calculate the betweenness centrality of the edges of the graph [40]. Another option would be to transform the directed graph  $G$  into a Line-graph ( $G_{LG}$ ), in which the nodes are constituted by the edges of  $G$  and the edges are the possible turns from one edge to other in the original graph [41]:

$$G_{LG}(V_L, E_L), \quad (3)$$

$$V_L = E, \quad (4)$$

where  $V_L$  is  $E$  from  $G$  and  $E_L$  is the set composed by the possible turns from one edge to other in  $G$ . We obtain the centrality measurement of  $G_{LG}$  in the set  $S_c$ :

$$S_c = \bigcup_{i=0}^n \{(v_{L_i}, c_B(v_{L_i}))\}. \quad (5)$$

Applying Equation (4) in Equation (5):

$$S_c = \bigcup_{i=0}^n \{(e_i, c_B(v_{L_i}))\}, \quad (6)$$

where  $e_i$  is each of the  $n$  elements in  $E$  (set of edges of the original graph  $G$ ) and  $c_B(v_{L_i})$  is the centrality value of  $e_i$ .

- (c)  $L_c$  is obtained from the ordination of the set  $S_c$  from the highest to the lowest centrality value, and the normalization of the centrality values over 1.
2.  $L_d$ : List of edges ordered by their historical occupation level. Obtained following the next steps:
- (a) Obtaining of the historical traffic values. These values might be already grouped or might be a traffic model that should be simulated to obtain numerical values. We define  $S_d$  as:

$$S_d = \bigcup_{i=0}^n \{(e_i, v_{d_i})\}, \quad (7)$$

where  $e_i$  is each of the  $n$  elements in  $E$  and  $v_{d_i}$  is the historical occupation value for  $e_i$ .

- (b)  $L_d$  is obtained from the ordination of the set  $S_d$  from the highest to the lowest historical occupation value, and the normalization of the historical occupation values over 1.
3.  $L_p$ : List obtained using the combination of  $L_c$  and  $L_d$ , and using a reliability factor for the historical values ( $\alpha$ ):

$$L_p = (1 - \alpha)L_c + \alpha L_d. \quad (8)$$

- (a)  $\alpha$  is a value between 0 and 1 that measures the reliability level of the historical data, where 1 would be the closest possible value to the real-life one.
- (b) Ordination of  $L_p$  from the highest to the lowest value.

Once this process is finished,  $L_p$  offers an ordered list of the most relevant edges in the network, where we can position the sensors to obtain better and more reliable information.

In Algorithm 1, we show the pseudo-code of the calculation of the  $L_p$  list to provide a more detailed view of this process. The algorithm takes a graph  $G(V, E)$  as input, and returns the  $L_p$  list as defined above. The first part of the code generates the line-graph ( $G_{LG}(V_L, E_L)$ ) used for the calculation

of the centrality values. The nodes of this graph ( $V_L$ ) are the edges of  $G$  (line 5), and the edge set ( $E_L$ ) is composed by the list of possible turns from one edge to the other in  $G$ , as it is shown in lines 6 to 11.

Once the line-graph is generated, it is used for the calculation of the  $L_c$  list, which is composed of tuples of edges and betweenness centrality values for those same edges ( $e_i, c_B(v_{L_i})$ ). For this calculation, we use an implementation of the betweenness centrality, which, in our pseudo-code, is represented by the function *calculate\_centrality* (line 13). The result set returned by the centrality function is ordered by the centrality values and normalized in the list  $L_c$  (lines 14 and 15).

Additionally, we calculate the  $L_d$  list, which is the result of formatting the historical data  $D$  into a set of tuples of edges and values ( $e_i, v_{d_i}$ ), and its ordination by the values  $v_{d_i}$  and normalization (lines 17 to 19).

Finally, for each edge, we calculate a combined value composed of the sum of the corresponding centrality value ( $c_B(v_{L_i})$ ) and historical value ( $v_{d_i}$ ) using the reliability factor  $\alpha$  to adjust which value will weigh more. The resultant set is ordered by this combined value, returning the desired  $L_p$  list. This is shown in lines 21 to 25.

---

**Algorithm 1:** Sensor location proposal pseudo-code.

---

```

1 Input:  $G(V, E)$ 
2 Output:  $L_p$ 
   Data:  $G_{LG}(V_L, E_L)$  //line-graph
            $turns$  //list of edges
            $S_c$  //unsorted set of edge and centrality value tuples
            $L_c$  //sorted list of edge and centrality value tuples
            $D$  //historical data
3            $S_d$  //unsorted set of edge and historical value tuples
            $L_d$  //sorted list of edge and historical value tuples
            $v$  //each value of combined centrality and historical data
            $\alpha$  //reliability factor
            $S_p$  //unsorted set of edge and combined value tuples
4 //line graph generation
5  $V_L \leftarrow E$ 
6 for  $e_i$  in  $E$  do
7   |  $turns \leftarrow \text{get\_possible\_turn\_edges\_for\_edge}(e_i)$ 
8   | for  $edge$  in  $turns$  do
9   | |  $\text{append}(e_i, edge) \rightarrow E_L$ 
10  | end
11 end
12 //  $L_c$  calculation
13  $S_c \leftarrow \text{calculate\_centrality}(G_{LG})$  //returns a list of tuples ( $e_i, c_B(v_{L_i})$ )
14  $L_c \leftarrow \text{order } S_c \text{ by } c_B(v_{L_i})$ 
15  $\text{normalize}(L_c)$ 
16 //  $L_d$  calculation
17  $S_d \leftarrow \text{get\_tuples\_from\_data}(D)$  //returns a list of tuples ( $e_i, v_{d_i}$ )
18  $L_d \leftarrow \text{order } S_d \text{ by } v_{d_i}$ 
19  $\text{normalize}(L_d)$ 
20 //  $L_p$  calculation
21 for  $e_i$  in  $L_c$  do
22 |  $v \leftarrow (1 - \alpha)c_B(v_{L_i}) + \alpha v_{d_i}$ 
23 |  $\text{append}(e_i, v) \rightarrow S_p$ 
24 end
25  $L_p \leftarrow \text{order } S_p \text{ by } v$ 

```

---

#### 4. Multi-Agent System for Intelligent Traffic Light Management

Once we have an optimized sensor network offering us a reliable data source from the traffic scenario, we are using it to feed an MAS for traffic light management. This system's goal is to reduce the duration of vehicle trips by modifying the intervals of the traffic lights (phases) in a traffic network.

In the following sections, we are going to describe the types of agents involved in the system and the general operation of the system itself.

##### 4.1. Agents

In this section, we describe each one of the three types of agents involved in the system. For each agent, we include its description, where it is located in a traffic scenario, and its behavior in the system.

##### 4.1.1. "TLAgent" Traffic Light Management Agent

- **Description:** This kind of agent is defined for the traffic light phases' changing management. It will be aware of the other agents of the same type in the intersection.
- **Location:** Each traffic light in each intersection of the network will be provided with one of these agents.
- **Behavior:** In an initial state, the traffic light changes will follow a static predefined pattern. Once the agent receives a message indicating a change, it will reconfigure the phases accordingly. The changes will always be performed within a given maximum and minimum time thresholds.

##### 4.1.2. "TJamAgent" Traffic Jam Detection Agent

- **Description:** These agents are in charge of obtaining data from their environment and then making a decision about the traffic state, with the ultimate goal of communicating a possible traffic jam in their surroundings.
- **Location:** These agents are deployed across the sensor network. They can be located both in sensors installed in the network or aboard sensors embedded in the vehicles that transit through it.
- **Behavior:** Agents located in vehicles will base their decision on the data provided by sensors (specifically, the geographic position and instantaneous velocity of the vehicle). A threshold will be defined for this data, and once the sensors provide higher values than expected for a period, the agent will consider a traffic jam situation and notify it. The sensors installed in the network will make the decision using the same mechanism, but relying on different types of data depending on the type of sensor installed.

##### 4.1.3. "IntersectionAgent"

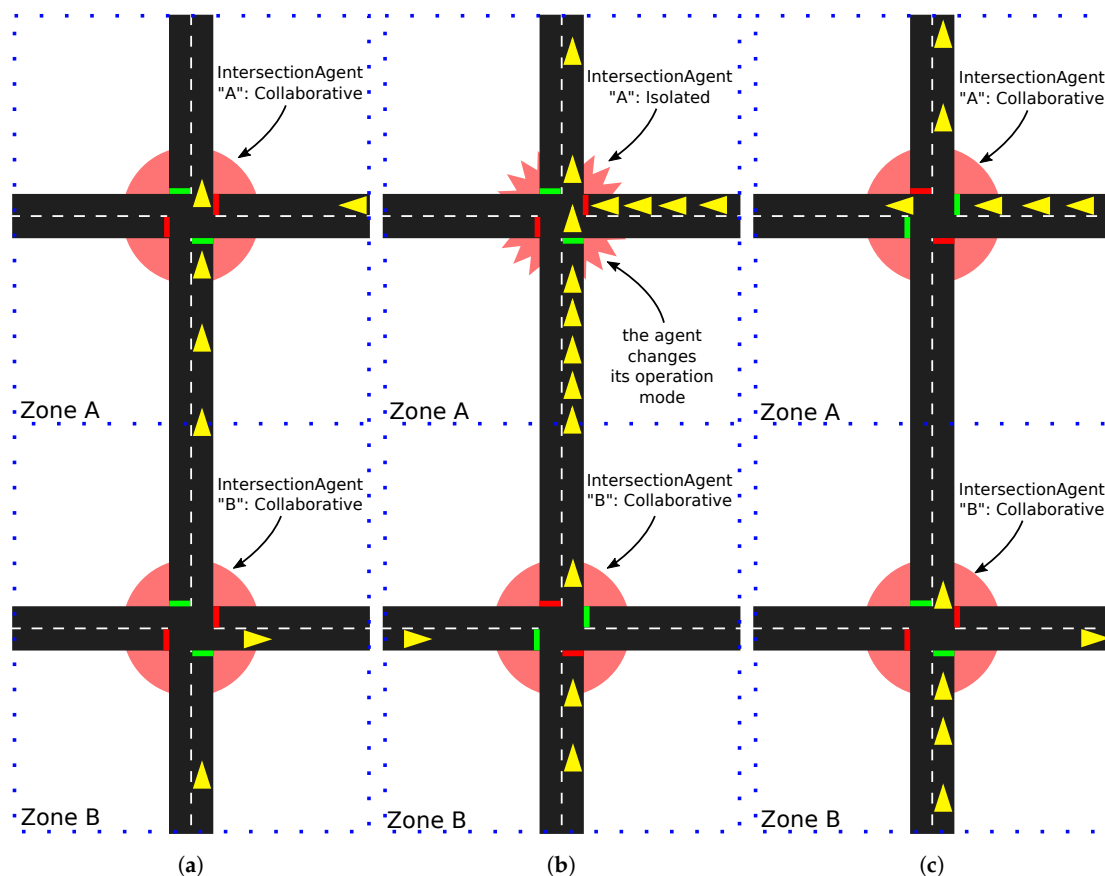
- **Description:** These agents are the ones that decide how the traffic lights of an intersection should be changed. They receive messages from the TLAgents near them, and then decide which traffic flow in the intersection should be prioritized.
- **Location:** They can be located anywhere within the communications coverage of the TLAgents of each intersection.
- **Behavior:** Using the current state of the TLAgents they control (the specific number of agents will depend on the number of lanes ending in the intersection) and information from TJamAgents, they will perform a decision-making process based on a state machine. The decisions will be sent again to the TLAgents. These agents are also communicating between them and they can operate in two modes regarding these communications:
  - **Altruistic or Collaborative:** This mode allow agents to listen to requests from other agents of the same type. The information given by the other IntersectionAgents will be added to the information obtained from the TJamAgents to make a more informed decision on how to prioritize flows.

- **Selfish or Isolated:** The agents enter this mode when they determine that the zone that they are managing is much too congested and therefore they should make their own decisions instead of pursuing a global cooperative goal. Once in this mode, the agents will ignore any message from other IntersectionAgents until the congestion is lower or a defined timer expires. This timer is set to avoid system blocking if the agents stay too much time in this mode.

#### 4.2. MAS Behaviour

From the agent definition shown in Section 4.1, the MAS is able to dynamically manage the traffic lights phases from a traffic scenario. The MAS is able to modify its general behavior to react to traffic jam situations, which can result in the blocking of vehicles for extended periods of time.

In this section, we are going to propose an example use case composed by a network with two intersections. In this scenario, we are forcing a congestion situation and then show how the agents change their behavior to solve the issue. The different states of the scenario during the system operation are shown in Figure 3. The three types of agents are represented in the figure using red or green lines (TLAgents), yellow triangles (TJamAgents in vehicles) and red areas (IntersectionAgents). In addition, we consider all traffic flows going from the left or the right in the figures are “horizontal flows”, and all traffic going from the bottom or the top as “vertical flows”.



**Figure 3.** Multi-Agent System operation use case. (a) initial state; (b) zone “A” congested; (c) zone “A” normal flow.

1. **Initial state** (Figure 3a): Both IntersectionAgents (“A” and “B”) will begin their operations in *altruistic* mode. This means that they will exchange information to help each other.

The vehicles in the vertical traffic flow inside the zone “A” circulate at a speed below the optimal in that road. Therefore, the TJamAgent included in each one, will report a possible traffic jam situation to IntersectionAgent of the zone “A”.

Once the IntersectionAgent collects the information from the TJamAgents in its zone, it starts the decision-making process and finally decides to prioritize the vertical traffic flow. The actions performed by the agent are: sending the new phases configuration to the TLAges in the intersection, and advertising the changes to the nearby IntersectionAgents (in this case, the agent in zone “B”) to maintain the synchronization of the adjacent intersections.

2. **Zone “A” Congested** (Figure 3b): Once an IntersectionAgent determines it is in a blocking situation due to a very high traffic volume, it changes its behavior by entering *selfish* mode. In this example, the IntersectionAgent in zone “A” determines that all the vehicles that are entering the intersection are blocked. Because of this, it enters *selfish* mode and stops using the information received from the nearby IntersectionAgents (in this case, the agent in zone “B”).

The actions performed by the agent are: It requests the IntersectionAgent in “B” to limit the vertical traffic flow (the IntersectionAgent in “B” will acknowledge this request because it is in *altruistic* mode yet), and then it sends the new phases configuration to the TLAges in the intersection.

3. **Zone “A” normal flow** (Figure 3c): The decisions taken by the IntersectionAgent in the previous case have been successful, and, therefore, the congestion situation is over. This allows the IntersectionAgent in zone “A” to change back its behavior to *altruistic* mode. Now, the vertical traffic flow in the zone “B” is higher than before. This is because of the traffic light phase duration changes performed in “B” to help the IntersectionAgent in “A”. Once the blocking state is solved, the phases duration can be reverted to the best possible values for intersection “B”.

## 5. Results

In this section, we are going to define the experiments carried out for evaluating the proposals of Sections 3 and 4.

Due to the high complexity of validating this kind of proposals in real-life scenarios, we have chosen to use a microscopic traffic simulator (SUMO [42]). This type of tool allows us to reproduce a scenario with conditions close to those expected from real scenarios, but with a very low cost.

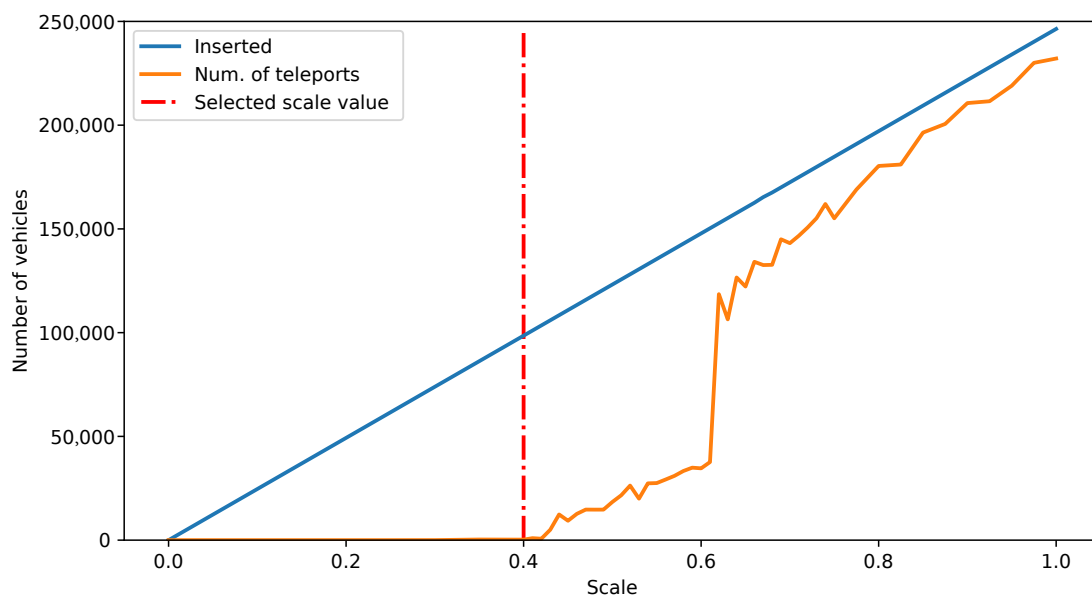
### 5.1. Simulation Scenario

To carry out experiments, we have chosen a realistic simulation model, as it will provide better and more reliable results. Specifically, we have chosen the scenario called “Travel and Activity PAtterns Simulation (TAPAS) Cologne” [43,44] that is referenced in the SUMO documentation. It is a complete simulation scenario of the German city of Cologne. It was created by the Institute of Transportation Systems at the German Aerospace Center (TIS-DLR), and its goal is to reproduce, with the maximum possible realism, the urban traffic of Cologne. It defines a map of 400 km<sup>2</sup> and 24 h of traffic.

The original simulation scenario is composed of a road network with 68,642 edges, 30,354 nodes and 1,547,333 routes, which causes very large simulation times. This is very inconvenient for the experiments that we have carried out, and, therefore, we have decided to crop the scenario in a smaller portion, still representative of the original one, but presenting lower simulation times. To obtain this sub-scenario, we have analyzed the features of the network used in [45], where they test a simulated network of the Lower Downtown Toronto.

Our sub-scenario is composed by 1416 edges and 716 nodes (from which 73 are intersections managed by traffic lights, and the rest are managed by priority rules). The routes modelling the traffic of the scenario have also been reduced to a more manageable number. We have selected the routes that are related to the cropped sub-scenario map, 246,374 in total.

According to the “TAPAS Cologne” scenario documentation, the traffic inserted in the simulation should be scaled for the model to work correctly. This can be configured by setting the parameter *scale*, and the recommended value is 0.3. In our case, as we are working with a sub-scenario of the full “TAPAS Cologne”, we must recalculate this recommended value. We have carried out a set of simulations while increasing the *scale* parameter by 0.01 until we start seeing teleported vehicles in the simulation (teleporting is an event defined by SUMO, in which vehicles are taken out of the simulation and inserted again later to avoid the simulation blocking due to a too high traffic rate [46]). Figure 4 show the results of the *scale* value calculation.



**Figure 4.** Evolution of the teleports number according to the scale parameter.

The results of these simulations have led us to decide to use a 0.4 value for the *scale* parameter. This is the value where the number of teleported vehicles starts rising, as Figure 4 shows. The specific value for teleports is 241 for that scale, which is still a reasonable low value and allows us to insert the highest possible volume of traffic without compromising the realism of the simulation results. Scaling the traffic volume by 0.4 means that the actual number of vehicles and routes inserted will be of 98,550.

Once the sub-scenario has been selected, we have integrated it into the MAS. In Figure 5 we show the roads of the sub-scenario (in black lines) and the IntersectionAgents (red dots). The size of the dots indicates the number of TLAGents managed by each IntersectionAgent.





**Figure 5.** Representation of network edges (black lines) and intersection agents (red dots).

### 5.2. Sensor Network Optimization Experiments

We have defined a set of SUMO simulations to evaluate the proposal of the Section 3.2. As a traffic scenario, we have used the sub-scenario defined in Section 5.1. These simulations are carried out to determine the maximum value of data that can be obtained by inserting  $N$  camera sensors in the network, and then compare it with the values obtained when we place the cameras in the edges indicated by the centrality lists and with the proposed location system. The definition of experiments we have carried out is based on the proposal in [17]. In this work, the authors compare their proposal with two location methods called Random Road Location Method (RRLM) and Arterial Road Location Method (ARLM). RRLM consists of selecting random locations in the network and repeating the experiments 10 times, using the mean values obtained. In ARLM, sensors are located in the roads depending on the priority degree of those roads. For roads with the same priority, the sensor location is also random, following the RRLM process. In our case, we only include the random selection of locations, as our scenario is a realistic city network where most of the streets share similar features (maximum velocity, length, etc.), which leads the ARLM system to become an RRLM in practice. Following this scheme, we define four experiments:

- **Max. value (Simulation data):** A simulation is carried out, and the position of every vehicle is stored in each simulation step (setting simulation step to 1s). If an edge is longer than 120 m, the edge is divided in sub-edges, so the limit of the camera vision is not surpassed. We use a counter for each edge (or sub-edge), and it is incremented each time there is a vehicle located in it. At the end of the simulation, we get a list of edges ordered by the occupation level.
- **Random locations:** In this experiment, we have simulated the results of applying a random location selection for sensors. We have selected from 1 to 32 random locations, and then we have obtained the results of the simulation. This process has been repeated 10 times, choosing new random locations each time. Once finished, the resultant values are the result of calculating the average value for the 10 repetitions.

- **Betweenness centrality:** Using the SUMO libraries, the network is loaded in the computer memory and then transformed into a graph using the NetworkX library. This directed graph is then transformed into a line-graph and then the “betweenness centrality” is applied. Using the resulting list, we carry out a set of simulations where we place from 1 to 32 cameras in the places indicated by the first positions of the list. Then, we store the number of vehicles that were counted by the cameras.
- **Proposed system:** The process followed in order to obtain the betweenness centrality list is repeated. Then, we create a list with historical traffic data (we use the results of a previous simulation) and then we generate a list that combines both using an  $\alpha$  factor equal to 0.5.

In Figure 6, the obtained results are shown after carrying out the four experiments. The  $x$ -axis represents the increment on the number of cameras used in the network, while the  $y$ -axis represents the number of values measured for each camera in each experiment. The red intermittent line shows the maximum theoretical value in case the cameras are on the edges with the highest occupation.

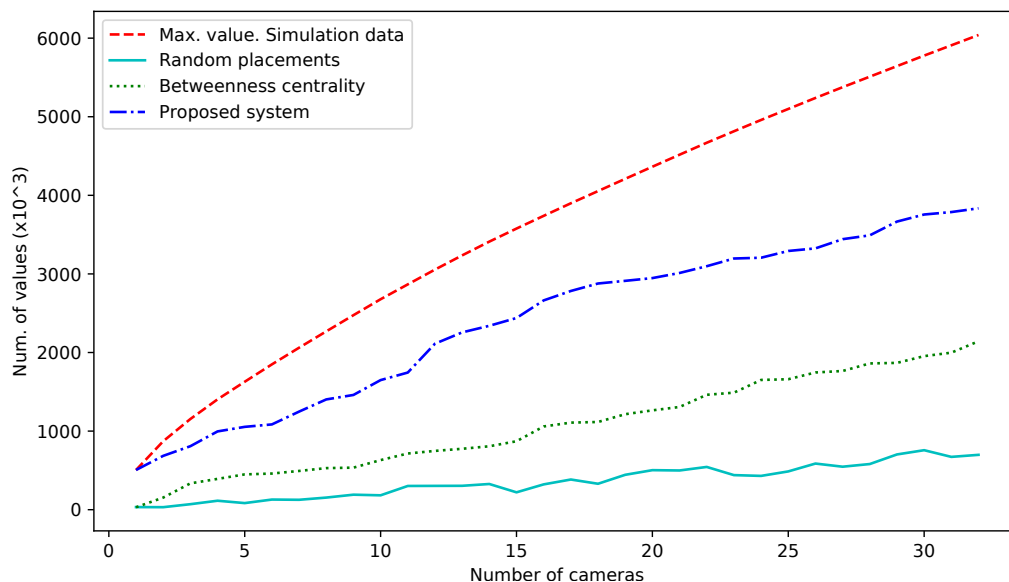


Figure 6. Number of values counted by camera sensors.

### 5.3. MAS Traffic Light Management Experiments

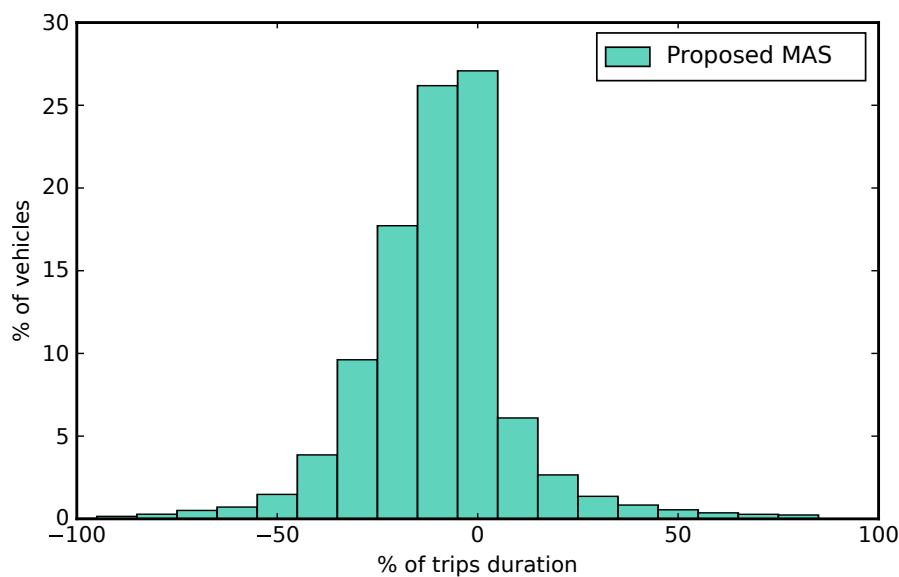
The experiments shown in this section have as main goals to measure the improvement of the system in terms of trip duration through the intelligent management of traffic lights, and to determine the number of messages generated by the sensors to feed each possible solution, in order to assess the improvement in each proposal (that is, using only the MAS or using the MAS with the optimized sensor network).

Using the set of traffic data shown in Section 5.1, we have defined and carried out a set of experiments to measure the trip duration of each vehicle and the number of messages generated by the sensors. We have used different system configurations in each case:

- **Static phases:** In this experiment, we use the default TAPAS scenario configuration. The TAPAS traffic network defines a set of static phases for traffic lights in each one of its intersections. These phases do not change during the simulations. There are no sensors deployed in the network.

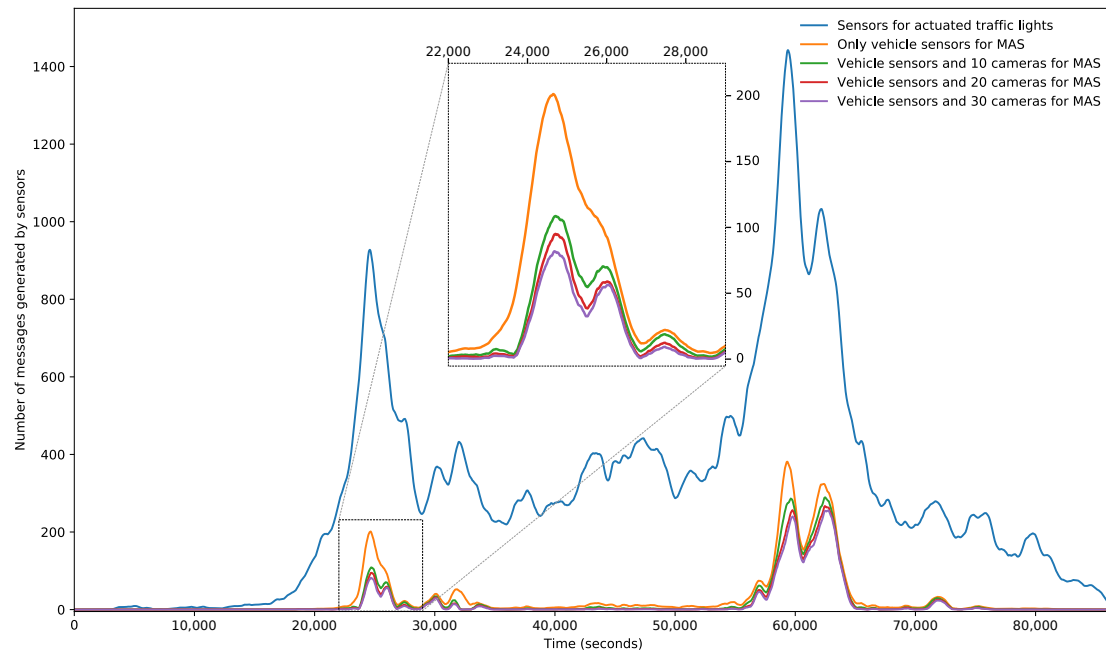
- **Actuated traffic lights:** This kind of traffic lights management is based on the location of sensors in a fixed distance to intersections. The traffic lights phases change depending on the information received from these sensors. We use in this experiment the SUMO traffic simulator actuated traffic lights feature, included among the network configuration possibilities. In this case, there are induction loop sensors deployed in the network at a fixed distance from each intersection controlled by traffic lights. These sensors send a message each time a vehicle activates them by passing over.
- **Proposed MAS:** We integrate the proposed MAS into SUMO to change the traffic lights phases dynamically during the simulation. When a camera sensor is deployed, it monitors the vehicles in its coverage area. If traffic jams are detected in that area, it sends a message. In addition, every vehicle in the system emulates a GPS sensor. If the velocity of the vehicle is less than half of the minimum between its maximum velocity and the maximum velocity of the road, it sends a message, notifying the situation. Note that, when the vehicle is in the coverage area of a camera sensor, it will not notify the situation, relying on the camera information.

In Figure 7, we show the results of comparing the “Static phases” experiment with the “Proposed MAS” experiment. The  $x$ -axis represents the percentage of increase or decrease in the trip duration (using steps of 10%) and the  $y$ -axis represents the percentage of vehicles inserted over the total vehicles in the simulation.



**Figure 7.** Percentage of increase or decrease in the trip duration over the percentage of vehicles.

The results of counting the number of messages generated by the sensor network in each experiment are shown in Figure 8. The  $x$ -axis represents the total simulation time (between 0 and 86,400 s: 24 h) and the  $y$ -axis shows the total number of messages generated in the system in each instant of time. To facilitate the visibility of the figure, we have zoomed the results between 22,000 and 29,000 s.



**Figure 8.** Number of messages generated by the sensor network during the simulation time.

## 6. Discussion

We are starting the section commenting on the results obtained in Section 5.2. Those experiments had as a goal to verify the viability of the methodology used for the selection of the sensor location in the traffic network and measure the obtained results. In Figure 6, we show the values obtained when we insert into the simulation from 1 to 32 camera sensors. These sensors can count vehicles inside of his vision range (a maximum of 120 m in front of the camera location).

We can see how, from the results from using the ordered list based in the network centrality (dotted green line), we obtain a curve whose slope is lower than the slope from the curve representing the theoretical maximum value (intermittent red line) but with a constant increment.

The most important comparison between the experimental results is the one with the random road location method (RRLM), which follows the methodology proposed in [17]. Comparing the results of our proposal with the results obtained in the RRLM experiment, the improvement of using a centrality-based selection versus a random selection is shown. In addition, the results of the centrality experiment can be considered favorable, given that only the network topology has been used in this case to decide the location of the cameras. We can conclude that these results show how the use of centrality calculations is useful for this kind of environment.

The list obtained with our proposed methodology offers better results than the list obtained only using the centrality calculation, and also present an improvement margin by adjusting the correction factor  $\alpha$ , which can be used to adapt the results depending on the reliability of the historical data used.

Once the sensor network has been defined and the sensors have been distributed along it for the support of the MAS, we have carried out the experiments described in Section 5.3.

We have measured the trip duration for each vehicle, and then we have compared the times obtained using the dynamic traffic lights management systems (that is, the actuated traffic lights and our proposal) with the trip duration obtained when using the static traffic light phases. The results of the comparison can be summarized by measuring the number of vehicles whose trip duration is reduced when compared with a static traffic lights scenario, the number of vehicles that do not see their trip duration modified, and, finally, the number of vehicles that show an increase in trip duration.

These measurements can be performed both for the actuated traffic lights system and our proposal, and they are shown in Table 2 in percentages.

**Table 2.** Results summary (Percentage of total vehicles).

Trip duration	Lower	Equal	Higher
Actuated Traffic Lights	58.70	28.41	12.89
Proposed Multi-Agent System	60.52	27.08	12.41

The table shows that not every vehicle has improved its trip duration when compared with the static traffic lights. The explanation for this can be found in the behavior of the dynamic systems, where some traffic flows will always be prioritized over others, causing an increase in waiting time for some vehicles. Nevertheless, the number of vehicles suffering for this increment is low (below 13% in both cases), and the time increase for half of those vehicles is below 10%. Therefore, it is possible to say that the dynamic change of traffic light phases offers a global improvement to the scenario in terms of trip duration.

If we compare the results between the actuated traffic lights and our proposal, it is true that the improvement of using the MAS might not seem too remarkable to justify using one system over the other, but there are some considerations that would indicate that our proposal is still better than the other one: first, the actuated traffic lights system depends on installing an induction loop in every edge of the traffic network that is attached to a traffic light managed intersection, while our proposal depends only on the optimized location of sensors across the network, which would noticeably decrease the overall infrastructure cost. Finally, the use of MAS provides a collaborative platform for the achievement of global goals across a traffic network, which cannot be obtained from isolated intersection management of actuated traffic lights.

From the generated messages volume point of view, and, using the results from Figure 8, we can obtain the following conclusions. First, we can observe how, around the time intervals centered in 25,000 and 60,000 s, there is a very noticeable increment of number of exchanged messages. These intervals of time correspond with peaks in the volume of traffic flows in the simulation scenario. As in the actuated traffic lights system (blue line in the figure), there is one message of each vehicle that passes over the sensors (induction loops), and the system generates a high volume of messages during all the simulation period, independently of the real traffic status (this system is not able to distinguish between congestion and not congestion states). On the other hand, we can see that the peaks of number of messages generated in the experiments based in the MAS (every other line in the figure) are centered only around the areas with peaks of traffic flows.

Regarding the possible improvements of using an optimized sensor network additionally to the MAS, we can observe in the zoomed section of Figure 8 how using the MAS without an optimized network (orange line) shows the maximum number of messages generated. When the experiment is repeated using an optimized network with 10 camera sensors positioned by following the proposed methodology (green line), it is possible to perceive a significant reduction in the generated messages quantity (specifically, there is a reduction of 47.4% or 216,860 messages less over the previous results). When more sensors are located in the network (red and purple lines), the number of messages exchanged is still being reduced, but with a lower effect. This result was predictable, as the first cameras are already positioned in the more relevant positions of the network. As a conclusion, the amount of data exchanged in the network needed for the decision-making process will be lower in the MAS, which will also reduce the communications overhead.

One of the most difficult situations to solve in traffic congestion states is the blocking of some vehicles in the intersections, where they stay stopped for very long periods of time. This kind of event has been reproduced during our experiments, and we have observed that the MAS is able to react specifically to avoid them. If we compare the results between experiments, we can conclude that 1790 vehicles have been “unlocked” by the MAS, and, consequently, their trip duration has been lowered. The reaction of agents to these blocking situations can also be considered an important advantage of MAS over actuated traffic light systems.

The proposal of this paper was to optimize a sensor network through the study of the traffic network topology and then use the information provided by this network to intelligently manage the traffic lights using an MAS.

After analyzing the results, we can conclude that the election of the locations of the sensors using centrality measurements from a graph provides useful and reliable information to be used in ITS applications. When compared with some of the solutions shown in Section 2 related to the positioning of sensors in vehicular traffic networks, one of the most relevant advantages of using our proposal is the low dependency from having historical data of traffic flows for it to work. The study based on the centrality of the network has shown that, using only the network topology, it is possible to find near-optimal sensor positions, where the sensors are able to provide a high volume of relevant information, while reducing the number of necessary message exchanges in the MAS.

Furthermore, when comparing the MAS results with other solutions like the static definition of traffic light scheduling or the dynamic management of traffic lights phases (actuated traffic lights system), we show that the use of our MAS is effective for the reduction of trip duration in vehicles.

The goals defined in Section 1 have been covered in the following way: demonstrating that the betweenness centrality is a valid measure for the selection of relevant zones in a traffic sensor network, validating the agent behavior proposed to modify the traffic lights phases and obtain a reduction in trip duration, and, finally, developing the needed tools for the model of sensors and agents in a microscopic traffic simulator such as SUMO.

**Acknowledgments:** This work has been supported by the Spanish Ministry of Economy and Competitiveness grants TIN2016-80622-P, TIN2014-61627-EXP and TEC2013-45183-R, and by the University of Alcalá through CCG2016/EXP-048 and UAH-AE-2017-3.

**Author Contributions:** All authors contributed to the manuscript and all authors have reviewed, read and approved the manuscript. L.C.-P. and I.M.-M. conceived the proposal. L.C.-P. designed and implemented it. L.C.-P. and D.R. designed and carried out the experiments. S.F. and D.R. provided the state of the art of sensor networks, traffic light management and Multi-Agent systems and validated the experimental results. All authors have contributed to the writing of this document.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ACTAM	Adaptive and Cooperative Traffic light Agent Model
ARLM	Arterial Road Location Method
ATMS	Advanced Traffic Management Systems
EA	Evolutionary Algorithms
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ITS	Intelligent Transportation Systems
MAS	Multi-Agent Systems
NSLP	Network Sensor Location Problem
O-D	Origin-Destination

PSO	Particle Swarm Optimization
RRLM	Random Road Location Method
SUMO	Simulation of Urban MObility
TAPAS	Travel and Activity PAtterns Simulation
TCP	Transmission Control Protocol
TIS-DLR	Institute of Transportation Systems at the German Aerospace Center
TraCI	Traffic Control Interface
WSN	Wireless Sensor Networks

## References

1. Cruz-Piris, L.; Rivera, D.; Marsa-Maestre, I.; de la Hoz, E.; Fernandez, S. Intelligent Traffic Light Management using Multi-Behavioral Agents. In Proceedings of the XIII Jornadas de Ingeniería Telemática (JITEL 2017), Valencia, Spain, 27–29 September 2017.
2. Lee, W.H.; Tseng, S.S.; Wang, C.H. Design and implementation of electronic toll collection system based on vehicle positioning system techniques. *Comput. Commun.* **2008**, *31*, 2925–2933.
3. Qi, L. Research on intelligent transportation system technologies and applications. In Proceedings of the PEITS'08 Workshop on Power Electronics and Intelligent Transportation System, Guangzhou, China, 2–3 August 2008; pp. 529–531.
4. Messelodi, S.; Modena, C.M.; Zanin, M.; De Natale, F.G.; Granelli, F.; Betterle, E.; Guarise, A. Intelligent extended floating car data collection. *Expert Syst. Appl.* **2009**, *36*, 4213–4227.
5. Chen, S.; Chen, F.; Liu, J.; Wu, J.; Bienkiewicz, B. Mobile mapping technology of wind velocity data along highway for traffic safety evaluation. *Transp. Res. Part C Emerg. Technol.* **2010**, *18*, 507–518.
6. Jun, J.; Guensler, R.; Ogle, J. Differences in observed speed patterns between crash-involved and crash-not-involved drivers: Application of in-vehicle monitoring technology. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 569–578.
7. Qin, X.; Khan, A.M. Control strategies of traffic signal timing transition for emergency vehicle preemption. *Transp. Res. Part C Emerg. Technol.* **2012**, *25*, 1–17.
8. Smith, H.R.; Hemily, B.; Ivanovic, M. *Transit Signal Priority (TSP): A Planning and Implementation Handbook*; ITS America: Washington, DC, USA, 2005; pp. 529–531.
9. Davey, E. Rail traffic management systems (TMS). In Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems (RSCS), London, UK, 21–24 May 2012; pp. 126–143.
10. Nellore, K.; Hancke, G.P. A Survey on Urban Traffic Management System Using Wireless Sensor Networks. *Sensors* **2016**, *16*, 157.
11. Faye, S.; Chaudet, C. Characterizing the topology of an urban wireless sensor network for road traffic management. *IEEE Trans. Veh. Technol.* **2016**, *65*, 5720–5725.
12. Zhou, B.; Cao, J.; Zeng, X.; Wu, H. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. In Proceedings of the 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), Ottawa, ON, Canada, 6–9 September 2010; pp. 1–5.
13. Yousef, K.M.; Al-Karaki, M.N.; Shatnawi, A.M. Intelligent traffic light flow control system using wireless sensors networks. *J. Inf. Sci. Eng.* **2010**, *26*, 753–768.
14. Fernandez, S.; Hadfi, R.; Ito, T.; Marsa-Maestre, I.; Velasco, J.R. Ontology-Based Architecture for Intelligent Transportation Systems Using a Traffic Sensor Network. *Sensors* **2016**, *16*, 1287.
15. Collotta, M.; Bello, L.L.; Pau, G. A novel approach for dynamic traffic lights management based on Wireless Sensor Networks and multiple fuzzy logic controllers. *Expert Syst. Appl.* **2015**, *42*, 5403–5415.
16. Kulkarni, R.V.; Venayagamoorthy, G.K. Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *41*, 262–267.
17. Shan, Z.; Zhu, Q. Camera location for real-time traffic state estimation in urban road network using big GPS data. *Neurocomputing* **2015**, *169*, 134–143.
18. Castillo, E.; Nogal, M.; Rivas, A.; Sánchez-Cambronero, S. Observability of traffic networks. Optimal location of counting and scanning devices. *Transp. B Transp. Dyn.* **2013**, *1*, 68–102. doi: 10.1080/21680566.2013.780987.
19. Hu, S.R.; Peeta, S.; Chu, C.H. Identification of vehicle sensor locations for link-based network traffic applications. *Transp. Res. Part B Methodol.* **2009**, *43*, 873–894.

