



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Optimización de tiempos semaforizados en un cruce urbano con
afluencia media.*

Grado en Ingeniería Mecánica

ALUMNO: Juan Díaz del Río Triguero

DIRECTORES: Débora Cores Carrera

CURSO ACADÉMICO: 2025-2026

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Optimización de tiempos semaforizados en un cruce urbano con
afluencia media.*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

UniversidadeVigo

RESUMEN

El presente Trabajo de Fin de Grado aborda la gestión del tráfico urbano, un desafío histórico que ha evolucionado desde las "walking cities" hasta la actual congestión de las metrópolis motorizadas. El objetivo principal es el desarrollo de un modelo de optimización lineal diseñado para regular los tiempos de luz verde en una intersección de cuatro fases que integra movimientos rectos y giros a la izquierda.

Se plantea un modelo matemático basado en técnicas de programación lineal con el objetivo de determinar una distribución eficiente de los tiempos de luz verde en un cruce urbano con afluencia media. El sistema está compuesto por cuatro fases, las cuales permiten tanto movimientos rectos como giros a la izquierda, tomando como variables de decisión los tiempos de luz verde asignados a cada fase del ciclo.

El modelo incorpora restricciones relacionadas con el tiempo total del ciclo, los tiempos mínimos y máximos de verde y las capacidades de los distintos movimientos del cruce. Para su resolución se utilizará el software MATLAB, empleando funciones como *linprog* e *intlinprog*, que permite implementar el modelo y obtener la solución óptima del problema.

Para ello, se analizarán distintos escenarios de tráfico con el fin de evaluar el comportamiento del modelo y estudiar cómo varían los resultados en función de los parámetros considerados.

PALABRAS CLAVE

Optimización entera, Programación lineal, Tráfico urbano, Tiempos semaforizados, MATLAB.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a toda mi familia, en especial a mis padres y a mi abuelo Ángel, por su apoyo incondicional durante esta etapa de formación tan importante en mi vida, la cual, sin su ejemplo de constancia y afán por hacer las cosas bien, no habría sido posible alcanzar donde hoy estoy.

A mi tutora, Débora Cores, por su continuo apoyo y ayuda durante la realización de este trabajo.

A los amigos que he hecho durante esta etapa y con los que tanto hemos aguantado y luchado juntos sin descanso, así como a mis amigos que, sin compartir esta profesión, han estado a mi lado y se han preocupado por mí.

A todos aquellos familiares y amigos que, desde pequeño, despertaron en mí el amor por España y por la Armada, y que dieron forma a mi vocación de ser marino de guerra. Hago una mención especial a mi abuelo Ángel; a mis tíos Ángel, Luis y Eugenio; y a mi primo Eugenio. A todos ellos, muchas gracias.

A la Virgen del Carmen, por ser faro en las horas de tempestad y continua ayuda.

CONTENIDO

Contenido	1
Índice de Figuras	3
Índice de Tablas.....	4
1 Introducción y objetivos	5
1.1 Contexto	5
1.2 Motivación y objetivo.	8
1.3 Estructura	8
2 Estado del arte	11
2.1 Modelos matemáticos	11
2.1.1 Definición	11
2.2 Optimización lineal	13
2.2.1 Definición	13
2.3 El método SIMPLEX	14
2.3.1 Origen.	14
2.3.2 Definición.	14
2.4 Programa MATLAB	16
2.4.1 Definición.	16
2.4.2 La optimización lineal con MATLAB.....	17
3 Desarrollo del TFG.....	19
3.1 Planteamiento del problema de forma matemática	19
3.1.1 Variables	19
3.1.2 Restricciones.....	22
3.1.3 Modelos propuestos	23
3.1.4 Consideraciones	28
3.1.5 Sistema matricial y vectorial a introducir en Matlab	28
3.1.6 Funcionamiento del código.....	30
3.1.7 Datos del problema.	31
4 Resultados obtenidos	35
4.1 Resultados del Escenario 1	35
4.2 Resultados del Escenario 2	36
4.3 Resultados del Escenario 3	38
4.4 Resultados del Escenario 4	39
4.5 Resultados del Escenario 5	40

4.6 Resultados del Escenario 6	41
5 Conclusiones y líneas futuras	45
5.1 Conclusiones	45
5.2 Líneas futuras	45
6 Bibliografía.....	47
Anexo I: Implicaciones Sociales, Económicas y Ambientales	49
Anexo II: Reflexiones Éticas y Sociales	50
Anexo III: Código del Programa	51
Anexo IV: Declaración de uso de Inteligencia Artificial	55

ÍNDICE DE FIGURAS

Figura 1-1: Ejemplo para resolver mediante el uso de programación lineal. Figura extraída de [5].	6
Figura 1-2: Uso provisional de la Plaza Mayor como estacionamiento entre 1961 y 1968. Figura extraída de [6].	7
Figura 1-3: Caos en el centro de Madrid durante el apagón de 2025. Fuente extraída de [10].	8
Figura 2-1 Adaptación del modelo matemático a la vida real. Figura extraída de [11].	12
Figura 2-2 Ejemplo de restricciones en frontera y región factible. Figura extraída de [22].	15
Figura 2-3 Poliedro asociado al método Simplex. Figura extraída de [25].	16
Figura 2-4 Logotipo de Matlab. Figura extraída de [26].	16
Figura 3-1 Representación giros a la izquierda en las diferentes vías. Figuras generadas con inteligencia artificial [31].	20
Figura 4-1 Representación resultados del Escenario 1.	36
Figura 4-2 Resultados obtenidos del Escenario 2.	37
Figura 4-3 Resultados obtenidos del Escenario 3.	38
Figura 4-4 Resultados obtenidos del Escenario 4.	40
Figura 4-5 Resultados obtenidos del Escenario 5.	41
Figura 4-6 Resultados obtenidos del Escenario 6.	42
Figura 4-7 Comparativa de los valores X en función de los tiempos muertos aplicados a los distintos escenarios.	43

ÍNDICE DE TABLAS

Tabla 2-1 Tipos de modelos matemáticos [14]	12
Tabla 2-2 Algunos de los comandos MATLAB aplicados a programación lineal. Datos extraídos de [28]	17
Tabla 3-1 Ventajas y desventajas de los modelos planteados	28
Tabla 3-2 Valores de ciclos en función de los flujos y capacidades de un cruce con afluencia media.	28
Tabla 3-3 Datos de partida del Escenario 1	31
Tabla 3-4 Datos de partida del Escenario 2	32
Tabla 3-5 Datos de partida del Escenario 3	32
Tabla 3-6 Datos de partida del Escenario 4	33
Tabla 3-7 Datos de partida del Escenario 5	33
Tabla 3-8 Datos de partida del Escenario 6	34
Tabla 4-1 Resultados obtenidos del Escenario 1	36
Tabla 4-2 Resultados obtenidos del Escenario 2	37
Tabla 4-3 Resultados obtenidos del Escenario 3	38
Tabla 4-4 Resultados obtenidos del Escenario 4	39
Tabla 4-5 Resultados obtenidos del Escenario 5	41
Tabla 4-6 Resultados obtenidos del Escenario 6	42

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto

Durante el último siglo, la población de las ciudades y con ello el número de vehículos particulares ha ido incrementándose de forma exponencial, dejando claro que la gestión del tráfico provocado por éstos resulta ser un problema que resolver cada día mayor, afectando no solo a las normas de circulación, sino a la planificación, diseño y construcción de ciudades o reformas de estas mismas.