20. Taale, H.; Bäck, T.; Preuss, M.; Eiben, A.; De Graaf, J.; Schippers, C. Optimizing traffic light controllers by means of evolutionary algorithms. In Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, 7–10 September 1998; Volume 3, pp. 1730–1734.
21. Garcia-Nieto, J.; Alba, E.; Olivera, A.C. Swarm intelligence for traffic light scheduling: Application to real urban areas. *Eng. Appl. Artif. Intell.* **2012**, *25*, 274–283.
22. Lau, S.P.; Merrett, G.V.; Weddell, A.S.; White, N.M. A traffic-aware street lighting scheme for Smart Cities using autonomous networked sensors. *Comput. Electr. Eng.* **2015**, *45*, 192–207.
23. Adler, J.L.; Blue, V.J. A cooperative multi-agent transportation management and route guidance system. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 433–454.
24. Bazzan, A.L.C.; Klugl, F. A review on agent-based technology for traffic and transportation. *Knowl. Eng. Rev.* **2014**, *29*, 375–403.
25. Chen, B.; Cheng, H.H. A review of the applications of agent technology in traffic and transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2010**, *11*, 485–497.
26. Tubaihat, M.; Shang, Y.; Shi, H. Adaptive traffic light control with wireless sensor networks. In Proceedings of the 4th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 11–13 January 2007; pp. 187–191.
27. Roozmond, D.A. Using intelligent agents for pro-active, real-time urban intersection control. *Eur. J. Oper. Res.* **2001**, *131*, 293–301.
28. Choy, M.C.; Srinivasan, D.; Cheu, R.L. Cooperative, hybrid agent architecture for real-time traffic signal control. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2003**, *33*, 597–607.
29. Ruey-Shun, C.; Duen-Kai, C.; Szu-Yin, L. Actam: Cooperative multi-agent system architecture for urban traffic signal control. *IEICE Trans. Inf. Syst.* **2005**, *88*, 119–126.
30. Logi, F.; Ritchie, S.G. A multi-agent architecture for cooperative inter-jurisdictional traffic congestion management. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 507–527.
31. Chen, B.; Cheng, H.H.; Palen, J. Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transp. Res. Part C Emerg. Technol.* **2009**, *17*, 1–10.
32. Cruz-Piris, L.; Rivera, D.; Marsa-Maestre, I.; de la Hoz, E. Optimizing vehicle trips using agent negotiation through a traffic matrix. In Proceedings of the The Tenth International Workshop on Agent-based Complex Automated Negotiations (ACAN2017), São Paulo, Brazil, 8–9 May 2017.
33. Sastre, D.; Torres, J.; Menendez, J.M. *Sistemas de Adquisición de Información de Tráfico: Estado Actual y Futuro*; Technical Report; Plataforma Tecnológica de la Carretera (PTC): Madrid, Spain, 2011. (In Spanish)
34. Kell, J.H.; Fullerton, I.J.; Mills, M.K. *Traffic Detector Handbook*; FHWA-HRT-06-108; Federal Highway Administration (US Department of Transportation): Washington, DC, USA, 2006; Volume I.
35. Wegener, A.; Piórkowski, M.; Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J.P. TraCI: An Interface for Coupling Road Traffic and Network Simulators. In *Proceedings of the 11th Communications and Networking Simulation Symposium*; ACM: New York, NY, USA, 2008; pp. 155–163.
36. Van Diggelen, F.; Enge, P. The worlds first gps mooc and worldwide laboratory using smartphones. In Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015), Tampa, FL, USA, 14–18 September 2015; pp. 361–369.
37. Kazerani, A.; Winter, S. Can betweenness centrality explain traffic flow. In Proceedings of the 12th AGILE International Conference on Geographic Information Science, Hannover, Germany, 2–5 June 2009; pp. 1–9.
38. Wang, P.; Hunter, T.; Bayen, A.M.; Schechtner, K.; González, M.C. Understanding road usage patterns in urban areas. *Sci. Rep.* **2012**, *2*, 1001.
39. Brandes, U. On variants of shortest-path betweenness centrality and their generic computation. *Soc. Netw.* **2008**, *30*, 136–145.
40. NetworkX. Edge\_Betweenness\_Centrality. Available online: [https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.centrality.edge\\_betweenness\\_centrality.html#networkx.algorithms.centrality.edge\\_betweenness\\_centrality](https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.centrality.edge_betweenness_centrality.html#networkx.algorithms.centrality.edge_betweenness_centrality) (accessed on 13 December 2017).
41. Winter, S. Modeling costs of turns in route planning. *Geoinformatica* **2002**, *6*, 345–361.
42. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent Development and Applications of SUMO—Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
43. TAPASCologne Scenario. Available online: <http://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne> (accessed on 14 July 2017).



44. Uppoor, S.; Trullols-Cruces, O.; Fiore, M.; Barcelo-Ordinas, J.M. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Trans. Mob. Comput.* **2014**, *13*, 1061–1075.
45. El-Tantawy, S.; Abdulhai, B.; Abdelgawad, H. Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1140–1150.
46. Simulation of Urban MObility Wiki: Simulation/Why Vehicles Are Teleporting. Available online: [http://sumo.dlr.de/wiki/Simulation/Why\\_Vehicles\\_are\\_teleporting](http://sumo.dlr.de/wiki/Simulation/Why_Vehicles_are_teleporting) (accessed on 7 December 2017).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

# Automated Optimization of Intersections Using a Genetic Algorithm

En este capítulo se muestra el artículo publicado “Optimización automatizada de intersecciones utilizando un algoritmo genético”. Tras un breve resumen de éste, se muestran los datos sobre la publicación y su relevancia. Por último, se ha incluido una copia de la versión publicada del artículo.

## 4.1 Introducción

Los problemas de congestión en las grandes ciudades, además de tener graves repercusiones económicas, provocan un incremento en las emisiones contaminantes generados por los vehículos. En los últimos años ha habido una constante y rápida evolución de las tecnologías aplicadas en el campo de los vehículos autónomos.

Existen soluciones comerciales con diferentes grados de conducción asistida y sistemas de condición semiautónoma, con previsiones muy favorables sobre obtener un posible coche autónomo completo en las próximas décadas. Este nuevo entorno genera nuevas oportunidades y retos para reducir los atascos en escenario con vehículos autónomos o semiautónomos.

Este artículo se centra en la optimización automática del paso de vehículos por intersección, las cuales son uno de los elementos que más problemas generan en una red de tráfico. El principal origen de estos problemas es debido a que las intersecciones deben gestionar múltiples flujos de tráfico, con diferentes prioridades y preferencias, cuyo resultado suele ser la generación de atascos. Este problema ha sido abordado proponiendo un mecanismo para modelar cualquier tipo de intersección, un método para calcular los caminos interiores con menores puntos de conflicto, y un proceso de optimización de los patrones de llegada de vehículos basada en un Algoritmo Genético. Este optimizador tiene como objetivo maximizar el rendimiento de la intersección. Para validar las soluciones propuestas se ha desarrollado un simulador de intersecciones que puede ser adaptado tanto a escenarios de vehículos autónomos

como convencionales. Este simulador permite proporcionar resultados realistas bajo ciertas condiciones.

Se han realizado un conjunto de experimentos para comparar la solución propuesta con otros métodos de coordinación ampliamente utilizados como las intersecciones basadas en prioridades y las gestionadas por semáforos, utilizando tanto el simulador creado como el simulador comercial SUMO. Los resultados obtenidos muestran la validez y la mejora de rendimiento del sistema propuesto, ofreciendo una mejora de entre el 9.21 % y el 36.98 % en comparación a los métodos de coordinación tradicionales.

## 4.2 Relevancia de la publicación

Título:	Automated Optimization of Intersections Using a Genetic Algorithm		
Autores:	Luis Cruz-Piris, Miguel A. Lopez-Carmona e Ivan Marsa-Maestre		
Estado:	Publicado		
Fecha de publicación:	25/01/2019		
DOI:	<a href="https://doi.org/10.1109/ACCESS.2019.2895370">https://doi.org/10.1109/ACCESS.2019.2895370</a>		
Revista:	Nombre:	IEEE Access	
	Editorial:	IEEE	
	País:	EEUU	
Relevancia	Revista:	Índice de impacto	4.098
		JCR (2018*):	Q1 (18/88)
	Artículo:	Lecturas desde su publicación <sup>1</sup> :	337
		Citas <sup>2</sup> :	2

\* Último JCR disponible a 15/09/2019

<sup>1</sup> Fuente: IEEE Xplore, Fecha de consulta: 15/09/2019

<sup>2</sup> Fuente: Google Scholar, Fecha de consulta: 15/09/2019

**Tabla 4.1.:** Datos asociados a la publicación: Automated Optimization of Intersections Using a Genetic Algorithm.

## 4.3 Artículo

Received December 21, 2018, accepted January 19, 2019, date of publication January 25, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2895370

# Automated Optimization of Intersections Using a Genetic Algorithm

LUIS CRUZ-PIRIS<sup>1</sup>, MIGUEL A. LOPEZ-CARMONA<sup>1</sup>, AND IVAN MARSA-MAESTRE<sup>1</sup>

Departamento de Automática, Escuela Politécnica Superior, Universidad de Alcalá, Campus Universitario, 28871 Alcalá de Henares, 28801 Madrid, Spain

Corresponding author: Luis Cruz-Piris (luis.cruz@uah.es)

This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Grant TIN2016-80622-P and Grant TEC2013-45183-R, and in part by the University of Alcalá under Grant UAH-AE-2017-3.

**ABSTRACT** Traffic jams in large cities, in addition to having a very high economic cost, cause an increase in emissions generated by vehicles over the same route being driven under normal conditions. In recent years, there has been a rapid evolution in the technologies applied to the field of autonomous vehicles. There are currently commercial solutions for assisted driving and semi-autonomous driving systems, with very favorable forecasts for reaching a completely autonomous vehicle scenario in the coming decades. This new environment generates opportunities and challenges to reduce congestion in scenarios with autonomous or semi-autonomous vehicles. This paper focuses on the automatic optimization of the passage of vehicles through intersections. The intersections are one of the most conflict-generating elements in a traffic network. This type of conflicts arises because the intersections must manage multiple traffic flows with different priorities and preferences, often leading to traffic jams. The problem has been addressed by proposing three mechanisms to model any type of intersection, to calculate the roads with fewer points of conflict between their inputs and outputs, and to optimize the arrival rate of vehicles using a Genetic Algorithm to achieve the maximum performance of the intersection. To validate this solution, a cellular automata simulator has been developed, which can be adapted to both autonomous and conventional vehicle scenarios and can provide realistic results when certain conditions are met. The results obtained have been compared with other traditional solutions (priority and traffic lights) using microscopic traffic simulations, and with those obtained in other studies showing the advantages of the proposed system. The proposed systems achieve a throughput improvement between 9.21% and 36.98% compared with the traditional solutions.

**INDEX TERMS** Autonomous systems, autonomous vehicles, genetic algorithms, traffic congestion.

## I. INTRODUCTION

Congestion is one of the big problems to solve in large cities. The INRIX Global Traffic Scorecard [1] assesses the impact of congestion 24 hours a day in 1,360 cities in 38 countries over 5 continents and calculates its cost as \$461 billion in 2017, only considering data from the US, UK, and Germany.

Traffic problems are very heterogeneous in terms of either their causes or their consequences. Road congestion is one of the most frequent causes, but it is not the only one. When congestion occurs, necessary mitigation actions should be taken to ensure that its duration is minimized. Another approach is to anticipate the possible formation of this congestion and guide the actions adopted to try to avoid it.

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

Due to the importance of the problem, there have been extensive efforts devoted to finding solutions for it. The successive advances in technology in this field have led to the design and implementation of solutions that improve upon the previous systems. During the eighties and nineties, the so-called Advanced Traffic Management Systems (ATMS) [2] were created, defined as the central nervous system of any transport system. ATMS receive and combine inputs from heterogeneous data sources and apply a series of algorithms to make technology-aided transportation decisions. These types of systems have evolved into an Intelligent Transportation System (ITS), which is the term commonly used today.

In the last years, improvements in technology in the field of Connected and Autonomous Vehicles (CAVs) have led to new opportunities to design and develop systems and tools focused on this new paradigm. The degree of autonomy

in driving a vehicle can be measured by levels using the scale defined by Society of Automotive Engineers (SAE) [3], ranging from no driving automation (level 0) to full driving automation (level 5), in the context of motor vehicles. One of the most important factors which would enable the driving automation is the final price of the technology embedded in the vehicles. For instance, in [4], taking into account the changes in the price of technology and the current regulations, it is predicted that by the year 2030, about 98% of the vehicles will have connectivity. These forecasts lead us to be optimistic about the increase of autonomous vehicles and their automation level in the next decades.

One of the study areas explored in autonomous vehicles scenarios is the definition of new techniques applied to reduce the conflict points in a road network. These points are those places where various vehicles have planned to pass at the same time. Intersections are one of these points, and the management and coordination of the vehicles that cross through these points pose a challenge in any type of scenario. From the point of view of the efficiency of a route, the ideal case would be that a vehicle could cross through an intersection without having to stop at all.

This work is focused on optimizing the passage of vehicles through a generic intersection automatically. We do this by means of an automatic optimization system that achieves the maximum performance of an intersection in a CAV scenario. Our contributions in this field are described below:

- We create a model capable of univocally labeling all the elements that compose an intersection, in an automated manner, which allows to apply the approach proposed in this paper to any intersection regardless of type.
- We propose a process to calculate the shortest paths with minimum conflict points between them in a cellular automata scenario.
- We define an algorithm that obtains the patterns or cadence of the entry of vehicles into the intersection, using the previously calculated paths, to achieve maximum performance.

The rest of this paper is organized as follows. First we will study intersection optimization alternatives such as: infrastructure designs that favor the passage of vehicles for a given way, management systems based on variable signaling, and optimizers that reduce the delays caused by intersections using bio-inspired algorithms (Section II). Starting with a known intersection, each of the components of the proposed system will be defined (cellular automata model for intersections, algorithm for calculating paths with minimum conflict points and optimizing crossing patterns through a Genetic Algorithm (GA) in Section III). The proposed system will be tested by analyzing the same intersection studied in [5] and [6] and comparing the results obtained with traditional intersection management methods as yield signs or traffic lights, which will validate its proper operation (Section IV). In conclusion, the advantages of the work will be highlighted and future lines of work will be indicated (Section V).

## II. RELATED WORK

Intersection studies are among the most common studies in transport engineering [7]. “Although intersections constitute a small segment of the highway network, just under a quarter of fatal collisions occur there” [8]. These works focus on improving features such as intersection delay or queue length and saturation flow in its inputs. In this section, we will review some of the existing work on how to improve vehicle traffic through intersection optimization.

The management of the cross points between vehicles can be performed through conventional strategies based on intersections controlled by traffic signs (*e.g.* yield signs) or by traffic lights. Another common solution is the replacement of intersections for roundabouts [9]. From the intersection design perspective, there are also unconventional strategies to improve the efficiency and safety of intersections. Median U-Turn and Continuous-Flow intersections [10] are examples of this trend. The first one removes the left-turns from the main intersection (these movements are converted to right turns using a median crossover to make a U-turn). The second type shifts the left-turn from the surroundings of the main intersection to left-turns across the opposing traffic lanes located prior to the main intersection. Another solution based on the variation of the road structure is the proposal of Xie and Jiang [11]. In this work, they describe a model that incorporates the allocation of exit lanes as decision variables. Their design increases the capacity of isolated intersections. These strategies can help to improve the *a priori* traffic planning but present a high cost due to the need for building new infrastructures. Besides, they lack flexibility when the traffic flows vary.

The technological evolution has led to the rising of new scenarios based on the use of autonomous or semi-autonomous vehicles in a fully connected environment. Cooperative work among vehicles to use shared resources such as intersections has given birth to the field of Cooperative Intersection Management (CIM). Generally, these management systems can be classified as centralized or distributed. Another dimension for classification may be the use of signalized or non-signalized intersection [12].

There are systems designed to optimize the control of traffic signaling (signalized intersection) to achieve higher levels of efficiency. SCATS (Sydney Coordinated Adaptive Traffic System) [13] and SCOOT (Split, Cycle and Offset Optimization Technique) [14] are two systems based on the Urban Traffic Control (UTC) methodology. The intersections are monitored through detectors placed at their inputs and outputs.

Solving problems related to the cross points using the conventional infrastructures raise new challenges. Hult *et al.* [15], model the vehicle coordination as a finite time constrained optimal control problem. The authors focus on the combinatorial part of the problem, that is, the vehicle crossing order. They use a central optimization-based coordinator to manage the commands sent to vehicles.

Although the theoretical results obtained in this work are very solid, the algorithm proposed could present scalability issues in more complex scenarios.

It is a well-known issue that left turns reduces the intersection capacity significantly. Xuan *et al.* [16] propose a strategy based on grouping batches of vehicles depending on the turn they are going to take in the next intersection. They define a sorting area, before the intersection, where the set of batches are sorted to improve the efficiency of the intersection flows. The intersection input points are managed using traffic lights. This proposal improves the intersection capacity over conventional solutions, which are penalized by the left turns. Reference [17] follows a similar strategy, and also [18], where Yan *et al.* focus on solving this problem by optimizing the traffic light phases to enable left-turns. In general, these solutions present some drawbacks such as the cost related to requiring physical changes at the intersection in some cases. Additionally, in high congestion scenarios, the generation of batches depending on the turns taken by vehicles can be a very complex task.

Following the CIM classification, an example approach for a non-signalized intersection in a distributed system that uses many of the advantages achieved with the new communication technologies are “virtual traffic lights” (VTL) [19]. In these systems, vehicles follow the same operating principle of traditional traffic lights but instead of reacting to a light signal, all vehicles communicate with each other to decide cooperatively which direction has priority of passage. This system has many advantages as it allows to adapt to different traffic conditions, but it may have limitations due to the previously defined rules (necessary to define the vehicle with priority and the cycle time).

Azimi, in his PhD Thesis [5], analyzes multiple aspects of co-operative driving at intersections. He makes contributions to the management of V2V communications in these scenarios and develops tools such as his own simulator. Besides that, the chapters dedicated to vehicles synchronization in intersections are especially interesting. He defines in [6] “a spatiotemporal technique called the Ballroom Intersection Protocol (BRIP) to manage the safe and efficient passage of autonomous vehicles through intersections”. This strategy uses the advantages offered by V2X communications, and the vehicle geo-positioning to generate a set of Synchronized Intersection Arrival Patterns (SIAP). This allows him to, avoiding collisions, obtain the maximum possible throughput in the intersection. To validate his proposal, Azimi uses a cellular automata simulator based on a rectangular two-dimensional lattice with von Neumann neighborhoods. Although Azimi’s work is very extensive, it presents some limitations on how to apply these techniques to other intersection types.

Focusing on conventional intersection optimization works, the use of traffic lights is a widely studied solution, and there are many applied techniques. The improvement in sensor-based technologies, and especially in the Wireless Sensor Networks (WSN) area, has offered new data sources which

can contribute to making more accurate decisions. The combination of WSN and Multi-Agent Systems (MAS) has led to solutions such as the one proposed in [20] for the autonomous management of traffic light phases depending on real-time traffic conditions. Younes and Boukerche, in [21], work over the dynamic optimization of traffic lights scheduling in specific scenarios where there is a special priority for emergency vehicles.

A more general approach for connected vehicle scenarios is shown in [22], where they generate a co-operative system between vehicles and infrastructures, achieving, in most of cases, important improvements in the vehicle delays. The proposed solution, however, is based in a too ideal use case, although the results show the importance of this work.

The bio-inspired algorithms [23] applied to the optimization area, are very useful to obtain solutions over nonlinear complex problems. Like the previously referenced works, the optimization of vehicles crossing intersections through the traffic light control is a problem where bio-inspired algorithms can be useful. Chuo *et al.* [24] propose an enhanced particle swarm algorithm to determine a traffic signal timing plan between intersections. This allows the reduction of congestion in the network. This work has been tested in a simple scenario, obtaining good results, but presents some flaws when high congestion scenarios are reached. Similarly, Hao *et al.* approach this problem in [25] using a Tabu search-artificial bee colony algorithm which works under unsaturated flow conditions.

Genetic Algorithms (GA), one of the most popular bio-inspired techniques, have already been used to solve some of the problems referenced previously in this section. For instance, in [26] they propose a GA-based solution for the autonomous vehicle sequencing problem at intersections, and in [27] they offer a solution using GA for priority vehicles, specifically emergency vehicles, in intersections.

Finally, it is crucial in these studies to count on tools which allow to validate the proposed solutions. Traffic simulation tools have been developed to reach this goal and can be categorized in three types: Microscopic, Macroscopic and Mesoscopic [28]. For a detailed testing of the vehicle’s behavior, the most used simulators are the microscopic ones. Traffic cellular automata (TCA) models have proven to offer very reliable results with a high computational efficiency [29]. The usefulness of applying this type of models has been widely demonstrated in urban traffic scenarios. For instance, Tonguz *et al.* build an urban traffic mobility model in [30], where they also focus on the intersection control mechanisms. Other solutions such as [31] include in their model parameters like the vehicle speed, the braking lights, and the possible conflicts between traffic flows.

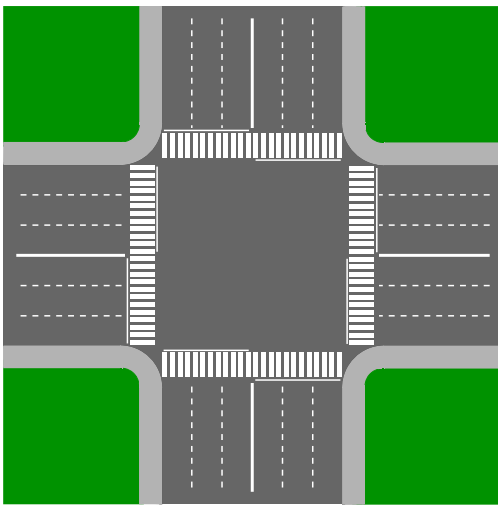
### III. MODELING AND OPTIMIZATION OF INTERSECTIONS

The main goal of the proposed system is to obtain automatically the optimal arrival patterns, that is, the sets of routes and time intervals for vehicle arrivals to use an intersection as efficiently as possible. This efficiency is measured as

the intersection average occupation. That is, in a cellular automata-based scenario where vehicles do not stop or vary their speed, maximizing the average occupation of the intersection when all input flows are equal, implies maximizing the vehicle flows through the intersection as we will show in Section IV. Furthermore, if all vehicles move deterministically with constant speed, the system reaches a metastable state [32].

In this section, we explain in depth the proposed solutions to automate the problem management. We show the intersection modeling (Section III-A), the calculation of the possible less conflicting paths between inputs and outputs in the intersection (Section III-B) and then the GA developed to optimize the vehicle arrival rates which would maximize the average intersection occupation.

We have set a sample intersection to facilitate the comprehension of the proposals and developments, which allow us to show the application of the methods to a specific use case after the theoretical descriptions of each one. We have selected the intersection shown in Figure 1, being a recurrent example used in the literature [5], [17], [22], [27], [31]. This intersection is composed of 4 arms, with 3 input lanes and 3 output lanes each one.



**FIGURE 1.** Use case: Intersection composed of 4 arms with 3 input and 3 output lanes.

#### A. CELLULAR AUTOMATA MODEL FOR INTERSECTIONS

The Traffic Cellular Automata (TCA) are very efficient models to simulate microscopic traffic flows [29]. Therefore, they are frequently used in vehicular traffic scenarios, when it is important to analyze the behavior of traffic in a specific situation, such as the behavior of the traffic crossing an intersection.

There are diverse approaches to model a traffic scenario using TCA. There are various lattice topologies for a cellular automaton in two dimensions. Some of these types are triangular or isometric, and hexagonal, although the

use of a rectangular two-dimensional lattice is a common approach. In this kind of models, it is necessary to represent the turning restrictions derived from the intersection.

For Cellular Automata neighborhoods with a radius of 1, there are two possible models for rectangular two-dimensional lattice: the von Neumann or the Moore neighborhoods. The first one consists of the central cell itself plus 4 adjacent cells while the second one includes 8 adjacent cells.

In this work, we focus in an intersection modeling where each intersection is composed of cells of enough size to contain an entire vehicle and the mandatory security distances to the vehicles located in adjacent cells. Each cell contains the information needed for the vehicles to know the cells they can move to in the next simulation step.

Cellular automata modeling process for intersections has been divided into three phases: intersection grid generation from the geographical data, generic intersection modeling proposal and conversion from the proposed model to a directed graph. Figure 2 shows a graphical example of each step.

#### 1) INTERSECTION GRID GENERATION

The first step to model the intersection is to generate the base grid using its defining features (shape, size, number of arms, number of lanes, distances between lanes, etc.). This grid will be used as the base of the cellular automata used to define the behavior of the vehicle in the intersection. Each cell of the grid must comply with minimum restrictions to guarantee the model correctness. The specific design features which should be taken into account to determine the cell size related with vehicles size, the possible conflict points, the turning radius, etc. have been covered in works such as [5, Ch. 7], [29, Sec. 1] or [31, Sec. 3].

Following the sample intersection is shown in Figure 1, and after applying the basic cellular automata modeling, the intersection is modeled as the grid shown in Figure 2.a. Each input or input lane in the intersection has been modeled as a fixed size cell sequence. Each vehicle can be located in one cell in each simulation instant and, in the next instant be moved to one of the adjacent cells, as long as the intersection rules allow it. The inner part of the intersection is composed by a cell grid which define the possible paths between inputs and outputs. In this case, the intersection geometry determines a 6x6 cell grid.

Although the example used throughout the article focuses on a TCA with very specific features, the modeling process proposed in section III-A.2 can be used in other types of environments. Section III-A.4 shows the application of this proposed modeling system to a TCA with smaller cells.

#### 2) GENERIC INTERSECTION MODELING PROPOSAL

In step 1 we describe the generic process for transforming the traffic network into a TCA. The theory about simulators based on cellular automata does not cover the nomenclature to be used to identify the model elements, and usually, each

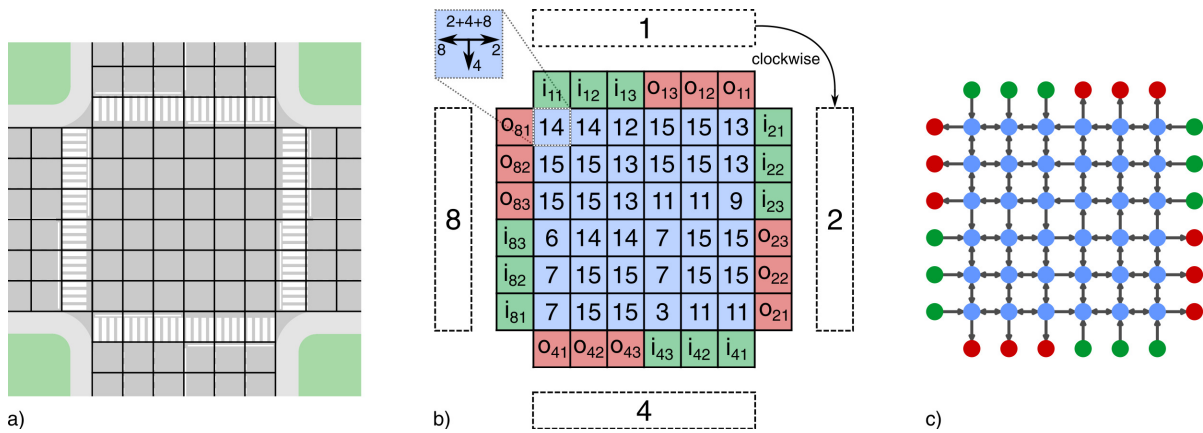


FIGURE 2. Intersection models. a) Cellular automata. b) Proposed system. c) Translation of the proposed model to a directed graph.

simulator uses a custom identifying system. One of the goals of this proposal is to provide a system able to automatically model the intersections of a traffic scenario. To achieve this, it is necessary to define a methodology which guarantees the unique denomination of each element in the model.

The proposal for the generic modeling is shown in Figure 2.b, and it is composed by the following steps:

- Identification of the input and output points in the intersection: Each arm in the intersection is a possible input point, output point, or input and output point for vehicles. The traffic flows can be then identified by these points, and therefore, they constitute the base elements of the model. The rest of the elements will be tagged according to these values.

The first arm to be tagged could be randomly selected, but we have decided to set criteria to be used in every intersection: The arm closest to the north will be selected as the first arm. If there is more than one arm at the same distance to the vector going from the center of the intersection to the north, we select the arm located in the right side of the vector as the first arm to be tagged. The next tags are assigned in order by following clockwise the next arms in the intersection. In the following sections, we use the combination of values given to each arm to tag the possible turns in the cells of the intersection. This means that the tags used for the arms must be selected in a way that allows to perform the opposite operation from the cell tag and obtain the possible turns. To allow this, we use base 2 exponents as arm tags:

$$Ta(j) = 2^j \tag{1}$$

where  $Ta$  is the tag of the arm and  $j$  is an integer between 0 and  $n$ , where 0 is assigned to the first arm and  $n$  is assigned to the last arm. According to this system, for the use case explained at the beginning of the section, we would have  $Ta([0, 1, 2, 3]) = [1, 2, 4, 8]$ , as shown in Figure 2.b.

- Input and output points tagging: The proposed intersection modeling system includes both the cells from the intersection itself and the first or last cell from the lanes that start or end at the intersection. That is, the cells used by vehicles when accessing the intersection or when they are leaving it. These cells are tagged from right to left in the direction of the traffic using integers and starting with 1. The notation for the cell tags is:

$$\begin{aligned} \text{inputs} &: i_{ab} \\ \text{outputs} &: o_{ab} \end{aligned} \tag{2}$$

where  $a$  is the value of the arm tag ( $Ta$ ) and  $b$  is the lane where the cell is located. In Figure 2.b we show the tagged input and output cells using green and red colors respectively.

- Turning values in intersection inner cells: Once a vehicle arrives at a given cell in the intersection, one of the most important parameters to take into account by the simulator is whether the movement of the vehicle from the cell to any of the adjacent cells is allowed or not. This task can be eased using as cell values numbers composed by the addition of the values of each arm tag in the intersection where it is possible to move the vehicle to:

$$Tv = \sum_{k \in \text{PossibleTurns}(\text{CurrentCell})} Ta(k) \tag{3}$$

where  $\text{PossibleTurns}(\text{CurrentCell})$  is the set of possible turns from the current cell to a specific output arm. In Figure 2.b, if from a given cell the vehicle can be moved to the left, we will add the value  $Ta = 8$ . If the vehicle can be moved towards the bottom of the intersection, the value  $Ta = 4$  is added to the previous value, and therefore the cell value will be  $Tv = 12$ . For instance, if we select the inner top left cell of the use case intersection, we can see that in that cell the vehicle can go to the right (towards the arm tagged as 8), go straight (towards the arm tagged as 4) and turn to the left (towards the arm tagged as 2). Therefore, the turning



value for that cell is  $Tv = Ta(1) + Ta(2) + Ta(3) = 2 + 4 + 8 = 14$ . This value can be easily decomposed at any moment in the original base 2 exponents composing it and can be used to obtain the possible turns in the cell.

Following these previous steps, we can model the intersection completely. This process can be performed automatically using the geographic and geometric data of an intersection.

### 3) CELLULAR AUTOMATA TO DIRECTED GRAPH CONVERSION

The possible paths are taken by vehicles in the intersection (respecting the intersection restrictions) are a very important factor to optimize the intersection behavior. In Section III-B these paths are going to be calculated, and one of the most powerful tools to be used for the calculation of a high number of paths are graph based algorithms. In this section we show an algorithm for the transformation of the proposed cellular automata model shown in Section III-A.2 into a directed graph which can be used to obtain possible paths between input and output points.

The algorithm steps are:

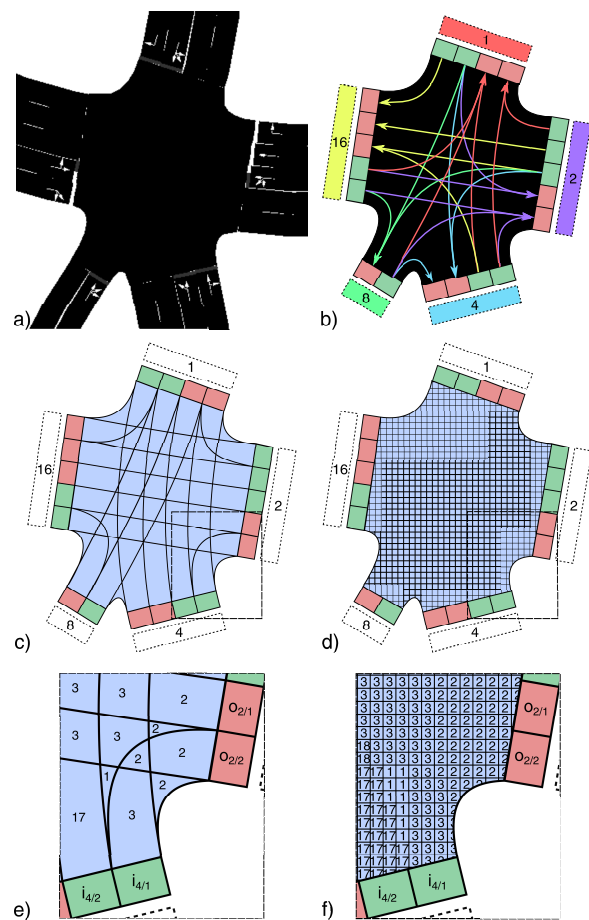
- 1) Each cell in the original model is transformed into a graph node.
- 2) The turning values are retrieved by going through each cell in the model:
  - The values are decomposed in their base 2 exponents.
  - For each possible turn obtained in the previous point, the destination cell corresponding to the value is located and a new directed edge is added to the graph. The edge goes from the current cell to the adjacent cell in the direction pointed out by the turning value.

After carrying out this process, we obtain a graph composed by a set of nodes which correspond to the model cells and a set of directed edges which represent the possible movements between cells in the intersection. The result of the conversion process for the use case we have been following is shown in Figure 2.c.

### 4) MODELING IRREGULAR-SHAPE INTERSECTIONS

The use case shown in Figure 1 allows for the application of the proposal to a very specific scenario, which facilitates the explanation of the modeling process. Due to the characteristics of some intersections (number of input and output points, restrictions, or shape), it is possible that the TCA used in which each cell can contain a single vehicle is not possible. In these cases, the size of the cell can be reduced to adapt to the restrictions of the intersection, defining how many cells each vehicle will occupy (e.g. 1x2 cells per vehicle in [31]).

Figure 3.a shows a real intersection with five branches and arms. Each of these branches has a number of input and output lanes that vary between one and three. For each input lane, there are possibilities for turning by following the intersection's signage (the ability to go from each input lane to a specific output lane). This information is necessary



**FIGURE 3.** Model of an asymmetric five-branch intersection. a) Real intersection. b) Trip trajectories. c) Inner virtual lanes. d) Cells of the traffic cellular automata. e) Detail of turn values (virtual lanes). f) Detail of turn values (cells).

for the definition of the trajectories generated between each input lane and the possible output lanes in order to cross the intersection (Figure 3.b). The intersection model follows the same steps explained above in section III-A.2. It begins by listing each arm of the intersection, starting with the one farthest north, by successive powers of 2. In this case, a five-branch intersection, the values used are 1, 2, 4, 8, and 16. Due to the fact that the shape of the intersection and its input/output connections do not allow for a homogeneous grid, it is necessary to draw the set of virtual lanes based on the input and output paths (Figure 3.c). Similarly, one must define a cell size that fits both the dimension of the vehicles and the shape of the intersection, as well as the virtual lanes that have been defined. To this end, Figure 3.d shows a grid with cells measuring 1.5x1.5 m (i.e. each vehicle would occupy 2x3 cells).

After naming each of the intersection's arms and its input and output points, the third step of the proposed modeling system is to label each cell with the possible turn values.

Figure 3.e shows the turn values for each region of the previously-defined virtual lanes (e.g. starting from input point  $i_{4/1}$ , the first region has a value of 3, which allows the vehicles that use it to continue straight toward  $o_{1/x}$ , or turn right toward  $o_{2/x}$ ). This assignment of turn values is transferred directly to the TCA so that each cell occupying the previously-labeled regions is given the same value. In the event that a cell occupies more than one region with different values, the assigned value will be that of the region that occupies the largest area of the cell. Finally, if two regions occupy the same area of the cell, the assigned value will be the one that coincides to a greater degree with the neighboring cells. This assignment is shown in Figure 3.f.

The movement of the vehicles follows the same principle described above but considering that each vehicle occupies six cells and that each of these cells must comply with the movement restrictions as defined by the labeling.

### B. ALGORITHM FOR CALCULATING PATHS WITH MINIMUM CONFLICT POINTS

The vehicles cross intersections from an input point to an output point following a given path. If two paths have common cells, it is possible that at the same time more than one vehicle tries to occupy the cell. This generates a conflict point and forces stopping the simulation, because it would generate an invalid solution in which there would be a collision between vehicles, given that in this scenario the vehicle speed is a static value.

To obtain the greatest possible vehicle flow in the intersection, it is crucial to remove these conflict points. This can be achieved by generating paths without common cells or by modifying the vehicle arrival patterns, ensuring that there is not a time instant where there is more than one vehicle in a cell.

For each vehicle, there must be possible paths from its input point and to any output point in the intersection in each direction that is not the input direction, except if the intersection forbids a specific turn. In this case, it is not possible to generate completely disjoint paths without any common cell, but it is possible to calculate, using the proposed algorithm, the paths that offer the less possible conflict points. This guarantees that it is possible to optimize the vehicle arrival patterns and that we start from the best possible case in Section III-C.

We start the algorithm explanation by defining the set of elements to be used during the process. The set of possible ways or intersection arms ( $D$ ) is shown in the Equation 4.

$$D = \{Ta(j) : j \text{ is an integer; and } 0 \leq j \leq n\} \quad (4)$$

where  $j$  identifies the specific arm, varying from 0 for the starting arm, to  $n$  for the last arm in the intersection, always following a clockwise order.

The set of input and output lanes in the way  $Ta(j)$  are denoted by  $L_{Ta(j)in}$  and  $L_{Ta(j)out}$  respectively

(Equations 5 and 6).

$$L_{Ta(j)in} = \{i_{Ta(j)b} : b \text{ is an integer; and } 0 \leq b \leq m\} \quad (5)$$

where  $b$  is a integer between 1 (lane more to the right of the way) and  $m$  (lane more to the left of the way).

$$L_{Ta(j)out} = \{o_{Ta(j)b} : b \text{ is an integer; and } 0 \leq b \leq p\} \quad (6)$$

where  $b$  is a integer between 1 (lane more to the right of the way) and  $p$  (lane more to the left of the way).

Previously, we have defined the sets of elements of the intersection model proposed in Section III-A.2, which are necessary to obtain the paths with less conflicting points. The algorithm steps for the path calculation are:

- 1) An element  $Ta(x) \in D$  is selected. This element is fixed as the starting input point of the intersection. The set  $I$  is composed by this unique element:

$$I = \{Ta(x)\} \quad (7)$$

- 2)  $O$  is the set of possible output ways, obtained as the difference between all the possible ways ( $D$ ) and the way selected as input way ( $I$ ). This set is shown in the Equation 8.

$$O = D - I \quad (8)$$

- 3) Let it be  $r$  the size of the set  $L_{Ta(x)in}$ . The permutations, with size  $r$ , of all possible output ways ( $O$ ) are calculated and is denoted by  $\mathfrak{S}(O)$ . That is,  $\mathfrak{S}(O)$  is composed by  $r$ -length tuples containing all possible orderings without repeated elements.
- 4) The Cartesian product of every set of lanes in each output way ( $L_{Ta(y)out}$ , where  $y \in O$ ) in the intersection is calculated:

$$L_{product} = \prod_{y \in O} L_{Ta(y)out}$$

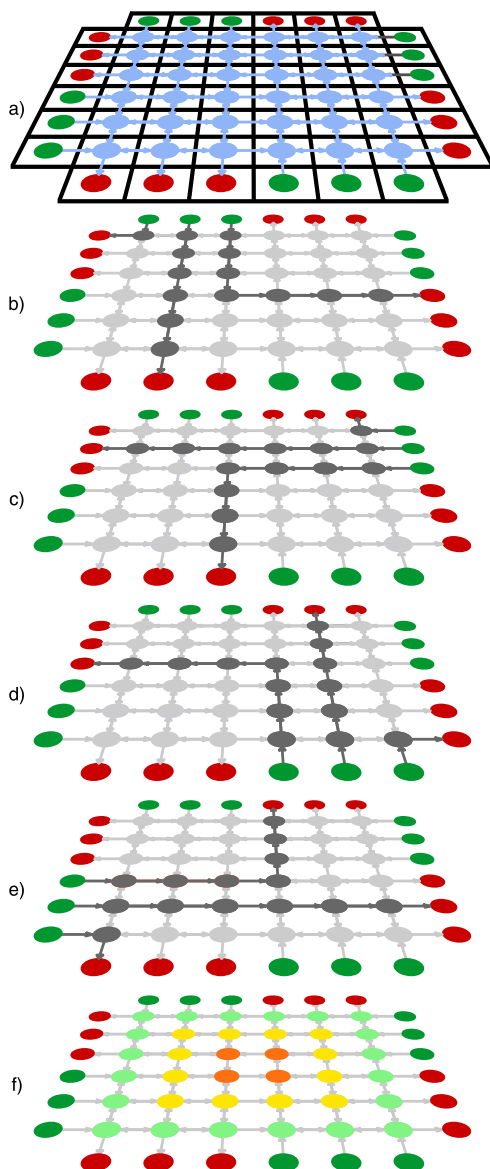
- 5) The Cartesian product of every set  $\mathfrak{S}(O)$  and  $L_{product}$  is calculated, and provides an ordered list of every possible combinations of output ways and lanes of each output way:  $C = \mathfrak{S}(O) \times L_{product}$ .
- 6) Each element  $c \in C$  is composed by the combination of one output way  $Ta(y)$ , and one output lane of this specific  $Ta(y)$ . We calculate every possible shortest path between a fixed input way  $Ta(x)$  (source node) and each  $c$  (target node), obtaining the set  $P_c$ .

Given that the size of each tuple in  $\mathfrak{S}(O)$  is equal to the size of  $L_{Ta(x)in}$ , the selection of a specific input lane in  $L_{Ta(x)in}$  as source node is determined by its index in the set (i.e.: if  $L_{Ta(x)in} = \{1, 2, 3\}$  and  $O = \{2, 4, 8\}$ , for a given element  $(4, 8, 2) \in \mathfrak{S}(O)$ , the input lanes 1, 2 and 3 have as destinations the ways 4, 8 and 2 respectively).

- 7) For each set  $P_c$  there is a matrix which is filled with the possible paths, obtaining the possible conflict points in the process. The number of possible conflicts is the cost of the specific combination of input and output points using the specific paths selected. The tuple formed by  $P_c$  and the calculated cost is added to the *paths* list.

- 8) The *paths* list is sorted from lower to higher cost. The element in the first position of the list is the set of input and output points in the intersection and their corresponding minimum conflict paths.
- 9) If the input flows are not symmetric (there is different traffic volume in each input way), the previous steps are repeated for the rest of elements in  $D$ .

In Figure 4 we show the results obtained by applying the algorithm to the use case of the paper. In this example, there is



**FIGURE 4.** Graphical representation of the optimal paths. a) Graph representing the intersection from the cells composing it. b, c, d, e) Minimum conflict paths for each input way. f) Representation of the conflict degree for each node (cell) in the graph: Green represents nodes without conflicts, yellow represents nodes with a single conflict and orange represents nodes with more than two conflicting routes.

an intersection composed of 4 arms or ways, and each one is composed of 3 input and 3 output lanes. Following the graph is shown in Figure 2.c, it is possible to determine that  $D = \{1, 2, 4, 8\}$  and that  $L_{Ta(j)in/out} = \{1, 2, 3\}$ , given that all the input or output ways are composed by the same number of lanes in this case. These features determine that the size of the set  $C$  for a single way is 162 ( $6 \times 3 \times 3 \times 3$ ) and therefore, the number of shortest paths for these 162 possible combinations is 81,067. Given that the size of  $D$  is 4, the size of the *paths* list is 324,268. The execution time for this algorithm applied to the use case has been 3.8242s (the testing equipment used for this calculations is the one described at the beginning of Section IV). In Figure 4, we show the graph modeling the intersection and the correspondence between this graph and the cell grid. In Figures 4.b, 4.c, 4.d and 4.e, we show the minimum conflict paths obtained for each input way in the intersection. Finally, in Figure 4 we show the conflict degree in each node when every route is used at the same time. Nodes without conflicts (there is zero or one route using the node) are colored green, nodes with a single conflict (there are two routes sharing the node) are colored yellow, and nodes with more than two conflicting routes are colored orange.

The paths calculated in this section are going to be used later in Section III-C.

### C. OPTIMIZING CROSSING PATTERNS THROUGH A GENETIC ALGORITHM

In section II we have shown various methods for traffic flow optimization in intersections. Some of these methods, such as the one proposed by Azimi *et al.* [5], are based in the empirical study of a specific intersection type. Genetic Algorithm based optimization system proposed in this work, on the other hand, has been designed to be used in any intersection type and to be able to autonomously generate optimal solutions for each one.

Using the model proposed in Section III-A.2, each intersection has a set of input and output points denoted by  $i_{ab}$  and  $o_{ab}$ . These points are in the external perimeter of the intersection and, once a vehicle arrives at one of these points, following the cellular automata scenario, there is a fixed path crossing the intersection which leads to one of the output points ( $o_{ab}$ ). While in the previous section the optimized calculation of these paths has been performed in an isolated manner, in this section we focus on obtaining the optimal input traffic flows to guarantee the maximum traffic rate in the intersection, without rising conflicts (collisions) between vehicles.

Genetic algorithms (GA), widely used to solve many complex problems in areas such as engineering or artificial intelligence, rely on the codification of the problem as an individual integrated into a population. Every individual is evaluated independently using a fitness function measuring the goodness of the individual according to the problem. This population, like what happens in nature, varies in each generation, favoring the survival of the individuals better adapted to the

environment (that is, the individuals which better solve the problem).

The arrival pattern in a cell  $i_{ab}$  determines the cadence of vehicle arrivals, that is, the time intervals between the arrival of vehicles to the intersection in that point. Although the size of the pattern can be from 2 to infinite cells, in symmetrical intersections they reach a stable state for patterns longer than the longest path run by the vehicles in the intersection. This stable state occurs when the sequence of occupation values is repeated every  $n$  simulation steps, where  $n$  is the pattern length. The set of vehicle queues of each input point (pattern) is the parameter to be optimized by the GA. Along with the codification of this problem, it is necessary to use a cellular automata-based simulator to validate the solutions obtained by each individual in each generation of the GA. This simulator is used to calculate the fitness value of the individuals in each iteration of the algorithm. In the next sections, we show the proposals for chromosome codification, the used GA, the methods used for selection, crossover and mutation, and the developed fitness function [33].

1) CHROMOSOME CODIFICATION

The codification of the problem must be carried out in a way that enables not only the evaluation of the individuals through the fitness function, but also to obtain viable crossover and mutation functions. In the case at hand, the variables of the problem are the sequences of vehicles composing the arrival patterns in each input cell of the intersection ( $i_{ab}$ ). The following parameters define the codification of the individuals:

- The pattern is a sequence of bits and represents the input queues of vehicles to an intersection.
- The value 0 represents the absence of a vehicle, and the value 1 represents the presence of a vehicle in a tuple composed by a cell and a simulation step (i.e.: a queue of vehicles for a one input cell of the intersection with the pattern “vehicle - void - vehicle” would generate the sequence 101).
- The length of the patterns of each  $i_{ab}$  in the intersection is the same.
- Across the simulation, the pattern is repeated as much as it is necessary until all the simulation steps are completed.

The chromosome codification of the individual is shown in Figure 5. An intersection is composed of arms from 0 to  $n$ , where 0 is the arm selected as the first one to be tagged (the tag value will be 2 elevated to the arm number). We use 0 to start the numbering of the arms to follow the criteria of Section III-A.2. Each lane is numbered from 1 to  $m$  from right to left following the way that enters the intersection in each case. Finally, each lane is provided with a pattern of the same fixed length for every lane in the intersection. If the length of the pattern is  $l_p$ , the codification will be a sequence of bits of length  $l_{ind} = (l_p \cdot m) \cdot (n + 1)$ , where each bit position in the string represents the presence or absence of vehicles in a specific position in the input arm and lane.

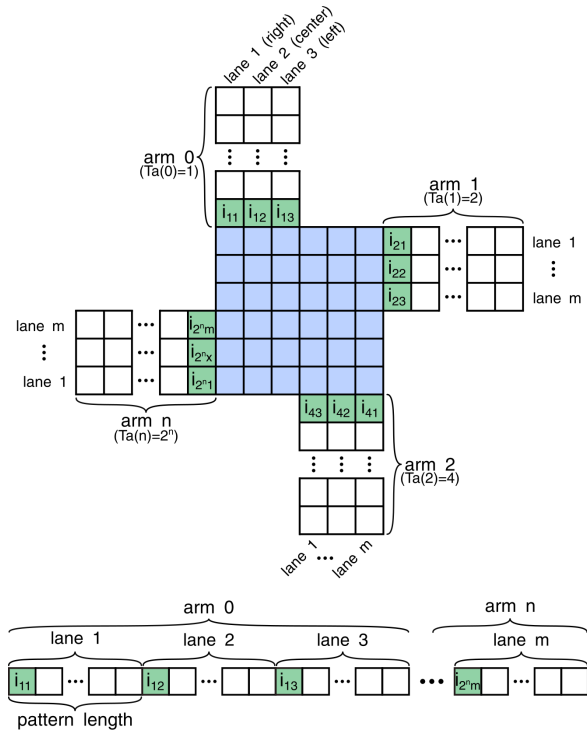


FIGURE 5. Individual chromosome scheme using the values for input patterns in the intersection.

2) GENETIC ALGORITHM

The basic GA behavior is, starting from an initial population, the representation of the evolution of the population in the next generations. Following this principle, the steps of the algorithm are:

- 1) Initial population generation: For a pattern length  $l_p$  a random bit string of length  $l_{ind}$  is generated for every-one. The number of individuals in the population is an important parameter. If it is too small, there would be a low diversity between individuals, which would difficult the generation of a valid solution, and if it is too large, the search space would be too big, and the algorithm could get stuck in local minima (sub-optimal solutions).
- 2) The GA takes the current population and computes the next generation. This is performed by following the following steps:
  - a) Each individual in the current generation is evaluated using the fitness function. This function is described in Section III-C.4 and returns values between 0 and 1, where 0 represents the maximum average intersection occupation without collisions.

In this case, the evaluation consists of a simulation for the input patterns proposed using the cellular

automata to obtain the occupation values for each simulation step.

- b) Every individual in the population is sorted from the lower (best value) to higher (worst value) results from the fitness function.
- c) The best individuals (elite) are automatically selected for the next generation.
- d) The rest of individuals are grouped in random couples and are combined using the crossover function. A new individual (child) of the next generation is created from parts of the chromosomes of each parent individual from the current generation.
- e) Finally, a small group of individuals from the current generation are mutated with random changes in the chromosomes. This process relies on a Gaussian distribution with mean 0 and a variable standard deviation.

The size of each set of individuals to be selected for the elite and crossover steps is defined as configuration parameters. The values used in this work are specified in Section IV-B.

- 3) The algorithm stores the best fitness value obtained in each generation and the differences in scores for every individual in the population.
- 4) The algorithm stops if one of the following conditions is met: i) A pre-established number of generations is reached (this limits the execution time of the algorithm), ii) The difference between the best fitness score and the average of a population is very low (there is no diversity between individuals), iii) The best possible fitness value is reached (0 in this case).
- 5) If the solution obtained at the end of algorithm execution generates collisions in the simulation, it is marked as invalid.

Given that in this problem there are two parameters to be optimized (the length of the pattern and the pattern itself), we have executed the GA repeatedly increasing  $l_p$  from 2 to the size of the maximum path which can be followed by a vehicle in the intersection. For each pattern length, the GA will obtain the pattern of that length which provides the lowest fitness value. At the end of the process, the lowest value of every pattern length tested will determine the best tuple of length and pattern.

### 3) Crossover AND MUTATION

The crossover function takes two individuals (parents) of the population  $p1$  and  $p2$  (randomly selected from those not classified as elite individuals) and exchanges their genes (values in the bit string of the individual) to obtain a new individual  $c1$  (child). The crossover is carried out randomly generating a crossover vector ( $cv$ ) of the same length of the parents. The positions with ones in the vector are the bits which are taken from parent  $p1$ , and the positions of the zeros

determine the genes were taken from parent  $p2$ :

$$\begin{aligned} p1 &= [A \ B \ C \ D \ E \ F \ G] \\ p2 &= [a \ b \ c \ d \ e \ f \ g] \\ cv &= [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1] \\ c1 &= [A \ B \ c \ d \ E \ f \ G] \end{aligned}$$

For the mutation function, we randomly select an individual from the population, and its genes are changed. The selected genes to be changed are selected using a Gaussian distribution with mean 0 and a standard deviation dependent on the individual size and the difference between individuals of the same generation. The result of this function is a new individual which is inserted in the next generation.

### 4) FITNESS FUNCTION

The fitness function design must take into account the goal set in Section III, which was the maximization of the average occupation of the intersection in order to maximize the vehicles flow through it. To achieve this, we have developed a cellular automata simulator that can simulate the vehicle behavior when crossing the intersection according to the proposed model.

This simulator takes as argument the intersection model (Section III-A.2), the optimized paths between inputs and outputs (section III-A.3) and the bit string corresponding to each individual in a GA generation. The simulator decomposes the string to obtain the patterns of each input point  $i_{ab}$  in the intersection and extends it to reach the total simulation time ( $T = t_n - 1$ ). This time is calculated depending on the intersection size and the pattern. Each time instant in the simulation (step) represents the time that a vehicle spends in a cell of the intersection. As every vehicle has the same speed in the scenario, in each simulation step, the position of every vehicle is updated.

The intersection occupation can be variable along the time, so for measuring the average occupation we consider every occupation value only after the intersection reaches a stable state in  $t_i$  (that is, we do not use the first occupation values at the initial simulation steps, when the vehicles are still arriving at the intersection). We guarantee the correct calculation of the average occupation value  $\bar{o}$  by calculating it using the arithmetic mean of the partial occupation values ( $o(t)$ ) from the instant when there has been a complete insertion of all the vehicles of a pattern ( $t_{stable}$ ) and until the end of the simulation ( $t_{end}$ ). This process to calculate the occupation is shown in the Equation 9.

$$\bar{o} = \frac{\sum_{i=t_{stable}}^{i=t_{end}} o(i)}{t_{end} - t_{stable} + 1} \quad (9)$$

The occupation value ( $o(t)$ ) is measured in relationship to the maximum possible occupation ( $I_s$ ) which depends on the number of cells composing the inner grid of the intersection. When two vehicles coincide in time and space during the simulation, there is a collision. We denote this collisions value

as  $C$ . Using these values, we generate the value  $F_V$  returned by the fitness function, following Equation 10.

$$F_V = \frac{(1 - \frac{\bar{o}}{I_s}) + \frac{C}{T}}{2} \quad (10)$$

The result of Equation 10 is a value between 0 and 1. For the maximum average intersection occupation without collisions it will return the minimum value ( $F_V = 0$ ).

#### IV. RESULTS AND DISCUSSION

In this section, we show the experiments carried out to validate the proposed system. In order to compare the results with other solutions, we take as reference a specific intersection type and the results applied to that type found in related works. Additionally, we show the advantages of using a completely automated system for the modeling and optimization of the intersection behavior.

We start by recreating the results of one of the study cases carried out by Azimi *et al.* [5]. After that, we define the experiment we have carried out to apply the proposed system to that same use case, which would allow the comparison of the results obtained in each case.

In addition to the comparative study, we have adapted the test scenario to Simulation of Urban MObility (SUMO) [34], a widely used microscopic and continuous road traffic simulation package. We have defined three simulation scenarios in which the intersection is managed using fixed signaling based on priorities, traffic lights, and the crossing patterns obtained in section IV-B (without signage). This set of tests will allow us to position the proposal in comparison to other systems that are widely used to manage traffic at intersections.

##### A. USE CASE BASED ON BRIP

In Section II the Ballroom Intersection Protocol (BRIP) [6] has been described. Among the intersection types described in Azimi's work, we have chosen the one named "multi-lane intersections of type III" (vehicles on the left lane must turn left, vehicles on the middle lane must go straight, and vehicles on the right lane must turn right). We have selected this type because it is the most complex scenario defined in that work. The author proposes using a Synchronized Intersection Arrival Pattern (SIAP) with a 5-unit length for that scenario. The SIAP proposed are, for each lane:

$$\begin{aligned} L_{right} &= [1 \ 1 \ 1 \ 1 \ 1] \\ L_{middle} &= [0 \ 1 \ 1 \ 0 \ 0] \\ L_{left} &= [1 \ 0 \ 0 \ 0 \ 0] \end{aligned}$$

If we translate these SIAP to the pattern of the model proposed we have:

$$P = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$$

In a symmetric intersection, this pattern would be repeated in each input. Azimi compares the obtained results for this pattern with the results of an intersection controlled by traffic

lights. The author states about these results: "BRIP outperforms the traffic light and stop sign models as it increases the parallelism and allows the simultaneous and continuous crossing of vehicles from all four directions". He includes a results graph which shows that in his simulation the delays (in seconds) of vehicles using his proposal are between a 90% and a 60% lower than using traffic lights.

For the comparison of the system proposed in this paper with the Azimi's work, we have used the pattern previously shown in the developed simulator. The obtained results show that the average occupation is  $\bar{o} = 53.33\%$ . The author uses the Intersection Capacity Utilization (ICU) concept, which is analogous to the  $\bar{o}$  definition in this work. Although for type III intersections Azimi establishes an ICU value of 56.25%, in appendix A-A we show the occupation values in each simulation step which confirm the value  $\bar{o} = 53.33\%$ .

##### B. EXPERIMENT DEFINITION

In this section we describe the experiment designed to validate the model described in Section III and compare it to Azimi's model.

The experiment uses a 6x6 cell intersection, with 4 input and output arms, and 3 lanes in each arm. The model generated for this intersection has been shown in Figure 2. The possible minimum conflict paths have been previously calculated and have been shown in Figure 4. Given that it is not possible to determine *a priori* the optimal length for the pattern, the experiment has been designed to search the best possible patterns with sizes from 2 to 14 (twice the length of the minimum conflict paths calculated) units.

Due to the use of GAs, which could lead to suboptimal solutions (the algorithm can stop at local minimums in the search space), we have repeated the GA execution 100 times for each pattern length. This allows the study of the solutions variability while obtaining results about the system performance.

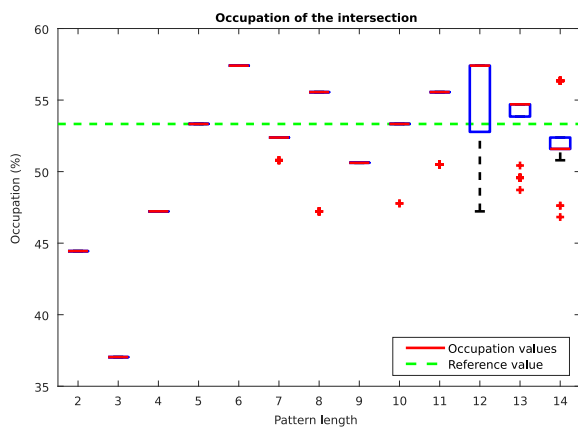
The GA's set up for the experiment is:

- Pattern size: 2 to 14 units.
- Population size: 100 individuals.
- Max. generations: 150.
- Tolerance value (stop condition):  $10^{-12}$ .
- Elite size: 5% of population size.
- Crossover size: 80% of population size.

The experiment has been carried out in a computer using MATLAB R2018a and the Global Optimization Toolbox for the GA development. The GA has been executed using the Parallel Computing Toolbox configured for 4 threads. The intersection modeling and the optimal paths have been implemented using Python 2.7 and the Numpy and NetworkX libraries. The computer specifications are Intel i7-7700 microprocessor with 32 GB (DDR4 2400 MHz) RAM memory and Ubuntu 18.04 operating system.

After running the proposed system over the use case under consideration, we obtain the results shown in Figure 6.

The reference value is drawn with a green dashed line and corresponds with to the average occupation value using



**FIGURE 6.** Average intersection occupations obtained for each pattern size by applying the proposed GA. The green dashed line represents the reference average occupation value for a 5 units pattern size. Red solid lines represent the median of every  $\bar{\delta}$ .

the Azimi proposal for a 5 units pattern size. This value is a 53.33%. For each set of average occupation values obtained for each pattern length, the graph shows, in red solid lines, the median of every  $\bar{\delta}$ . Additionally, for each pattern, the graph shows, in blue solid line boxes, the values between the 25 and the 75 percent of the data in each set. Finally, it shows the outliers with the “+” symbol.

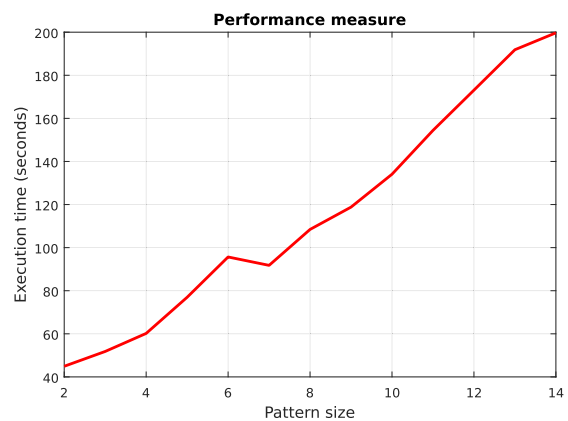
The discussion of this results can be summarized in the following statements:

- The proposed system has been able to obtain automatically the same value than the solution proposed by Azimi, for the same pattern length.
- There are pattern lengths for which this proposal obtains better average occupation values than the Azimi’s solution. For a pattern size of 6 units, we obtain the best average occupation (57.407%). This value is an improvement of almost an 8% from the empirical solution provided by Azimi.
- The set of obtained solutions has a very low variability for patterns sizes from 2 to 11 units.
- When the system reaches a maximum average occupation value for a given pattern size, this same value is obtained for multiples of the size (for instance, for 5 and 10, or for 6 and 12).

In parallel, we have stored the execution times of the GA for each pattern size for the evaluation of the system’s performance. In Figure 7 it is shown the time in seconds spent on average (of the 100 repetitions of the execution) for each pattern size. An almost linear time increase can be observed when the search space is increased due to the pattern size variation.

### C. SIMULATIONS BASED ON A TIME-CONTINUOUS MODEL

The primary objective of traffic simulators is to reproduce the behavior of drivers under certain conditions.



**FIGURE 7.** GA execution performance values for increasing pattern size.

Although SUMO can work with different models, the Krauss car following model [35] is the most common. Generally, the car following model is a method for determining how vehicles follow one another on a specific road. The Krauss model focuses on defining the desired velocity ( $v_{des}$ ), taking a safe velocity ( $v_{safe}$ ) into account. Calculating these velocities requires the use of values including the maximum deceleration possible, reaction time, the maximum velocity of the vehicle, and the time step.

A network has been created in order to adapt the simulation scenario proposed in section IV-B. It consists of an intersection featuring four branches, with three lanes per branch and direction. Due to the characteristics of the simulator, the input and output lanes of the intersection are defined with a length of 300 meters and with a maximum speed of 13.89 m/s (50 km/h). The vehicles are inserted at the start of the intersection’s input lanes at intervals of four, three, or two seconds (corresponding to low, medium, and high traffic volumes, respectively) following the vehicles pattern obtained through the previous experiment.

Three configurations have been made to manage the passage of vehicles through the intersection:

- Priority ( $P$ ): The passage of vehicles is managed using yield signs along the horizontal direction of the intersection (see Figure 8.a). The vehicles traveling vertically have priority over the vehicles traveling horizontally.
- Traffic lights ( $TL$ ): The intersection is managed by traffic lights with 42-second cycles. Vehicles that turn right from any outside lane have a green light throughout the cycle. The possibilities of continuing straight ahead or turning left are divided into periods of equal duration between the horizontal and vertical directions.
- Optimized crossing patterns ( $OCP$ ): Following the proposed system, vehicles reduce their velocity and form groups that follow the patterns established in the meters leading up to the intersection. Velocity is reduced for safety reasons. Once they have crossed the intersection completely or partially (past the possible conflict points

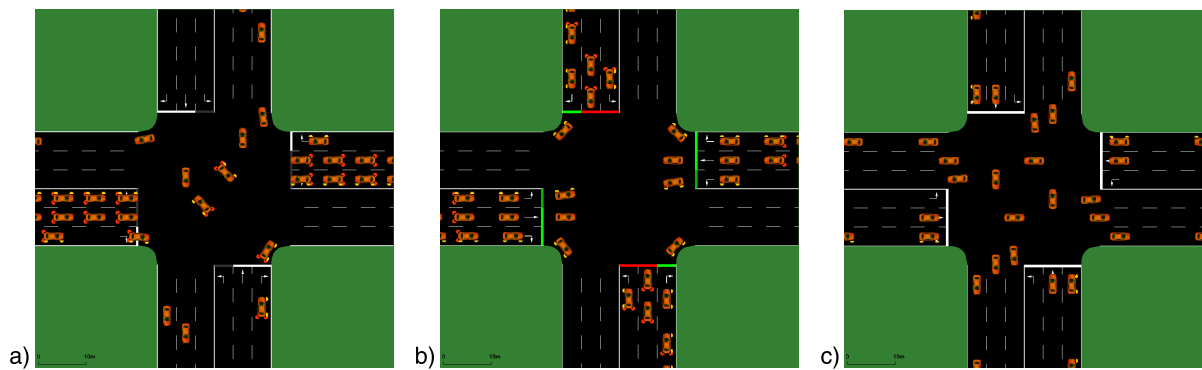


FIGURE 8. Intersection management systems. a) Priority. b) Traffic lights. c) Optimized crossing patterns.

inside it), the vehicles accelerate in order to recover the maximum speed allowed by the system. A simulation is considered valid only if no collisions have occurred by its conclusion.

In order to measure the results of this set of simulations, we decided to carry out a scenario with the vehicles ending their trips between the simulation times of 0s and 3,600s. Since the three methods of managing the intersection involve inserting the same number of vehicles at low, medium, and high traffic volumes, the number of vehicles arriving at their destination gives us a valid system performance metric.

Table 1 shows the results that were obtained, grouped according to whether they resulted from low, medium, or high traffic volumes. Each result is labeled according the origin and destination of the trip and the type of movement it performs within the intersection.

TABLE 1. Simulation results based on the vehicles that have finished their trip during one hour. A total of nine simulations have been carried out combining intersections managed by priority (P), traffic lights (TL), and the optimized crossing patterns (OCP), and low, medium and high traffic volumes.

Turn type	From*	To*	Low			Medium			High		
			P	TL	OCP	P	TL	OCP	P	TL	OCP
Right	1	8	884	887	872	1181	1180	1255	1419	1255	1396
	2	1	887	886	880	1182	1181	1242	1418	1242	1408
	4	2	886	887	872	1182	1180	1250	1417	1250	1396
	8	4	843	887	872	1180	1179	1250	1417	1250	1396
Go straight ahead	1	4	355	440	437	474	586	804	708	804	873
	2	8	269	442	437	168	590	798	52	798	873
	4	1	354	440	438	473	586	796	710	796	877
	8	2	255	444	437	174	857	800	55	800	873
Left	1	2	178	147	146	237	196	189	354	189	291
	2	4	148	146	146	64	194	180	23	180	291
	4	8	177	147	146	236	196	179	354	179	291
	8	1	122	146	147	65	194	183	22	183	293

\* 1,2,4, and 8 values correspond with the top, right, bottom, and left intersection sides, respectively.

Note: All values for P, TL and OCP are indicated in number of vehicles that finish their trips during simulation time.

Observing the values for the vehicles making a right turn (moving without interference), we can see how the value

obtained for all volume levels are very similar. For vehicles that continue to travel straight ahead, one can begin to see the effect caused by intersections that are managed based solely on priority. Even with low traffic volumes, the intersection’s vertical direction (1 to 4 and 4 to 1) allows more vehicles to pass than its horizontal direction (2 to 8 and 8 to 2). This effect becomes more significant as the traffic volume increases. Comparing the results of traffic lights with the proposed system for this type of movement, it can be observed that the proposed system always allows a greater number of vehicles to pass. Finally, the path that presents the most conflict points (turning left) maximizes the effects observed so far. The disparity between the different directions of the priority-based system becomes greater, and for high traffic volumes the number of vehicles managed by the proposed system is 50% higher than what can be achieved using traffic lights.

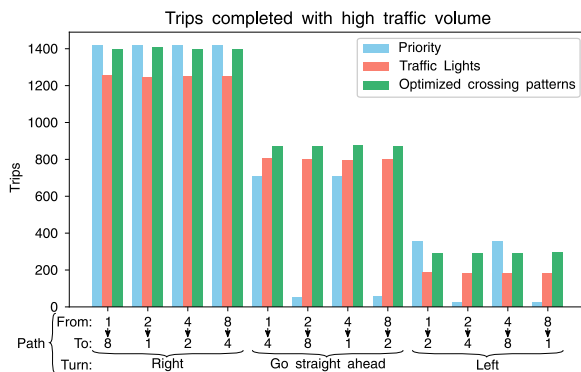
Summarizing the results, the OCP guarantees the input flows to the intersection are almost equal with a 5% of maximum difference. Furthermore, the proposed systems achieve a throughput improvement between 9.21% and 36.98% when compared with P and TL.

Although the advantages of the proposed system have already become evident, Figure 9 provides a graphic comparison of the results obtained for high traffic volumes, seeing as the experiment tests the intersection management systems at their limit. On the x-axis, we have the path followed by each vehicle within the intersection, as well as its branch of origin and destination. On the y-axis, we have the number of completed trips, or in other words, the vehicles that have reached their destination during the first hour of the simulation.

Using this graph, one can make several assertions:

- The paths without interference (turning right) present similar results under the three management systems.
- Both OCP and TL allow for equal use of the intersection among all vehicles, regardless of their point of origin. Meanwhile, the priority-based intersection creates a blockage situation for vehicles originating from nonpriority branches.





**FIGURE 9.** Simulation results for managed intersection with priorities, traffic lights, and the proposed solution (crossing patterns) in high traffic volume conditions.

- The proposed system offers the best average values for vehicles regardless of their origin or destination.
- The reduction in velocity required of vehicles under the proposed system in order to allow for the formation of groups is compensated for by the fact that it also provides a constant rate of vehicle passage.

## V. CONCLUSION

In this article, we have described an automated traffic intersection optimization system which is able to model, generate paths with a lower degree of interference between them, and optimize the arrival patterns of vehicles at an intersection, automatically to maximize their performance. Based on the objectives set in Section I, we can conclude the following:

- The proposed modeling system is capable, automatically, of labeling each element of any type of intersection univocally, containing the necessary information for the next steps of the system.
- The methodology proposed to calculate the shortest paths with minimum conflict points between them is capable of obtaining the best possible solution. The performance level of this methodology is very high. In the experiment performed, the methodology was able to calculate more than 80,000 routes in less than 4 seconds.
- The GA proposal used has shown its utility in getting to the same result for the same length of vehicle pattern as the reference work, but automatically. In addition, it has achieved better results using other pattern sizes.

In general, this system has provided better results than previous studies based on the empirical study of specific intersections, with the advantage that this method can be applied to any type of intersection. This factor is fundamental since, in a real traffic scenario, the number of intersections present in a road network can be very high and having an automatic optimization system facilitates the global optimization of the network. In addition, the comparison of the results with other traditional intersection management methods based on priorities and traffic lights shows the advantages of the proposed system.

This system has been designed to be generic, so that it can be applied to all sorts of intersections, not just those where the entrance flows are symmetrical. Our most immediate future work will be to study the variation the functions of the route calculations and the fitness function of the GA to prioritize specific flows at the intersection according to the convenience of the network design.

## APPENDIX A GRAPHICAL RESULTS

In this appendix, we include the graphical results of the simulation steps carried out with the cellular automata model developed. In Figures 10 and 11 we show the occupation of cells composing the inner space of an intersection and the 5 cells closer to the intersection, associated to the input and output lanes. The white empty cells represent the absence of vehicles. The occupation value ( $o(t)$ ) is shown over each simulation step, being  $t$  the current simulation step. In appendices V-A and V-B we explain the calculation of the average occupation value ( $\bar{o}$ ) in each case, using the occupation values shown in the figures.

### A. BRIP APPLIED TO TYPE III INTERSECTION

Figure 10 shows the results obtained when inserting the following 5 unit pattern in the intersection in a symmetric way:

$$P = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]$$

Which, once decomposed in lanes, would be:

$$L_{right} = [1 \ 1 \ 1 \ 1 \ 1]$$

$$L_{middle} = [0 \ 1 \ 1 \ 0 \ 0]$$

$$L_{left} = [1 \ 0 \ 0 \ 0 \ 0]$$

When  $t = 6$ , all vehicles, following the pattern, have arrived at the cells adjacent to the inner intersection. Given that the maximum size obtained for the minimum conflict paths is 7 units, we must wait for 6 simulation steps to ensure that the vehicle following the longest path has been able to complete a cycle (from  $t = 7$  to  $t = 12$ ).

After  $T = 12$ , the simulation enters a stable state, where it will receive, in a repetitive way, the same input patterns. Is in this step ( $t = 13$ ) where we start measuring the occupation values to obtain the average value during the simulation, which applying Equation 9 results as follows:

$$\bar{o} = \frac{\sum_{t=13}^{t=32} o(t)}{(32 - 13 + 1)} = 53.33\%$$

The simulation time is 12 units (step when the simulation enters the stable state) plus 4 times the pattern size.

### B. PROPOSED SOLUTION APPLIED TO TYPE III INTERSECTION

Figure 11 shows the results obtained when inserting the following 6 unit pattern in the intersection in a symmetric way:

$$P = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$$

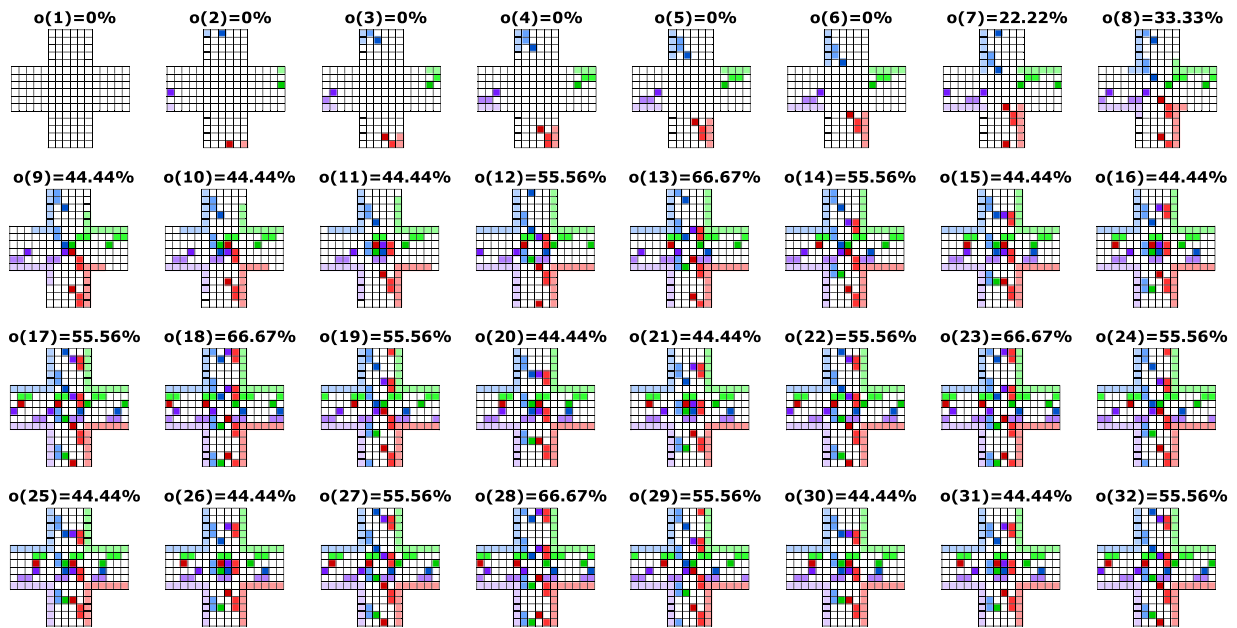


FIGURE 10. Graphical representation of the vehicles positions in each simulation step following the SIAP proposed by Azimi.