Son varios y distintos los temas sociales relacionados con el problema del tráfico que abordar dependiendo del momento de la historia en que pongamos el foco de atención, como por ejemplo; la preocupación por disponer de calles más anchas en la Europa del siglo XIX; la necesidad de un transporte público y unas normas reguladoras del tráfico en el primer tercio del siglo XX o la preocupación por la contaminación acústica y atmosférica en el presente siglo junto a un crecimiento vertiginoso de la población en lugares como la India o China. [1] [2]

A finales del siglo XIX, España al igual que Europa, aunque de una forma más moderada y en un contexto marcado por la industrialización, la aparición del ferrocarril y el descenso de la mortalidad gracias a los avances en medicina experimentó un éxodo rural hacia las principales ciudades industriales del momento, como Barcelona, Madrid y Bilbao.

En este contexto y poniendo el foco de atención en la ciudad de Madrid, en 1860 se lleva a cabo el Plan de Ensanche de Madrid, elaborado por Carlos María de Castro. Uno de los principales objetivos que se buscaban con este proyecto eran ordenar el crecimiento urbano que Madrid empezaba a experimentar, así como para facilitar la circulación.

Sin embargo, no será hasta 1933, con la celebración del primer Congreso Municipal de Circulación, donde se aborde la circulación urbana desde un punto de vista técnico, motivada a resolver los crecientes problemas de tráfico debidos al aumento del número de vehículos particulares y la “convivencia” con tranvías, peatones y vehículos de tracción animal, tal y como reflejó la diputación de Barcelona en dicho congreso:

“¿Qué habría de acontecer a aquellos estrechos senderos, caminos, callejuelas y luego, la mayoría de las veces, calles de nuestras ciudades, no superiores ni en mucho a los ocho metros de calzada, cuando en un periodo tan corto como el que media entre 1910 y 1930, es decir, en un

espacio de tiempo de veinte años, pasaron a ser cauce del paso diario de 20, 30 o 50.000 automóviles, además de otros tantos vehículos de tracción animal, carretones, bicicletas y tranvías? ¿Cuál tenía que ser el resultado?

Evidentemente, el caos.

Y nos encontramos tan sólo en la génesis del problema. Dentro de diez años ¿qué sucederá?” [3]

Se dejaba así clara la realidad de que Madrid, al igual que el resto de las capitales europeas estaban pasando de ser unas *walking cities* a unas ciudades “motorizadas”, donde el tráfico ya era una realidad que gestionar. La aparición de estos nuevos problemas urbanos puso de manifiesto la necesidad de desarrollar herramientas que permitieran organizar y optimizar sistemas complejos en los que intervienen numerosos recursos y restricciones. Precisamente en este tipo de contextos, donde es necesario encontrar soluciones eficientes bajo determinadas limitaciones, comienzan a cobrar importancia los métodos matemáticos de optimización, entre los que destaca la programación lineal.

Durante la segunda Guerra Mundial, ante la necesidad de optimizar el uso de recursos limitados en operaciones militares surge la programación lineal. Los aliados debían resolver problemas complejos de planificación logística, como la asignación de materias primas, la organización del transporte de suministros o la distribución eficiente del personal, bajo fuertes restricciones materiales y temporales.

Estos problemas podían formularse mediante relaciones lineales entre variables, lo que permitió el desarrollo de modelos matemáticos orientados a maximizar la eficiencia o minimizar costes (Figura 1-1) [4]. En este marco surge la programación lineal.

El avance decisivo se produce en 1947, cuando George Dantzig desarrolla el método SIMPLEX, un algoritmo capaz de resolver de manera eficiente problemas de programación lineal con múltiples variables y restricciones. El método permite identificar la solución óptima explorando los vértices del dominio factible definido por las restricciones.

De este modo, la programación lineal queda formalmente establecida como una técnica matemática rigurosa, estrechamente ligada al método SIMPLEX, y consolidada a partir de las necesidades estratégicas surgidas durante la Segunda Guerra Mundial, sentando las bases del análisis moderno de optimización. [5]

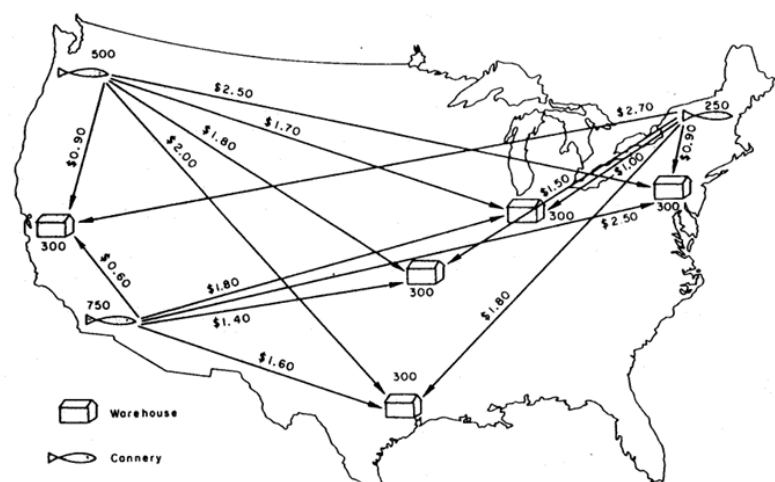


Figure 1-2-I. The Problem: Find a least cost plan of shipping from canneries to warehouses (the costs per case, availabilities and requirements are as indicated).

Figura 1-1: Ejemplo para resolver mediante el uso de programación lineal. Figura extraída de [5]

Tras el final de la segunda Guerra Mundial y a partir de la década de los 50 y debido al crecimiento demográfico en la periferia, se comienzan a construir en las grandes ciudades los primeros pasos elevados, concebidos para eliminar cruces semaforizados, asegurar la continuidad del tráfico y separar flujos en función del destino de los usuarios de la vía. Ciudades como París, Roma o Madrid adoptaron esta medida. Sin embargo, esto no hizo sino aumentar el efecto llamada de más vehículos particulares, provocando la marginación de los peatones (Figura 1-2). [6] [2]



Figura 1-2: Uso provisional de la Plaza Mayor como estacionamiento entre 1961 y 1968. Figura extraída de [6]

A lo largo del siglo XXI, la movilidad urbana ha experimentado cambios significativos derivados del crecimiento de las áreas metropolitanas, el aumento de los desplazamientos diarios y la consolidación del vehículo privado como modo dominante en muchas ciudades, intensificándose los problemas de congestión del tráfico, consumo energético y las emisiones asociadas al transporte, situando la movilidad en el centro del debate técnico y político. [1] [7]

La pandemia de la COVID-19 supuso un punto de inflexión en estos patrones de desplazamiento. Las restricciones a la movilidad y la generalización del teletrabajo provocaron una reducción significativa de los viajes dentro de las urbes, debido a la implementación del teletrabajo y el ocio. Según el informe de la Comisión Europea, el transporte público fue el modo más afectado, con una caída notable de la demanda, mientras que el uso del vehículo privado mostró una recuperación más rápida tras el levantamiento de las restricciones. [8]

En el periodo postpandemia, algunos de estos cambios se han consolidado parcialmente. La mayor flexibilidad laboral ha modificado la frecuencia y distribución temporal de los desplazamientos, sin eliminar completamente la demanda de movilidad urbana. Al mismo tiempo, se ha observado un mayor uso de medios alternativos, como patinetes eléctricos, caminar o la bicicleta, especialmente en trayectos cortos, favorecido por políticas urbanas orientadas a la movilidad sostenible.

Sin embargo, una vez los países fueron levantando las restricciones de movilidad se volvió a sufrir un drástico aumento en el uso del vehículo privado, motivado también por la aún vigente opción de muchas empresas del trabajo desde casa cuando el trabajador lo requiera o decida. Esto por tanto es un claro competidor contra el uso del transporte público y un riesgo impredecible a la hora de determinar qué día y a qué hora se producirán aumentos en la densidad del tráfico y su consiguiente congestión.