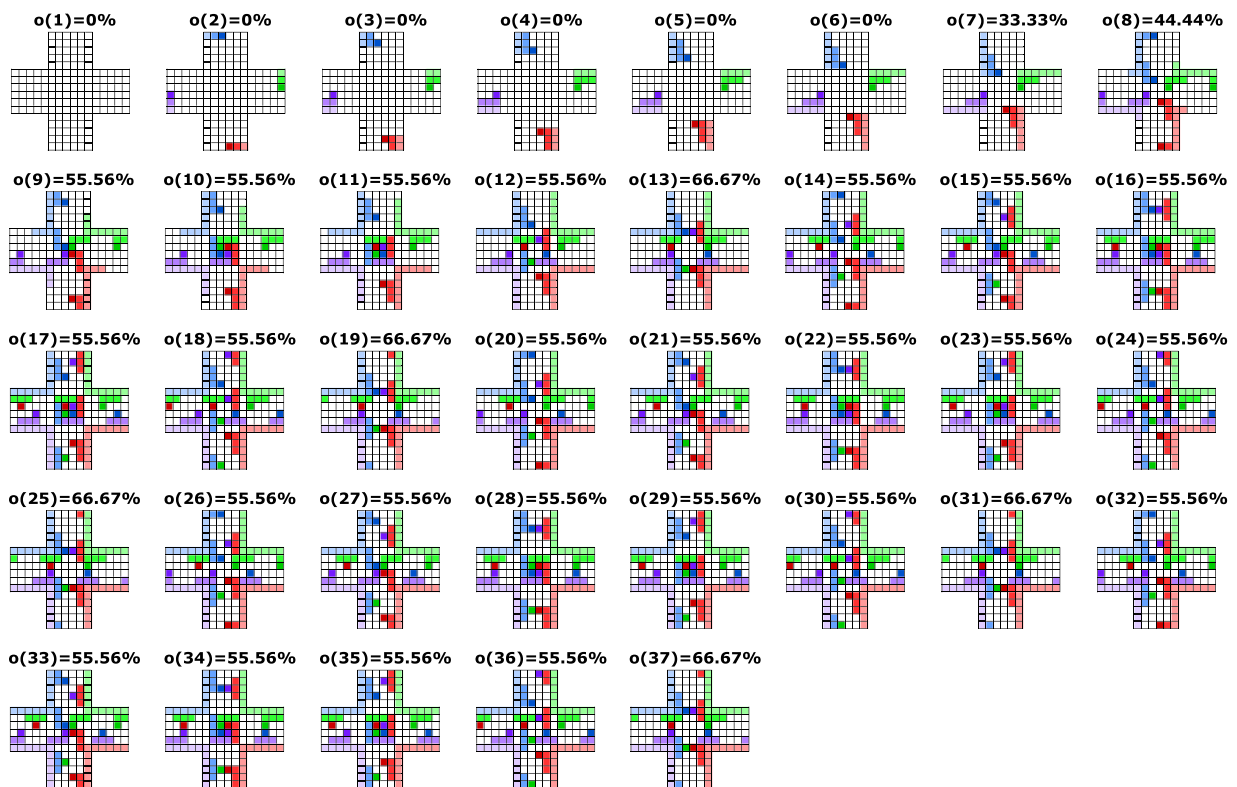


FIGURE 11. Graphical representation of the vehicles positions in each simulation step following the best pattern obtained in the carried-out experiments.

Which, once decomposed in lanes, would be:

$$\begin{aligned} L_{right} &= [1 \ 1 \ 1 \ 1 \ 1 \ 1] \\ L_{middle} &= [1 \ 1 \ 1 \ 0 \ 0 \ 0] \\ L_{left} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0] \end{aligned}$$

Following the same reasoning than in the previous section, when  $t = 6$ , the vehicles, following the pattern, have arrived at the cells adjacent to the inner intersection. As in this case, the maximum size obtained for the minimum conflict paths is also 7 units, we must wait for these 7 simulation steps (the pattern length is greater than in the previous case) to ensure that the vehicle following the longest path has been able to complete a cycle (from  $t = 7$  to  $t = 13$ ).

After this instant, the simulation enters a stable state, where it will receive, in a repetitive way, the same input patterns. Is in this step ( $t = 14$ ) where we start measuring the occupation values to obtain the average value during the simulation, being the calculation:

From  $T = 14$ , the simulation enters a stable state and therefore the process to obtain the average intersection occupation begins. For this pattern size, the simulation ends in  $T = 37$ . Applying these values to the Equation 9, the average occupation is:

$$\bar{o} = \frac{\sum_{t=14}^{t=37} o(t)}{(37 - 14 + 1)} = 57.407\%$$

## REFERENCES

- [1] G. Cookson. (Feb. 2018). *INRIX Global Traffic Scorecard*. [Online]. Available: <http://inrix.com/scorecard/>
- [2] J. D. Leonard, II, B. Ramanathan, and W. Recker, "A real-time information processing algorithm for the evaluation and implementation of ATMS strategies," in *Proc. Intell. Vehicles Symp.*, Jun. 1992, pp. 225–229.
- [3] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Standards J3016\_201806, Society of Automotive Engineers (SAE), Jun. 2018. [Online]. Available: [https://www.sae.org/standards/content/j3016\\_201806/](https://www.sae.org/standards/content/j3016_201806/)
- [4] P. Bansal and K. M. Kockelman, "Forecasting Americans' long-term adoption of connected and autonomous vehicle technologies," *Transp. Res. A, Policy Pract.*, vol. 95, pp. 49–63, Jan. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0965856415300628>
- [5] R. Azimi, "Co-operative driving at intersections using vehicular networks and vehicle-resident sensing," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, Jul. 2015. [Online]. Available: [https://figshare.com/articles/Co-operative\\_Driving\\_at\\_Intersections\\_using\\_Vehicular\\_Networks\\_and\\_Vehicle-Resident\\_Sensing/6715379](https://figshare.com/articles/Co-operative_Driving_at_Intersections_using_Vehicular_Networks_and_Vehicle-Resident_Sensing/6715379)
- [6] R. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Ballroom intersection protocol: Synchronous autonomous driving at intersections," in *Proc. IEEE 21st Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2015, pp. 167–175, doi: [10.1109/RTCSA.2015.20](https://doi.org/10.1109/RTCSA.2015.20).
- [7] D. J. Findley, "Traffic engineering studies," in *Traffic Engineering Handbook: Institute of Transportation Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2016, pp. 109–146.
- [8] A. Smiley and R. E. Dewar, "Road users," in *Traffic Engineering Handbook: Institute of Transportation Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2016, pp. 51–107.
- [9] J. R. Riegner, "Design and operation of complete streets and intersections," in *Traffic Engineering Handbook: Institute of Transportation Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2016, pp. 367–398.
- [10] A. Pande and B. Wolshon, "Design and control for interrupted traffic flow through intersections," in *Traffic Engineering Handbook: Institute of Transportation Engineers*, 7th ed. Hoboken, NJ, USA: Wiley, 2016, pp. 321–366.
- [11] S. Xie and H. Jiang, "Increasing the capacity of signalized intersections by allocating exit lanes to turning movements," *J. Adv. Transp.*, vol. 50, no. 8, pp. 2239–2265, Dec. 2016, doi: [10.1002/atr.1457](https://doi.org/10.1002/atr.1457).
- [12] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.
- [13] R. Akcelik, M. Besley, and E. Chung, "An evaluation of scats master isolated control," in *Proc. 19th ARRB Transp. Res. Conf. (Transport)*, 1998, pp. 1–24.
- [14] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time—the SCOOT method," *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 11–15, Feb. 1991.
- [15] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 763–768.
- [16] Y. Xuan, C. F. Daganzo, and M. J. Cassidy, "Increasing the capacity of signalized intersections with separate left turn phases," *Transp. Res. B, Methodol.*, vol. 45, no. 5, pp. 769–781, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0191261511000312>
- [17] W. Ma, Y. Liu, J. Zhao, and N. Wu, "Increasing the capacity of signalized intersections with left-turn waiting areas," *Transp. Res. A, Policy Pract.*, vol. 105, pp. 181–196, Nov. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0965856415300112>
- [18] C. Yan, H. Jiang, and S. Xie, "Capacity optimization of an isolated intersection under the phase swap sorting strategy," *Transp. Res. B, Methodol.*, vol. 60, pp. 85–106, Feb. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0191261513002208>
- [19] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitvat, and O. K. Tonguz, "Self-organized traffic control," in *Proc. 7th ACM Int. Workshop Veh. InterNetworking (VANET)*, New York, NY, USA, 2010, pp. 85–90, doi: [10.1145/1860058.1860077](https://doi.org/10.1145/1860058.1860077).
- [20] L. Cruz-Piris, D. Rivera, S. Fernandez, and I. Marsa-Maestre, "Optimized sensor network and multi-agent decision support for smart traffic light management," *Sensors*, vol. 18, no. 2, p. 435, 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/2/435>
- [21] M. B. Younes and A. Boukerche, "An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems," *Wireless Netw.*, vol. 24, no. 7, pp. 2451–2463, 2018, doi: [10.1007/s11276-017-1482-5](https://doi.org/10.1007/s11276-017-1482-5).
- [22] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 81–90, Mar. 2012.
- [23] O. Castillo, R. Martínez-Marroquín, P. Melin, F. Valdez, and J. Soria, "Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot," *Inf. Sci.*, vol. 192, pp. 19–38, Jun. 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025510001003>
- [24] H. S. E. Chuo, M. K. Tan, A. C. H. Chong, R. K. Y. Chin, and K. T. K. Teo, "Evolvable traffic signal control for intersection congestion alleviation with enhanced particle swarm optimisation," in *Proc. IEEE 2nd Int. Conf. Autom. Control Intell. Syst. (I2CACIS)*, Oct. 2017, pp. 92–97.
- [25] W. Hao, C. Ma, B. Moghimi, Y. Fan, and Z. Gao, "Robust optimization of signal control parameters for unsaturated intersection based on tabu search-artificial bee colony algorithm," *IEEE Access*, vol. 6, pp. 32015–32022, 2018.
- [26] F. Yan, M. Dridi, and A. El Moudni, "An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach," *Int. J. Appl. Math. Comput. Sci.*, vol. 23, no. 1, pp. 183–200, 2013. [Online]. Available: <https://content.sciendo.com/view/journals/amcs/23/1/article-p183.xml>
- [27] Q. Lu and K.-D. Kim, "A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic," *J. Adv. Transp.*, vol. 2017, Oct. 2017, Art. no. 7318917. [Online]. Available: <https://www.hindawi.com/journals/jat/2017/7318917/cta/>
- [28] S. P. Hoogendoorn and P. H. L. Bovy, "State-of-the-art of vehicular traffic flow modelling," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 215, no. 4, pp. 283–303, 2001, doi: [10.1177/095965180121500402](https://doi.org/10.1177/095965180121500402).
- [29] S. Maerivoet and B. De Moor, "Cellular automata models of road traffic," *Phys. Rep.*, vol. 419, no. 1, pp. 1–64, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0370157305003315>

- [30] O. K. Tonguz, W. Viriyasitavat, and F. Bai, "Modeling urban traffic: A cellular automata approach," *IEEE Commun. Mag.*, vol. 47, no. 5, pp. 142–150, May 2009.
- [31] H.-T. Zhao, X.-R. Liu, X.-X. Chen, and J.-C. Lu, "Cellular automata model for traffic flow at intersections in Internet of vehicles," *Phys. A, Stat. Mech. Appl.*, vol. 494, pp. 40–51, Mar. 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378437117312335>
- [32] D. E. Wolf, "Cellular automata for traffic simulations," *Phys. A, Stat. Mech. Appl.*, vol. 263, nos. 1–4, pp. 438–451, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378437198005366>
- [33] D. Whitley, "A genetic algorithm tutorial," *Statist. Comput.*, vol. 4, no. 2, pp. 65–85, Jun. 1994, doi: 10.1007/BF00175354.
- [34] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, nos. 3–4, pp. 128–138, Dec. 2012.
- [35] S. Krauss, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," Ph.D. dissertation, Fac. Math. Natural Sci., Univ. Cologne, Köln, Germany, 1998.



**LUIS CRUZ-PIRIS** received the Telematics Engineering degree and the master's degree in telecommunication engineering from the Universidad de Alcalá, in 2012 and 2016, respectively, where he is currently pursuing the Ph.D. degree. Since 2013, he has been involved in multiple research projects related to multimodal transportation systems, simulation, and cybersecurity. He is currently a Research Staff with the Telematics Service Engineering Research Group, Universidad de Alcalá. His research interests include intelligent agents, the IoT architectures, and intelligent transportation systems.



**MIGUEL A. LOPEZ-CARMONA** received the B.S. degree in electronics engineering from the Universidad de Alcalá, in 1994, the M.S. degree in telecommunication engineering from the Polytechnic University of Madrid, in 1999, and the Ph.D. degree in telecommunication engineering from the Universidad de Alcalá, in 2006. From 1995 to 2000, he held positions in Logytel and Alcatel as a Project Manager and a Research Scientist in the areas of railways and defense.

Since 2000, he has been an Associate Professor with the Department of Computer Engineering, Universidad de Alcalá. From 2010 to 2017, he was the Vice-Dean of the Polytechnic School, Universidad de Alcalá. He was a Visiting Researcher with the Center for Collective Intelligence, Massachusetts Institute of Technology, from 2010 to 2011, where he was involved in complex automated negotiations and optimization of complex networks. He has around 100 publications in journals, book chapters, and conferences. His research interests include intelligent transport systems, automated negotiation, multi-agent systems, and the optimization of complex networks.



**IVAN MARSMA-MAESTRE** received the degree in telecommunication engineering and the Ph.D. degree from the Universidad de Alcalá, Spain, in 2003 and 2009, respectively, where he is currently an Associate Professor with the Computer Engineering Department. He has taken part in many public and private research projects in these matters, has a number of publications in high impact international conferences and journals, and serves as the Program Chair and a Reviewer for some of them. His research interests include the use of negotiation and nonlinear optimization techniques for distributed coordination of complex systems, such as computer networks, supply chains, or vehicle management systems. From his research have emerged collaborative research lines with international research groups, such as the Center for Green Computing, the Nagoya Institute of Technology, Japan, the Technical University of Delft, The Netherlands, or the Center for Collective Intelligence, Massachusetts Institute of Technology, USA.

...



# A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem

En este capítulo se muestra el artículo publicado “Un Algoritmo Genético con cromosomas de longitud variable para resolver el problema de coordinación tráfico urbano multicamino”. Tras un breve resumen de éste, se muestran los datos sobre la publicación y su relevancia. Por último, se ha incluido una copia de la versión publicada del artículo.

## 5.1 Introducción

Los problemas relacionados con la coordinación del tráfico en intersecciones son muy habituales en las grandes ciudades. Las soluciones más comunes que se le dan a estos problemas actualmente son la gestión de prioridades de forma estática a través de intersecciones sin señalizar o señalizadas utilizando cedas el paso, el uso de señalización variable como los semáforos, o la transformación de las intersecciones en otras estructuras como pueden ser la rotondas.

La creación, evolución y consolidación de tecnologías que dan soporte al paradigma el coche autónomo y conectado ha permitido el desarrollo de nuevas soluciones donde los vehículos siguen caminos prefijados sin detenerse a su paso por los cruces o intersecciones.

En este trabajo se propone el uso de un Algoritmo Genético con cromosomas de longitud variable para resolver el problema de la coordinación de vehículos multicamino en intersecciones. El algoritmo propuesto se centra en el proceso de optimización de la secuencia de llegada de vehículos a cada entrada de la intersección siguiendo los flujos preestablecidos. Mientras otras soluciones asumen que las tasas de llegada son iguales en todas las entradas de la intersección, nuestra propuesta permite trabajar con flujos de tráfico asimétricos. Para esta solución, extendemos nuestro modelo de intersección anterior basado en caminos internos fijos a otro con múltiples caminos.

Es decir, en este nuevo modelo cada vehículo puede ir desde cualquier punto de entrada de la intersección hacia cualquiera de sus ramas de salida. Además, se han propuesto operadores de selección, cruce y mutación especialmente adaptados a este problema. De igual forma, se ha diseñado una nueva metodología para llevar a cabo el cruce entre individuos de la población de diferente tamaño teniendo en cuenta las peculiaridades de su codificación.

Para validar esta propuesta se han realizado diferentes pruebas utilizando valores de entrada al sistema con solución conocida y valores aleatorios. Los resultados han sido comparados con otros optimizadores conocidos obteniendo una mejora en los puntuaciones de más del 9.1 % y del 126 % en cuanto a tiempo de ejecución.

## 5.2 Relevancia de la publicación

Título:	A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem		
Autores:	Luis Cruz-Piris, Ivan Marsa-Maestre y Miguel A. Lopez-Carmona		
Estado:	Publicado		
Fecha de publicación:	15/08/2019		
DOI:	<a href="https://doi.org/10.1109/ACCESS.2019.2935041">https://doi.org/10.1109/ACCESS.2019.2935041</a>		
Revista:	Nombre:	IEEE Access	
	Editorial:	IEEE	
	País:	EEUU	
Relevancia	Revista:	Índice de impacto	4.098
		JCR (2018*):	Q1 (18/88)
	Artículo:	Lecturas desde su publicación <sup>1</sup> :	54
		Citas <sup>2</sup> :	-

\* Último JCR disponible a 15/09/2019

<sup>1</sup> Fuente: IEEE Xplore, Fecha de consulta: 15/09/2019

<sup>2</sup> Sin valores a 15/09/2019

**Tabla 5.1.:** Datos asociados a la publicación: A Variable-Length Chromosome Genetic Algorithm to Solve a Road Traffic Coordination Multipath Problem.

## 5.3 Artículo

Received July 27, 2019, accepted August 9, 2019, date of publication August 13, 2019, date of current version August 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935041

# A Variable-Length Chromosome Genetic Algorithm to Solve a Road Traffic Coordination Multipath Problem

LUIS CRUZ-PIRIS<sup>1</sup>, IVAN MARSA-MAESTRE<sup>1</sup>, AND MIGUEL A. LOPEZ-CARMONA<sup>1</sup>

Departamento de Automática, Escuela Politécnica Superior, Universidad de Alcalá, 28871 Alcalá de Henares, Spain

Corresponding author: Luis Cruz-Piris (luis.cruz@uah.es)

This work was supported in part by the Spanish Ministry of Economy, Industry, and Competitiveness under Grant TIN2016-80622-P (AEI/FEDER, UE) and Grant TEC2013-45183-R (AEI/FEDER, UE), and in part by the University of Alcalá under Grant CCG2018-EXP-041.

**ABSTRACT** The problems related to traffic coordination in intersections are quite common in large cities. Current solutions are based on the utilization of static priorities (i.e. yield signs), on variable signaling like traffic lights, or even on the physical modification of the road structures by transforming intersections in roundabouts. The emergence, evolution, and consolidation of technologies that enable the paradigm of connected and autonomous vehicles have allowed the development of new solutions where the vehicles' coordination follow a preset path without stopping when entering the intersections. In this work, we propose using a genetic algorithm with variable-length chromosomes to solve the vehicle coordination multipath problem in intersections. The proposed algorithm is focused on optimizing the vehicles' arrival sequencing according to preset flow rates. While other solutions assume the same flow rates in every branch of the intersection, in our proposal the traffic flows can be asymmetric. We extend one of the existent intersection models, based on fixed paths, to allow multiple paths. This means that each vehicle can go from any input point to any output branch in the intersection. Moreover, we have designed specific selection, crossover and mutation operators, and a new methodology to carry out the crossover function between different sized individuals, which are adapted to the specific peculiarities of the problem. Our proposal has been validated by carrying out tests using input data with known solutions and with random data. The results have been compared with systems based on other optimizers, obtaining improved results in the fitness outcome up to 9.1%, and up to 126% in computation time.

**INDEX TERMS** Cooperative systems, genetic algorithms, intelligent vehicles, road traffic intersection, optimization.

## I. INTRODUCTION

In road traffic systems, a particular problem in terms of security and efficiency is to achieve the coordination of vehicles in intersections, roundabouts, and ramps [1]. In the United States, over 40% of the almost 6 million accidents produced over a year are intersection-related crashes [2]. This is due to the fact that intersections are a natural path confluence zone, and therefore, are prone to collisions. The evolution of the technologies used in Intelligent Transport Systems (ITS) has allowed an increment on the generation and exchange of

information between vehicles (V2V), and between vehicles and the infrastructure (V2I). In the same way, new paradigms such as Connected and Autonomous Vehicles (CAVs), are expected to experiment a huge expansion in the following years. In the year 2030, about 98% of vehicles are expected to be connected, due to the decreasing prices of the related technology and a favorable legislation [3]. These technologies are being applied to reduce the accidents in confluence zones through the improvement in traffic coordination techniques.

This work is focused on obtaining synchronized vehicle sequences according to desired preset traffic flows. This optimization process must guarantee a safe crossing while

The associate editor coordinating the review of this article and approving it for publication was Yang Tang.



complying with the destination preferences (that is, their turn selection) of each vehicle.

According to the available literature, there are multiple works related to the vehicles' coordination techniques [4]. For instance, the vehicle decision-making process about choosing to cross or to give way to an oncoming vehicle is studied in works as [5]. Also, in [6] the authors apply game theory techniques over "Cooperative Adaptive Cruise Control" equipped vehicles to solve this problem. It is possible to classify the works related to the vehicles' coordination in intersections in those which perform a study of the vehicle coordination in a single intersection [7], [8], and those focused on the coordination of two or more intersections [9]–[11]. Another classification criterion could divide them into systems for the management of intersections using fixed or variable traffic signaling [12], and systems which opt to variate the arrival times of vehicles to the conflict zones, without using any visible signaling [13]. Traffic lights based on constant time cycles, along with the yield signs, have been the most common way of managing intersections. Nowadays, and partially because of the emergence of vehicle-to-everything (V2X) communication systems [14], other dynamic programming schemes for traffic lights are possible. The use of fuzzy logic traffic light systems considering the traffic levels in each input point of an intersection has been found useful in [15] applied to a specific use case in Kuwait city. Other fuzzy logic controllers working in cooperation with Wireless Sensor Networks (WSN) can be found in [16] which analyze possible turns of vehicles and manage green time and cycle change of traffic lights. Although these works improve traffic light programming for the management of isolated intersections, some vehicles still must wait to access them.

From the point of view of possible heuristic search systems which could be used to solve the vehicle coordination problems, Genetic Algorithms (GA) have been extensively used and have provided good results. Following a traditional focus on traffic lights management, in [17] the authors propose a GA-based architecture, traffic micro-simulations and cluster computing mechanisms for the adjustment of traffic lights phases in a city. The proposed system is effective in reducing the waiting times of vehicles. Specifically, there are solutions that manage the vehicle crossing at an intersection by calculating an orderly sequence of arrivals, where each vehicle follows a specific path that avoids collisions without the need to alter their speed when they enter the intersection. We refer to these arrival sequences as patterns in this work. In [18] the authors propose an intersection hybrid management system where, when an emergency vehicle is close to the intersection, a GA calculates efficient sequences to prioritize this type of vehicle. The rest of the time, the system uses a conventional management system depending on the vehicle's trajectories. In a similar way, [19] uses a GA to obtain optimal vehicle sequences. Vehicle speed varies based on these sequences, allowing the vehicles to cross the intersections without stopping. These works behave in a reactive way depending on the

network conditions. In [20], instead, the research focuses on generating the incoming vehicle sequences (or patterns) that maximize resource utilization in the intersection. If vehicles in the network adapt themselves to these optimal patterns, the global efficiency of the system may be increased. Taking these previous works into account, we have chosen to use a genetic algorithm as the basis of the proposed optimization process.

Regarding the codification of the problem, the works referenced in the previous paragraphs that rely on a GA use fixed-length individuals. However, this has the drawback that the pattern length that allows an optimal solution to the optimization problem is not known *a priori*. Because of that, if we want to obtain the optimal arrival sequences of vehicles at an intersection, we must initiate several optimization processes. Harvey studies in [21] the possible techniques which could be applied to crossover operators in GA where variable-length genotypes (VLGs) are used. That work is focused mainly on how the crossover points should be selected to guarantee that the parents' genotypes represent the same type of information. He proposes using a recombination operator able to, given a crossover point in the primary genotype, choose a complementary crossover point in the other parent. For a different domain, in [22], the authors review the existing solutions using variable-length genomes and the effects of this codification in the algorithm convergence. Another example of this type of genomes applied to event planning can be found in [23], where the solution of a problem can be obtained applying a variable number of steps. Focusing on traffic-related problems we find examples such as [24], where Qiongbing and Lixin use a GA with variable-length chromosomes to solve path optimization problems. Each gene of the chromosome represents a point in the traffic network, and a set of genes determine a path. In this case, the authors improve the crossover operator that can be used in this kind of scenarios. The previously referenced methods for variable-length genotype crossover can be summarized in three main concepts: optimal selection of crossover points (in different-size genotypes, which points refer to the same type of information), crossover process (how to generate a new individual and update its genotype after the process), and the addition or removal of independent value genes (for instance, adding an extra step in a process or adding a new node in a path).

Every intersection may have a different shape and restrictions. In our previous work on intersection management [20], based on fixed paths between input and output points, we showed that the diversity in geometrical shape and usage patterns in real-world intersections pose a challenge for optimization and that careful modeling is critical for optimization to succeed. In this work, we generalize this approach which allows us to use one single optimization process to reach an optimal or suboptimal solution, independently of the vehicle paths and rates that are desired to be managed by the intersection, and of the pattern size needed to achieve those goals. Consequently, in this paper, we extend the

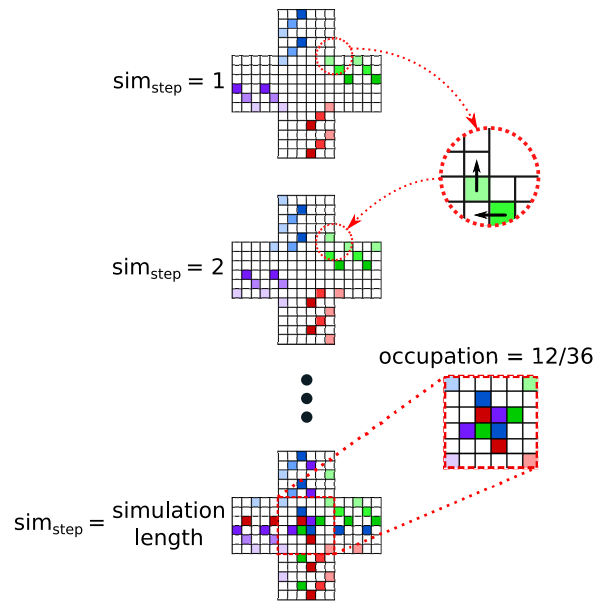
Optimized Crossing Patterns (OCP) system by removing the turn restrictions and allowing to go from any input point to any output branch in the intersection. Moreover, to improve the algorithm performance, we have modeled the problem using a variable-length genotype Genetic Algorithm. This system allows obtaining the optimal vehicles' arrival sequencing according to preset flow rates along with the shortest pattern length in a single optimization process. Our contributions to this field are the following:

- We have extended our previous intersection model (OCP) by adding the possibility for each vehicle to go freely from any input point to any output branch. We have named this new version Optimized Crossing Patterns with Multiple Paths (OCP-MP). This allows expanding the search space, and therefore potentially improve the system performance.
- We have proposed an optimizer based on a variable-length chromosomes GA that allows obtaining the optimal arrival sequences of vehicles in a single execution (Optimized Crossing Patterns with Multiple Paths and Variable Length or OCP-MP-VL).
- We have defined a custom method to perform the crossover process between individuals of different lengths based on the contraction and expansion of chromosome genes adapted to our problem.
- We have proposed a new methodology for the generation of initial populations and custom crossover and mutation operators adapted to our particular problem.
- Our proposal has been validated using data with a known solution and random data. The results have been compared with the results obtained using other optimizers.

The rest of this paper has been organized in the following way. In section II we show the fundamental elements of the microscopic simulator used in this work and the OCP system [20] which is used as the base for our proposal. In section III we describe the proposed solution for solving the traffic coordination problem. In that section, we define the improvements adopted over the OCP model and the new methodology to obtain the best vehicle arrival rate using a GA (the initial population generation mechanism, and the selection, mutation and crossover operators designed to use variable-length chromosomes). Moreover, we define the fitness function which results will be the value to minimize by the proposed GA. After that, we define the set of experiments for the validation of the proposal by comparing it to other optimization methods for the same problem (Section IV). We discuss these results in Section V. Finally, we explain our conclusions and future work lines (Section VI).

**II. BACKGROUND**

In this section, we briefly describe the microscopic simulator based on Traffic Cellular Automata (TCA) and the OCP method we proposed in [20], which will be used as the basis for this work.



**FIGURE 1. Diagram of the basic behavior of the microscopic simulator based on a Traffic Cellular Automata (TCA).**

The modeling process for a cellular automata which aims to simulate the behavior of vehicles crossing an intersection is divided into two phases: the modeling of the intersection and the behavior of the simulator itself. In the modeling phase, the geographic space is divided using a two-dimension grid. Depending on the specific shape of the intersection, the size of each division will be different. The movements available for vehicles are modeled following the von Neumann model, which means that they can continue from one cell to any one of the 4 adjacent cells. Each cell stores the specific turn restrictions applied in each case, so it is possible that some of the movements are restricted in some cells. In each simulation step, the vehicles can advance just one cell, and the total simulation time will be dependent on the size of the pattern to simulate and the size of the maximum route in the system. The implemented simulator can store the cells' occupation in each simulation step, being able to return metrics about minimum occupation, maximum occupation and average occupation in the intersection, turn flows from each branch, and the minimum, maximum and average collisions. A collision is detected when 2 or more vehicles use the same cell in the same simulation step. The process for moving two vehicles and the collision detection processes are shown in Figure 1.

The optimization problem raised in [20] had as a goal the generation of the maximum possible intersection utilization. To achieve this goal, it used a fitness function based on the maximum average occupation of the intersection during the simulation time. The OCP methodology includes the intersection modeling process and a labeling system for each element of the model, the calculation of the paths from input to output points in the intersection, and the optimizer based on a genetic

algorithm which obtains the ideal crossing patterns. OCP uses the vehicle position in a specific lane to determine its route. This method facilitates the algorithm convergence and simplifies the problem codification but introduces a restriction in the maximum traffic flow that can be directed to a specific output. OCP has three main limitations: The first one is that vehicle routes are limited by their position in the input lanes of the intersection. The second one is that the optimization process must be executed for each pattern length, so it is necessary to repeat the GA for each pattern length to test. Finally, OCP is focused on obtaining the maximum average occupancy of the intersection, regardless of desired traffic flow rates.

### III. PROPOSED GENETIC ALGORITHM WITH AUTOMATED CHROMOSOME LENGTH EXPANSION AND CONTRACTION

#### A. PROBLEM FORMULATION

We model the problem of traffic coordination in road intersections in a similar way as [1], [11]. Accordingly, we define  $\Gamma_i$  as the space occupied by a vehicle when following a path between two points in the traffic network depending on the vehicle size. Given two or more vehicles, if there exists a non-null zone such that  $\Gamma_c = \Gamma_1 \cap \Gamma_2 \cap \dots \cap \Gamma_n$ ,  $\Gamma_c$  represents a critical region in the traffic network, where it is possible that collisions between vehicles are produced. Once the  $\Gamma_i$  paths which can generate critical regions (affecting the safety constraints of the problem) are established, the cost for each vehicle is denoted by  $J_i(x_i(t), u_i(t)) = \int_0^{+\infty} \Lambda_i(x_i(t), u_i(t), t) dt$  where  $x_i$  and  $u_i$  are the state and input/control vectors, respectively, and the stage cost  $\Lambda_i(x_i(t), u_i(t), t) dt$  varies depending on the parameter to optimize (instantaneous power consumption, deviation from a target speed, etc.). Using this formulation, the N-vehicle optimal coordination problem can be treated as a constrained optimal control problem where the goal is to minimize  $\sum_{i=1}^N J_i(x_i(t), u_i(t))$ . In our case, we need to solve the problem as a combinatorial problem with constraints which grows exponentially depending on the path options and pattern length, or features of the intersection such as its number of branches, lanes per branch and its shape. That is, any variation of these parameters can change the number of critical regions  $\Gamma_c$ , modifying the complexity of the problem. An example of convergence and complexity results of a similar problem is shown in [25]. In our research work, the vehicle dynamics and the associated metrics described mathematically in the previous lines (route times, intersection occupation, collisions, etc.) are managed by the cellular automata intersection simulator shown in section II.

In the following, we define the proposed methods and techniques to obtain the best vehicle patterns according to their turn preferences. The resulting patterns induce vehicle arrival rates which enable safe crossings at the intersection while respecting destination preferences.

#### B. PROBLEM ENCODING

The method we proposed in [20], which defines the OCP system, allows the univocal identification of each element in an intersection. OCP encodes the presence or absence of vehi-

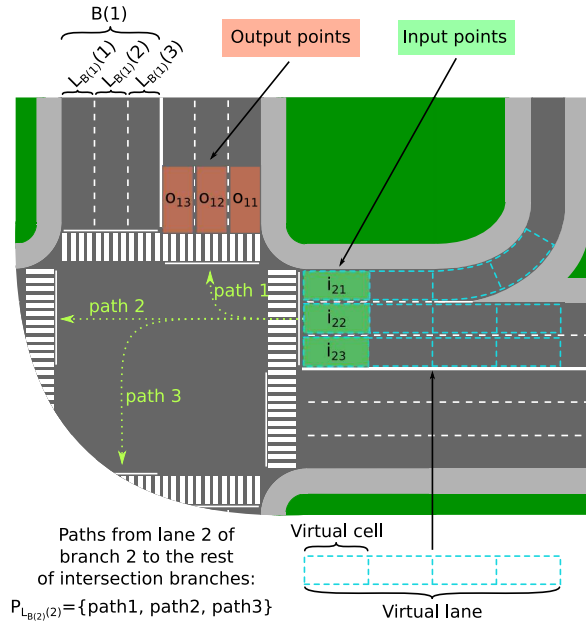


FIGURE 2. Labeling of the elements of an intersection of 4 branches and 3 lanes per branch.

cles in the patterns as a binary value, obtaining a fixed-length chromosome composed of a bit string. The codification of each GA individual is carried out by using a bit string where 1 represents the presence of a vehicle and 0 its absence. The paths of the vehicles in the intersection are fixed by their access lane. The input and output points are labeled with the tags  $i_{BL}$  and  $o_{BL}$ , respectively. The trajectories or routes followed by the vehicles inside of the intersection are denoted by the possible routes between each  $i_{BL}$  and each  $o_{BL}$ . These routes must comply with the turn restrictions ( $T_v$ ), which represent the movement options from each cell to its adjacent cells. In [20], the routes of the vehicles were determined by the input lane used by the vehicle when accessing the intersection, so the maximum flows for a given turn option were determined by the capacity of the correspondent lane.

In this work, we propose an extended optimization algorithm based on OCP, where, in contrast, the vehicles can use different paths at the intersection even if they use the same entry point (OCP-MP). In addition, we use a GA based on chromosomes with variable-length (OCP-MP-VL) to obtain the optimal pattern length in a single optimization process. In OCP-MP, an intersection is composed of a given number of branches or arms. Each branch is also divided into input and output lanes. Along this section, for easier comprehension of the proposed operators and algorithms, we use an intersection composed of 4 branches with 3 input lanes and 3 output lanes in each branch. The main elements of this intersection are shown in Figure 2.

The elements of the intersection are denoted as follows:

$$B(x) \mid x = \{1, 2, \dots, n\} \quad (1)$$

$$L_{B(x)}(y) \mid y = \{1, 2, \dots, m\} \quad (2)$$

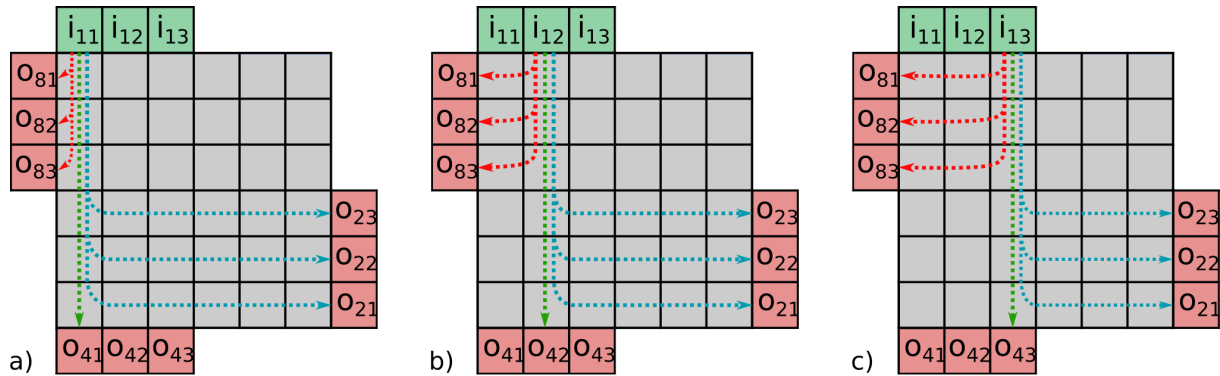


FIGURE 3. Paths with a single turn option from all input points of a specific branch. a) Paths from  $i_{11}$ . b) Paths from  $i_{12}$ . c) Paths from  $i_{13}$ .

where  $x$  is a set of integer numbers from 1 to  $n$ , being  $n$  the total number of branches in the intersection. In the same way,  $y$  is a set of integer numbers from 1 to  $m$  which identifies the  $m$  input lanes from the branch  $B(x)$ . Only the input lanes must be identified, given that the output lanes do not intervene in the problem formulation.

Each input lane is divided into virtual cells, as shown in Figure 2. These cells represent the position of a single vehicle in the platoon accessing the intersection. The cell's size is fixed and has been selected to ensure enough separation between vehicles for them to maneuver in the intersection. That is, the position of a vehicle in a specific virtual cell determines both the access point to the intersection (denoted by  $i_{BL}$ ) and the instant for this access. Every vehicle at the same distance in cells from the intersection will access it at the same time instant.

The values associated with each virtual cell of the platoons indicate the route or trajectory which each vehicle will follow in the intersection. Equation (3) defines these values.

$$P_{L_{B(x)}(y)} = \{0, 1, \dots, r\} \tag{3}$$

where  $P_{L_{B(x)}(y)}$  is the set of possible routes from the input point of the branch  $B(x)$  and the lane  $L_{B(x)}(y)$  to every possible output point in the intersection. These possible routes are labeled using an integer number from 0 to  $r$ , where 0 represents the absence of a vehicle.

$P$ 's size may vary depending on the total number of branches in the intersection, the number of lanes in each branch, and the possible turn restrictions in each case. These variables add complexity to the problem. The methodology used in OCP to obtain the  $P$  sets is based on determining the routes with less possible conflicts between them. This methodology allows the generation of a limited set of routes, selecting just one option for each input point. As a drawback, it introduces a limitation on the number of vehicles that can continue to a destination. In contrast, in OCP-MP we propose that each vehicle can go from any input point to any output branch, adding just one restriction: that in their route they change their way just once. This restriction is added

because any change in the trajectory means that the vehicle must adjust and reduce its velocity. By limiting the trajectory changes to the minimum, it is possible to maintain a constant velocity in the intersection.

Following the intersection example shown in Figure 2, we have an intersection with the same number of lanes in each branch. This means that it is possible to determine the set of paths for only one branch and then translate the values to the other three branches. In Figure 3 we show the paths calculated for the example. Starting from a given branch, the possible paths to the rest of branches are shown in Figure 3.a, Figure 3.b, and Figure 3.c, each one with origin in a specific lane of the input branch. Considering the restriction of performing at most one change in the path trajectory, values 1, 2 and 3 represent the turn to the right of a vehicle. Value 4 represents going straight ahead, and values 5, 6 and 7 points out that the vehicle turns to its left. Therefore,  $L_{B(x)}(y) = \{0, 1, 2, 3, 4, 5, 6, 7\}$ . This set of identifiers will be the same for any origin and destination, because of the symmetry characteristics of the example intersection, although the routes will be different depending on the origin point  $i_{BL}$ .

Using these definitions, it is possible to determine that the chromosome of an individual in the genetic algorithm is encoded with as many genes as virtual cells are defined in each input lane of the intersection. That is, given a pattern length of  $l_p$ , the size of an individual is calculated as shown in (4).

$$I_l = \sum_{i=1}^n m_i \cdot l_p \tag{4}$$

where  $m_i$  is the number of input lanes in each  $i$  branch of the intersection.

In Figure 4 we show a representation of the proposed codification where each position of a gene in the chromosome represents a virtual cell of a lane and branch. The value or genotype of each gene is contained in the set  $P_{L_{B(x)}(y)}$  of the branch/lane represented depending on the position. The phenotype of each gene is the route followed by a given vehicle in the intersection.