El 28 de abril de 2025, aproximadamente a mediodía, España y Portugal sufrieron un apagón que dejó inoperativo los sistemas de transporte tanto aéreos como terrestres (Figura 1-3). Debido al fallo en los semáforos y demás sistemas de regulación del tráfico, en Madrid, se reportaron grandes congestiones y con el consiguiente cierre de accesos de entrada/salida de la capital como el anillo de la M-30. [9]

1.2 Motivación y objetivo.

Como ya se ha expuesto en el punto anterior, estos cambios configuran un nuevo escenario de movilidad urbana a nivel mundial, caracterizado por patrones más diversos y menos predecibles, lo que plantea nuevos retos para el análisis, la gestión del tráfico y la planificación de sistemas de transporte en las ciudades contemporáneas.

Es por esto por lo que este trabajo tiene como objetivo desarrollar un modelo que optimice un tiempo semaforizado mediante el uso del programa MATLAB, aplicado a un cruce urbano, donde se permitan giros a la izquierda y el tráfico se encuentre dirigido de norte a sur y de este a oeste. Se hará uso de la optimización lineal, y se tomarán como variables los tiempos en verde que permitan a los vehículos seguir recto o hacer el giro hacia la izquierda.



Figura 1-3: Caos en el centro de Madrid durante el apagón de 2025. Fuente extraída de [10]

1.3 Estructura

El presente trabajo está dividido en cinco capítulos diferenciados con el propósito de organizar y permitir una mejor comprensión de este, además de uno dedicado a presentar de forma clara los resultados obtenidos. Dichos capítulos se estructuran de la siguiente forma:

- **Capítulo 1:** Introducción y objetivos. En este capítulo se da un breve repaso histórico al surgimiento de la circulación con la llegada de los vehículos a motor particulares a las ciudades, y su evolución a lo largo del siglo XX, poniendo el foco en la ciudad de Madrid y en los principales cambios que marcaron su desarrollo. Asimismo, se analiza la evolución del transporte hasta el presente siglo teniendo en cuenta episodios recientes como la pandemia del COVID-19 o el apagón de 2025 en la Península Ibérica. Termina por tanto este capítulo manifestando la necesidad de mejorar un modelo matemático que sea capaz de adaptarse a un entorno urbano en constante cambio, motivado y respaldado tanto por los acontecimientos históricos como por los recientes sucesos anteriormente mencionados.

- **Capítulo 2:** Estado del arte: En este capítulo se lleva a cabo un análisis del modelo de optimización lineal, de los mecanismos de la programación lineal, así como una introducción al software Matlab.
- **Capítulo 3:** Desarrollo del TFG. En este capítulo se enuncia el problema y la situación de este, planteándose a continuación de manera matemática, exponiéndose el método de resolución con MATLAB y obteniéndose finalmente el resultado.
- **Capítulo 4:** En este capítulo se exponen e interpretan los resultados obtenidos a partir del modelo matemático desarrollado y analizando el comportamiento bajo las condiciones impuestas del cruce semaforizado.
- **Capítulo 5:** Conclusiones y líneas futuras. En el último capítulo se presentan las conclusiones obtenidas en este TFG y se exponen las posibles líneas futuras para mejorar y desarrollar este modelo, teniendo en cuenta la posibilidad de un aumento de la complejidad en el cruce semaforizado.
- **Capítulo 6:** Por último, se incluyen las referencias y fuentes consultadas para la realización de este trabajo.

2 ESTADO DEL ARTE

En este capítulo, se introduce la noción de modelo matemático como una herramienta que permite representar y analizar sistemas reales de forma simplificada, manteniendo el rigor necesario para su estudio. A continuación, se aborda la programación lineal, describiendo sus fundamentos básicos y su utilidad en la resolución de problemas de optimización propios del ámbito de la ingeniería, haciendo mención del método *simplex*.

Por último, se presenta el entorno de cálculo MATLAB, destacando su papel como herramienta de apoyo para la implementación práctica de los modelos matemáticos y la resolución computacional de los problemas planteados.

Este conjunto de conceptos constituye la base teórica sobre la que se apoya el desarrollo del trabajo en los capítulos siguientes.

2.1 Modelos matemáticos

2.1.1 Definición

Un modelo matemático es una representación simplificada de un problema o situación de la vida real que utiliza el lenguaje matemático para describir la relación entre distintas variables y parámetros. Se considera una abstracción del sistema real que permite analizar fenómenos complejos de forma comprensible y predecible. [11] [12]

Todo modelo matemático, independientemente de su complejidad, se apoya en los siguientes componentes: [11] [13] [14]

1. Variables. Son los objetos o conceptos que se desea analizar, a menudo divididos en variables de entrada, de salida y de estado.
2. Parámetros. Valores conocidos o controlables que permanecen constantes dentro de la escala del modelo.
3. Restricciones. Límites lógicos que aseguran que los resultados sean razonables.
4. Relaciones. El establecimiento de vínculos entre variables basados en teorías físicas, económicas o sociales.

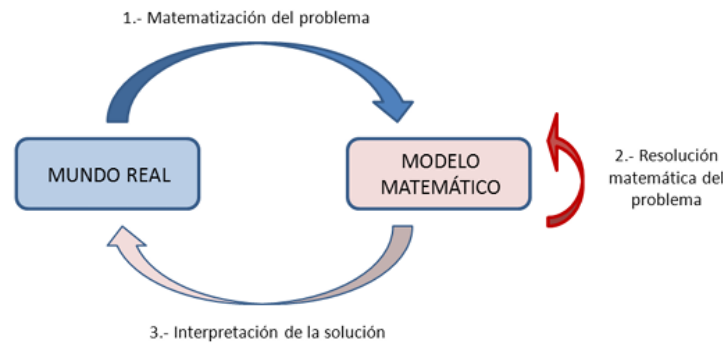


Figura 2-1 Adaptación del modelo matemático a la vida real. Figura extraída de [11]

La siguiente tabla muestra diversas formas de categorizar estos modelos según su propósito o funcionamiento:

Tipo de clasificación	Variantes principales	Herramienta matemática común
Fundamento	Deterministas/Físicos	Regresión/Ecuaciones diferenciales
Azar	Estocásticos	Funciones de distribución y azar
Espacio	Agregados/Distribuidos	Modelos de datos espaciales
Tiempo	Estáticos/Dinámicos	Algoritmos y series temporales

Tabla 2-1 Tipos de modelos matemáticos [14]

La construcción de un modelo matemático se apoya en una serie de etapas bien definidas. La primera de ellas consiste en un proceso de **abstracción**, cuyo objetivo es identificar los elementos más relevantes de la situación o problema que se desea analizar. Este proceso permite simplificar la realidad, centrándose únicamente en aquellos componentes que influyen de manera significativa en el comportamiento del sistema.

Para poder identificar dichos componentes, es necesario llevar a cabo una conceptualización del problema, en la que se definan con claridad sus características fundamentales. Esta conceptualización debe contemplar, como mínimo, los siguientes aspectos:

1. Identificación y conceptualización. Definir objetivos, identificar variables relevantes y establecer hipótesis simples.
2. Matematización. Traducir el problema al lenguaje simbólico mediante ecuaciones algebraicas o diferenciales.
3. Resolución y calibración. Una vez obtenidas las soluciones, ajustar los parámetros basándose en datos reales.
4. Validación y verificación. Comparar los resultados con la realidad para determinar la credibilidad del modelo; si hay errores, se deben ajustar las hipótesis iniciales.

Por último, para cerrar el ciclo de modelización, se lleva a cabo la etapa de **interpretación**, en la cual los resultados numéricos o simbólicos obtenidos, de vuelta al lenguaje de la realidad son traducidos para verificar si el problema original ha sido resuelto. En caso negativo, se iniciaría el llamado *proceso cíclico de retroalimentación*, durante el cual se deberán de replantear las hipótesis y ecuaciones, revisar el modelo conceptual y/o modificar los parámetros.

Una vez definidos los objetivos del modelo, sus componentes y fases, el problema puede ser formulado de manera matemática y abordarse mediante técnicas de optimización. En este contexto, la programación lineal se presenta como una herramienta especialmente adecuada cuando las relaciones entre las variables pueden expresarse de forma lineal, permitiendo obtener soluciones óptimas de manera eficiente y sistemática.

2.2 Optimización lineal

2.2.1 Definición

“La Programación Matemática, en general, aborda el problema de determinar asignaciones óptimas de recursos limitados para cumplir un objetivo dado. El objetivo debe representar la meta del decisor. Los recursos pueden corresponder, por ejemplo, a personas, materiales, dinero o terrenos. Entre todas las asignaciones de recursos admisibles, queremos encontrar la/s que maximiza/n o minimiza/n alguna cantidad numérica tal como ganancias o costos.” [15], p.6.

La optimización lineal (o programación lineal), es una técnica matemática destinada a determinar la mejor solución posible, maximizando o minimizando una magnitud concreta. Su planteamiento se basa en la definición de una función objetivo de carácter lineal, cuyo valor se ha de optimizar, sujeto a un conjunto de restricciones lineal siendo estas limitaciones reales del sistema analizado. [16] [17] [18]

A la hora de usar este método, se deberán de tener en cuenta tres principios o requisitos [19] [15]

1. Linealidad. La contribución de cada variable a la función objetivo y a las restricciones es proporcional al su valor y únicamente deben ser sumadas o restadas.
2. Maximizar o minimizar la función lineal será el objetivo de este modelo matemático.
3. Las restricciones impuestas deberán ser igualmente lineales.

De forma general, un problema de programación lineal puede formularse matemáticamente como:

$$\begin{aligned} \text{Max (Min)} \quad & c^T x = \sum_{j=1}^n c_j x_j \\ \text{tal que} \quad & \begin{cases} A \cdot x \leq b, \\ Aeq \cdot x = beq, \\ lb \leq x \leq ub, \end{cases} \quad (1) \end{aligned}$$

donde f, x, b, beq, lb y ub son vectores y A y Aeq son matrices [20].

Conviene destacar que, en la práctica, un problema de optimización puede reformularse con facilidad pasando de maximizar a minimizar. Este cambio se consigue simplemente multiplicando por -1 los coeficientes de la función objetivo, esto es:

$$\arg \text{Max (Min)} \quad c^T x = \sum_{j=1}^n c_j x_j \leftrightarrow \arg \text{Min} \quad -c^T x = -\sum_{j=1}^n c_j x_j$$

2.3 El método SIMPLEX

2.3.1 Origen.

El método Simplex fue desarrollado por el matemático y científico estadounidense George Dantzig durante su tiempo trabajando en el pentágono, cuando miembros de la Fuerza Aérea estadounidense le plantearon el desafío de idear un procedimiento que les permitiera acelerar y automatizar sus tareas de planificación.

En la terminología empleada por Dantzig, el concepto de *programación* hace referencia al proceso de planificación, mientras que el término *lineal* describe la relación proporcional y aditiva existente entre las distintas actividades, los recursos que requieren y los costes asociados a las mismas. [21]

El método simplex surge como una técnica capaz de determinar la mejor combinación posible entre un conjunto muy amplio de alternativas de decisión. A partir de este enfoque, Dantzig puso de manifiesto que una gran variedad de problemas de decisión, procedentes de ámbitos tan diversos como la empresa, la administración pública o el sector militar, podían formularse de manera sistemática como problemas de programación lineal. Este planteamiento supuso un avance decisivo para la investigación operativa, al proporcionar una herramienta eficaz para abordar problemas ya conocidos y otros nuevos, además de introducir una forma innovadora de plantear y demostrar resultados matemáticos.

2.3.2 Definición.

Se define el método Simplex como un algoritmo de optimización diseñado para resolver problemas de programación lineal, buscando obtener la solución óptima de la función objetivo que cumpla el conjunto de restricciones impuestas. Cabe destacar que este método lleva implícito una gran carga de conceptos geométricos. [22]

La base conceptual del método se apoya en la interpretación geométrica del problema: el conjunto de soluciones factibles define una región convexa, cuyos vértices representan soluciones extremas candidatas a ser óptimas.

El método simplex parte de la idea fundamental de que, si existe una solución óptima, esta se alcanzará en alguno de los vértices de la región factible (Figura 2-3). A partir de esto, el algoritmo comienza identificando una **solución básica factible inicial**, que corresponde a uno de estos vértices y que puede obtenerse directamente cuando el problema se encuentra en forma estándar. Del mismo modo, esta solución se puede obtener mediante procedimientos auxiliares, como el método de las dos fases o el método de la gran M, cuando dicha solución no es evidente.

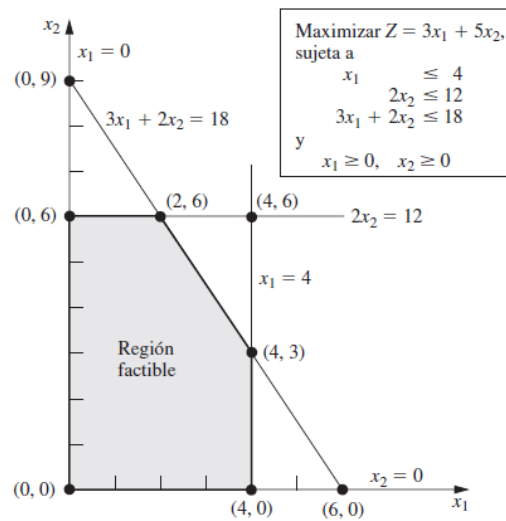


Figura 2-2 Ejemplo de restricciones en frontera y región factible. Figura extraída de [22]

Una vez establecida la solución inicial, el método avanza de forma iterativa, analizando cómo modificar la base actual con el objetivo de mejorar el valor de la función objetivo. Para ello, se selecciona una **variable no básica entrante**, cuya incorporación a la base permitiría aumentar o disminuir el valor de la función objetivo al mayor ritmo posible. De manera simultánea, se determina la **variable básica saliente**, garantizando que la nueva solución siga siendo factible y respete todas las restricciones del modelo.

Este proceso puede interpretarse como un desplazamiento controlado a lo largo de las aristas de la región factible, pasando de un vértice a otro adyacente que ofrece una mejora en el valor de la función objetivo (Figura 2-4). El procedimiento se repite hasta que no es posible encontrar ninguna variable entrante que mejore el resultado actual, momento en el cual se alcanza la **solución óptima** del problema. En ese punto, el algoritmo concluye, certificando que la solución obtenida es óptima dentro del conjunto factible. [4] [5] [22] [23] [24]

Sin embargo, la aplicación de este método permite, además de obtener la ya mencionada solución óptima, dar como resultado los siguientes tipos: [22]

1. **Múltiples soluciones óptimas.** Existen varias soluciones factibles con el mismo valor óptimo, lo que ocurre cuando la función objetivo es paralela a una arista de la región factible.
2. **Soluciones degeneradas.** En donde una o más variables toman el valor cero, provocando que el método avance sin mejorar el valor de la función objetivo en determinadas iteraciones, aunque no invalida el método.
3. **Problema no acotado.** Cuando la función objetivo puede mejorar indefinidamente sin violar ninguna restricción y por tanto no existe una solución óptima finita. Se da cuando no es posible determinar una variable básica saliente al intentar introducir una variable entrante que mejora la función objetivo.
4. **Problema no factible.** Si no existe ningún punto que satisfaga simultáneamente todas las restricciones del modelo.