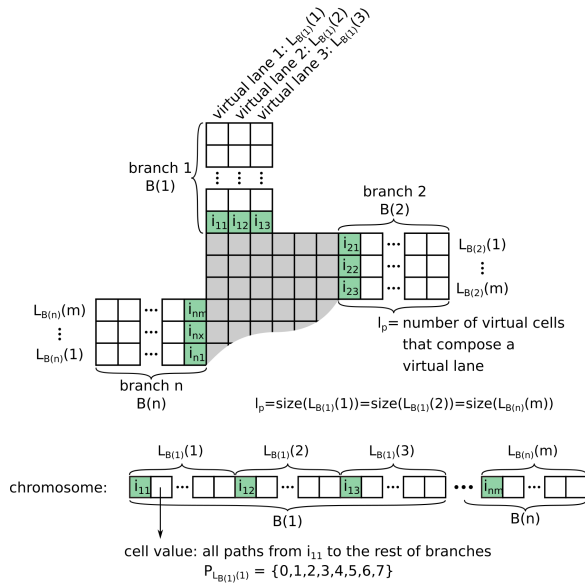


FIGURE 4. Traffic coordination problem encoding for a genetic algorithm.

To guarantee the correct synchronization of every vehicle accessing the intersection, the size of the pattern is equal for every input lane. The set of values  $P_{L_{B(x)}(y)}$  might vary depending on the number of possible path combinations in each case.

### C. FITNESS FUNCTION

The proposed optimization process has the goal of generating the set of vehicle input patterns for an intersection that guarantees the requirements of the system. This requirement is that, under a given vehicle arrival rate from a specific branch of the intersection, it is possible to route all vehicles to their destination without any collision.

The proposed system, additionally to the intersection descriptive data, receives as input parameter a set of “desirable” flows for each one of the intersection inputs  $q_{from}(to)$ , where “from” denotes the input branch and “to” is the destination branch for that flow. Equation (5) defines this data structure.

$$Q = [q_1(n), \dots, q_1(3), q_1(2), \dots, q_n(1)] \quad (5)$$

where  $Q(i)$  is the  $i$ -th element of the  $Q$  array (for instance,  $Q(1) = q_1(n)$ ). Each one of these proportions is defined over the maximum capacity of vehicles that can access to the intersection from a specific branch, and therefore, they must comply with the restriction that their sum is lower than or equal to 1. Assuming that a vehicle cannot enter and exit an intersection through the same branch, the number of elements of the array  $Q$  will be  $n \cdot (n - 1)$ .

The fitness function implemented in our proposal uses the microscopic traffic simulator based on the cellular automata described in section II. This simulator allows us, from the

patterns generated for each individual in the population, to reproduce the behavior of the vehicles when crossing the intersection and to count the number of collisions that would have happened. Besides, the function evaluates the similarity between the flows obtained for each population and the desired flows. When the simulation ends, we obtain the following parameters:

- **Average Collisions ( $C_{avg}$ ):** In each simulation step  $i$ , the number of vehicles occupying the same cell (that is, the collisions  $c_i$ ) are stored. When the simulation steps  $l_{sim}$  is over, the average collision score is calculated as the division between the total number of collisions and the number of simulation steps executed.

$$C_{avg} = \frac{\sum_{i=1}^{l_{sim}} c_i}{l_{sim}} \quad (6)$$

- **Sum of Absolute Errors (SAE):** When the simulation is finished, the input flows  $Q$  are compared with those obtained from the optimizer  $Q_{opt}$ . The sum of the absolute value of each one of the sub-flows is defined as the value  $SAE$ .

$$SAE = \sum_{i=1}^{size(Q)} |Q(i) - Q_{opt}(i)| \quad (7)$$

- **Normalized Mean Absolute Error by flows ( $NMAE_{flows}$ ):** This parameter measures the difference between  $Q$  and  $Q_{opt}$ , evaluating each sub-flow separately. Each sub-flow is evaluated regarding the difference between the desired flow and the flow obtained from the optimization process.

$$NMAE_{flows} = \frac{\sum_{i=1}^{size(Q)} \frac{|Q(i) - Q_{opt}(i)|}{Q(i)}}{size(Q)} \quad (8)$$

Taking these metrics into account, the goal of the optimization process is to minimize the result of the fitness function, which means getting the minimum difference between  $Q$  and  $Q_{opt}$ . We can give different weights to  $C_{avg}$ ,  $SAE$  and  $NMAE_{flows}$ , in order to achieve solutions where the focus is on obtaining the minimum global difference or obtaining solutions where the minimum difference for each turn option considered independently is guaranteed. Therefore, the fitness function can be expressed as:

$$f = \alpha \cdot C_{avg} + \beta \cdot SAE + \gamma \cdot NMAE_{flows} \quad (9)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are the weights associated with each one of the aforementioned metrics. Each of these weights can be used for different purposes. The  $\alpha$  value is used for obtaining valid solutions without collisions. The parameters  $\beta$  and  $\gamma$  allow changing the optimization goals. If the results sought are based on minimizing the total difference between all traffic flow rates, the weight of  $\beta$  will be predominant. Conversely, if the global goal is that each traffic flow rate that composes the  $Q$  array has the same importance,  $\gamma$  will be greater than  $\beta$ .

#### D. GENERATION OF THE INITIAL POPULATION

The convergence, performance, and effectiveness of a genetic algorithm are factors which are highly affected by the definition of the initial population as indicated in [26], [27]. It is quite common to perform a random selection of the genes composing the initial population, but in multiple restriction environments it is necessary to guarantee that the restrictions are met in this population and that there is enough diversity between individuals for the correct convergence of the GA.

To define a function for the generation of initial populations, in our scenario we must consider the following restrictions:

- The length of each individual depends on the number of branches  $B(x)$  and lanes  $L_{B(x)}$  composing the intersection (defined in (1) and (2), respectively), and also on the number of cells composing the virtual lane  $l_p$ .
- The possible values for each gene must be elements of the set  $P_{L_{B(x)}(y)}$ .

Taking these restrictions into account, the generation process for the initial population follows these steps:

- 1) The function receives as parameters the number of branches and lanes of the intersection, the minimum and maximum length of  $l_p$ , the number of individuals of the maximum pattern length to generate, and the set of flows (turn options) to be searched in the optimization process.
- 2) The population size depends on the interval between the minimum and maximum value of the pattern  $l_p$ , and on the number of individuals of maximum size received as the input parameter. The number of individuals of each length is obtained in the following way:
  - The minimum pattern length is  $\min(l_p)$ , the maximum pattern length is  $\max(l_p)$  and maximum of individuals with  $\max(l_p)$  length is  $\max(subP)$ .
  - The array of pattern lengths is:  
 $L = [\min(l_p), \min(l_p) + 1, \dots, \max(l_p)]$   
 where the size of  $L$  is  $\max(l_p) - \min(l_p) + 1$ .
  - The size of the set of individuals for the  $\min(l_p)$  pattern length is  $\max(subP)/3$ . This value has been obtained empirically to balance the subpopulation sizes and the total size of the final population.
  - The array of subpopulation sizes is:  
 $subP = [\max(subP)/3, (\max(subP)/3) + inc, \dots]$   
 where  $inc = \frac{\max(subP) - \frac{\max(subP)}{3}}{\max(l_p) - \min(l_p)}$
- 3) Considering the expected flows, an array containing the occurrence probabilities for each value in the  $P_{L_{B(x)}(y)}$  set is generated.
- 4) Each gene of each individual is set by assigning the values of  $P_{L_{B(x)}(y)}$ , taking into account the probabilities calculated in the previous step. Each individual will be of the length calculated in step 2.

#### E. SELECTION OPERATOR

The selection operator used is the function named *selectionremainder* included in the *global optimization* toolbox

of Matlab R2018b. This operator is based on getting an expectations array using the fitness values obtained for each individual in the current population. The parents are selected in a deterministic way, based on the integer part of each value from the expectations array. Once this process is finished, if there are still parents to select, the decimal part of the expectation values is used to stochastically select them [28].

#### F. CROSSOVER OPERATOR

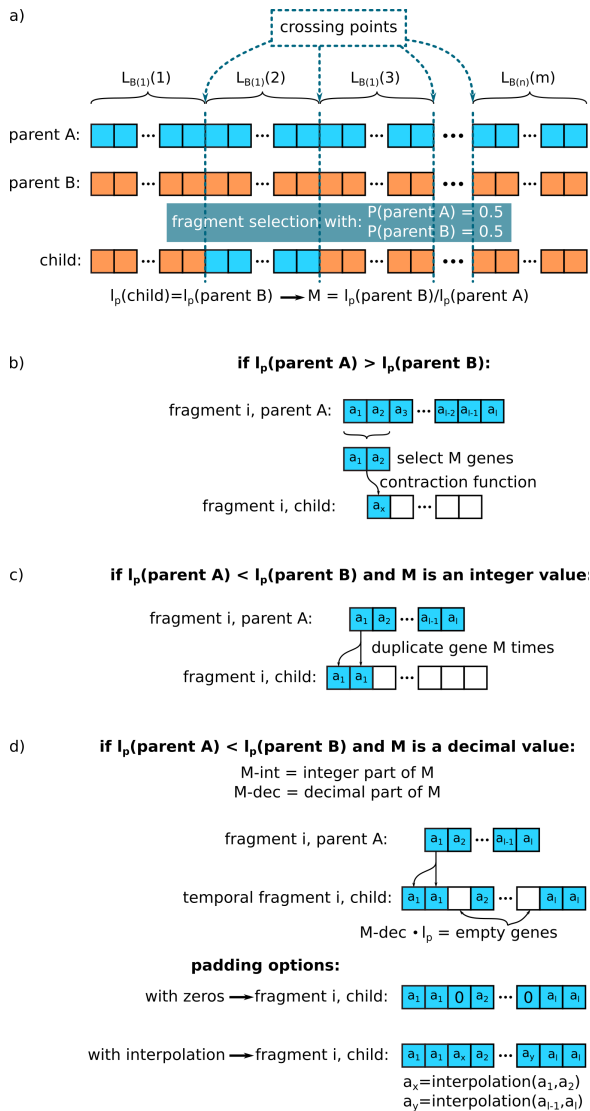
There are multiple techniques for the crossover function, such as the selection of a single crossover point, the selection of multiple crossover points, or the gene-by-gene combination [29]. Given that our problem requires performing an intensive search in the solution space, we have selected the gene-by-gene combination option, adapting it to the specific proposed codification of the problem. To guarantee the fulfillment of the restrictions, the crossover will be performed pattern-by-pattern. That is, it will be the combination of sets of adjacent genes representing the vehicle pattern in a given branch and lane. The crossover process for variable-length chromosome (OCP-MP-VL) is as follows (see Figure 5):

- 1) The two selected individuals (*parentA* and *parentB*) are divided into the same number of subsets. This number is obtained by multiplying the number of branches in the intersection with the number of lanes composing each branch.
- 2) A new child individual is generated copying the genes and length of *parent B*. The relationship between the two parents lengths is noted as  $M$  and is shown in (10).

$$M = \frac{l_p(\text{parentB})}{l_p(\text{parentA})} \quad (10)$$

- 3) The fragments generated in step 1 are simultaneously evaluated for *parent A* and *parent B*. With a probability of 0.5, the process decides if each fragment of the child stays as the value corresponding to *parent B* or is substituted with the value of the equivalent fragment from *parent A*. In this last case:
  - a) If  $l_p(\text{parentA}) = l_p(\text{parentB})$ , the genes of the fragment are directly substituted (Figure 5.a).
  - b) If  $l_p(\text{parentA}) > l_p(\text{parentB})$ , a contraction process must be carried out for the fragment (Figure 5.b).
  - c) If  $l_p(\text{parentA}) < l_p(\text{parentB})$ , an expansion process must be carried out for the fragment (Figure 5.c and Figure 5.d).
- 4) The process ends when all the fragments have been evaluated.

The contraction process selects  $M$  genes from parent A to obtain a gene from parent B (Figure 5.b). If the  $M$  genes of parent A have the same value, this value is assigned to the gene of parent B. When the  $M$  genes have different values, but the paths that represent these values have the same destination, the gene of parent B is chosen randomly between the  $M$  genes of parent A. Finally, if the  $M$  genes from parent A



**FIGURE 5. Crossover operator description. a) Crossing points, b) contraction method, c) expansion method if M is an integer value, d) expansion method if M is a decimal value including padding options.**

are different among themselves and their phenotype represent paths with different destinations, one of the M genes values is selected and used in parent B. This selection will take into account the proportion of appearance for each value in the parent A chromosome.

On the other hand, for the expansion process, if the size of one fragment is a multiple of the other (M is an integer value), it is possible to carry out a direct correspondence (Figure 5.c). When this is not the case, for the expansion process we need to add padding genes to achieve the needed size. Two different strategies have been defined to add the padding genes: adding zeros or adding values depending on the values of the adjacent genes. In the first case, the advantage is that the added zeros will not generate new collisions in the simulation scenario,

but they present the problem of adding a slight deviation from the original phenotype. In the second case, the isolated behavior of the fragment is maintained, but the new genes might introduce new collisions during the simulation with a higher probability. This process for padding empty genes is shown in Figure 5.d.

### G. MUTATION OPERATOR

The mutation operator receives as parameters the genes composing an individual, the mutation rate which indicates the proportion of genes which could be mutated, and the possible values of variation for each gene. This last parameter guarantees the validity of the mutated individual.

Each gene mutates with probability equal to the mutation rate, selecting randomly a value from the possible values regarding its position (possible values of  $P_{L_{B(x)}(y)}$  depending on the branch and lane that represents the gene). The mutation process is executed for a number of individuals once the elite population has been stored and the rest of individuals have been part of the crossover process.

For a number of the first generations, the mutation operator is not applied in order to favor a faster convergence to a sub-optimal solution, given the diversity of the initial population defined in section III-D.

## IV. EXPERIMENTS AND RESULTS

To test and validate the behavior of the proposal, we have conducted a series of experiments. The sets of input data for each experiment have been selected in order to obtain diversity to validate our proposal in different traffic circumstances.

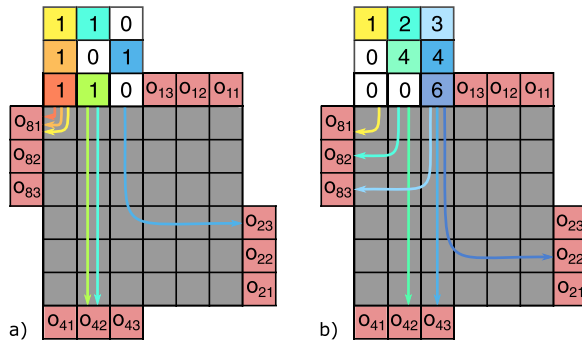
### A. EVALUATION METRICS

In each experiment, we have selected the following metrics to evaluate the results:

- **Fitness score:** Value of the fitness function used for the optimization process. This function was described in section III-C and its values represent the weighted sum shown in (9). The best possible value is 0.
- **Computation time:** Measured in seconds, is the time that the optimizer takes to return a solution.
- **Sum of Absolute Errors:** Value representing the difference between the flows or turn options for each input of the intersection obtained in the solution and the values requested at the input of the optimizer. The best possible value is 0.
- **Collisions:** Sum of the collisions between vehicles (that is, coincidences of two or more vehicles in the same cell during a simulation step) produced in each simulation step, divided by the total simulation steps. Any value different from 0 is considered an invalid solution.

### B. EXPERIMENTAL SETUP

The solution proposed in section III allows us to obtain the vehicle arrival patterns for each input point in an intersection



**FIGURE 6.** Solutions to the same problem using different optimization processes. a) OCP optimizer, b) proposal optimizer (OCP-MP).

that is minimizing the distance to the specific requirements of the system. Figure 6 shows the resolution of an example case using OCP and OCP-MP. In this case, the turning preferences of the vehicles are 3/9 turn right, 2/9 go straight and 1/9 turn left. OCP defines a set of fixed paths and all vehicles with the same input point follow the same path (Figure 6.a). On the other hand, OCP-MP allows using different paths from different input points to the intersection (Figure 6.b).

To evaluate the OCP-MP and OCP-MP-VL optimization processes, we have defined two sets of experiments. The first one is composed of values of input flows for which a candidate solution is already known. In the second set, 100 random flows following a specific distribution have been generated. Besides OCP-MP and OCP-MP-VL, two other known optimizers have been tested with the same input data: Hill Climber (HC) optimization algorithm [27], [30] and OCP as defined in [20].

In section III-F we defined two possible crossover operators depending on how the expansion and contraction processes were implemented. In the first set of experiments, we have tested the results obtained for each possible crossover operator. In the second set, however, we have just used the crossover operator producing the best results. The parameters used for the OCP-MP-VL optimizer are the following:

- **Fitness function:** Based on (9), we have decided to favor the global goal of obtaining the smallest possible difference between  $Q$  and  $Q_{opt}$ , establishing  $\alpha = 0.5$ ,  $\beta = 0.5$  y  $\gamma = 0$ .
- **Initial population:** We have established that  $\min(l_p) = 3$ ,  $\max(l_p) = 13$ , which generates a population of 462 individuals.
- **Mutation operator:** The operator starts acting after the 10 first generations and the mutation rate parameter is set to 0.05.

The selection of these parameters is based on the sensitivity analysis shown in appendix A.

### C. EXPERIMENTS DEFINED

For the first set of experiments (input data with known solution) we have defined the following three possible input data:

- **Experiment A1:** Every vehicle wants to turn to the right when arriving the intersection. The input value for the optimizer is:

$$Q_{A1} = [1, 0, 0; 1, 0, 0; 1, 0, 0; 1, 0, 0]$$

- **Experiment A2:** A third part of the vehicles want to turn to the right from each input point and 2/3 want to continue straight in two ways situated one in front of the other, so:

$$Q_{A2} = \left[ \frac{1}{3}, \frac{2}{3}, 0; \frac{1}{3}, \frac{2}{3}, 0; \frac{1}{3}, \frac{2}{3}, 0; \frac{1}{3}, \frac{2}{3}, 0 \right]$$

- **Experiment A3:** The third combination of flow values corresponds with the solution obtained for this type of intersection with 4 branches and 3 lanes/branch shown in [31]. This solution is described by the following input flows:

$$Q_{A3} = \left[ \frac{5}{15}, \frac{2}{15}, \frac{1}{15}; \frac{5}{15}, \frac{2}{15}, \frac{1}{15}; \frac{5}{15}, \frac{2}{15}, \frac{1}{15}; \frac{5}{15}, \frac{2}{15}, \frac{1}{15} \right]$$

The graphical representation of the solutions for these values of  $Q_{Ax}$  is shown in Figure 7.

The second set of experiments (random input preferences) have been carried out for the use case intersection (UCI) with 4 branches and 3 lanes per branch. Because of this,  $Q$  has 12 elements. Each branch has been clockwise labeled with the values of 1 to 4 so when applying (5), the following values for  $Q$  are obtained:

$$Q_{UCI} = [q_1(4), q_1(3), q_1(2); q_2(1), q_2(4), q_2(3); q_3(2), q_3(1), q_3(4); q_4(3), q_4(2), q_4(1)]$$

In this scenario, there are three turning options: right, go straight and left, where:

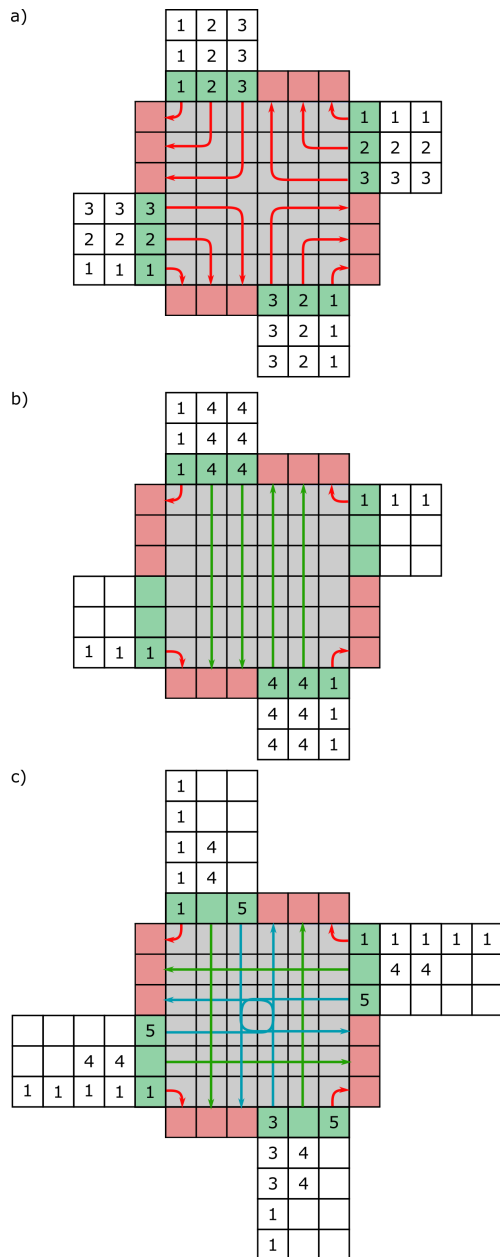
$$\begin{aligned} q_{right} &= [q_1(4), q_2(1), q_3(2), q_4(3)] \\ q_{go} &= [q_1(3), q_2(4), q_3(1), q_4(2)] \\ q_{left} &= [q_1(2), q_2(3), q_3(2), q_4(1)] \end{aligned}$$

To count with more extensive testing for our proposal, we have defined three sets of testing data composed each of 100 input flows following these distributions:

- **Experiment B1:** Each set of input flows for a given branch in the intersection is generated following these steps:

- We generate a random value  $q_{rand1}$  such that:  $0 < q_{rand1} \leq 0.6$ .
- We generate a random value  $q_{rand2}$  such that:  $0 < q_{rand2} \leq 1 - q_{rand1}$ .
- We generate a random value  $q_{rand3}$  such that:  $0 < q_{rand3} \leq 1 - q_{rand1} - q_{rand2}$ .





**FIGURE 7.** Pattern complying with the input requirements of the experiments Ax. The cell values represent the following options: No vehicle (0), vehicle turns right (1, 2, or 3), vehicle goes straight (4), and vehicle turns left (5, 6, or 7). a) Possible solution for  $Q_{A1}$ , b) Possible solution for  $Q_{A2}$ , and c) Possible solution for  $Q_{A3}$ .

- $q_{rand1}$ ,  $q_{rand2}$  and  $q_{rand3}$  are assigned randomly to  $q_{right}(i)$ ,  $q_{go}(i)$  and  $q_{left}(i)$ .
- The process finishes when all values of  $q_{UCI}$  have been filled.
- **Experiment B2:** Following the process described for the previous experiment, we generate random flow rates following the proportions of 0.7, 0.2 and 0.1 for each

**TABLE 1.** Results obtained for the experiments A1, A2 and A3 using the four defined optimization processes.

Value	Optimizer	Crossover type	Experiment A1	Experiment A2	Experiment A3
Fitness score	HC	-	1.3387	0.6356	0.1223
	OCP	-	0.1197	0.0438	0
	OCP-MP	-	0	0	0.0417
	OCP-MP-VL	Zeros	0.0694	0.0556	0
	OCP-MP-VL	No zeros	0	0	0.0417
	Computation time (s)	HC	-	466.5469	285.9582
OCP		-	2763.1273	3148.6648	2812.8252
OCP-MP		-	1841.2997	1459.9598	1737.867
OCP-MP-VL		Zeros	615.7362	310.6485	1585.902
OCP-MP-VL		No zeros	306.1409	118.2788	283.3829
Sum of Absolute Errors		HC	-	0	0.0972
	OCP	-	0.2393	0.0876	0
	OCP-MP	-	0	0	0.0833
	OCP-MP-VL	Zeros	0.1389	0.1111	0
	OCP-MP-VL	No zeros	0	0	0.0833
	Collisions	HC	-	83	27
OCP		-	0	0	0
OCP-MP		-	0	0	0
OCP-MP-VL		Zeros	0	0	0
OCP-MP-VL		No zeros	0	0	0

possible turn direction. The calculated values are also shuffled to assign randomly the specific turn option. The total sum of all the turn options from a given branch must be equal or below 1.

- **Experiment B3:** Finally, in this experiment, we add an extreme case in which the vehicles only want to use a specific turn option. In this case, we assign a random value between 0 and 1 to a turn option between the three available in each branch.

#### D. EXPERIMENTS RESULTS

The HC, OCP, OCP-MP, and OCP-MP-VL optimizers have been run in each experiment. Of these, the first two, HC and OCP, allow obtaining reference results using known optimizers. The improvements proposed in this work are included in both the OCP-ML and OCP-ML-VL optimizers, but while the first uses a fixed pattern length, the second uses the proposed variable-length chromosome GA to solve the problem.

In Table 1 we show the results obtained when executing each optimizer on each set of flows in experiments A1, A2, and A3. The results of experiments B1, B2, and B3 are shown in Table 2. In this second set of results, given that there is more than one testing value for each experiment, the results show the average value for each case and the corresponding standard deviation (average  $\pm$  standard deviation).

#### V. DISCUSSION

The experiments A1, A2, and A3 defined in section IV-D are based on vehicles preferences with a known solution, which means that *a priori*, we know that it would be possible to obtain a 0 result for the fitness function value and also in the total difference measured at the end of the optimization process. This result is an optimal solution for the problem, and therefore, the optimizers would have achieved

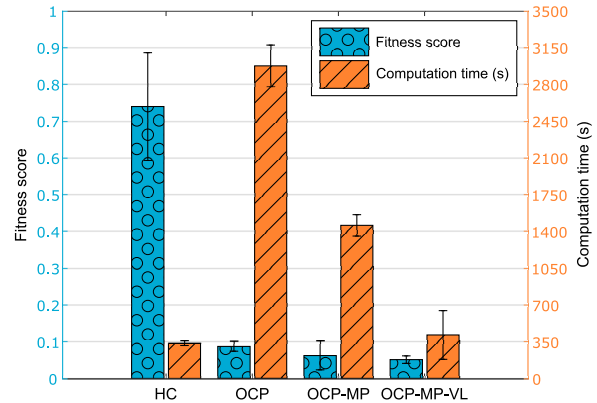
**TABLE 2. Results obtained for the experiments B1, B2 and B3, using the four defined optimization processes.**

Value	Optimizer	Experiment B1	Experiment B2	Experiment B3
Fitness score	HC	1.169 ± 0.209	0.6278 ± 0.1572	0.5433 ± 0.2456
	OCP	0.0642 ± 0.0057	0.1186 ± 0.0138	0.1746 ± 0.0282
	OCP-MP	0.0649 ± 0.0031	0.1126 ± 0.0390	0.1536 ± 0.0867
	OCP-MP-VL	0.0683 ± 0.008	0.1072 ± 0.013	0.1312 ± 0.0199
Computation time (s)	HC	375.2 ± 7.402	320.6 ± 23.83	326.8 ± 44.83
	OCP	3022 ± 135.5	2992 ± 358	3104 ± 295.1
	OCP-MP	1426.57 ± 134.8	1208.89 ± 114.39	1064.5 ± 175.49
	OCP-MP-VL	496.3 ± 207.01	886.6 ± 433.7	389.1 ± 301.8
Sum of Absolute Errors	HC	0.1628 ± 0.0376	0.2133 ± 0.0257	0.273 ± 0.039
	OCP	0.1283 ± 0.0113	0.2371 ± 0.0275	0.3491 ± 0.0565
	OCP-MP	0.1298 ± 0.0061	0.2253 ± 0.078	0.3072 ± 0.1734
	OCP-MP-VL	0.1366 ± 0.016	0.2144 ± 0.0259	0.2625 ± 0.0397
Collisions	HC	70.14 ± 26.51	38.95 ± 12.86	29.58 ± 22.9
	OCP	0	0	0
	OCP-MP	0	0	0
	OCP-MP-VL	0	0	0

input patterns which comply with the restrictions defined in the desired input flows, while maintaining the simulations collision-free. As shown in Table 1, the only optimizers able to reach this goal are those proposed in this article, the OCP-MP and OCP-MP-VL optimizers. These optimizers achieve a result of  $f = 0$  for experiments A1 and A2 when using the crossover function based on adding non-zero values in the expansion operations, and it also achieves a result of  $f = 0$  for experiment A3 using the crossover function that adds zeros when expanding the individual during the crossover process. These results are explained because in A1 and A2 one or more branches use the maximum vehicle capacity that they can process (adding zeros to the individuals in the crossover process to pad the individual size generates sub-optimal solutions, given that the optimal solution would require that the individual had every gene with non-zero values). On the other hand, in experiment A3, when it is required to route traffic to any branch and there is no requirement to use the 100% of the capacity of any branch, the crossover with the zero-padding based expansion offers better results, as a zero value guarantees that there are no collisions in that position. OCP-MP and OCP-MP-VL optimizers obtain the same fitness score, but the second one finished in lower computing time. OCP-MP-VL is up to 12 times faster in some cases.

The results obtained with OCP can be seen as acceptable regarding the fitness values, but they are limited by the use of fixed paths. This restriction determines the maximum traffic capacity that can be routed towards each output point of the intersection. Besides, the computation times are much higher than in the rest of the optimizers, because it must solve the problem for each pattern length to be tested as the optimal lengths are not known *a priori*.

Finally, the values obtained for HC are the worst among the compared optimizers. This algorithm can converge in very short computation time, but the solutions returned are far from the optimal solution. Moreover, in every experiment,



**FIGURE 8. Average fitness scores (blue bars with circles) and computation times (orange bars with lines) of all the experiments. The value of each bar shows the average value of each optimizer. With a black vertical line on each bar, the standard deviation of the samples is represented.**

the results present collisions, and therefore, cannot be considered valid solutions.

In a similar way to the previous experiments, the results obtained by the experiments B1, B2, and B3 (random input preferences) using HC present the lowest computation times, but with the fitness scores farther from the optimal solutions. It is possible to notice that in this case, none of the optimizers can reach the value  $f = 0$ . This is the expected result, as the input flows have been selected randomly, and therefore the optimizers can only offer a solution as close as possible as the requested, without adding collisions to the simulation. The computation times of OCP-MP and OCP-MP-VL are 2.5 and 8 times lower than the times measured for OCP, respectively. This result, and fitness values very close (B1) or lower (B2 and B3) than the results of the OCP optimizer, show the advantages of using our proposed solution, which can calculate both the optimal vehicle combination and the best pattern size in a single process.

The results of all the experiments have been grouped to obtain the average and standard deviations values of the fitness scores and the computation times of each optimizer. These values are shown in Figure 8.

This summary allows the overall evaluation of the results obtained. OCP-MP and OCP-MP-VL achieve the best fitness scores, getting optimal or quasi-optimal solutions. In addition, the values in the figure show the improvement, in terms of reduction in computation time, of using the optimized with variable-length chromosome versus fixed-length chromosome. The small difference between the average fitness scores obtained with OCP-MP versus OCP-MP-VL is due to OCP-MP optimizer finishing upon reaching the maximum number of generations in some cases, before getting the best solution.

## VI. CONCLUSIONS

In this work, we have presented a new methodology to obtain patterns which induce vehicle arrival rates enabling

safe crossings at the intersection while respecting destination preferences.

The proposed methods generalize our previous work by allowing the vehicles to have freedom of path choice in an intersection. This advantage, however, increases the problem complexity, as now the number of possible combinations depends directly on the turn options in each intersection. To mitigate this drawback, we have improved the GA performance by designing, implementing and testing custom crossover and mutation operators adapted to this scenario. Moreover, we have designed a novel methodology for the generation of initial populations based on the searched solutions that preserve the necessary randomness in these algorithms. Given that the pattern length of the solution is not known *a priori*, we have introduced the use of variable-length chromosomes to obtain, at the same time, the best possible fitness score and the shortest pattern length. To achieve this, we have proposed techniques for the expansion and contraction of the chromosome's genes.

After evaluating the optimizers both with known solutions and random data, the results demonstrate the viability of the OCP-MP and OCP-MP-VL optimizers. The second version of our proposal is the optimizer process which achieves the best solutions with the lowest computation times. In addition, the comparison between the obtained results using the proposed system and other solutions such as OCP or widely used algorithms like Hill Climber has concluded that our system can show improvements of up to a 9.1% and computation times 126% lower in average, which proves the advantages of the work.

Although the results of this paper show an improvement from previous solutions in this field, there are certain limitations in the optimization system that should be highlighted. The optimization process is applied to a single intersection, the set of trajectories or paths in the intersection cannot vary from those set at the beginning of process, and the proposed techniques are adapted specifically to the vehicle coordination problem in intersections. Following these statements, we can clearly establish three possible work lines. The first one would be extending this methodology to patterns generated between two or more intersections considering as goal obtaining the minimum vehicle position changes (between lanes) to adapt them to the optimal patterns. The second one would be using the new crossover operator to solve other problems in which it would be useful to codify variable-length population. Finally, we would like to explore the possibility of adapting the proposed system to invalidate the individuals that generate collisions. In this case, if the optimization process cannot be completed due to time restrictions, we can still get a pseudo-optimal solution.

#### APPENDIX A PARAMETER SENSITIVITY ANALYSIS

In order to calibrate the optimization process based on the GA, we have carried out a sensitivity analysis for the main parameters of the algorithm. In Figures 9, 10, and 11 we show the results corresponding to the experiments performed

VOLUME 7, 2019

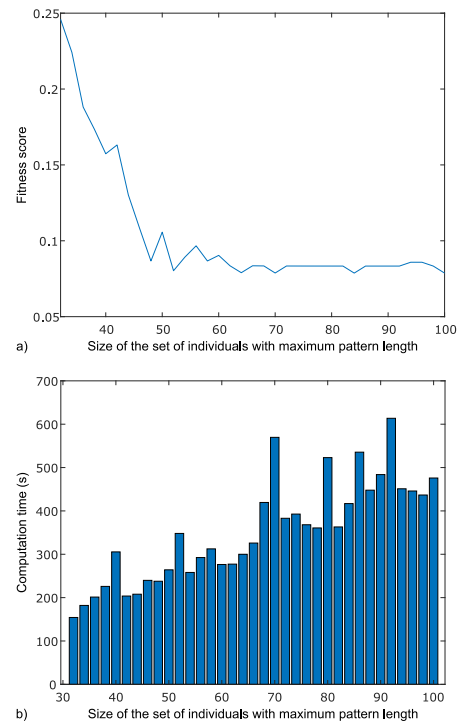


FIGURE 9. Average values for fitness scores (a) and computation times (b) when evaluating the population size.

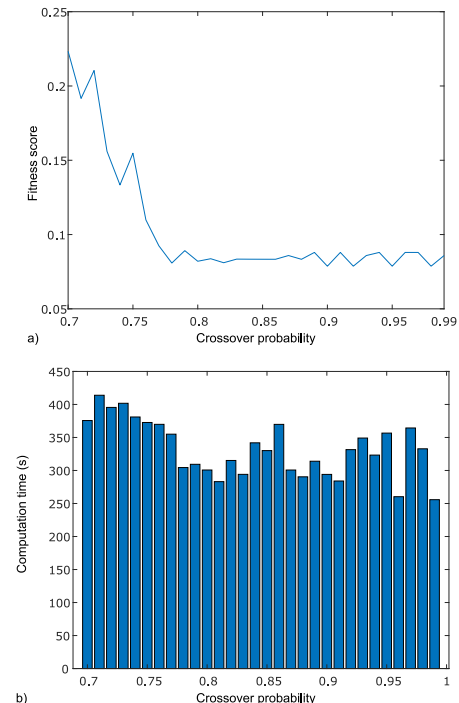
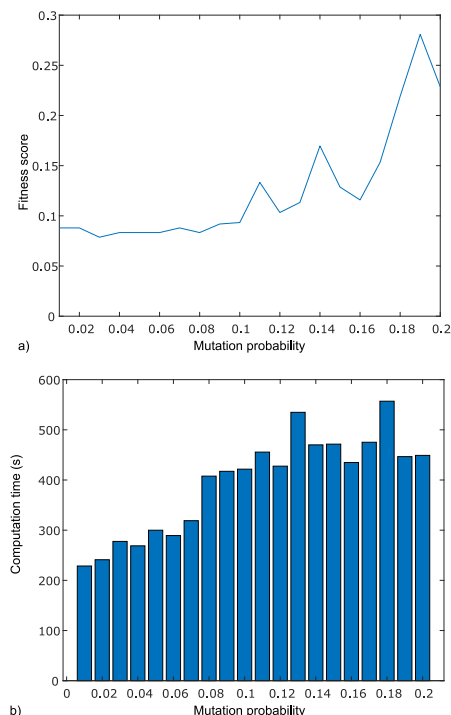


FIGURE 10. Average values for fitness scores (a) and computation times (b) when evaluating the crossover probability.

over the population size, the crossover probability, and the mutation probability.

111979



**FIGURE 11.** Average values for fitness scores (a) and computation times (b) when evaluating the mutation probability.

For each one of the parameters we have randomly selected 10  $Q$  values extracted from the data defined in experiments B1, B2 and B3 described in section IV-C. Using this set of ten values we have executed the optimization process selecting fixed values for two parameters and varying the third one in the following way:

- **Population size:** The initial population of the GA is defined by the size of the set of individuals with the maximum pattern length, as explained in section III-D. The values used for the crossover probability and mutation probability are 0.8 and 0.05 respectively. The size of the set of individuals with maximum pattern has been changed from 32 to 100.
- **Crossover:** The crossover probability has been changed in each optimization process between 0.7 and 0.99. We have selected 64 as the number of initial individuals with maximum pattern length (which generates a population of 462 individuals), and a mutation probability of 0.05.
- **Mutation:** The mutation probability has been changed in each optimization process between 0.01 and 0.2. We have selected 64 as the number of initial individuals with maximum pattern length (which generates a population of 462 individuals), and a crossover probability of 0.8.

Once all experiments have been carried out, we have calculated the average values for fitness scores and computation times. Regarding the population size, in

Figures 9.a and 9.b it is shown how above 60, the fitness score remains stable. On the other hand, there is a local minimum for a value of 64 in computation times. Due to these results, we have selected 64 as the input parameter which generates an initial population of 462 individuals.

Evaluating in a similar way the results shown in Figures 10 and 11, we can establish that the optimal values for crossover and mutation probabilities are 0.8 and 0.05 respectively.