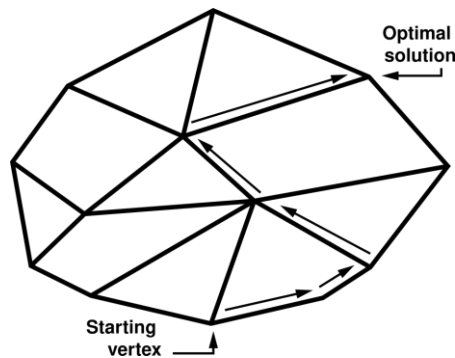


Figura 2-3 Poliedro asociado al método Simplex. Figura extraída de [25]

2.4 Programa MATLAB

2.4.1 Definición.

MATLAB (MATriz LABoratory) es un software matemático desarrollado y mantenido por la empresa MathWorks, que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Este programa se puede emplear para el cálculo matricial, programación de algoritmos, análisis simbólicos, representación y análisis de datos y funciones, etc.

Dentro del entorno de trabajo de MATLAB, además de sus capacidades técnicas, destaca la red de soporte al usuario a través de una documentación exhaustiva, actualizada y accesible, que cubre desde los fundamentos del entorno hasta el uso avanzado de *toolboxes* específicas, como las orientadas a la optimización y la programación lineal y que se describirán en el siguiente subapartado.

Asimismo, MATLAB dispone de una extensa base de ejemplos, tutoriales y casos de estudio que facilitan la comprensión y aplicación práctica de los distintos métodos, entre ellos el método simplex. A esto se suma la existencia de foros y comunidades de usuarios, así como soporte técnico especializado, que contribuyen a resolver dudas y a validar la correcta implementación de los modelos. Todo ello convierte a MATLAB en una herramienta sólida y bien respaldada, especialmente adecuada para su utilización en trabajos académicos y de investigación.

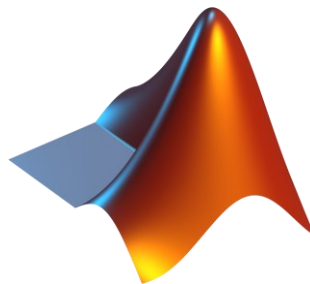


Figura 2-4 Logotipo de Matlab. Figura extraída de [26]

Otra de las ventajas de MATLAB, es la integración en su software del llamado **Simulink**, un “entorno de diagramas de bloques para la simulación multidominio y el diseño basado en modelos. Permite el diseño a nivel de sistema, la simulación, la generación automática de código y la verificación y prueba continua de sistemas embebidos. Simulink proporciona un editor gráfico, bibliotecas de bloques personalizables y solvers para el modelado y la simulación de sistemas dinámicos.” [27]

2.4.2 La optimización lineal con MATLAB.

A la hora de usar este programa, se deberá tener descargado lo que se conoce como *toolbox*, que, por definición, son conjuntos de herramientas diseñadas para ampliar las funcionalidades básicas del programa y especializadas en áreas concretas, resultando especialmente útiles en el ámbito de la **programación lineal**. Algunas de estas funciones aplicadas a la programación lineal son:

Función	Propósito
$x = \text{linprog}(f, A, b)$	Resuelve el mínimo de f^*x de forma que $A*x \leq b$
$x = \text{linprog}(f, A, b, Aeq, beq)$	Incluye restricciones de igualdad $Aeq*x = beq$.
$x = \text{linprog}(f, A, b, Aeq, beq, lb, ub, options)$	Minimiza con las opciones de optimización especificadas por <i>options</i> .

Tabla 2-2 Algunos de los comandos MATLAB aplicados a programación lineal. Datos extraídos de [28]

Desde un punto de vista académico y profesional, MATLAB actúa como un puente entre la teoría de la programación lineal y su aplicación práctica.

Mientras que el método simplex proporciona la base algorítmica, MATLAB abstrae los detalles de programación del sistema, permitiendo al usuario centrarse en la correcta modelización del problema y en la interpretación de la solución obtenida.

3 DESARROLLO DEL TFG

3.1 Planteamiento del problema de forma matemática

En este trabajo se considera un cruce urbano controlado por semáforos. Dicho cruce consta de dos vías principales, cada una con doble carril, uno en cada sentido. Este sistema funciona según la convención estándar de circulación por la derecha, donde se permiten giros a la izquierda en ambas vías (vía Norte-Sur y vía Este-Oeste) como se muestra en la Figura 3-1. Según esta convención un giro a la izquierda se define como un giro de noventa grados (90°) sentido antihorario respecto al sentido de avance del vehículo.

Veamos en qué consiste un ciclo de un semáforo de este tipo. En este caso se consideran cuatro fases (fase 1: Norte-Sur, fase 2 Este-oeste, fase 3: Giros izquierda en Norte-sur, fase 4: giros a la izquierda de Este-oeste). Se desea minimizar la demora de vehículos en un ciclo de **tiempo total CT** de un semáforo. Para ello se quiere modelar el problema como un problema de programación lineal, es decir tanto la función objetivo como las restricciones del problema son lineales.

Para ello se definen los siguientes movimientos para las 4 fases:

1. **Fase $i=1$.**
 - a. Movimiento $i1$: Ir de Norte a Sur.
 - b. Movimiento $i2$: Ir de Sur a Norte.
2. **Fase $i=2$.**
 - a. Movimiento $i1$: Ir de Este a Oeste.
 - b. Movimiento $i2$: Ir de Oeste a Este.
3. **Fase $i=3$.**
 - a. Movimiento $i1$: Girar de Norte a Este.
 - b. Movimiento $i2$: Girar de Sur a Oeste.
4. **Fase $i=4$.**
 - a. Movimiento $i1$: Girar de Este a Sur.
 - b. Movimiento $i2$: Girar de Oeste a Norte.

3.1.1 Variables

Para poder plantear el problema de programación lineal definiremos las siguientes variables:

1. g_i = Tiempo en segundos que dura la luz verde en la fase i , $i=1,2,3,4$.

2. **CT**= Segundos máximos del ciclo de 4 fases, con 8 movimientos (incluye tiempos perdidos de luz roja y amarilla). Como máximo en cada fase i , los 2 movimientos ocurren simultáneamente sin afectarse. La luz verde tiene la misma duración y es la misma para los 2 movimientos de la misma fase.
3. **f_{ij}** = Flujo de vehículos promedio (veh/s) en la fase i movimiento j . ($i=1:4$; $j=1,2$).
4. **q_{ij}** = Capacidad (saturación) de vehículos que quiere el movimiento j de la fase i (veh/seg).
5. **g_i** = Como los movimientos $i1$ e $i2$ de cada fase i comparten la misma luz verde, esta variable corresponde al tiempo en segundos que dura la luz verde en la fase i ($i=1:4$).
6. **L** = Tiempo total muerto en el ciclo del semáforo (amarillas + rojas de seguridad).
7. **n** = Número de fases.

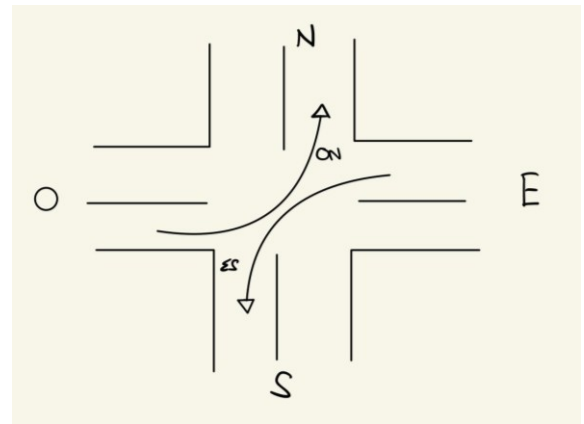
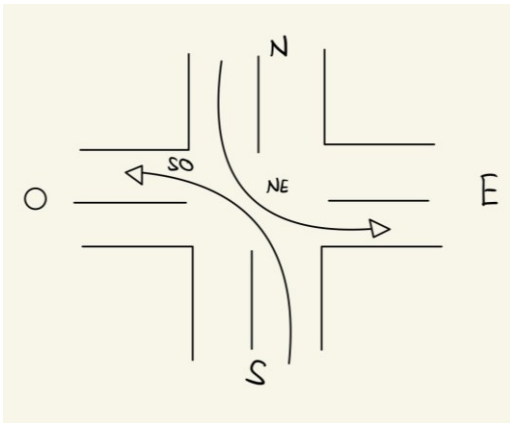


Figura 3-1 Representación giros a la izquierda en las diferentes vías.
Figuras generadas con inteligencia artificial [31].