## REFERENCES

- [1] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 74–84, Nov. 2016.
- [2] E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS-811 366, 2010. [Online]. Available: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811366>
- [3] P. Bansal and K. M. Kockelman, "Forecasting Americans' long-term adoption of connected and autonomous vehicle technologies," *Transp. Res. A, Policy Pract.*, vol. 95, pp. 49–63, Jan. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0965856415300628>
- [4] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.
- [5] G. Marti, A. H. Morice, and G. Montagne, "Drivers' decision-making when attempting to cross an intersection results from choice between affordances," *Frontiers Hum. Neurosci.*, vol. 8, p. 1026, Jan. 2015.
- [6] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Dec. 2015, pp. 343–347.
- [7] W. Sun, J. Zheng, and H. X. Liu, "A capacity maximization scheme for intersection management with automated vehicles," *Transp. Res. Procedia*, vol. 23, pp. 121–136, Jul. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2352146517302855>
- [8] F. Yan, M. Dridi, and A. E. Moudni, "Autonomous vehicle sequencing algorithm at isolated intersections," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.
- [9] F. Yan, M. Dridi, and A. El Moudni, "A scheduling approach for autonomous vehicle sequencing problem at multi-intersections," *Int. J. Oper. Res.*, vol. 8, no. 1, pp. 57–68, 2011.
- [10] F. Yan, M. Dridi, and A. El Moudni, "An autonomous vehicle sequencing problem at intersections: A genetic algorithm approach," *Int. J. Appl. Math. Comput. Sci.*, vol. 23, no. 1, pp. 183–200, 2013.
- [11] G. R. de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 8–21, Jan. 2017.
- [12] L. Cruz-Piris, D. Rivera, S. Fernandez, and I. Marsa-Maestre, "Optimized sensor network and multi-agent decision support for smart traffic light management," *Sensors*, vol. 18, no. 2, p. 435, 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/2/435>
- [13] M. Bashiri and C. H. Fleming, "A platoon-based intersection management system for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 667–672.
- [14] J. Lianghai, A. Weinand, B. Han, and H. D. Schotten, "Applying multiradio access technologies for reliability enhancement in vehicle-to-everything communication," *IEEE Access*, vol. 6, pp. 23079–23094, 2018.
- [15] A. N. H. Zaied and W. Al Othman, "Development of a fuzzy logic traffic system for isolated signalized intersections in the State of Kuwait," *Expert Syst. Appl.*, vol. 38, pp. 9434–9441, Aug. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417411001503>

- [16] M. Collotta, L. L. Bello, and G. Pau, "A novel approach for dynamic traffic lights management based on wireless sensor networks and multiple fuzzy logic controllers," *Expert Syst. Appl.*, vol. 42, pp. 5403–5415, Aug. 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417415001104>
- [17] J. J. Sanchez-Medina, M. J. Galan-Moreno, and E. Rubio-Royo, "Traffic signal optimization in 'La Almozara' district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing," *J. Adv. Transp. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 132–141, Mar. 2010.
- [18] Q. Lu and K.-D. Kim, "A genetic algorithm approach for expedited crossing of emergency vehicles in connected and autonomous intersection traffic," *J. Adv. Transp.*, vol. 2017, Oct. 2017, Art. no. 7318917. doi: [10.1155/2017/7318917](https://doi.org/10.1155/2017/7318917).
- [19] J. Li, M. Dridi, and A. El-Moudni, "A cooperative traffic control for the vehicles in the intersection based on the Genetic algorithm," in *Proc. 4th IEEE Int. Colloq. Inf. Sci. Technol. (CIST)*, Oct. 2016, pp. 627–632.
- [20] L. Cruz-Piris, M. A. López-Carmona, and I. Marsá-Maestre, "Automated optimization of intersections using a genetic algorithm," *IEEE Access*, vol. 7, pp. 15452–15468, 2019.
- [21] I. Harvey, "The SAGA cross: The mechanics of recombination for species with variable-length genotypes," in *Proc. PPSN*, 1992, p. 271.
- [22] C.-Y. Lee and E. K. Antonsson, "Variable length genomes for evolutionary algorithms," in *Proc. GECCO*, 2000, p. 806.
- [23] A. H. Brie and P. Morignot, "Genetic planning using variable length chromosomes," in *Proc. ICAPS*, 2005, pp. 320–329.
- [24] Z. Qiongbing and D. Lixin, "A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems," *Expert Syst. Appl.*, vol. 60, pp. 183–189, Oct. 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417416301634>
- [25] X. Wu and K. Zhang, "Constrained optimal control problems of nonlinear systems based on improved Newton algorithms," in *Proc. 3rd Int. Conf. Inf. Comput. Social Syst. (ICSS)*, Aug. 2016, pp. 229–234.
- [26] V. Toğan and A. T. Daloğlu, "An improved genetic algorithm with initial population strategy and self-adaptive member grouping," *Comput. Struct.*, vol. 86, no. 11, pp. 1204–1218, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S004579490700301X>
- [27] F. Viveros-Jiménez, J. A. León-Borges, and N. Cruz-Cortés, "An adaptive single-point algorithm for global numerical optimization," *Expert Syst. Appl.*, vol. 41, no. 3, pp. 877–885, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417413006301>
- [28] The MathWorks Inc. (2015). *Selection Options—Selectionremainder*. [Online]. Available: <https://es.mathworks.com/help/gads/genetic-algorithm-options.html#f6593>
- [29] A. J. Umbarkar and P. D. Sheth, "Crossover operators in genetic algorithms: A review," *ICTACT J. Soft Comput.*, vol. 6, no. 1, pp. 1083–1092, 2015.
- [30] U. Derigs and R. Kaiser, "Applying the attribute based hill climber heuristic to the vehicle routing problem," *Eur. J. Oper. Res.*, vol. 177, no. 2, pp. 719–732, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377221705009446>
- [31] R. Azimi, "Co-operative driving at intersections using vehicular networks and vehicle-resident sensing," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, USA, Jul. 2015. [Online]. Available: [https://figshare.com/articles/Co-operative\\_Driving\\_at\\_Intersections\\_using\\_Vehicular\\_Networks\\_and\\_Vehicle-Resident\\_Sensing/6715379](https://figshare.com/articles/Co-operative_Driving_at_Intersections_using_Vehicular_Networks_and_Vehicle-Resident_Sensing/6715379)



**LUIS CRUZ-PIRIS** received the degree in telematics engineering and the master's degree in telecommunication engineering from the University of Alcalá, in 2012 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Telematics Service Engineering Research Group and also a Research Staff. Since 2013, he has been working in multiple research projects related with multimodal transportation systems, simulation, and cybersecurity. His research interests include intelligent agents, the IoT architectures, and intelligent transportation systems.



**IVAN MARSÁ-MAESTRE** received the degree in telecommunication engineering and the Ph.D. degree from the University of Alcalá, Spain, in 2003 and 2009, respectively, where he is currently an Associate Professor with the Computer Engineering Department. His research interests include the use of negotiation and nonlinear optimization techniques for distributed coordination of complex systems, such as computer networks, supply chains or vehicle management systems. He has taken part in many public and private research projects in these matters, has a number of publications in high impact international conferences and journals, and serves as a Program Chair and a Reviewer for some of them. From his research have emerged collaborative research lines with international research groups like the Center for Green Computing, at the Nagoya Institute of Technology in Japan, the Technical University of Delft (TUDelft) or the Center for Collective Intelligence, at the Massachusetts Institute of Technology in USA.



**MIGUEL A. LOPEZ-CARMONA** received the B.S. degree in electronics engineering from the University of Alcalá, in 1994, the M.S. degree in telecommunication engineering from the Polytechnic University of Madrid, in 1999, and the Ph.D. degree in telecommunication engineering from the University of Alcalá, in 2006. From 1995 to 2000, he held positions in Logytel and Alcatel as a Project Manager and a Research Scientist in the areas of railways and defense. Since 2000, he has been an Associate Professor with the Department of Computer Engineering, University of Alcalá, where he was the Vice-Dean of the Polytechnic School, from 2010 to 2017. He has been a Visiting Researcher with the Center for Collective Intelligence, Massachusetts Institute of Technology (MIT), where he did research in complex automated negotiations and optimization of complex networks. His research interests include intelligent transport systems, automated negotiation, multi-agent systems, and optimization of complex networks.

...

# Parte III

---

Conclusiones



## Conclusiones y líneas futuras

En este capítulo, además de mostrar las conclusiones de la investigación realizada e introducir las posibles futuras líneas de trabajo que pueden surgir a partir de las contribuciones de esta tesis, se realizará una discusión sobre los resultados obtenidos durante su desarrollo. Aunque en el resumen amplio se han realizado valoraciones sobre cada propuesta realizada, en este capítulo final es posible dar una visión global de los logros alcanzados con mayor claridad.

### 6.1 Discusión y conclusiones

En las secciones 2.1.3, 2.3.2.3, 2.3.4.4 y 2.3.5.5 se han mostrado, de forma resumida, los experimentos realizados para validar cada una de las contribuciones de esta tesis. En este apartado, se discutirán los resultados obtenidos realizando una valoración global de estos.

Respecto a la contribución sobre los métodos para la elección de la ubicación de sensores en una red de tráfico, la figura 2.2 permite analizar los resultados del sistema propuesto. Los valores obtenidos al aplicar únicamente las puntuaciones de centralidad muestran que, aunque los resultados están por debajo del valor de referencia, existe una relación directa entre estas dos curvas al analizar su evolución cuando aumenta el número de cámaras utilizadas. Este hecho valida la elección de esta métrica como base para el sistema de cálculo de ubicaciones. Además, cuando este sistema se despliega en un escenario simulado realista, como el utilizado para obtener los resultados mostrados en la figura 2.9, incluso en el caso de uso de la aplicación de un sistema multiagente muy optimizado, la aplicación de este método consigue reducir el número de mensajes (línea verde) necesarios para la toma de decisiones.

Por su parte, la contribución realizada en el ámbito de la coordinación de vehículos basada en la gestión de los semáforos en las intersecciones ha obtenido mejores valores que otros métodos tradicionalmente utilizados, tal y como se muestra en la tabla 2.2. Estos valores pueden verse desagregados en la figura 6.1, lo cual permite realizar un análisis de los mismos en profundidad. Casi un 27% de los vehículos no han variado el tiempo que tardan en realizar la misma ruta. Esto se debe a dos



circunstancias: es que esas rutas estén alejadas de los puntos de la red donde se generan las situaciones de congestión o que los desplazamientos se produzcan en franjas horarias diferentes a las de las horas punta de tráfico. En relación con los vehículos que se han visto perjudicados por el MAS (12.41 % del total), la mayor parte de ellos sufren incrementos de menos del 25 % de la duración de sus rutas. Por el contrario, casi un 45 % de los vehículos reducen sus tiempos en hasta un 25 %. Es importante destacar que entorno al 10 % de ellos recuden sus tiempos de desplazamiento sobre un 35 %, llegando a lograrse mejoras de casi el 90 %.

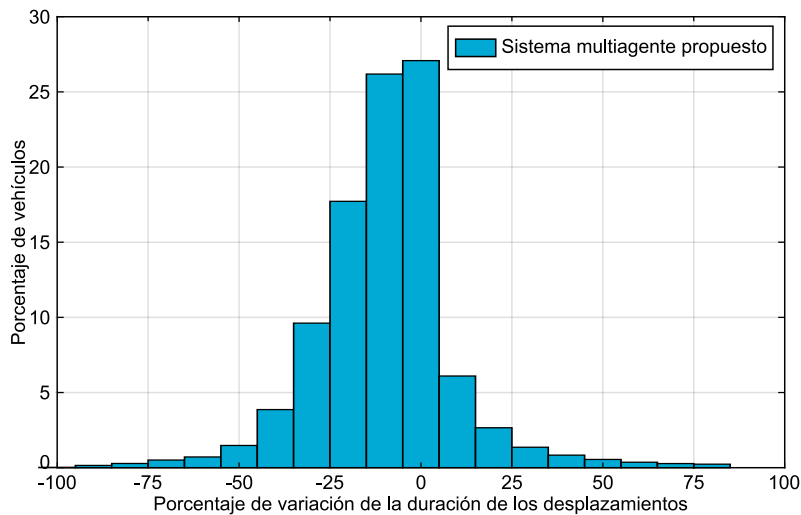


Figura 6.1.: Relación del número de vehículos que mejora, iguala o empeora los tiempos de sus desplazamientos al utilizar el sistema de gestión de semáforos propuesto.

La siguiente contribución de la tesis, la propuesta de modelado de intersecciones, ha sido validada al ser utilizada como base de los sistemas de optimización propuesto. La figura 6.2 agrupo los resultados de los experimentos realizados para cada uno de estos procesos de optimización (OCP, OCP-MP y OCP-MP-VL). Adicionalmente a estos resultados, las pruebas realizadas con OCP se han comparado con el valor máximo obtenido para un caso de uso concreto mostrado en el trabajo [27].

Lo anterior pone de manifiesto que estos optimizadores son capaces de obtener mejores resultados, de forma automática, que el estudio de referencia utilizado y frente a otros algoritmos de búsqueda local de soluciones. Tal y como se muestra en la figura 6.2, aunque existe una mejora en las puntuaciones medias obtenidas por cada uno de los optimizadores (barras azules), las diferentes contribuciones que se han ido añadiendo el proceso de optimización como, ampliar el espacio de búsqueda utilizando múltiples posibles caminos o el uso del algoritmo genético con cromosomas de longitud variable, ha repercutido directamente en la reducción del tiempo de cómputo necesario para resolver el problema (barras naranjas).

En el planteamiento inicial del problema que se realizaba en el capítulo 1, se señalaba la heterogeneidad de los causas que pueden dar lugar a problemas en un escenario

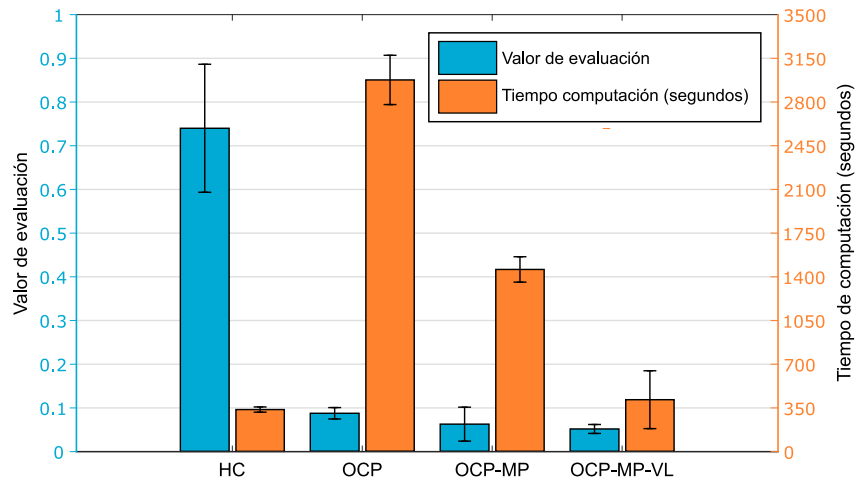


Figura 6.2.: Media de los valores de evaluación (color azul) y tiempos de computación (color naranja) obtenidos en cada experimento.

de tráfico y, cómo estas circunstancias no desembocaban siempre en las mismas situaciones. Dentro de este ámbito, esta tesis se ha centrado en las estrategias de optimización multiobjetivo para la coordinación de vehículos en uno de los elementos de las redes de carreteras que más conflictos genera: las intersecciones. Debido a la dimensión de los escenarios típicos cuando nos encontramos en un núcleo urbano, este objetivo principal siempre se ha abordado desde el planteamiento de soluciones que además de resolver el problema, realizaran el proceso de una forma automatizada, lo cual facilitara su escalabilidad. Para la consecución de esta meta, la investigación de esta tesis se inició definiendo los siguientes tres objetivos:

1. *Mejorar los métodos de ubicación de sensores en una red de tráfico de vehículos a través del desarrollo de una propuesta basada en el estudio de medidas de la propia red, como puede ser la centralidad de cada uno de sus nodos o enlaces, para obtener una metodología de búsqueda de ubicaciones idóneas que dependa en menor grado de los datos históricos de la propia red.*
2. *Realizar una propuesta de metodología de modelado de intersecciones que permita, a partir de características conocidas de ella como su forma, número de brazos, carriles por brazo y restricciones de giro, crear un modelo donde cada elemento esté etiquetado unívocamente utilizando un proceso automatizado.*
3. *Proponer, tanto una metodología para codificar el problema de la coordinación de vehículos en intersecciones de tal forma que pueda ser resuelto con procesos de optimización ampliamente utilizados y mejorar las técnicas utilizadas para este propósito.*

Para abordar el primero de los objetivos de la tesis, tras analizar el estado del arte en el ámbito de las redes de sensores en escenario de tráfico, se comenzó por desarrollar los métodos que permitieran representar las redes de carretas como grafos. Dadas las peculiaridades de estas redes se definió el denominado *pseudo-dual graph* que reunía las características de ser dirigido, multigrafo y basado en un grafo línea, lo cual solucionaba el problema de codificación relacionado con los giros prohibidos. Tras obtener un tipo de grafo que representa fielmente las redes de tráfico de vehículos, se comenzó con el estudio de métricas como la centralidad, evaluando sobre un escenario con tráfico realista, los resultados ofrecidos. Se evaluaron las ventajas que proporciona la medida de centralidad de intermediación o *betweenness* de cada enlace de la red para, posteriormente, realizar la propuesta para la ubicación de sensores que utiliza esta métrica y datos obtenidos del propio escenario. Esta solución se validó simulando un caso de uso que utilizaba cámaras para monitorizar el tráfico como sensores en la red de carreteras.

En relación con el segundo de los objetivos, el estudio de los trabajos relacionados con el modelado de intersecciones mostró la dificultad de encontrar sistemas que realizaran el proceso de modelado de forma automática y definieran, a la vez, todos los elementos que componen la intersección. Se comenzó por analizar algunos de los tipos de simuladores microscópicos existentes para, tras su análisis, decidir el que más se adaptaba a las características del tráfico que discurre por una intersección. La elección final fue utilizar los denominados TCA o autómatas celulares de tráfico. A partir de esta elección, se propuso una metodología para realizar la división del espacio en celdas de igual tamaño, compatibles con un TCA. También se definieron los pasos para nombrar a cada uno de los elementos que componen una intersección, poniendo especial cuidado en el valor que recibirían las celdas interiores, que indicarían las restricciones de giro de los vehículos que estuvieran situados sobre ellas. De igual forma, este modelo se extendió para poder ser utilizado en intersecciones con formas geométricas irregular, detallando las peculiaridades del proceso de modelado en estos casos. La validación de la propuesta se realizó de dos formas, la primera aplicando el modelo a un tipo de intersección ampliamente utilizada en la literatura como es la que cuenta con cuatro brazos, tres carriles de entrada y salida por brazo, y forma regular cuadrada. También se hizo lo mismo con una intersección real de cinco brazos y forma irregular. La segunda de las pruebas de validación fue utilizar el modelo desarrollado para simular las intersecciones utilizadas en los procesos de optimización de cruce de vehículos posteriores. Estas dos vías, la prueba teórica sobre un caso de uso y su simulación, confirman la validez y utilidad del modelo propuesto.

Sobre el tercero y último de los objetivos de esta tesis, se ha propuesto un conjunto de optimizadores capaces de contribuir a la coordinación de vehículos a su paso por intersecciones. El problema se abordó desde el punto de vista de dos metodologías

de gestión de tráfico diferentes, basadas en intersecciones señalizadas y sin señalizar. Tras realizar un estudio sobre las soluciones existentes, se desarrolló un sistema multiagente (MAS) para la gestión del tráfico basado en la variación de las fases de los semáforos, cubriendo de esta forma los sistemas de gestión basados en señalización. Este MAS está compuesto por tres tipos de agentes: los encargados de detectar atascos o situaciones de congestión, los que realizan los cambios en la señalización de los semáforos, y los controladores de la intersección, encargados de realizar la toma de decisiones. Respecto a las intersecciones sin señalizar, se propusieron varios optimizadores capaces de generar secuencias ordenadas de llegada de vehículos (patrones) a las entradas de una intersección, garantizando un cruce ordenado entre ellos, siguiendo las preferencias fijadas en cada caso. El problema de la coordinación de los vehículos se codificó para que fuera compatible con la aplicación de algoritmos genéticos. El primero de los optimizadores propuestos fue OCP (*Optimized Crossing Patterns*), diseñado para coordinar flujos de vehículos iguales por todas las ramas de la intersección, donde los vehículos seguían caminos fijados por el punto de acceso a ella, y cuyo objetivo era maximizar su uso. Utilizando el simulador de intersección desarrollado, esta maximización se mide por la ocupación media de la intersección durante el tiempo de simulación. Tras este primer optimizador, el sistema fue extendido para permitir libertad de elección de destino desde al cualquier punto de entrada a la intersección. Este segundo optimizador se denominó OCP-MP (*Optimized Crossing Patterns with Multiple Paths*) y varió también el objetivo global del mismo, permitiendo buscar los flujos de vehículos que cumplieran con los requisitos fijados por el sistema, sin la restricción de que tuvieran que ser iguales entre las diferentes entradas de la intersección. Estas variaciones hicieron que el espacio de búsqueda del algoritmo aumentara exponencialmente. Por este motivo se propuso el uso de un algoritmo genético con cromosomas de longitud variable, lo cual permitió resolver el problema en un único proceso de ejecución, obteniendo tanto los patrones de vehículos que más se aproximaban a los flujos de vehículos buscados y la longitud de patrón ideal. Todas estas contribuciones fueron evaluadas de forma exhaustiva a través de su prueba en escenario de tráfico simulados, arrojando resultados positivos en todos los casos.

## 6.2 Difusión de las contribuciones de la tesis

En este apartado se muestran las publicaciones a las que ha dado lugar, de forma directa, las contribuciones realizadas durante el desarrollo de esta tesis.

Las tres publicaciones que han dado lugar al compendio de artículos de esta tesis doctoral son las siguientes:

- **Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management** [50]. En este artículo (capítulo 3) se muestran las contribuciones relacionadas con la aplicación de medidas de centralidad de los enlaces de la red de carreteras para ubicar sensores de tráfico y el sistema multiagente propuesto para gestionar las fases de los semáforos de las intersecciones.

L. Cruz-Piris, D. Rivera, S. Fernandez, and I. Marsa-Maestre, “Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management”, *Sensors*, vol. 18, no. 2, p. 435, Feb. 2018.

- **Automated Optimization of Intersections Using a Genetic Algorithm** [67]. Las contribuciones sobre el método para modelar intersecciones y la propuesta de codificación del problema de coordinación de vehículos en intersección generando patrones de llegada de vehículos se recogen en esta publicación (capítulo 4).

L. Cruz-Piris, M. A. Lopez-Carmona and I. Marsa-Maestre, “Automated Optimization of Intersections Using a Genetic Algorithm”, in *IEEE Access*, vol. 7, pp. 15452-15468, 2019.

- **A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem** [68]. En este tercer artículo (capítulo 5) se muestran las contribuciones realizadas para la obtención de métodos de optimización del cruce de vehículos en intersecciones. Recoge los sistemas de optimización que extienden la metodología de codificación del problema anteriormente desarrollada, ampliando el espacio de búsqueda exponencialmente al permitir múltiples caminos y flujos de entrada diferentes por cada brazo y al generalizar el objetivo del optimizador para lograr las preferencias del sistema. En este artículo se recoge también el algoritmo genético con cromosomas de longitud variable desarrollado.

L. Cruz-Piris, I. Marsa-Maestre and M. A. Lopez-Carmona, “A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem” in *IEEE Access*, vol. 7, pp. 111968-111981, 2019.

De forma adicional a estas publicaciones principales, a continuación, se muestran otras publicaciones realizadas durante el desarrollo de esta tesis:

- **Intelligent Traffic Light Management using Multi-Behavioral Agents** [76]. En esta congreso se presentó la primera versión del sistema multiagente

propuesto para la gestión de las fases de los semáforos y sistema de simulación/emulación desarrollado para realizar las pruebas.

L. Cruz-Piris, D. Rivera, I. Marsa-Maestre, E. De La Hoz and S. Fernandez. “Intelligent Traffic Light Management using Multi-Behavioral Agents”, en *XIII Jornadas de Ingeniería telemática (JITEL 2017)*. Libro de actas. Editorial Universitat Politècnica de València. pp. 110-117.

- **Estrategias de control de tráfico de vehículos en un entorno urbano** [63]. En estas jornadas se presentó el estudio preliminar sobre los efectos del paso de vehículos por las intersecciones. También se mostró una versión preliminar de la propuesta de generación de patrones de llegada de vehículos para garantizar su cruce coordinado.

L. Cruz-Piris, “Estrategias de control de tráfico de vehículos en un entorno urbano” in *Séptimas Jornadas de Jóvenes Investigadores de la Universidad de Alcalá: (Ciencias e Ingenierías)*, (UAH obras colectivas. Ciencias 18). Alcalá de Henares: Universidad de Alcalá, Servicio de Publicaciones, 2019, pp. 17-25.

### 6.3 Líneas de investigación futuras

Los problemas relacionados con el tráfico de vehículos son uno de los retos más importantes a los que se enfrentan las grandes ciudades hoy en día. Debido a la gran diversidad de situaciones o causas que inician estos problemas, es fundamental focalizar los esfuerzos realizados en tareas específicas. El trabajo realizado en esta tesis se ha centrado en el problema concreto de las estrategias de coordinación multiobjetivo para la coordinación de vehículos en intersecciones. A partir de las contribuciones realizadas en este ámbito y de la experiencia adquirida durante la realización de esta tesis, se han definido algunas líneas de investigación futuras.

En relación con el uso de métricas basada en la centralidad de los enlaces de una red de tráfico, en la sección 2.1 se ha demostrado su utilidad para obtener ubicaciones candidatas a albergar sensores de conteo de vehículos. De forma implícita, esta propuesta presenta un modo de determinar, *a priori*, qué enlaces de la red van a ser los más utilizados por su valor de centralidad, de forma independiente a los datos de desplazamientos reales. Una posible línea futura de trabajo donde es posible aplicar esta contribución sería, a partir de las herramientas desarrolladas para la creación de grafos que representen redes de tráfico de vehículos, realizar el proceso inverso para diseñar las redes de carreteras a partir de grafos cuya medida de centralidad de sus enlaces esté compensada con la capacidad real de esos puntos. Esto aseguraría en cierta medida, que el tráfico potencial que va a circular por un determinado

punto se ajusta a la capacidad que puede llegar a asumir. Este enfoque de realizar modificaciones en las redes de carreteras en base a las medidas de centralidad puede utilizarse no sólo en nuevas carreteras, sino también en redes actuales, modificando los valores de centralidad a través de la variación de características de la red (velocidad máxima de la vía, situación de la señalización, etc.).

Las contribuciones realizadas por esta tesis en el campo de los métodos de optimización para la gestión de intersecciones, los cuales han mostrado muy buenos resultados al aplicarse a intersecciones aisladas, pueden ser extendidos a la coordinación entre intersecciones. De forma directa, puede extenderse el sistema multiagente mostrado en la sección 2.3.2.1 para que haga uso de los patrones de vehículos obtenidos del optimizador OCP-MP-VL. Los agentes de este MAS ya pueden ofrecer la inteligencia necesaria para decidir los flujos que debe gestionar cada intersección. A partir de esa información se obtendrían los patrones y puntos de entrada cada una de ellas, y sería también el MAS el encargado de comunicar a cada vehículo a qué medidas debe adoptar para sincronizarse con su patrón. Esta línea de investigación lleva una segunda que debería estudiar como aplicar estos optimizadores de forma masiva en un entorno que deba responder en tiempo real. Una posible dirección en la que trabajar es estudiar el número mínimo de combinaciones de flujos de entrada que se deben precalcular sobre cada intersección, para adaptarse a las futuras peticiones de un sistema gestor. Es decir, ya que las características físicas de la intersección permanecen constantes, es posible calcular un conjunto de combinaciones de patrones de llegada que sean almacenados, para poder saber que patrones aplicar sin necesidad de realizar el cálculo en ese momento.

Por último, una línea de investigación más transversal consistiría en aplicar el algoritmo genético con cromosomas de longitud variable desarrollado (sección 2.3.5.3) y sus funciones de expansión y contracción de genes, a problemas de otro ámbito que tengan características similares al problema tratado en esta tesis.

# Parte IV

---

Apéndices





## Recursos y herramientas para el estudio de escenarios de tráfico

Durante el desarrollo de la investigación realizada en esta tesis, ha sido necesario utilizar recursos, aplicaciones o herramientas para probar, validar y/o evaluar las contribuciones de este trabajo. La realización de experimentos en áreas donde su reproducción en la vida real no es viable requiere del uso de herramientas o técnicas basadas en la simulación. Estas limitaciones pueden darse por razones de diversa índole tales como su coste, el tiempo ejecución necesario o limitaciones tecnológicas. En este sentido, los simuladores son herramientas cuya finalidad es reproducir el comportamiento de los elementos que componen el escenario de un experimento concreto, con la mayor precisión posible, y en un entorno controlado. Cuando su funcionalidad se centra en la simulación de tráfico rodado, es necesario conocer las técnicas existentes para conseguir modelar de forma teórica su comportamiento.

Desde las primeras investigaciones en los años 50, donde se proponían modelos para el tráfico rodado asimilando los flujos de vehículos a la teoría de la mecánica de fluidos [77], existe una clasificación ampliamente aceptada entre los denominados modelos **microscópicos**, **macroscópicos**, y **mesoscópicos**. Estos modelos son descritos en base a los flujos de tráfico en [78] teniendo en cuenta si se basan en consideraciones del comportamiento espacio-tiempo de cada conductor en relación a los vehículos de su entorno (microscópico), el comportamiento de los conductores sin distinguir explícitamente su comportamiento espacio-tiempo (mesoscópico), o bien, desde un punto de vista de los flujos colectivos de vehículos (macroscópico).

Los requisitos del propio experimento pueden determinar el tipo de modelo a utilizar. Mientras que los de tipo microscópico nos proporcionan información sobre la posición y comportamiento de cada vehículo en cada instante de simulación, los modelos macroscópicos únicamente nos pueden ofrecer valores promedio o resultados finales (por ejemplo, la estimación de vehículos que han pasado por un determinado punto, en un periodo de tiempo concreto). Por este mismo principio de funcionamiento, los simuladores basados en modelados microscópicos requieren mayores tiempos de ejecución debido a la alta carga de computación que tienen respecto a los que utilizan modelos macroscópicos. Los simuladores mesoscópicos suelen basarse en la definición de entidades mayores que un único vehículo, por ejemplo, un grupo de vehículos, funcionando todos ellos como una única entidad. De esta forma ofrecen

más información que los macroscópicos, pero sin llegar al nivel de detalle de los microscópicos.

Los trabajos realizados en esta tesis se centran de forma directa o indirecta en el estudio e interacción del comportamiento individual de los vehículos bajo diferentes entornos. Por este motivo, es obligatorio utilizar herramientas de simulación de tráfico basados en modelados de tipo microscópico.

Centrándonos en los modelos microscópicos, cada vehículo puede representarse como una partícula, donde la interacción entre estas partículas viene determinada por cómo influye el movimiento de cada vehículo en el resto [79]. De entre los múltiples modelos de este tipo, la mayor parte de los simuladores se basan de la teoría denominada *car-following* o en el modelo basado en autómatas celulares (*Cellular Automata*, CA por sus siglas en inglés). En [79] ofrecen las siguientes definiciones para cada uno de ellos:

- **Modelos basados en *Car-following*:** Para cada vehículo se define una ecuación de movimiento que es análoga a las ecuaciones newtonianas de la dinámica de partículas. La aceleración de una de estas “partículas” es una respuesta a un estímulo que recibe desde una fuerza externa. En la práctica, este tipo de modelado se basa en cómo un vehículo reacciona a los movimientos de los vehículos que le preceden, en posición y tiempo, durante la simulación.
- **Modelos basados en CA:** Parte de una idealización de un sistema donde se presupone un espacio y tiempo discreto (o se ha producido previamente un proceso de discretización), de forma que cada unidad de interacción puede tomar únicamente un número finito de estados. En este tipo de modelos se realiza una división del espacio en porciones finitas, donde un vehículo puede o no estar, en un instante de simulación concreto.

En las siguientes secciones se analizarán las dos herramientas utilizadas de forma mayoritaria para llevar a cabo los experimentos mostrados en esta tesis. Debido al grado de novedad de alguna de las contribuciones, ha sido necesario desarrollar un simulador propio, basado en autómatas celulares, para analizar el comportamiento de los vehículos a su paso por una intersección. En este caso, se describirán los principales elementos, así como los modelos utilizados y sus diagramas de funcionamiento. Cuando el simulador utilizado se trate de una herramienta ya existente y creado por un tercero, se expondrán los motivos para su elección y las adaptaciones que se hayan realizado para cubrir las necesidades de los experimentos en los que ha sido utilizado.

## A.1 Simulador de tráfico basados en el modelo de *car-following*

En la introducción de este apéndice se hablaba sobre las diferencias de funcionamiento entre el modelo microscópico basado en Autómatas Celulares y el basado en *car-following*. Este segundo presenta algunas ventajas respecto al primero, como que sus resultados suelen ser más próximos al comportamiento real de los vehículos para escenarios de gran tamaño. Además, las variantes existentes de modelos de *car-following* lo hacen más versátil. En este apartado expondremos un breve resumen sobre algunos de los simuladores microscópicos más utilizados, el motivo para elegir uno de ellos y describiremos como generar la infraestructura necesaria para comunicar el sistema multiagente desarrollado con el simulador de tráfico elegido.

### A.1.1 Simuladores microscópicos comerciales

En [80] tenemos una enumeración de los principales simuladores de este tipo que existen en la actualidad. A continuación veremos una descripción de cada uno de ellos:

- **VISSIM[81]**: desarrollado por PTV en Alemania. Sus aplicaciones se extienden a la ingeniería de tráfico, sistemas de transporte público, y planificación urbana para la protección contra incendios.
- **PARAMICS[82]**: es una herramienta de tráfico microscópico desarrollada por SIAS Ltd y Quadstone Ltd de Escocia para una amplia gama de aplicaciones donde la congestión del tráfico es su principal característica.
- **AIMSUN[83]**: desarrollado y comercializado por TSS, es utilizado para mejorar las infraestructuras de carreteras, reducir las emisiones, recudir las congestiones y diseñar entornos urbanos para vehículos y peatones.
- **MITSIM[84]**: es una herramienta de simulación que se desarrolló para evaluar el impacto del diseño de sistemas de gestión de tráfico alternativo a nivel operativo, ayudando en el refinamiento del diseño posterior.
- **SUMO[47]**: *Simulation of Urban Mobility* es un simulador de tráfico de vehículos de tipo microscópico, de código abierto y altamente portable, diseñado para manejar grandes escenarios. Está desarrollado por el Instituto de Sistemas de Transporte del Centro Aeroespacial Alemán.

- **MATSIM[85]**: *Multi-Agent Transport Simulator*, es un simulador de transporte modular, capaz de trabajar con grandes escenarios y miles de agentes. Los diferentes módulos en los que está compuesto, pueden ser usados de manera individual o combinados para adaptarse a una necesidad específica. Está desarrollado en Java y se distribuye bajo la Licencia Pública General de GNU.

VISSIM es uno de los simuladores más utilizados a nivel empresarial, ya que fue uno de los primeros y de los que más tipos de productos proporciona. En nuestro caso nos centraremos en los dos últimos, SUMO y MATSIM, ya que ambos se distribuyen bajo licencia de código abierto, permitiendo su libre utilización y modificación. Aunque los dos proporcionan resultados muy realistas, hoy en día la comunidad de desarrolladores que tiene SUMO es mucho mayor a la de MATSIM, lo que facilita su evolución y la corrección de posibles fallos. Además, SUMO incluye herramientas adicionales para simular redes VANETs y para interactuar con el simulador durante su ejecución. Finalmente, cuenta con varios escenarios de tráfico realista cuyo uso está muy extendido en el estado del arte.

Por todo ello, el simulador elegido y utilizado durante esta investigación, cuando se ha necesitado reproducir escenarios de tráfico realista, es SUMO.

### A.1.2 Comunicación entre el sistema multiagente y el simulador de tráfico

En general, el funcionamiento habitual de un simulador de tráfico consiste en generar un escenario inicial a partir de una descripción detallada de este, en forma de datos de entrada. A partir de este estado inicial, el simulador aplica sus principios de funcionamiento para hacer evolucionar el escenario. Durante este proceso de evolución, el simulador recoge información, pudiendo además insertar nuevos elementos en él. Al finalizar la simulación se obtienen los datos que ha ido recogiendo.

En la sección 2.3.2.1 se propone un sistema multiagente para la gestión de las fases de los semáforos que controlan algunas de las intersecciones de un escenario de tráfico. Este MAS cuenta con varios tipos de agentes que recogen información del escenario, avisan sobre situaciones de congestión, toman decisiones sobre las acciones a realizar o varían las fases de los semáforos. Para llevar a cabo los experimentos relacionados con esta propuesta, es necesario contar con las herramientas que permitan simular estas situaciones.

Separando por funcionalidad las tareas que realizan estos agentes en relación con el escenario de tráfico, están los que necesitan recoger información de sensores y

los que deben interactuar con elementos del escenario para variar las fases de los semáforos. Por tanto, es imprescindible contar con herramientas que permitan la conexión con SUMO durante el tiempo de simulación. Esta tarea se realiza mediante el uso del módulo *Traffic Control Interface* (TraCI) [86], incluido en SUMO. Esta herramienta proporciona una arquitectura cliente/servidor basada en mensajes TCP (*Transmission Control Protocol*) que permite a una aplicación externa al simulador, interactuar con él. Existen clientes de TraCI implementados en varios lenguajes de programación, aunque el que recibe soporte oficial es el desarrollado en Python. Debido a esto, el emulador del sistema multiagente ha sido desarrollado también en Python, para facilitar el uso de esta herramienta.

En la figura A.1 se muestra el diagrama de ejecución básico del sistema conjunto de emulador y simulador.

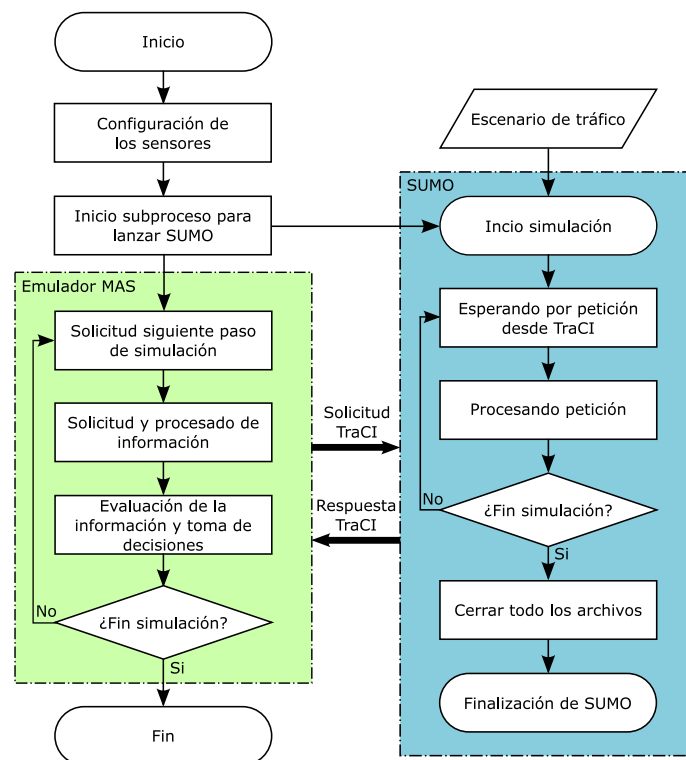


Figura A.1.: Diagrama de bloques con los principales pasos del emulador del sistema multi-agente y su interacción con el simulador de tráfico.

Cada agente incluido en el emulador del MAS es implementado en Python como una clase con un conjunto de atributos y métodos. El propio programa es el encargado de generar tantos objetos como agentes haya que desplegar en el escenario. También se encarga de la coordinación entre el MAS y SUMO. Los principales pasos durante el proceso de simulación conjunta de SUMO con el emulador del sistema multiagente son:

1. Inicialización: La aplicación lee los datos del escenario de tráfico, inicializa la configuración de los sensores incluyendo su definición en el escenario y lanza un subproceso con el simulador de tráfico SUMO. Para iniciar SUMO se le indica la ruta de los archivos que contiene el escenario de tráfico y los parámetros de configuración del simulador (duración de la simulación, tipo de modelo utilizado, ruta de los archivos que almacenarán los datos de salida, etc.). Para este caso concreto y poder recibir llamadas desde TraCI, SUMO se inicia con el parámetro `-remote-port` que hace que SUMO se inicie en modo servidor, esperando recibir peticiones por el puerto que se le haya indicado.
2. Bucle hasta el final de la simulación: Durante la configuración de SUMO se ha tenido que indicar el tiempo que representa cada paso de simulación (por defecto vale 1 segundo). Al inicio de cada bucle, el programa debe indicar a SUMO que desea ejecutar un número de pasos de simulación concretos, normalmente uno. Esta orden se realiza con la llamada a la función `traci.simulationStep(step)`, donde `step` indica el número de pasos de simulación que se van a ejecutar. Las tareas que realiza el emulador del MAS en cada iteración del bucle son:
  - a) Solicitar a SUMO que ejecute un paso de simulación y esperar su respuesta de fin de ese paso.
  - b) Solicitar la posición y velocidad de todos los vehículos activos en ese paso de simulación.
  - c) Procesar la información recibida y modelarla según los tipos de agentes que se hayan definido. Para un agente que utilice un sensor de tipo GPS, se almacenará un registro unitario correspondiente a la velocidad instantánea y la posición del vehículo. Para un agente que utilice un sensor tipo cámara, se hará un recuento de todos los vehículos cuya posición estén dentro del área de visión de la cámara. Se almacenará el número de vehículos y el instante temporal de la medición.
  - d) A partir de los datos recibidos y procesados, cada agente realizará las acciones que considere oportunas.
  - e) En el caso de que los agentes controladores de las intersecciones decidan que deben variar las fases de sus semáforos, generarán un nuevo conjunto de fases con el formato de SUMO, y utilizando la función `traci.trafficlight.setPhase(TLID, phase)` comunicarán a cada semáforo (identificado por un TLID único) que realice el cambio en su programación.
3. Fin de simulación: El bucle se repite tantas veces como pasos de simulación haya.

## A.2 Simulador de intersecciones basado en *Cellular Automata*

Las contribuciones realizadas en las secciones 2.3.4 y 2.3.5 se centran en la optimización del paso de vehículos a través de intersecciones, utilizando como base algoritmos genéticos. Este tipo de algoritmos requieren evaluar en cada iteración o generación las soluciones propuestas. En un escenario de tráfico como puede ser el estudio de una intersección, esta valoración se realiza a través de la simulación de la solución candidata sobre el escenario de tráfico. Las simulaciones deben realizarse en un tiempo de computación muy pequeño debido a que serán utilizadas de forma recurrente durante todo el proceso de resolución del problema y, por tanto, tienen una influencia muy importante sobre el tiempo total de convergencia de los optimizadores.

Los TCA - *Traffic Cellular Automata* (Autómatas Celulares de Tráfico) son modelos muy eficientes para la simulación microscópica de flujos de tráfico [28] en términos de tiempos de ejecución. Son utilizados de forma habitual en escenarios de tráfico de vehículos cuando analizar el comportamiento del tráfico, en una situación concreta, es un factor fundamental.

Existen diferentes propuestas para modelar este tipo de escenarios utilizando *TCA*, siendo necesario realizar en todos ellos la división del espacio en dos dimensiones con un número discreto de celdas en forma de rejilla homogénea (cada celda debe tener el mismo tamaño). En [87] se muestran aspectos fundamentales de los TCA tales como la forma de realizar la división de un espacio bidimensional o los tipos de movimiento permitidos. La forma que toma cada celda puede variar entre triangular o isométrica, hexagonal o rectangular, tal y como se aparece en el ejemplo de la figura A.2.a en color amarillo, verde o azul respectivamente.

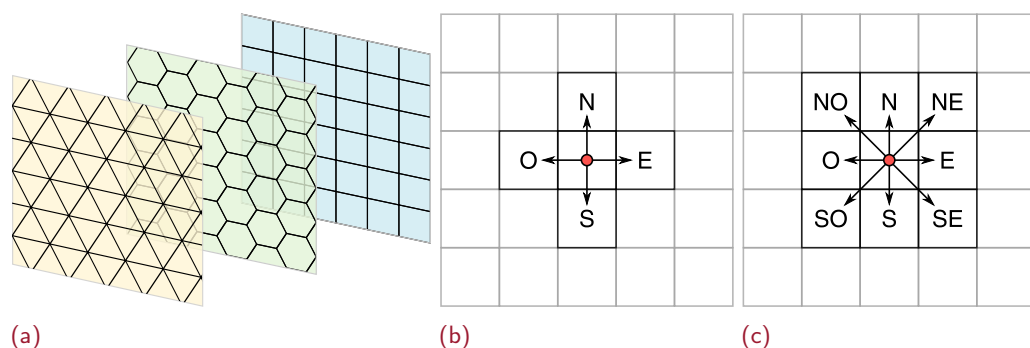


Figura A.2.: Principios básicos de un modelo basado de autómatas celulares. (a) Posibles tipos de división del espacio. (b) Movimientos de von Neumann. (c) Movimientos de Moore.



En este tipo de modelos es necesario indicar las reglas de movimiento permitidas entre una celda concreta y su vecindario, tanto a nivel del funcionamiento del propio modelo (cómo son los movimientos de los vehículos desde una celda a sus adyacentes), como de las restricciones propias que puede plantear un escenario concreto (opciones de giro prohibidas en una intersección en nuestro caso). Partiendo de un CA con movimientos con radio 1, existen dos posibles metodologías de movimiento a seguir para rejillas bidimensionales de forma rectangular. Por un lado, tenemos la definición de von Neumann (ver figura A.2.b) donde desde una celda central, los movimientos permitidos se restringen a 4 de sus celdas adyacentes (las situadas al norte, sur, este u oeste de esta). Por otro lado, la aproximación de Moore (ver figura A.2.c) permite movimientos sobre todas las celdas adyacentes a una celda central (8 posibilidades donde además de las 4 anteriores se añaden las opciones de noreste, noroeste, sureste y suroeste).

Un planteamiento similar al de un TCA aplicado a intersecciones se puede ver en [27]. En este trabajo se realiza división en celdas del interior de una intersección regular. Cada una de estas celdas recibe un valor numérico que se va incrementado de izquierda a derecha y con cada nueva fila de celdas. A partir de esta identificación básica de las celdas, el autor define el segmento que ocupa en un instante temporal un vehículo, el punto de salida al que llegará, y en base a ello crea una lista ordena de celdas por las que el vehículo ha de pasar para llegar de origen a destino. El camino o la trayectoria que sigue está fijado por la información que contiene una tabla que ha definido previamente. Aunque el funcionamiento básico de esta propuesta se asemeja al de un TCA, su simulador no puede funcionar por si mismo para representar trayectorias que no hayan sido precalculadas.

En los siguientes apartados mostraremos las estructuras de datos, funcionamiento, métricas y pruebas realizadas, sobre el simulador de intersecciones desarrollado.

### A.2.1 Estructuras de datos

El simulador de intersecciones basado en TCA, tras analizar las opciones disponibles, ha sido implementado como un conjunto de funciones de MATLAB. Esto facilita su integración y uso en combinación con los optimizadores propuestos en la sección 2.3.

Como se ha mencionado anteriormente, los TCA suelen hacer una división bidimensional del espacio en celdas, formando una cuadrícula o rejilla. En la sección 2.2.2 se ha tratado en profundidad la propuesta de modelado de una intersección de tráfico que permita identificar, de forma unívoca, todos los elementos que la componen. Siguiendo esta metodología, el simulador propuesto utiliza estructuras de datos

basadas en matrices, cuya dimensión depende del número de filas y columnas en las que se haya dividido el escenario de simulación. Todas las matrices que utiliza el simulador tendrán por tanto la misma dimensión, dependiente esta del tamaño del escenario y del tipo de división que se haya realizado del mismo.

Para ilustrar de forma clara el cometido de cada una de estas matrices, se utilizará un caso de uso concreto basado en una intersección con 4 ramas, con 3 carriles de entrada y de salida por cada una de ellas (figura A.3). Tomando como tamaño de celda el área de coincidencia entre los diferentes sentidos de la marcha, obtendremos una matriz de tamaño  $6 \times 6$  para modelar el interior de la intersección (zona azul). Además, se ha definido que cada carril de entrada (zona verde) y salida (zona roja) tenga una longitud de 6 celdas. Estos carriles, aunque no son imprescindibles desde el punto de vista de la simulación debido a que los eventos que se van a evaluar suceden en el interior de la intersección, proporcionan información útil a la hora de visualizar los patrones de entrada.

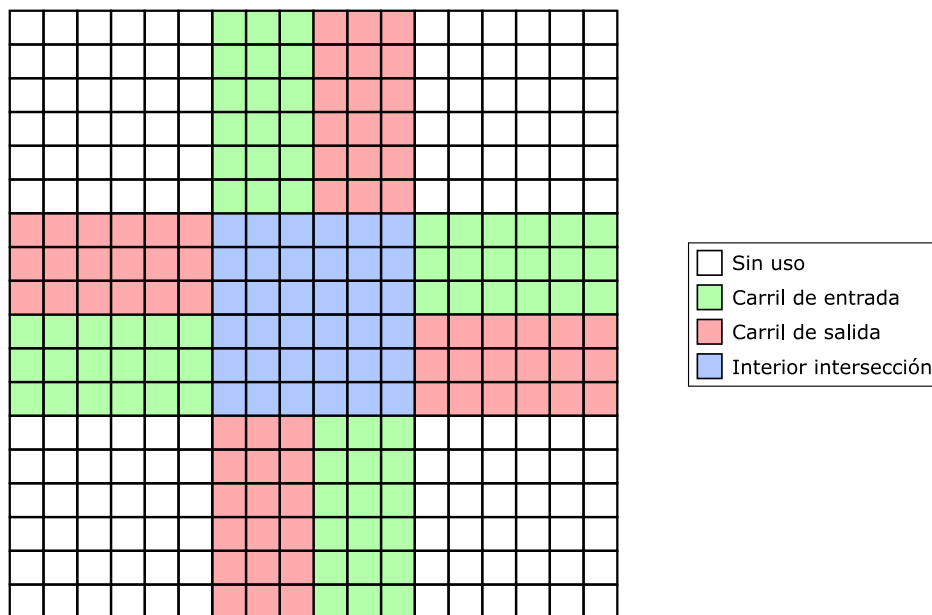


Figura A.3.: División del espacio de una intersección de 4 ramas, con 3 carriles de entrada y 3 de salida por cada rama.

El simulador ha sido diseñado para utilizar dos tipos de estructuras de datos en formato matricial. Las dimensiones de estas matrices están formadas por un conjunto de filas de 1 a  $f$  y un número de columna de 1 a  $c$ . Cada valor de la matriz se corresponde con una celda de la rejilla del escenario. A continuación, se explican estas dos matrices y se aplican al caso de uso definido:

- **moveRestriction:** Esta matriz se define como parámetro de entrada del simulador y tiene el siguiente formato:

$$moveRestriction = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1c} \\ t_{21} & t_{22} & \cdots & t_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ t_{f1} & t_{f2} & \cdots & t_{fc} \end{bmatrix}$$

El valor de la fila 1 y columna 1 se corresponde con la posición superior izquierda del escenario. Cada  $t_{xy}$  de la matriz contiene el valor del posible de giro desde la celda que ocupa la fila  $x$  y la columna  $y$  del escenario de simulación, según se especifica en la propuesta de modelado de la sección 2.2.2. De esta forma, las celdas cuya posición está fuera de la estructura que compone la intersección, son etiquetados con el valor 0 (los vehículos no pueden entrar ni transitar por ellas). El resto de las celdas almacenan el valor de giro posible desde esa celda, hacia sus celdas adyacentes, utilizando la codificación definida en la sección 2.2.2.1. Para el caso de uso, el valor de *moveRestriction* (*mR*) es:

$$mR = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 8 & 8 & 8 & 8 & 8 & 12 & 12 & 12 & 9 & 9 & 9 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & 12 & 12 & 12 & 9 & 9 & 9 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & 12 & 12 & 12 & 9 & 9 & 9 & 8 & 8 & 8 & 8 & 8 & 8 \\ 2 & 2 & 2 & 2 & 2 & 2 & 6 & 6 & 6 & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 6 & 6 & 6 & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 & 6 & 6 & 6 & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Esta matriz tiene dos utilidades, la primera de ella es definir el área de simulación ya que sus dimensiones representan el tamaño y estructura de la intersección. La segunda es la de poder ser consultada en cada movimiento que realice un vehículo, ya que contiene el valores de giro posible para cada

caso, entendiendo estos giros como las posibilidades de ir desde una celda a sus adyacentes.

- **currentState**: Tiene la misma dimensión que *moveRestriction* y es generada para cada instante  $i$  de simulación. La estructura de esta matriz es:

$$currentState_i = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1c} \\ v_{21} & v_{22} & \cdots & v_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ v_{f1} & v_{f2} & \cdots & v_{fc} \end{bmatrix}$$

Cada valor  $v_{xy}$  representa el valor actual de ocupación de cada celda del escenario de simulación. En un TCA convencional,  $v_{xy}$  sería un valor binario donde 0 representaría la ausencia de vehículo, y 1 la presencia. En este desarrollo se ha optado por permitir que más de un vehículo cohabite en una misma celda, marcando ese caso como una colisión. Esta forma de funcionamiento está diseñada para evaluar todas las posibles combinaciones de caminos en el interior de la intersección, sin eliminar *a priori* las que ocasionen alguna colisión. De esta forma, un camino que genere una única colisión tendrá una mejor puntuación que otro que genere varias colisiones, aunque los dos serán soluciones inválidas al finalizar el proceso de optimización.

Una vez se han definido las estructuras de datos que describen las características de la intersección y almacenan las posiciones de cada vehículo, es necesario terminar de completar la información que requerirá el programa para llevar a cabo la simulación completa del escenario.

Según se ha definido en la sección 2.2.2, el resultado de la optimización será una secuencia de llegada de vehículos a cada punto de entrada de una intersección. Estas secuencias, denominadas patrones de vehículos, serán por tanto un parámetro de entrada del simulador e indicarán, por un lado, cuando (en qué paso de simulación) se inserta un vehículo en un punto concreto de la intersección, y por otro lado, la ruta que deberá seguir el vehículo por su interior. Existen dos variaciones en cuenta a la definición de las rutas que siguen los vehículos siguiendo la definición de los procesos de optimización de la sección 2.3.4. Con el optimizador *Optimized Crossing Pattern* (OCP), los patrones de entrada son una cadena de bits (0 para ausencia de vehículo y 1 para presencia de vehículo) y la ruta que siguen los vehículos viene marcada por su punto de acceso a la intersección. Sin embargo, utilizando el optimizador *Optimized Crossing Patterns with Multiple Paths and Variable Length* (OCP-MP-VL), los patrones presentan valores entre 0 y el valor máximo de número de caminos posibles desde un punto de entrada hacia todos los posibles puntos de salida. Esta información del camino que puede seguir cada vehículos debe ser

indicado al simulador como parámetro al comienzo de cada proceso. Con este fin se ha definido una estructura con tres campos: coordenada  $x$ , coordenada  $y$ , y caminos posibles. Esta estructura tiene un fila por cada punto de entrada de la intersección identificado por los índices  $x$  e  $y$ . Para cada una de estas entradas se define un tipo de campo *cell* en Matlab que permite almacenar todos los caminos posibles desde ese punto de entrada, hacia los posibles puntos de salida. En el caso de OCP, esta *cell* contendrá un único camino, mientras que para OCP-MP-VL llega a contener hasta siete posibles opciones (siguiendo el principio de permitir caminos con un único cambio de dirección).

Con esta información, el simulador puede comenzar su tarea. Los pasos que sigue se describen a continuación:

1. A partir de los patrones de entrada que recibe como argumentos, los divide para obtener los fragmentos que corresponden a cada punto de entrada de la intersección. Estos fragmentos son extendidos teniendo en cuenta el tiempo de simulación que se haya configurado. Esta extensión se realiza concatenando el mismo fragmento tantas veces como sea necesario.
2. Se inicia la simulación con la rejilla de define al escenario vacía. En cada paso de simulación se inserta un vehículo siguiendo lo indicado por los fragmentos de patrón extendidos obtenidos en el paso anterior.
3. En cada paso de simulación se evalúan todas las celdas que estén ocupadas y se mueven los vehículos teniendo en cuenta la ruta que tiene fijada cada vehículo según el valor que tenían en su patrón de acceso, y las restricciones del propio TCA.
4. Este proceso se repite hasta final el tiempo de simulación.

## A.2.2 Métricas realizadas durante la simulación

El tiempo de simulación es calculado en base al tamaño de la intersección y del patrón a simular. La simulación llegue a un estado estable cuando, los valores calculados sobre la ocupación o la diferencia sobre las preferencias de tasas de llegada buscadas, sean fiables.

### A.2.2.1. Ocupación media

El valor de ocupación puede variar durante el tiempo de simulación. Por este motivo, para que el simulado pueda devolver una medida válida, se debe devolver un valor medio de ocupaciones.

Se define  $t_i$  como el instante temporal donde la simulación alcanza un estado estable. Ante de ese momento, no se ha rellenado la intersección con los vehículos que se están insertando, por lo que los valores de ocupación van a ser menores ya que la simulación se inicia con la rejilla vacía. Cada valor de ocupación en un paso de simulación ( $o(t)$ ), se obtiene de la suma de todos los valores de la matriz *currentState* dividida entre el número total de celdas de la intersección (ver figura A.4) . Por tanto, el valor medio de ocupación  $\bar{o}$  se calcula como la media aritmética de cada  $o(t)$  entre  $t_i$  y  $t_{end}$  (último paso de simulación), tal y como se mostró en la ecuación 2.17:

$$\bar{o} = \frac{\sum_{i=t_{stable}}^{i=t_{end}} o(i)}{t_{end} - t_{stable} + 1}$$

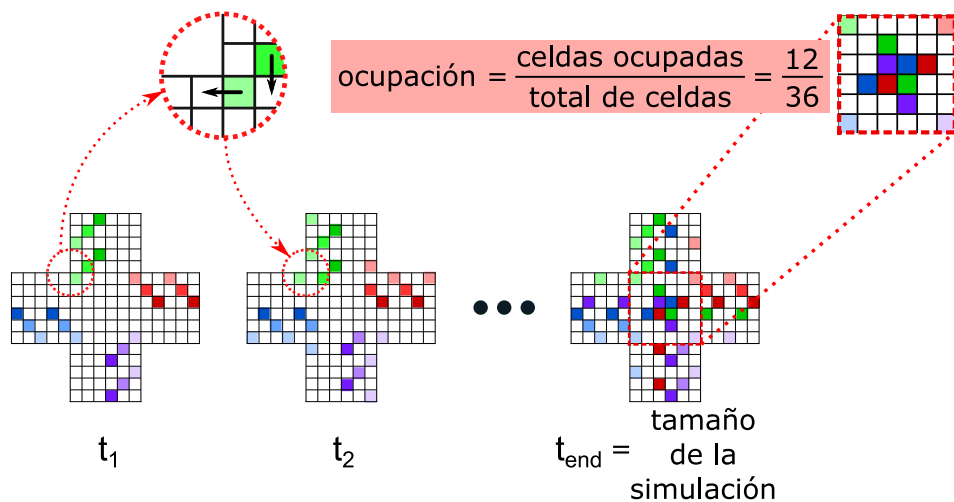


Figura A.4.: Representación de los movimientos que realizan los vehículos y el cálculo de la ocupación de un paso de simulación.

### A.2.2.2. Medidas de error

Además de la medida de ocupación, para los optimizadores cuyo objetivo es obtener una tasa de vehículos determinada, es necesario que el simulador calcule métricas sobre las posibles desviaciones entre el vector  $Q$  buscado y el obtenido. En el simulador se han incluido el cálculo de los siguientes valores:

- **Colisiones medias** ( $C_{avg}$ ): En cada paso de simulación  $i$ , se almacena el número de celdas ocupadas por más de un vehículo. Cuando el tiempo de simulación  $l_{sim}$  finaliza, se calcula la media todos los valores de colisión parciales obtenidos.

$$C_{avg} = \frac{\sum_{i=1}^{l_{sim}} c_i}{l_{sim}} \quad (A.1)$$

- **Suma de errores absolutos** ( $SAE$ ): Al finalizar la simulación, se comparan los flujos preestablecidos definidos en  $Q$  por los obtenidos al final del proceso de optimización  $q_{opt}$ . La suma del valor absoluto de la diferencia de estos valores proporciona el valor final de  $SAE$ .

$$SAE = \sum_{i=1}^{size(Q)} |Q(i) - Q_{opt}(i)| \quad (A.2)$$

- **Error medio absoluto normalizado por flujos** ( $NMAE_{flows}$ ): Este parámetro mide la diferencia entre  $Q$  y  $Q_{opt}$ , evaluando cada subflujo de forma independiente.

$$NMAE_{flows} = \frac{\sum_{i=1}^{size(Q)} \frac{|Q(i) - Q_{opt}(i)|}{Q(i)}}{size(Q)} \quad (A.3)$$

### A.2.3 Visualización de resultados

Aunque las métricas definidas en el apartado anterior ofrecen una cuantificación numérica referida a la simulación realizada, en muchas ocasiones es de utilidad poder representar la simulación para poder entender mejor el comportamiento de los patrones.

Utilizando las opciones para visualizar con las que cuenta MATLAB, es posible realizar una impresión de cada matriz *currentState* utilizando las funciones implementadas. Un ejemplo de la visualización de una simulación se muestra en la figura A.5. Esta función permite configurar el tamaño de la cuadrícula donde se imprimirán las ocupaciones parciales (6×7 en el caso del ejemplo). Además, como título de cada figura aparece la ocupación  $o(t)$  de ese paso de simulación.

Con el objetivo de poder seguir los movimientos, todos los vehículos que acceden a la intersección por el mismo brazo, se pintan con el mismo color, variando el tono entre los diferentes carriles.

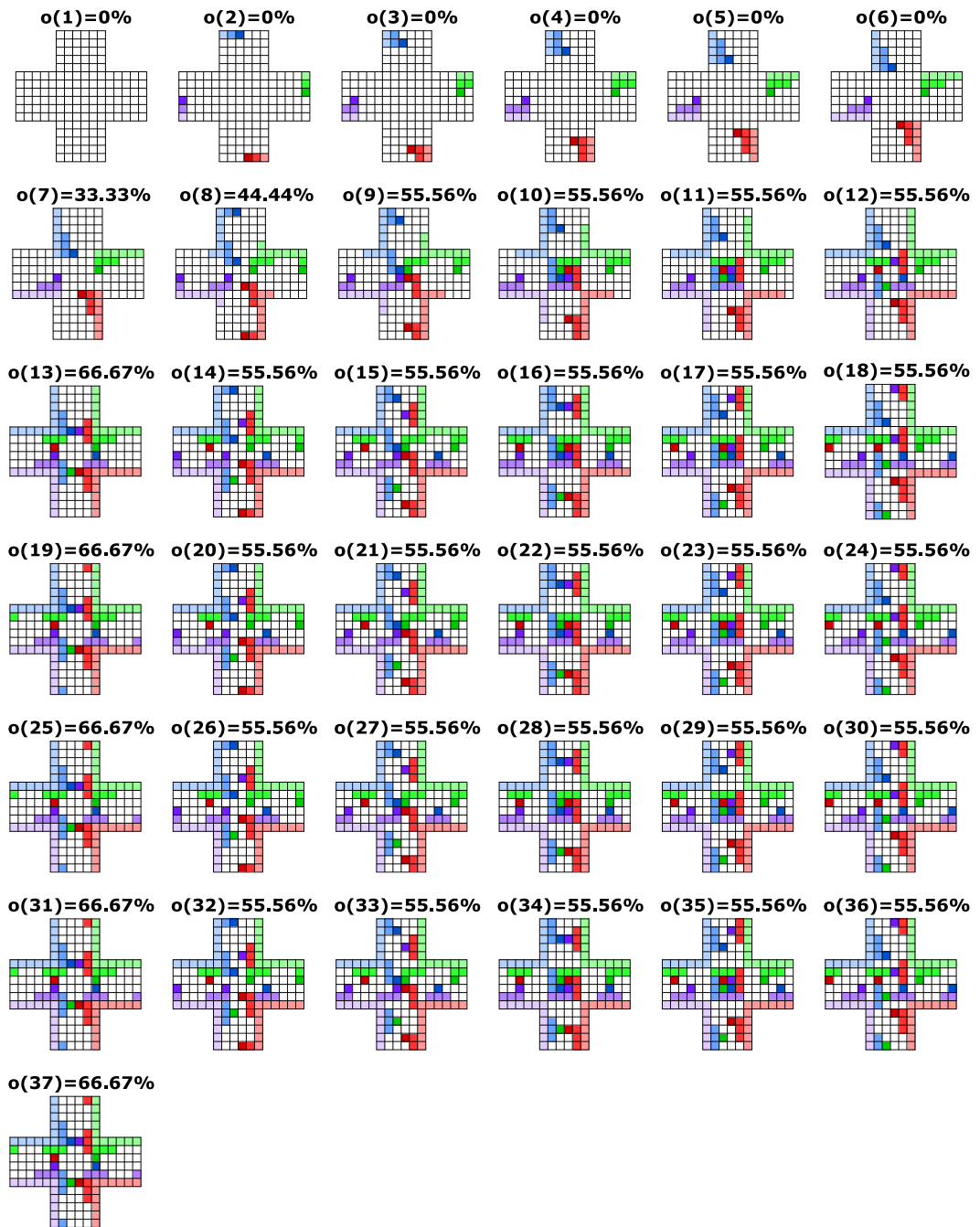


Figura A.5.: Representación de la posición de cada vehículo según el instante de simulación.

De igual forma, esta representación es posible visualizarla en movimiento, presentando cada figura de ocupación parcial, como un fotograma de un vídeo.





## Escenarios de tráfico urbano y datos de entrada

Durante el desarrollo de esta tesis se han llevado a cabo experimentos para poder validar, evaluar y cuantificar la mejora que suponen las contribuciones realizadas. Estos experimentos, en su mayoría, se basan en la simulación de escenarios de tráfico de vehículos, los cuales permitan comparar los resultados obtenidos al aplicar las técnicas que componen las contribuciones de este trabajo, frente a otras soluciones de referencia. En esta sección se describirán en detalle los escenarios o datos de entrada utilizados, indicando su origen, características y las posibles adaptaciones que se hayan realizado sobre los datos originales.

Las herramientas de simulación utilizadas en esta investigación y descritas en el apéndice A se pueden agrupar en las que tienen como objetivo recrear un escenario de tráfico de gran tamaño utilizando el simulador microscópico SUMO, y las que se basan en reproducir el comportamiento de los vehículos en una única intersección, utilizando el simulador desarrollado basado en un autómata celular. Debido a esta diferenciación, en las siguientes secciones se mostrarán los escenarios de tráfico utilizados en ambos casos.

### B.1 Escenario de tráfico urbano realista: TAPASCologne

Las pruebas sobre escenarios reales son un elemento básico de la validación de cualquier propuesta realizada. En determinados casos, como los experimentos sobre tráfico de vehículos que deben ser llevados a cabo en su mayoría en simuladores (sección A.1), este tipo de pruebas realistas se logran utilizando escenarios de tráfico basados en datos históricos recogidos en situaciones reales. En este sentido, en la documentación del simulador SUMO aparecen varios escenarios de tráfico urbano que la comunidad de desarrolladores ha ido implementando sobre diferentes ciudades. Uno de los más populares, tanto por su dimensión como por ser el primero de gran tamaño que se realizó para este simulador, es el denominado “Travel and Activity Patterns Simulation (TAPAS) Cologne” [46], [88]. Este escenario surge por la iniciativa del Instituto de Sistemas de Transporte del Centro Aeroespacial Alemán

(TIS-DLR) con el objetivo de reproducir, con el máximo de realismo posible, el tráfico vehicular de una zona urbana de gran tamaño de la ciudad alemana de Colonia, durante un periodo de 24 horas.

Los elementos básicos que forman un escenario de simulación son una red vial y un conjunto de rutas. Los desarrolladores del escenario utilizaron las fuentes de datos y métodos que se detallan a continuación para construirlo:

- Aunque inicialmente, para generar la red de carreteras que componen el escenario, se utilizaron datos de origen no abierto, según fue evolucionando el proyecto se realizó un mapeo de estos datos a los provenientes *OpenStreetMap* (OSM). Esta fuente de datos de información geográfica se distribuye con licencia ODbL (Licencia Abierta de Bases de Datos), la cual garantiza que tanto el acceso como su posterior distribución sea libre. La representación de esta red de carreteras puede verse en la figura B.1.



Figura B.1.: Representación con SUMO-GUI del mapa de TAPASCologne

- La simulación microscópica de la movilidad de los vehículos se ha realizado con SUMO.

- La información de la demanda de tráfico (flujos macroscópicos de tráfico tales como matrices de orígenes/destinos) se ha obtenido utilizando la metodología de *Travel and Activity PAtterns Simulation* (TAPAS) [46].
- Los flujos de tráfico obtenidos con el paso anterior se han mejorado de forma iterativa utilizando el algoritmo de asignación de usuarios dinámica de Gawron [89].

El resultado final es una captura de tráfico de la ciudad de Colonia, de una superficie de 400 km cuadrados, durante un periodo de 24 horas.

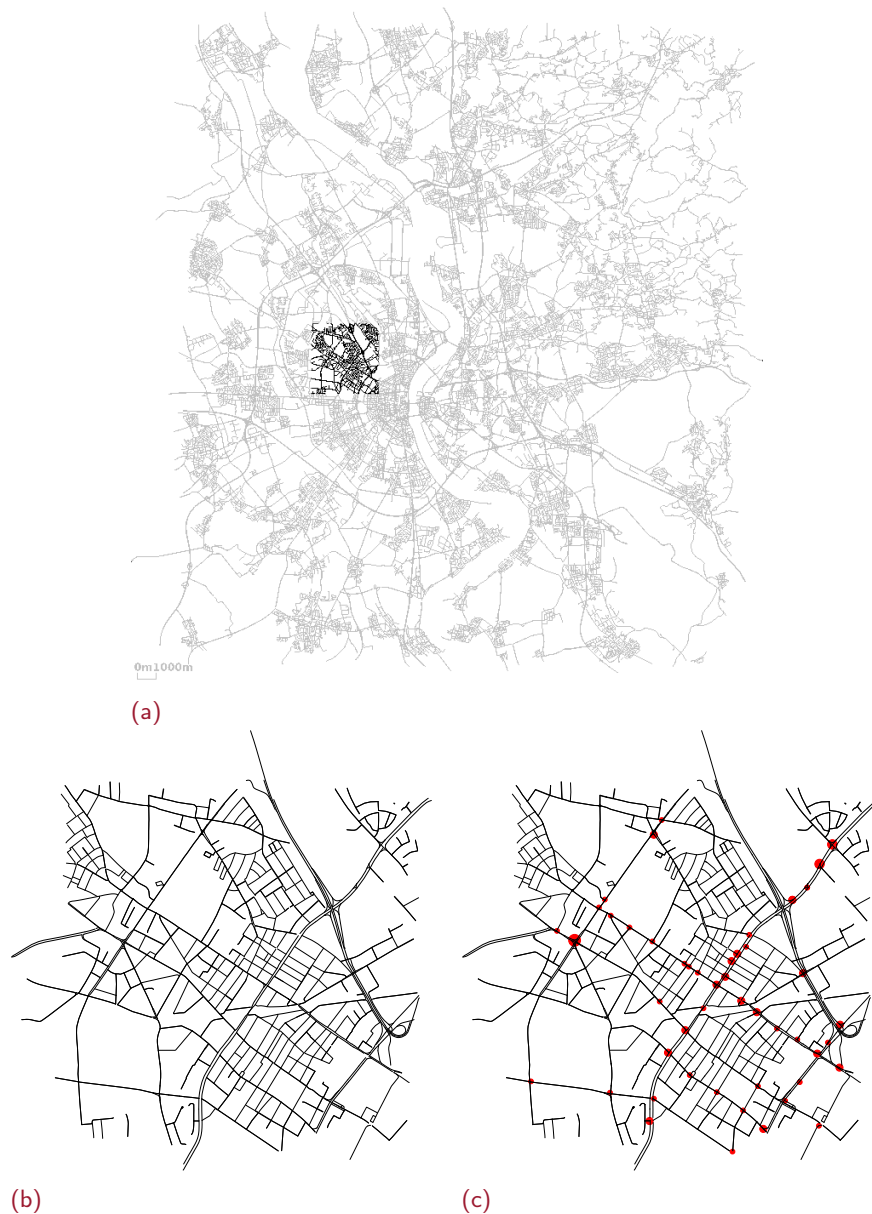
### B.1.1 Escenario de tráfico reducido basado en TAPASCologne

El escenario de simulación original está compuesto por una red de carreteras con 68 642 enlaces, 30 354 nodos y 1 547 333 rutas, lo que genera tiempos de simulación elevados. Esto puede ser un inconveniente para llevar a cabo procesos que necesiten hacer un uso intensivo de este escenario, debiendo realizar muchas simulaciones antes de obtener un resultado. Por este motivo se ha optado por realizar un recorte del escenario (red de carreteras y rutas), que mantenga la complejidad que es necesaria para la evaluación de las propuestas realizadas, pero reduzca los tiempos de simulación. Para obtener este subescenario, se han analizado las propiedades y características que debería cumplir ese fragmento de mapa candidato siguiendo lo establecido en [90], donde los autores comprueban estos valores en una red simulada del centro de la ciudad de Toronto.

En la figura B.2a se puede observar el mapa original de TAPASCologne y la zona elegida como candidata para realizar este recorte. Esta región ampliada puede verse en la figura B.2b y está compuesta por 1 416 enlaces y 716 nodos. Una de las razones para elegir esta porción del mapa es que cuenta con 73 intersecciones gestionadas por semáforos, las cuales se representan en color rojo en la figura B.2c. Para obtener las rutas del nuevo escenario, se han seleccionado todas aquellas que se inician, finalizan o atraviesan ese fragmento del mapa, obteniendo un total de 246 374 rutas.

### B.1.2 Ajustes y calibración

Tras definir el escenario de tráfico que se va a utilizar para las pruebas que requieran de simulaciones realistas, se debe comprobar su correcto funcionamiento y adaptarlo a las necesidades de este trabajo. La documentación incluida en el propio escenario,



**Figura B.2.:** Fragmento del mapa de Colonia recortado para la versión reducida del escenario. (a) Área seleccionada del mapa original. (b) Recorte del mapa. (c) Intersecciones del mapa recortado gestionadas por semáforos.

indica que se debe utilizar siempre con un valor de *scale* igual al '0.3'. Este parámetro permite reducir el número de rutas que son cargadas desde el archivo de entrada, utilizándose normalmente para variar el grado de carga de un escenario sin tener que modificar el archivo de rutas.

Este proceso de escalado puede ser contradictorio con lo comentado anteriormente, ya que si el escenario de TAPASCologne está ajustado para que sea lo más parecido a la realidad, no parece haber motivo para tener que realizar una reducción del número de rutas insertadas en la simulación. La explicación que dan sus desarrolladores para realizar este escalado es que con el número total de rutas que tiene el archivo

cologne.rou.xml, la red se satura muy rápido y la simulación no puede terminar de forma correcta. De forma empírica han realizado este escalado al 30 % de las rutas originales, y han comprobado que los datos ofrecidos al finalizar la simulación son extrapolables a la realidad.

El escalado se produce descartando algunas de las rutas que el simulador va leyendo del archivo de rutas, en orden temporal de inicio de la ruta (para el valor 0.3, por cada ruta que se inserta en la simulación, se descartan dos). Este valor de 0.3 ha sido probado para el escenario completo de TAPASCologne por lo que, para el escenario reducido que se ha realizado en esta sección, debe ser recalculado. Para realizar este estudio de la capacidad que tiene este subescenario, se han realizado un conjunto de simulaciones en las cuales se ha ido incrementando el parámetro *scale* en pasos de 0,01.

Durante el proceso de simulación en SUMO, en ocasiones se produce un evento denominado *teleport* [91], el cual es muy útil para identificar si hay algún tipo de incidencia en la simulación por saturación de vehículos. Los *teleports* son un mecanismo que utiliza SUMO para evitar situaciones de congestión anómalas debidas el propio funcionamiento del simulador. Cuando un vehículo permanece demasiado tiempo esperando para acceder a una intersección o el vehículo colisiona con su líder (vehículo al que sigue al utilizar el modelo de simulación de *car-following*), SUMO trata de evitar un punto de la red que permanecería cerrado, “teletransportando” a ese vehículo a una posición posterior de su ruta. Este tipo de evento puede darse en cantidades muy reducidas por errores en la simulación. Cuando el valor *teleports* aumenta, es previsible que la carga de tráfico que se esté insertando en el escenario sea excesiva. Teniendo cuenta esto, para el conjunto de experimentos de calibración del parámetro *scale*, se almacenará el valor de *teleports* total que se han dado lugar durante el tiempo de simulación.

En la figura B.3 se muestran los resultados obtenidos para este proceso de ajuste del parámetro *scale*.

El número de vehículos insertados es directamente proporcional al propio parámetro (línea azul). Para valores inferiores a 0.4, el número de *teleports* permanece en valores muy reducidos (241 para casi 100 000 vehículos insertados). Sin embargo, entre los valores de *scale* de 0.4 a 0.6, este valor empieza a incrementarse considerablemente (en torno al 15 % del total de vehículos insertados). Para valores superiores a 0.6, el número de *teleports* se aproxima al número total de vehículos insertado, por lo que en este último caso los resultados de la simulación no representan el comportamiento real de los vehículos. Por este motivo, el valor de *scale* elegido para este escenario reducido denominado *TAPASCologne-Lite* será de 0.4, insertando un total de 98 550 vehículos durante la 24 horas de simulación.

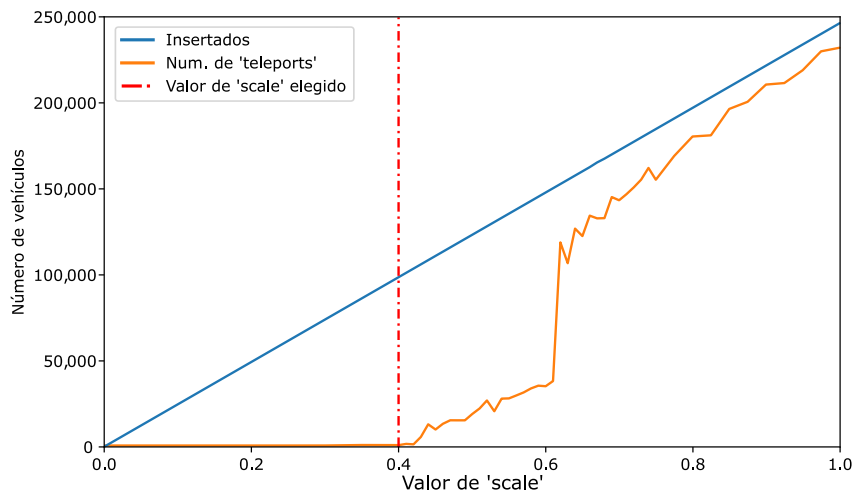


Figura B.3.: Evolución del número de *teleports* registrados según se modifica el parámetro *scale*.

## B.2 Generación de flujos de tráfico para el simulador de intersecciones

Para realizar los experimentos del sistema multiagente de la sección 2.3.2.1 se recurrió a un escenario basado en tráfico real debido que uno de los retos de esa contribución era validar la propuesta en un sistema realista. El enfoque de los experimentos de la coordinación de vehículos en intersecciones utilizando patrones de llegada debe ser diferente. Los objetivos de los optimizadores utilizados se centran en obtener un comportamiento determinado de los vehículos a su paso por la intersección. En uno de los casos el objetivo es maximizar la ocupación de la intersección (sección 2.3.4) por lo que no es necesario contar con flujos de entrada de prueba, ya que el fin que se persigue es obtener los patrones de vehículos que, en de media, más celdas de la intersección ocupen. Sin embargo, al variar el objetivo en el optimizador de la sección 2.3.5 para buscar los patrones que generen los flujos más cercanos a los que marque el sistema, se hace imprescindible contar con un conjunto de flujos precalculados que permitan validar su funcionamiento frente a diferentes condiciones.

Para validar los optimizadores OCP, OCP-MP y OCP-MP-VL se han generado un conjunto de datos de entrada, que pueden ser utilizados en cada uno de los experimentos, permitiendo la comparación de los resultados obtenidos. Existen dos posibles enfoques a la hora de generar estos conjuntos de datos de entrada. El primero de ellos es definir un conjunto de flujos de vehículos candidatos de los cuales se conoce al menos una posible solución. La segunda metodología se centra en la generación aleatoria de flujos, siguiendo determinadas preferencias que permitan probar los casos de resolución límites.

Todos los flujos que contienen estos conjuntos de datos de entrada están diseñados para una intersección de cuatro brazos y, por tanto desde cada brazo existen tres posibles giros formado cada combinación un vector  $Q$  con 12 elementos tal que:

$$Q_{UCI} = [q_1(4), q_1(3), q_1(2); q_2(1), q_2(4), q_2(3); q_3(2), q_3(1), q_3(4); q_4(3), q_4(2), q_4(1)],$$

donde  $Q_{UCI}$  es el vector de valores de entrada al optimizados y cada  $q_a(b)$  indica la proporción de vehículos que componen el flujo que va desde el brazo  $a$  hasta el brazo  $b$  de la intersección. De esta forma, es posible agrupar el total de las opciones globales de giro de cada vector  $Q$  de la siguiente forma:

$$\begin{aligned} q_{right} &= [q_1(4), q_2(1), q_3(2), q_4(3)] \\ q_{go} &= [q_1(3), q_2(4), q_3(1), q_4(2)] \\ q_{left} &= [q_1(2), q_2(3), q_3(2), q_4(1)] \end{aligned}$$

### B.2.1 Flujos de vehículos candidatos con soluciones conocidas

Para realizar este conjunto de datos se han definido patrones de vehículos que permitan el cruce de los vehículos sin colisiones. Estos patrones pueden ser validados de forma empírica a través de su representación, tal y como se muestra en la figura B.4.