A continuación, se exponen algunas expresiones matemáticas para plantear el problema de programación lineal:

La demora en este tipo de problema puede escribirse mediante funciones no lineales, algunas más precisas que otras.

Fórmula de Webster: La demora media de la fase i , viene dada por

$$d_i = \frac{CT(1 - \frac{g_i}{CT})}{2(1 - \frac{f_i}{q_i})} + \frac{900T(\frac{f_i}{q_i} - 1)^2}{CT(1 - \frac{f_i}{q_i})} \quad (2)$$

El primer término de **d_i** en la ecuación (2), mide la demora de espera durante un ciclo y asume que los vehículos llegan uniformemente durante el ciclo. Este primer término crece cuando la luz verde disminuye o cuando el flujo (f_i) se acerca a la capacidad (q_i).

El segundo término es la demora por saturación o aleatoriedad (overflow). Se activa si $f_i/g_i > 1$. Este término representa colas acumuladas por saturación. Este término suele omitirse si el flujo es menor que la capacidad.

Por consiguiente, si $f_i/g_i < 1$ y la fase no está saturada, generalmente la demora se aproxima por el primer término:

$$d_i \approx \frac{CT \left(1 - \frac{g_i}{CT}\right)^2}{2 \left(1 - \frac{f_i}{g_i}\right)} (s)$$

La demanda total de la fase i vendrá dado por:

$$\left\{ \begin{array}{l} Di = f_i \cdot CT \cdot d_i \\ = f_i \cdot CT \cdot \left[\frac{CT(1 - g_i/CT)^2}{2(1 - f_i/q_i)} \right] \\ = \left(\frac{veh}{s} \right) \cdot s \cdot s = \left(\frac{veh}{s} \right) \end{array} \right.$$

Por ejemplo: tener 10 vehículos esperando 5 s equivale a: $Di = 50 \left(\frac{veh}{s} \right)$

Aproximación lineal de la demora:

Una forma de definir la demora consiste en calcular el tiempo perdido en la fase i, es decir, el tiempo de la fase i correspondiente a periodo sin luz verde, esto es:

$$r_j = CT - g_j (s) \rightarrow \text{Tiempo perdido fase } i$$

Por otra parte, el tiempo en segundos del ciclo completo (CT) viene dado por:

$$CT = L + \sum_{i=1}^n g_i (s)$$

Donde L es el tiempo muerto, esto es, el tiempo de seguridad durante el cual no pasa ningún vehículo de ninguna fase del ciclo (consiste en tiempos pequeños de luz roja y luz amarilla de seguridad), entonces:

$$r_j = L + \sum_{i=1}^n g_i - g_j = L + \sum_{\substack{i=1 \\ i \neq j}}^n g_i (s)$$

Por ejemplo, si para cada fase $i=1:4$, el tiempo de amarillo es de tres segundos y la fase de seguridad en rojo es de tres segundos,

$$L = (4 \cdot 3) + (4 \cdot 3) = 24 \text{ segundos}$$

La demora en segundos de la fase i viene dada por:

$$d_i = K \cdot r_i$$

Por tanto, la demora total en fase i (**veh/s**) será:

$$D_i = f_i \cdot d_i \cdot CT = K f_i \cdot r_i \cdot CT$$

Generalmente $K=1/2$ o $K=1$.

La aproximación lineal de la demora total en fase i en función de las variables g_i , viene dada por:

$$\begin{cases} D_i = f_i \cdot (CT - g_i) \cdot K \cdot CT \\ D_i = K f_i \cdot CT (CT - g_i) \left(\frac{veh}{s}\right) \end{cases}$$

Cabe recordar que cada fase i tiene dos flujos (f_{i1}, f_{i2}) y dos capacidades (q_{i1}, q_{i2}). Entonces:

$$\begin{cases} f_i = f_{i1} + f_{i2} \\ q_i = q_{i1} + q_{i2} \end{cases}$$

3.1.2 Restricciones

A continuación, presentamos las restricciones básicas para un modelo de semáforo.

1. La primera restricción corresponde a la distribución del tiempo del ciclo CT ; para las variables L y CT del ciclo:

$$CT = L + \sum_{i=1}^n g_i \text{ (s)}$$

Los vehículos que pasan en luz verde en la fase i , vienen dados por:

$$f_i \cdot g_i$$

Los vehículos que pueden pasar en el ciclo son:

$$CT \cdot q_i$$

Entonces existe una condición de capacidad que dice que los vehículos que pueden llegar en un ciclo deben ser menor o igual que el máximo número de vehículos que pasan en luz verde, de donde:

$$g_i \cdot q_i \geq f_i \cdot CT$$

Sin embargo, debido a que esta restricción en muchos casos es difícil satisfacer, podría relajarse de la siguiente manera:

$$g_i \cdot q_i \geq \alpha \cdot f_i \cdot CT ; \text{ con } 0 \leq \alpha \leq 1$$

2. Restricción de tiempo mínimo de duración de luz verde por fase i , viene dada por:

$$g_i \geq T_i^{\min}$$

También se puede agregar una restricción de tiempo máximo de duración de luz verde por fase:

$$g_i \leq T_i^{\max}$$

3. Para evitar mucha diferencia entre algunas fases, cuyo flujo es parecido, se puede agregar restricciones del tipo:

$$0 \leq g_i - g_j \leq a_i$$

$$\text{siendo } i = 1, 2, \dots, n \text{ e } i \neq j$$

3.1.3 Modelos propuestos

A continuación, se definen tres modelos para determinar el tiempo óptimo de luz verde en las distintas fases del semáforo. Para ello, se asume que los flujos y capacidad de saturación son iguales en todas las líneas de la misma fase i , además las dos líneas de la misma fase i comparten la misma luz verde, roja y luz muerta (**L**).

Se quiere minimizar la demora total en las n fases del problema:

$$\text{Min} \sum_{i=1}^n Di = \text{Min} \sum_{i=1}^n K \cdot CT \cdot f_i \cdot (CT - g_i) = \text{Min}_{(g_i; i=1,2,\dots,n)} \sum_{i=1}^n K \cdot CT^2 \cdot f_i - \sum_{i=1}^n K \cdot CT \cdot f_i \cdot g_i$$

Como que el término $\sum_{i=1}^n K \cdot CT^2 \cdot f_i$ es constante, el $g = (g_1, g_2, \dots, g_n)^T \in \mathbb{R}^n$ que minimiza la suma de las demoras ($\sum_{i=1}^n Di$) coincide con el mínimo de la función $\sum_{i=1}^n K \cdot CT \cdot f_i \cdot g_i$. Por consiguiente, la minimización de las sumas de las demoras es equivalente a:

$$\text{Min}_{(g_i; i=1,2,\dots,n)} Z_1(g) = -K \cdot CT \sum_{i=1}^n f_i \cdot g_i \quad (H1)$$

1. Modelo P1:

$$\left\{ \begin{array}{ll} \text{Min}_{(g_i; i=1,2,\dots,n)} Z_1(g) = -K \cdot CT \sum_{i=1}^n f_i \cdot g_i & (H1) \\ \text{s. a} \quad \sum_{i=1}^n g_i = CT - L & (A) \\ q_i \cdot g_i \geq f_i \cdot CT, i = 1, 2, \dots, n & (B) \\ T_i^{\min} \leq g_i \leq T_i^{\max}, i = 1, 2, \dots, n & (D) \end{array} \right.$$