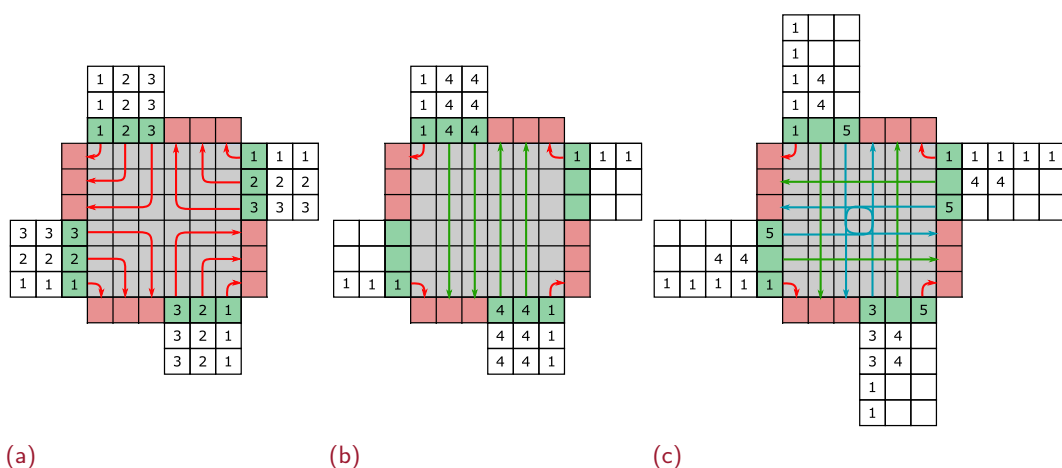


Figura B.4.: Flujos de tráfico (volumen y tipo de giro) con patrones de vehículos conocido (rojo=giro derecha, verde=continuar recto y azul=giro izquierda). (a) Todos los vehículos giran a la derecha. (b) Los vehículos giran a la derecha y continúan recto. (c) Los vehículos giran a la derecha, continúan recto y giran a la izquierda.



Se han diseñado tres posibles  $Q_{Ax}$  con soluciones conocidas y comportamiento que permitan comprobar el funcionamiento de los optimizadores en diferentes aspectos. Los vectores definidos son:

- **Experimento A1:** Representado en la figura B.4a representa la preferencia de los vehículos de que todos ellos quieren girar a la derecha. Este tipo de movimiento es el que menos puntos de conflicto presenta ya que existen caminos para cursar todo el tráfico sin que se produzcan puntos de conflicto. El valor del vector en este caso sería:

$$Q_{A1} = [1, 0, 0; 1, 0, 0; 1, 0, 0; 1, 0, 0]$$

- **Experimento A2:** En este caso, mostrado en la figura B.4b se van a combinar dos tipos de opciones de giro, continuar recto o girar a la derecha. El caso buscado tampoco presenta puntos posibles de conflicto. El valor de  $Q$  se define de la siguiente forma:

$$Q_{A2} = \left[ \frac{1}{3}, \frac{2}{3}, 0; \frac{1}{3}, \frac{2}{3}, 0; \frac{1}{3}, \frac{2}{3}, 0; \frac{1}{3}, \frac{2}{3}, 0 \right]$$

- **Experimento A3:** El tercero de los vectores de flujos con solución conocida ha sido extraído de la solución planteadas en [27] (figura B.4c). En este caso, se utilizan para todos los brazos de la intersección las tres opciones de giro existentes, produciéndose puntos de conflicto que el optimizador deberá resolver. El vector queda definido como:

$$Q_{A3} = \left[ \frac{5}{15}, \frac{2}{15}, \frac{1}{15}; \frac{5}{15}, \frac{2}{15}, \frac{1}{15}; \frac{5}{15}, \frac{2}{15}, \frac{1}{15}; \frac{5}{15}, \frac{2}{15}, \frac{1}{15} \right]$$

## B.2.2 Flujos de vehículos generados de forma aleatoria

En el segundo grupo de experimentos, es necesario definir flujos de vehículos candidatos. Al tratarse de flujos aleatorios, existe una probabilidad muy baja de que se genere una combinación de valores para un vector  $Q$  que sea compatible con una solución perfecta. Por este motivo, la forma de evaluar cada optimización será midiendo el método que obtenga soluciones más próximas al valor buscado.

Dada la aleatoriedad de la forma de generar los flujos, no es posible generar únicamente un vector para cada proporción probada, ya que podría darse fuera un caso extremo o muy sencillo de resolver. Debido a esto, para cada una de las proporciones que se van a generar, se crean conjuntos de 100 vectores de entrada  $Q$ . Esto permitirá

probar de forma intensiva cada optimizador y obtener posteriormente valores medios de diferencias.

De esta forma se han definido tres métodos para la generación de patrones que siguen las siguientes distribuciones:

- **Experimento B1:** Cada conjunto de flujos de entrada para un brazo concreto de la intersección, se calcula siguiendo estos tres pasos:
  - Se genera un valor aleatorio  $q_{rand1}$  tal que:  
 $0 < q_{rand1} \leq 0,6$ .
  - Se genera un valor aleatorio  $q_{rand2}$  tal que:  
 $0 < q_{rand2} \leq 1 - q_{rand1}$ .
  - Se genera un valor aleatorio  $q_{rand3}$  tal que:  
 $0 < q_{rand3} \leq 1 - q_{rand1} - q_{rand2}$ .
  - $q_{rand1}$ ,  $q_{rand2}$  y  $q_{rand3}$  son asignados de forma aleatoria a las posiciones del vector que representa cada opción de giro ( $q_{right}(i)$ ,  $q_{go}(i)$  y  $q_{left}(i)$ ).
  - Este proceso se repite tantas veces como brazos tenga la intersección. Al finalizar, el vector  $Q_{UCI}$  ha sido rellenado por completo.
- **Experimento B2:** En este caso se calculan tres valores independientes para  $q_{rand1}$ ,  $q_{rand2}$  y  $q_{rand3}$ , obteniendo valores aleatorios entre 0 y 0.7, 0.2, y 0.1 respectivamente. Los valores obtenidos son barajados para su asignación a las tres opciones de giro de cada brazo.
- **Experimento B3:** El último de los conjuntos de datos de entrada está centrado en un caso extremo, donde todos los vehículos de cada rama quieren realizar el mismo tipo de giro con una proporción entre 0 y 1.

Para validar los conjunto de datos generados se ha realizado la representación de sus valores utilizando los diagramas de cajas mostrados en la figura B.5. Las gráficas de esta figura pueden dividirse en dos tipos. Las primeras (situadas a la izquierda) muestran los valores  $q_{randX}$  generados siguiendo cada una de las proporciones definidas anteriormente. El otro tipo de gráficos (situadas a la derecha) se centran en representar las tasas de flujos asignados, en valor medio, a cada opción de giro de cada brazo.

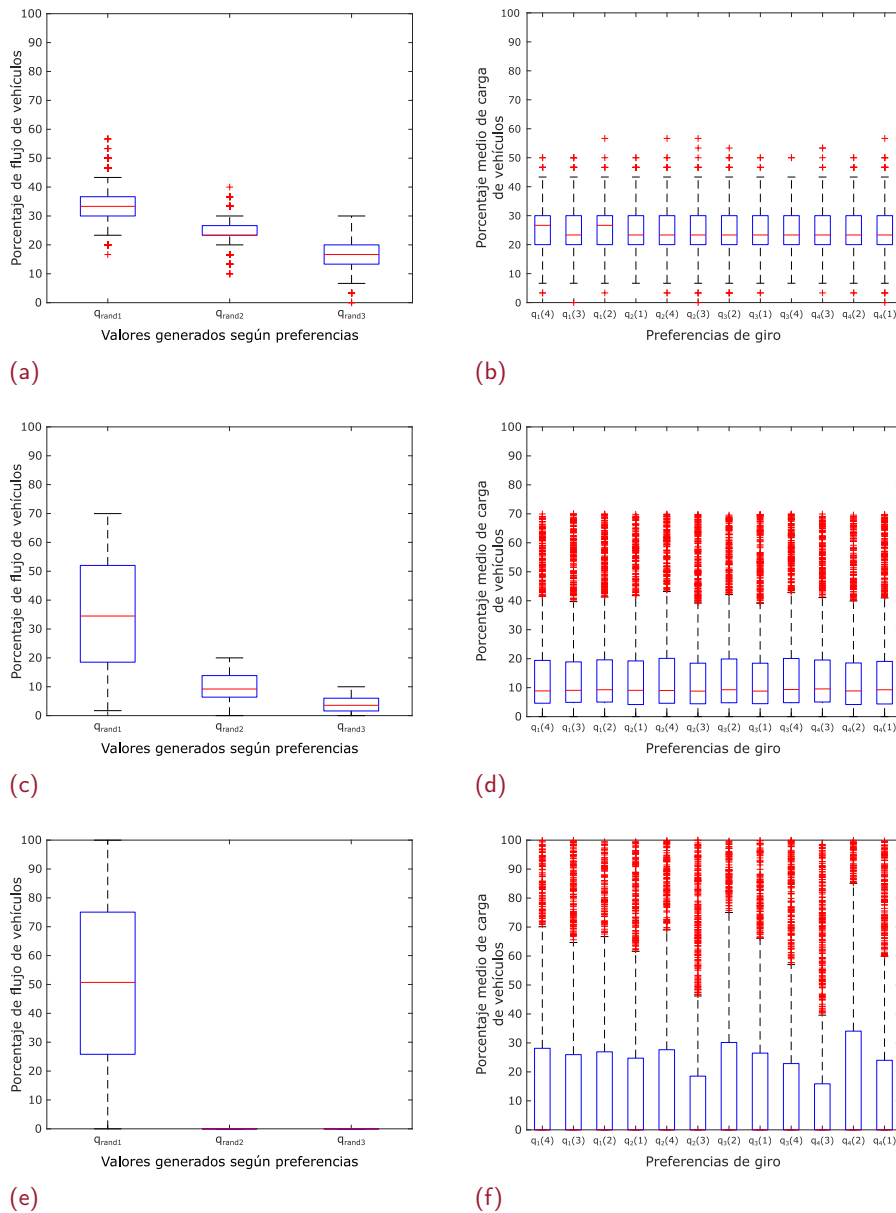


Figura B.5.: Diagramas de caja donde se analizan los conjuntos de flujos de entrada generados aleatoriamente para los experimentos Bx. (a) (c) (d) Según las preferencias de cada conjunto se generan grupos de tres valores. (b) (d) (f) Estos tres valores se distribuyen de forma aleatoria entre las preferencias de una rama.

Analizando los valores obtenidos para el conjunto de datos del experimento B1 (figura B.5a) se puede observar como, al tratarse de valores aleatorios que se van generando sobre la diferencia del total de la capacidad máxima de cada brazo (representado por el valor 1), las dimensiones de sus cajas son similares con medias decrecientes según se van asignando valores. Para el experimento B2 (figura B.5c), al generarse los números aleatorios de forma aislada, cada valor no es influenciado por el valor anteriormente calculado. Por último, para el experimento B3 (figura B.5e) se puede ver como todos los valores obtenidos se centran en el primer número aleatorio calculado.

Ya que en la definición de estos conjuntos de datos se buscaba que no hubiera una opción de giro predeterminada, para comprobar si el barajado de los valores calculados ha tenido efecto, debemos observar las figuras B.5b, B.5d y B.5f, que representan la media de los porcentajes de flujos asignados para cada tupla de opción y giro, y brazo de entrada. Tal y como se aprecia en las tres gráficas, se obtienen valores similares para cada valor del vector  $Q$ , por lo que se cumplen también las preferencias de giro que se habían marcado como restricción. Estas tres gráficas nos dan una visión de la dispersión de los valores, donde cuando plantemos proporciones de flujos más extremas, aparecen en la gráfica un mayor número de valores atípicos.



# Bibliografía

- [1]T. Reed y J. Kidd, «Global traffic scorecard», INRIX RESEARCH, inf. téc., 2019.
- [2]I. Leonard J.D., B. Ramanathan y W. Recker, «A real-time information processing algorithm for the evaluation and implementation of ATMS strategies», en *Intelligent Vehicles '92 Symposium., Proceedings of the*, jun. de 1992, págs. 225-229.
- [3]S. Singh, «Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey», National Highway Traffic Safety Administration, Washington, DC, inf. téc. DOT HS 812 506, mar. de 2018.
- [4]M. Simon, T. Hermitte e Y. Page, «Intersection road accident causation: A European view», en *21st International Technical Conference on the Enhanced Safety of Vehicles*, 2009, págs. 1-10.
- [5]J. Rios-Torres y A. A. Malikopoulos, «A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps», *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, n.º 5, págs. 1066-1077, mayo de 2017.
- [6]L. Chen y C. Englund, «Cooperative Intersection Management: A Survey», *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, n.º 2, págs. 570-586, feb. de 2016.
- [7]B. Chen y H. H. Cheng, «A Review of the Applications of Agent Technology in Traffic and Transportation Systems», *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, n.º 2, págs. 485-497, jun. de 2010.
- [8]«IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture», *IEEE Std 1609.0-2019*, feb. de 2019.
- [9]E. Hall, «V2X Communication in 3GPP», Qualcomm Incorporated, San Francisco, inf. téc. S1-144374, nov. de 2014.
- [10]F. Yang, J. Li, T. Lei y S. Wang, «Architecture and key technologies for Internet of Vehicles: a survey», *Journal of Communications and Information Networks*, vol. 2, n.º 2, págs. 1-17, jun. de 2017.
- [11]T. S. J. Darwish, K. Abu Bakar y K. Haseeb, «Reliable Intersection-Based Traffic Aware Routing Protocol for Urban Areas Vehicular Ad Hoc Networks», *IEEE Intelligent Transportation Systems Magazine*, vol. 10, n.º 1, págs. 60-73, Spring de 2018.

- [12]R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone y H. Wymeersch, «Coordination of Cooperative Autonomous Vehicles: Toward safer and more efficient road transportation», *IEEE Signal Processing Magazine*, vol. 33, n.º 6, págs. 74-84, nov. de 2016.
- [13]G. Rodrigues de Campos, P. Falcone, R. Hult, H. Wymeersch y J. Sjöberg, «Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics», *IEEE Intelligent Transportation Systems Magazine*, vol. 9, n.º 1, págs. 8-21, Spring de 2017.
- [14]Y. Bichiou y H. A. Rakha, «Developing an Optimal Intersection Control System for Automated Connected Vehicles», *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, n.º 5, págs. 1908-1916, mayo de 2019.
- [15]T. U. Kuzubaş, I. Ömercikoğlu y B. Saltoğlu, «Network centrality measures and systemic risk: An application to the Turkish financial crisis», *Physica A: Statistical Mechanics and its Applications*, vol. 405, págs. 203-215, 2014.
- [16]A. B. M. Nasiruzzaman, H. R. Pota y M. A. Mahmud, «Application of centrality measures of complex network framework in power grid», en *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, nov. de 2011, págs. 4660-4665.
- [17]M. Shulman y R. K. Deering, «Second Annual Report of the Crash Avoidance Metrics Partnership», National Highway Traffic Safety Administration, Washington, DC, inf. téc. DOT HS 809 663, ene. de 2003.
- [18]R. Azimi, G. Bhatia, R. Rajkumar y P. Mudalige, «Ballroom Intersection Protocol: Synchronous Autonomous Driving at Intersections», en *2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications*, ago. de 2015, págs. 167-175.
- [19]M. Bashiri y C. H. Fleming, «A platoon-based intersection management system for autonomous vehicles», en *2017 IEEE Intelligent Vehicles Symposium (IV)*, jun. de 2017, págs. 667-672.
- [20]J. Lioris, R. Pedarsani, F. Y. Tascikaraoglu y P. Varaiya, «Platoons of connected vehicles can double throughput in urban roads», *Transportation Research Part C: Emerging Technologies*, vol. 77, págs. 292-305, 2017.
- [21]K. M. Yousef, M. N. Al-Karaki y A. M. Shatnawi, «Intelligent traffic light flow control system using wireless sensors networks.», *Journal of Information Science and Engineering*, vol. 26, n.º 3, págs. 753-768, 2010.
- [22]S. Faye y C. Chaudet, «Characterizing the topology of an urban wireless sensor network for road traffic management», *IEEE Transactions on Vehicular Technology*, vol. 65, n.º 7, págs. 5720-5725, 2016.
- [23]R. V. Kulkarni y G. K. Venayagamoorthy, «Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey», *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, n.º 2, págs. 262-267, 2011, ID: 1.
- [24]Z. Shan y Q. Zhu, «Camera location for real-time traffic state estimation in urban road network using big GPS data», *Neurocomputing*, vol. 169, págs. 134-143, 2015, Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis.

- [25] M. Ahmane, A. Abbas-Turki, F. Perronnet, J. Wu, A. E. Moudni, J. Buisson y R. Zeo, «Modeling and controlling an isolated urban intersection based on cooperative vehicles», *Transportation Research Part C: Emerging Technologies*, vol. 28, págs. 44-62, 2013, Euro Transportation: selected paper from the EWGT Meeting, Padova, September 2009.
- [26] A. Mirheli, M. Tajalli, L. Hajibabai y A. Hajbabaie, «A consensus-based distributed trajectory control in a signal-free intersection», *Transportation Research Part C: Emerging Technologies*, vol. 100, págs. 161-176, 2019.
- [27] R. Azimi, «Co-operative Driving at Intersections using Vehicular Networks and Vehicle-Resident Sensing», Tesis doct., Carnegie Mellon University, jul. de 2015.
- [28] S. Maerivoet y B. D. Moor, «Cellular automata models of road traffic», *Physics Reports*, vol. 419, n.º 1, págs. 1-64, 2005, ID: 271542.
- [29] R. Akcelik, M. Besley y E. Chung, «An evaluation of SCATS Master Isolated control», en *Proceedings of the 19th ARRB Transport Research Conference (Transport 98)*, Citeseer, 1998, págs. 1-24.
- [30] D. I. Robertson y R. D. Bretherton, «Optimizing networks of traffic signals in real time-the SCOOT method», *IEEE Transactions on Vehicular Technology*, vol. 40, n.º 1, págs. 11-15, feb. de 1991.
- [31] R. Hult, G. R. Campos, P. Falcone y H. Wymeersch, «An approximate solution to the optimal coordination problem for autonomous vehicles at intersections», en *2015 American Control Conference (ACC)*, jul. de 2015, págs. 763-768.
- [32] M. Ferreira, R. Fernandes, H. Conceição, W. Viriyasitavat y O. K. Tonguz, «Self-organized Traffic Control», en *Proceedings of the Seventh ACM International Workshop on Vehicular InterNetworking*, ép. VANET '10, Chicago, Illinois, USA: ACM, 2010, págs. 85-90.
- [33] W. Lin, H. Liu y H. K. Lo, «Guest Editorial: Big Data for Driver, Vehicle, and System Control in ITS», *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, n.º 6, págs. 1663-1665, jun. de 2016.
- [34] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen y X. Shen, «Internet of vehicles in big data era», *IEEE/CAA Journal of Automatica Sinica*, vol. 5, n.º 1, págs. 19-35, ene. de 2018.
- [35] E. Castillo, M. Nogal, A. Rivas y S. Sánchez-Cambronero, «Observability of traffic networks. Optimal location of counting and scanning devices», *Transportmetrica B: Transport Dynamics*, vol. 1, n.º 1, págs. 68-102, 2013.
- [36] S.-R. Hu, S. Peeta y C.-H. Chu, «Identification of vehicle sensor locations for link-based network traffic applications», *Transportation Research Part B: Methodological*, vol. 43, n.º 8, págs. 873-894, 2009.
- [37] R. V. Kulkarni y G. K. Venayagamoorthy, «Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey», *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, n.º 2, págs. 262-267, mar. de 2011.
- [38] L. Cruz-Piris, «Mejora en la eficiencia y escalabilidad en sistema de gestión de tráfico basados en simulaciones», Tesis de mtría., Universidad de Alcalá, EPS, Alcalá de Henares, Madrid, 2015.
- [39] S. Winter, «Modeling Costs of Turns in Route Planning», *GeoInformatica*, vol. 6, n.º 4, págs. 345-361, dic. de 2002.



- [40]A. Kazerani y S. Winter, «Can betweenness centrality explain traffic flow», en *12th AGILE International Conference on Geographic Information Science*, 2009, págs. 1-9.
- [41]U. Brandes, «On Variants of Shortest-Path Betweenness Centrality and their Generic Computation», *SOCIAL NETWORKS*, vol. 30, n.º 2, 2008.
- [42]D. L. Moura, R. S. Cabral, T. Sales y A. L. Aquino, «An evolutionary algorithm for roadside unit deployment with betweenness centrality preprocessing», *Future Generation Computer Systems*, vol. 88, págs. 776-784, 2018.
- [43]P. Wang, T. Hunter, A. M. Bayen, K. Schechtner y M. C. González, «Understanding Road Usage Patterns in Urban Areas», *Scientific Reports*, vol. 2, dic. de 2012.
- [44]U. Brandes, «On variants of shortest-path betweenness centrality and their generic computation», *Social Networks*, vol. 30, n.º 2, págs. 136-145, 2008.
- [45]NetworkX, *edge\_betweenness\_centrality*, [https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.centrality.edge\\_betweenness\\_centrality.html](https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.centrality.edge_betweenness_centrality.html), Online; accedido el 30 de agosto de 2019.
- [46]DLR - Institute of Transportation Systems. (2018). «TAPAS Cologne» Scenario, dirección: <https://sumo.dlr.de/wiki/Data/Scenarios/TAPASCologne> (visitado 4 de jul. de 2019).
- [47]P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner y E. Wießner, «Microscopic Traffic Simulation using SUMO», en *The 21st IEEE International Conference on Intelligent Transportation Systems*, IEEE, 2018.
- [48]D. Sastre, J. Torres y J. M. Menendez, «Sistemas de adquisición de información de tráfico: Estado actual y futuro», Plataforma Tecnológica de la Carretera (PTC), inf. téc., ene. de 2011.
- [49]Z. Shan y Q. Zhu, «Camera location for real-time traffic state estimation in urban road network using big GPS data», *Neurocomputing*, vol. 169, págs. 134-143, 2015, Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis.
- [50]L. Cruz-Piris, D. Rivera, S. Fernandez e I. Marsa-Maestre, «Optimized Sensor Network and Multi-Agent Decision Support for Smart Traffic Light Management», *Sensors*, vol. 18, n.º 2, 2018.
- [51]A. L. C. Bazzan y F. Klügl, «A review on agent-based technology for traffic and transportation», *The Knowledge Engineering Review*, vol. 29, n.º 3, págs. 375-403, 2014.
- [52]J. L. Adler y V. J. Blue, «A cooperative multi-agent transportation management and route guidance system», *Transportation Research Part C: Emerging Technologies*, vol. 10, n.º 5, págs. 433-454, 2002.
- [53]S. Maerivoet y B. D. Moor, «Cellular automata models of road traffic», *Physics Reports*, vol. 419, n.º 1, págs. 1-64, 2005.
- [54]R. Azimi, «Co-operative Driving at Intersections using Vehicular Networks and Vehicle-Resident Sensing», Tesis doct., jul. de 2015.
- [55]S. Maerivoet y B. D. Moor, «Cellular automata models of road traffic», *Physics Reports*, vol. 419, n.º 1, págs. 1-64, 2005, ID: 271542.

- [56]H.-T. Zhao, X.-R. Liu, X.-X. Chen y J.-C. Lu, «Cellular automata model for traffic flow at intersections in internet of vehicles», *Physica A: Statistical Mechanics and its Applications*, vol. 494, págs. 40-51, 2018.
- [57]Q. Lu y K.-D. Kim, «A Genetic Algorithm Approach for Expedited Crossing of Emergency Vehicles in Connected and Autonomous Intersection Traffic», *Journal of Advanced Transportation*, oct. de 2017.
- [58]J. Lee y B. Park, «Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment», *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, n.º 1, págs. 81-90, 2012.
- [59]W. Ma, Y. Liu, J. Zhao y N. Wu, «Increasing the capacity of signalized intersections with left-turn waiting areas», *Transportation Research Part A: Policy and Practice*, vol. 105, págs. 181-196, 2017.
- [60]A. Smiley y R. E. Dewar, «The Institute of Transportation Engineers, Traffic Engineering Handbook, 7th Edition», en *Traffic Engineering Handbook*. Wiley-Blackwell, 2016, cap. Road Users, págs. 51-107.
- [61]J. R. Riegner, «The Institute of Transportation Engineers, Traffic Engineering Handbook, 7th Edition», en *Traffic Engineering Handbook*. Wiley-Blackwell, 2016, cap. Design and Operation of Complete Streets and Intersections, págs. 367-398.
- [62]A. Pande y B. Wolshon, «The Institute of Transportation Engineers, Traffic Engineering Handbook, 7th Edition», en *Traffic Engineering Handbook*. Wiley-Blackwell, 2016, cap. Design and Control for Interrupted Traffic Flow through Intersections, págs. 321-366.
- [63]L. Cruz-Piris, «Estrategias de control de tráfico de vehículos en un entorno urbano», en *Séptimas Jornadas de Jóvenes Investigadores de la Universidad de Alcalá (Ciencias e Ingenierías)*, A. G. Orteg, G. R. Magán, P. R. Benito, F. P. Vives, C. T. Martínez y V. T. Magro, eds., Alcalá de Henares: Universidad de Alcalá. Servicio de publicaciones, 2019, págs. 17-25.
- [64]«Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles», Society of Automotive Engineers (SAE), inf. téc., jun. de 2018.
- [65]P. Bansal y K. M. Kockelman, «Forecasting Americans' long-term adoption of connected and autonomous vehicle technologies», *Transportation Research Part A: Policy and Practice*, vol. 95, págs. 49-63, 2017.
- [66]Y. Xuan, C. F. Daganzo y M. J. Cassidy, «Increasing the capacity of signalized intersections with separate left turn phases», *Transportation Research Part B: Methodological*, vol. 45, n.º 5, págs. 769-781, 2011.
- [67]L. Cruz-Piris, M. A. Lopez-Carmona e I. Marsa-Maestre, «Automated Optimization of Intersections Using a Genetic Algorithm», *IEEE Access*, vol. 7, págs. 15 452-15 468, 2019.
- [68]L. Cruz-Piris, I. Marsa-Maestre y M. A. Lopez-Carmona, «A Variable-Length Chromosome Genetic Algorithm to Solve a Road Traffic Coordination Multipath Problem», *IEEE Access*, vol. 7, págs. 111 968-111 981, 2019.
- [69]C.-Y. Lee y E. K. Antonsson, «Variable Length Genomes for Evolutionary Algorithms», en *Proceedings of the 2Nd Annual Conference on Genetic and Evolutionary Computation*, ép. GECCO'00, Las Vegas, Nevada: Morgan Kaufmann Publishers Inc., 2000, págs. 806-806.

- [70]A. H. Brie y P. Morignot, «Genetic Planning Using Variable Length Chromosomes», en *Proceedings of the Fifteenth International Conference on International Conference on Automated Planning and Scheduling*, ép. ICAPS'05, Monterey, California, USA: AAAI Press, 2005, págs. 320-329.
- [71]Z. Qiongbing y D. Lixin, «A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems», *Expert Systems with Applications*, vol. 60, págs. 183-189, 2016.
- [72]The MathWorks Inc., *Selection Options - SELECTIONREMAINDER*, 2015.
- [73]A. J. Umbarkar y P. Sheth, «Crossover Operators in Genetic Algorithms:A Review», en *SOCO 2015*, 2015.
- [74]U. Derigs y R. Kaiser, «Applying the attribute based hill climber heuristic to the vehicle routing problem», *European Journal of Operational Research*, vol. 177, n.º 2, págs. 719-732, 2007.
- [75]F. Viveros-Jiménez, J. A. León-Borges y N. Cruz-Cortés, «An adaptive single-point algorithm for global numerical optimization», *Expert Systems with Applications*, vol. 41, n.º 3, págs. 877-885, 2014, Methods and Applications of Artificial and Computational Intelligence.
- [76]L. Cruz-Piris, D. Rivera, I. Marsa-Maestre, E. de la Hoz y S. Fernandez, «Intelligent Traffic Light Management using Multi-Behavioral Agents», en *XIII Jornadas de Ingeniería telemática (JITEL 2017)*. Libro de actas, Editorial Universitat Politècnica de València, 2018, págs. 110-117.
- [77]M. J. Lighthill y G. B. Whitham, «On kinematic waves II. A theory of traffic flow on long crowded roads», *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, n.º 1178, págs. 317-345, 1955. eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1955.0089>.
- [78]S. P. Hoogendoorn y P. H. L. Bovy, «State-of-the-art of vehicular traffic flow modelling», *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 215, n.º 4, págs. 283-303, 2001. eprint: <https://doi.org/10.1177/095965180121500402>.
- [79]D. Chowdhury, L. Santen y A. Schadschneider, «Statistical physics of vehicular traffic and some related systems», *Physics Reports*, vol. 329, n.º 4, págs. 199-329, 2000.
- [80]L. S. Passos, R. J. Rossetti y Z. Kokkinoginis, «Towards the next-generation traffic simulation tools: a first appraisal», en *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on*, IEEE, 2011, págs. 1-6.
- [81]M. Fellendorf, «VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority», en *64th Institute of Transportation Engineers Annual Meeting*, 1994, págs. 1-9.
- [82]R. Bertini, R. Lindgren y S. Tantiyanugulchai, «Application of PARAMICS simulation at a diamond interchange», *Portland State University, Transportation Research Group, Research Report PSU-CE-TRG-02-02*, 2002.
- [83]J. Barceló, «Microscopic traffic simulation: A tool for the analysis and assessment of its systems», en *Highway Capacity Committee, Half Year Meeting.*, 2001.

- [84]M. Ben-Akiva, M. Cortes, A. Davol, H. Koutsopoulos y T. Toledo, «Mitsimlab: Enhancements and applications for urban networks», en *9th World Conference on Transportation Research (WCTR)*, Seoul, Korea, 2001.
- [85]*Multi-Agent Transport Simulation (MATSIM)*, <https://www.matsim.org/>, Recurso web; accedido 17 de julio de 2019.
- [86]A. Wegener, M. Piórkowski Michałand Raya, H. Hellbrück, S. Fischer y J.-P. Hubaux, «TraCI: An Interface for Coupling Road Traffic and Network Simulators», en *Proceedings of the 11th Communications and Networking Simulation Symposium*, ép. CNS '08, Ottawa, Canada: ACM, 2008, págs. 155-163.
- [87]M. Batty, H. Couclelis y M. Eichen, «Urban systems as cellular automata», *Environment and Planning B: Planning and Design*, vol. 24, n.º 2, págs. 159-164, 1997.
- [88]S. Uppoor, O. Trullols-Cruces, M. Fiore y J. M. Barcelo-Ordinas, «Generation and Analysis of a Large-Scale Urban Vehicular Mobility Dataset», *IEEE Transactions on Mobile Computing*, vol. 13, n.º 5, págs. 1061-1075, mayo de 2014.
- [89]C. Gawron, «An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model», *International Journal of Modern Physics C*, vol. 9, n.º 03, págs. 393-407, 1998.
- [90]S. El-Tantawy, B. Abdulhai y H. Abdelgawad, «Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto», *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, n.º 3, págs. 1140-1150, sep. de 2013.
- [91]*Simulation of Urban MObility Wiki: Simulation/Why Vehicles are teleporting*, [https://sumo.dlr.de/docs/Simulation/Why\\_Vehicles\\_are\\_teleporting.html](https://sumo.dlr.de/docs/Simulation/Why_Vehicles_are_teleporting.html), Recurso web; accedido 15 de septiembre de 2019.



# Acrónimos

<b>ARLM</b>	<i>Arterial Road Location Method</i> Método de ubicación en carreteras arteriales
<b>ATMS</b>	<i>Advanced Traffic Management Systems</i> Sistemas de gestión del tráfico avanzados
<b>BRIP</b>	<i>Ballroom Intersection Protocol</i> Protocolo del “salón de baile” en intersecciones
<b>CA</b>	<i>Cellular Automata</i> Autómata celular
<b>CAV</b>	<i>Connected and Autonomous Vehicle</i> Vehículos conectados y autónomos
<b>CIM</b>	<i>Cooperative Intersection Management</i> Gestión de intersección cooperativa
<b>C-ITS</b>	<i>Cooperative Intelligent Transportation Systems</i> Sistemas de transporte inteligentes cooperativos
<b>DSRC</b>	<i>Dedicated Short-Range Communications</i> Comunicaciones dedicadas de corto alcance
<b>GA</b>	<i>Genetic Algorithm</i> Algoritmo genético
<b>GPS</b>	<i>Global Positioning System</i> Sistema de posicionamiento global
<b>ICT</b>	<i>Information and Communications Technology</i> Tecnologías de la información y la comunicación
<b>IoT</b>	<i>Internet of Things</i> Internet de las cosas
<b>IoV</b>	<i>Internet of Vehicles</i> Internet de los vehículos
<b>ITS</b>	<i>Intelligent Transportation Systems</i> Sistemas inteligentes de transporte
<b>MAS</b>	<i>Multi-Agent Systems</i> Sistemas multiagente

<b>MATSIM</b>	<i>Multi-Agent Transport Simulator</i> Simulador de transporte multiagente
<b>NSLP</b>	<i>Network Sensor Location Problem</i> Problema de ubicación de sensores en red
<b>OCF</b>	<i>Optimized Crossing Patterns</i> Patrones de cruce optimizados
<b>OCF-MP</b>	<i>Optimized Crossing Patterns with Multiple Paths</i> Patrones de cruce optimizados con caminos múltiples
<b>OCF-MP-VL</b>	<i>Optimized Crossing Patterns with Multiple Paths and Variable Length</i> Patrones de cruce optimizados con caminos múltiples y longitud variable
<b>ODbL</b>	<i>Open Database License</i> Licencia Abierta de Bases de Datos
<b>OSM</b>	OpenStreetMap
<b>PSO</b>	<i>Particle Swarm Optimization</i> Optimización por enjambre de partículas
<b>RRLM</b>	<i>Random Road Location Method</i> Método de ubicación en carreteras aleatorias
<b>RTAR</b>	<i>Reliable Traffic Aware Routing</i> Encaminamiento de confianza basado en tráfico
<b>SAE</b>	<i>Sum of Absolute Errors</i> Suma de errores absolutos
<b>SCATS</b>	<i>Sydney Coordinated Adaptive Traffic System</i> Sistema de tráfico adaptativo coordinado de Sydney
<b>SCOOT</b>	<i>Split, Cycle and Offset Optimization Technique</i> Técnica de optimización de división, ciclo y ajuste
<b>SIAP</b>	<i>Synchronized Intersection Arrival Patterns</i> Patrones sincronizados de llegada a la intersección
<b>SUMO</b>	<i>Simulation of Urban Mobility</i> Simulador de movilidad urbana
<b>TAPAS</b>	<i>Travel and Activity Patterns Simulation</i> Simulación de patrones de viajes y actividades
<b>TCA</b>	<i>Traffic Cellular Automata</i> Autómata celular de tráfico
<b>TCP</b>	<i>Transmission Control Protocol</i> Protocolo de control de transmisión
<b>TIC</b>	Tecnologías de la Información y las Comunicaciones
<b>TRACI</b>	<i>Traffic Control Interface</i> Interfaz de control de tráfico
<b>UTC</b>	<i>Urban Traffic Control</i> Control del tráfico urbano
<b>V2I</b>	<i>Vehicle-to-infrastructure</i> Vehículo a infraestructura

<b>V2P</b>	<i>Vehicle-to-persons</i> Vehículo a persona
<b>V2V</b>	<i>Vehicle-to-vehicle</i> Vehículo a vehículo
<b>V2X</b>	<i>Vehicle-to-everything</i> Vehículo a todo
<b>VANET</b>	<i>Vehicular Ad-Hoc Networks</i> Redes vehiculares ad-hoc
<b>VTL</b>	<i>Virtual Traffic Lights</i> Semáforos virtuales
<b>WAVE</b>	<i>Wireless Access in Vehicular Environments</i> Acceso inalámbrico en entornos vehiculares
<b>WSN</b>	<i>Wireless Sensor Networks</i> Sensor de redes inalámbricas







# Contribución a las estrategias de optimización multiobjetivo para la coordinación de vehículos en intersecciones urbanas

Esta tesis aborda el problema de la coordinación de vehículos a su paso por cruces de carreteras o intersecciones en un entorno urbano. El estudio de este problema comienza por el análisis de las posibles fuentes de información existentes para el caso particular de un escenario de tráfico urbano, teniendo en cuenta las tecnologías para su recolección. En este ámbito se realiza la primera de las contribuciones, proponiendo una metodología basada en la centralidad de la red de carretera para obtener las mejores ubicaciones para la instalación de sensores. Los dispositivos instalados en esas posiciones proporcionarán la información más relevante del escenario de tráfico. Tras conocer las fuentes de datos disponibles, la hipótesis central de esta tesis es que es posible mejorar los procesos de optimización utilizados para la gestión actual de las intersecciones. Este problema se aborda comenzando por definir un modelo capaz de etiquetar de forma unívoca todos los elementos que componen una intersección. El modelo será la base de las propuestas de optimización posteriores.

En relación con la coordinación de vehículos, en esta tesis se estudian dos posibles escenarios, dependiendo de si la coordinación de vehículos se realiza a través de elementos externos como los semáforos o si, por el contrario, esta coordinación se produce en un escenario de vehículos autónomos, donde cada uno de ellos es capaz de pasar por la intersección, en condiciones seguras y sin detenerse. En el primer caso, la propuesta realizada consiste en el desarrollo de un sistema multiagente, desplegado sobre un escenario simulado generado con tráfico real, capaz de gestionar las fases de los semáforos. Los agentes controladores de cada intersección tienen la capacidad de variar su comportamiento en base a los indicadores de tráfico que reciben. En el segundo de los casos, la investigación estudia el comportamiento de los vehículos en el interior de las intersecciones y los posibles métodos para la generación de patrones de llegada de vehículos. La obtención de estos patrones permite el cruce en condiciones seguras, sin que los vehículos deban detenerse, y cumpliendo las preferencias del sistema. Para lograr este objetivo se comenzó por diseñar un proceso de optimización basado en un algoritmo genético con caminos fijos entre los puntos de entrada y salida de la intersección, centrado en conseguir el mayor rendimiento posible para el caso en que todos los flujos de entrada fueran iguales. Posteriormente esta solución es extendida, permitiendo cualquier origen y destino posible dentro de la intersección, y generalizando el objetivo de la optimización para lograr los flujos de entrada deseados, sin la restricción de que sean iguales. Debido al aumento exponencial de la complejidad del problema en esta última solución, se propone un algoritmo genético de longitud variable de cromosomas, adaptado a este problema.

Para validar las propuestas realizadas se han utilizado simuladores y escenarios de tráfico realistas, que permitan probar de forma intensiva cada una de ellas. Además del uso de simuladores comerciales, ha sido necesario implementar un simulador de intersecciones propio, el cual fuera capaz de reproducir sus particularidades. En cada caso, los resultados obtenidos utilizando las propuestas realizadas han sido comparados frente a otras soluciones ampliamente utilizados. Los buenos resultados obtenidos con los métodos propuestos en esta tesis permiten confirmar las hipótesis planteadas al inicio de la investigación.

Las imágenes de fondo de la portada y la contraportada corresponden a la posición y diagramas de iluminación para gestionar los flujos de tráfico del área entre la Calle Alcalá y la Avenida del Conde de Peñalver (actual Gran Vía) de la ciudad de Madrid. Se trata del primer sistema de señales luminosas para la gestión del tráfico en España.

Fuente: La ordenación del tráfico en Madrid. *Revista España Automóvil y Aeronáutica*, 1926, p 121-123. (Hemeroteca Municipal)



Universidad de Alcalá