La restricción (B) impone la condición de que los vehículos que llegan son siempre menor o igual al número máximo de vehículos que pueden pasar en la luz verde de la fase i . Esta restricción suele ser muy teórica pero difícil de satisfacer en muchas simulaciones. Para que la restricción (B) se satisfaga, deben verificarse en cada fase i del ciclo:

$$g_i \geq \frac{f_i \cdot CT}{q_i}$$

Por tanto,

$$\sum_{i=1}^n g_i \geq \sum_{i=1}^n \frac{f_i \cdot CT}{q_i}$$

Usando la restricción (A), se tiene que

$$\sum_{i=1}^n g_i \leq \sum_{i=1}^n \frac{f_i \cdot CT}{q_i} = CT - L$$

Entonces

$$\sum_{i=1}^n \frac{f_i}{q_i} \leq \frac{CT - L}{CT} = 1 - \frac{L}{CT} \quad (3)$$

La restricción (3) puede no cumplirse en algunas ocasiones y las formas de conseguir que (3) se satisfaga implica:

- Aumentar el valor de CT.
- Reducir el valor de fi.
- Aumentar el valor de qi.
- Relajar la restricción (B).

Una forma de relajar la restricción (B) es considerar:

$$q_i \cdot g_i \geq \alpha \cdot CT \cdot f_i \quad 0 < \alpha \leq 1 \quad i = 1, 2, \dots, n$$

Por todo lo expuesto anteriormente en este apartado, se plantea a continuación el siguiente modelo.

2. Modelo P2: Relajación modelo P1.

$$\left\{ \begin{array}{ll} \text{Min}_{(g_i; i=1,2,\dots,n)} Z_1(g) = -K \cdot CT \sum_{i=1}^n f_i \cdot g_i & (H1) \\ \text{s. a} \quad \sum_{i=1}^n g_i = CT - L & (A) \\ q_i \cdot g_i \geq \alpha \cdot f_i \cdot CT, i = 1, 2, \dots, n, \quad 0 < \alpha \leq 1 & (B1) \\ T_i^{\min} \leq g_i \leq T_i^{\max}, i = 1, 2, \dots, n & (D) \end{array} \right.$$

En los Modelos P1 y P2 se minimiza la suma de las demoras, la cual viene dada por la expresión (H1). Si en lugar de minimizar la suma de las demoras se quiere minimizar la duración de las luces rojas del ciclo, se define una nueva variable, Ri, que representa los segundos de luz roja de la fase i en el ciclo. La luz roja de cada fase está completamente determinada por la luz verde en cada fase i, de la siguiente forma:

$$Ri = \sum_{\substack{j=1 \\ j \neq i}}^n g_j + L \quad (4)$$

Ahora, si además se quiere maximizar los segundos de luz verde en cada fase i del ciclo, el problema se plantea como:

$$\text{Min } Z_2(Ri, gi; i=1,2,\dots,n) = \sum_{i=1}^n fi \cdot Ri - M \sum_{i=1}^n gi \quad M > 0 \quad (H2)$$

Como Ri viene dado por (4), sustituyendo en la expresión dada por (H2), se tiene:

$$\begin{aligned} Z_2(g) &= \sum_{i=1}^n fi \cdot \left(\sum_{\substack{j=1 \\ j \neq i}}^n g_j + L \right) - M \sum_{i=1}^n gi \\ Z_2(g) &= \sum_{i=1}^n fi \cdot L + \sum_{i=1}^n fi \cdot \sum_{\substack{j=1 \\ j \neq i}}^n g_j - M \sum_{i=1}^n gi \end{aligned}$$

Como $\sum_{i=1}^n fi \cdot L$ es constante, se tiene que minimizar $Z_2(g)$ es equivalente a:

$$\text{Min } (gi; i=1,2,\dots,n) Z_2(g) = \sum_{i=1}^n fi \cdot \left(\sum_{\substack{j=1 \\ j \neq i}}^n g_j \right) - M \sum_{i=1}^n gi \quad (H2)$$

La función objetivo (H2), para $n=4$, que es el tiempo de ciclo metafórico que se considera en este trabajo, se tiene que:

$$\begin{aligned} Z_2(g) &= f_1(g_2 + g_3 + g_4) + f_2(g_1 + g_3 + g_4) + f_3(g_1 + g_2 + g_4) + f_4(g_1 + g_2 + g_3) \\ &\quad + f_1(g_2 + g_3 + g_4) - M(g_1 + g_2 + g_3 + g_4) \\ &= g_1(-M + f_2 + f_3 + f_4) + g_2(-M + f_1 + f_3 + f_4) + g_3(-M + f_1 + f_2 + f_4) \\ &\quad + g_4(-M + f_1 + f_2 + f_3) \\ &= \sum_{i=1}^{n=4} gi \cdot \left(-M + \sum_{\substack{j=1 \\ j \neq i}}^{n=4} fj \right) \quad (H2) \end{aligned}$$

Por otro lado, la relajación de la restricción (B) del modelo (P1), se puede plantear de forma diferente como:

$$q_i \cdot g_i + u_i \geq CT \cdot f_i \quad i = 1, 2, \dots, n \quad (B2)$$

donde u_i representa los vehículos retenidos al terminar la fase i del ciclo.

Entonces, se plantea un nuevo modelo de optimización que relaja la restricción de capacidad (B) y que además toma en consideración la luz roja de la fase i del ciclo, escribiéndose tal y como se indica en el siguiente apartado.

3. **Modelo P3:** Se pueden mejorar los modelos (P1) (P2) definiendo 4 variables más:

$$\left\{ \begin{array}{ll} \text{Min}_{((g_i; u_i)_{i=1,2,\dots,n})} Z_1(g) \text{ (o Min}_{((g_i; u_i)_{i=1,2,\dots,n})} Z_2(g) \text{)} & (H1 \text{ o } H2) \\ s. a \sum_{i=1}^n g_i = CT - L & (A) \\ q_i \cdot g_i + u_i \geq f_i \cdot CT & (B2) \\ 0 \leq g_i - g_j \leq a_i, \text{ para algún } i = 1, 2, \dots, n \text{ y } j \neq i & (C) \\ T_i^{\min} \leq g_i \leq T_i^{\max} & (D) \\ 0 \leq u_i & (E) \\ u_i \text{ y } g_i \text{ enteros, } i = 1, 2, \dots, 4 \end{array} \right.$$

Las restricciones dadas en (C), fuerzan a que las luces verdes de ciertas fases i sean mayores que ciertas fases j . Esa condición se puede agregar si se quiere dar prioridad de tiempo a alguna fase.

La siguiente tabla trata de resumir las ventajas y desventajas de los tres modelos propuestos anteriormente.

Modelo	Factible	Colas Residuales	Uso
P1	No siempre	No	Diseño teórico
P2	Si (para algún α y valores de f_i y q_i)	Existen, pero no se pueden determinar (no son variables del modelo)	Control real

P3	Siempre	Si, y además se pueden determinar Si $u_i=0$ no hay cola en la fase i	Control real
----	---------	--	--------------

Tabla 3-1 Ventajas y desventajas de los modelos planteados

La Tabla 3-2 muestra valores reales de ciclo de semáforos urbanos de afluencia media:

Movimiento	Flujo (f_i) (s)	Capacidad (q_i) (s)	Comentarios
Recto	0,2-0,4	0,5-0,6	Tránsito urbano medio
Recto (alto pero estable)	0,4-0,5	0,55-0,6	Cerca de la saturación
Giro izquierda	0,05-0,15	0,25-0,35	Menos capacidad por maniobra
Giro izquierda (alto)	0,15-0,25	0,3-0,35	Giro protegido

Tabla 3-2 Valores de ciclos en función de los flujos y capacidades de un cruce con afluencia media.

3.1.4 Consideraciones

En general, se asume que el flujo de llegada (f_i) es menor que la capacidad del sistema (q_i). No obstante, pueden presentarse los siguientes casos:

$$\begin{cases} f_i = q_i \rightarrow \text{Sistema crítico} \\ f_i > q_i \rightarrow \text{Colas crecen sin límites} \end{cases}$$

3.1.5 Sistema matricial y vectorial a introducir en Matlab

A continuación, se presenta la forma matricial de escribir las diferentes restricciones de los modelos (P1) (P2) para $n=4$.

La restricción (A) se escribe como $A_{eq} \cdot x = b_{eq}$; donde:

$$A_{eq} \cdot x = (1 \ 1 \ 1 \ 1) \cdot \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} = (CT - L) = b_{eq}$$

Las restricciones (B)/(B1) para $n=4$ vienen dadas por:

$$\begin{cases} -q_1 \cdot g_1 \leq \alpha \cdot -CT \cdot f_1 \\ -q_2 \cdot g_2 \leq \alpha \cdot -CT \cdot f_2 \\ -q_3 \cdot g_3 \leq \alpha \cdot -CT \cdot f_3 \\ -q_4 \cdot g_4 \leq \alpha \cdot -CT \cdot f_4 \end{cases} \quad 0 < \alpha \leq 1$$

Su forma matricial será $A_1 \cdot x \leq \alpha \cdot b_1$, donde:

$$A_1 \cdot x = \begin{pmatrix} -q_1 & 0 & 0 & 0 \\ 0 & -q_2 & 0 & 0 \\ 0 & 0 & -q_3 & 0 \\ 0 & 0 & 0 & -q_4 \end{pmatrix} \cdot \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} \leq \alpha \cdot (-CT) \cdot \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \alpha \cdot b_1$$

Para el modelo (P3), la restricción (B2) para $n=4$ viene dada por

$$(B2) \begin{cases} -q_1 \cdot g_1 - u_1 \leq -CT \cdot f_1 \\ -q_2 \cdot g_2 - u_2 \leq -CT \cdot f_2 \\ -q_3 \cdot g_3 - u_3 \leq -CT \cdot f_3 \\ -q_4 \cdot g_4 - u_4 \leq -CT \cdot f_4 \end{cases}$$

Se escribe de forma matricial como $\tilde{A}_1 \cdot \tilde{x} = b_1$, donde:

$$\tilde{A}_1 \cdot \tilde{x} = \begin{pmatrix} -q_1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -q_2 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -q_3 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -q_4 & 0 & 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} g_1 \\ \vdots \\ g_4 \\ u_1 \\ \vdots \\ u_4 \end{pmatrix} \leq -CT \cdot \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = b_1$$

La restricción (C), para los pares $(i,j) = (1,2)$ y $(i,j) = (3,4)$, se escribe de forma matricial como $A_2 \cdot \tilde{x} \leq b_2$, donde:

$$A_2 \cdot \tilde{x} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} \leq b_2 = \begin{pmatrix} a_1 \\ a_2 \\ 0 \\ 0 \end{pmatrix}$$

3.1.6 Funcionamiento del código.

A continuación, se describe el procedimiento seguido para el desarrollo e implementación, en MATLAB, del modelo de optimización lineal destinado a calcular los tiempos óptimos de verde en una intersección semaforizada de cuatro fases, cuyo objetivo principal es minimizar las retenciones vehiculares durante cada ciclo completo, garantizando simultáneamente el cumplimiento de las restricciones físicas y operativas del sistema.

En primer lugar, se definen los **parámetros del problema**, como los flujos de saturación (q), el flujo de tráfico (f), el tiempo total de ciclo (CT), el tiempo muerto en cambios de fase (L) y el número de fases (N=4). También se establecen cotas mínimas y máximas para los tiempos de verde (Tmin y Tmax) con el objetivo de garantizar valores físicamente razonables y enteros.

Para encontrar la solución del modelo, se utiliza la función **linprog** de MATLAB para resolver el problema de programación lineal asociado. Esta devuelve el vector de tiempos óptimos de verde (g) y el valor óptimo de la función objetivo (fval). Para obtener soluciones enteras del modelo, se usa la función **intlinprog** que garantiza que el vector de tiempos óptimos en verde es entero, al igual que los coches retenidos en las fases.

Posteriormente, se generan los parámetros y se construyen las **matrices del modelo de programación lineal** dado por (1). En particular:

1. La matriz A y el vector b representan las **restricciones de capacidad**, en los modelos (P1), (P2) que aseguran que el número de vehículos que llegan no supere la capacidad de evacuación de cada fase. En el caso del modelo (P3) esta matriz también incluye las restricciones de desigualdad dadas por (C).
2. La matriz A_{eq} y el valor b_{eq} imponen la **restricción de ciclo**, es decir, que la suma de los tiempos de verde sea igual al tiempo efectivo disponible (CT - L).
3. Los vectores lb y ub definen las **cotas inferiores y superiores** de las variables.

La función objetivo se define mediante el vector f, el cual varía dependiendo si se utiliza la función objetivo $Z_1(g)$ o $Z_2(g)$.

Finalmente, el código calcula y muestra:

1. Los tiempos óptimos de verde de cada fase.
2. Las restricciones de capacidad considerando los vehículos retenidos.
3. Los tiempos en rojo de cada fase.
4. La suma total de tiempos de verde.
5. El valor de la función objetivo.

3.1.7 Datos del problema.

En el presente subapartado, se exponen los datos seleccionados para el desarrollo del problema y cuyas soluciones serán expuestas y discutidas en el posterior apartado.

Para la formulación y resolución del problema se ha utilizado el programa **MATLAB**, ejecutado en un ordenador personal, con licencia académica de la Universidad de Vigo. Este software permite implementar de forma eficiente el modelo de programación lineal planteado, así como procesar los datos necesarios y obtener la solución óptima del sistema. Mediante su entorno de cálculo numérico y sus herramientas de optimización, MATLAB facilita la resolución computacional del problema y el análisis de los resultados obtenidos.

Se han elaborado cuatro escenarios diferentes de un cruce urbano, donde, a partir de modificaciones en los datos de partida, se obtendrán diferentes resultados de tiempos en verde y holguras del cruce analizando dichos resultados obtenidos, escogiéndose la solución óptima. Se deberá tener en cuenta que independientemente de los tiempos en verde que se vayan a obtener, se ha decidido fijar una prioridad de paso en los trayectos norte/sur – sur/norte.

A continuación, se expondrán los datos de partida de los diferentes escenarios planteados, usando como base de trabajo el **Modelo P3** y como función objetivo $Z_1(g)$.

La capacidad típica para movimientos rectos de un carril suele ser de $q = 1900 \text{ veh/h}$, y la de giros a la izquierda de $q = 1700 \text{ veh/h}$. Ambos valores, teniendo en cuenta que se puede circular en ambos sentidos, han de multiplicarse por dos.

A continuación, la Tabla 3-3 muestra los datos de partida para el llamado Escenario 1, donde se mantendrán constantes tanto los tiempos máximos y mínimos, siendo estos de valor $T_i^{max} = (40,40,20,20)$ y $T_i^{min} = (24,23,6,5)$, como el valor de $CT = 90$. Sin embargo, los valores de L irán variando.

Escenario 1	Flujo (ff) (veh/h)	Capacidad (q) (veh/h)	L	CT
1.1	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	90
1.2	1000; 950; 220; 200	3800; 3800; 3400; 3400	27	90
1.3	1000; 950; 220; 200	3800; 3800; 3400; 3400	30	90
1.4	1000; 950; 220; 200	3800; 3800; 3400; 3400	32	90
1.5	1000; 950; 220; 200	3800; 3800; 3400; 3400	34	90

Tabla 3-3 Datos de partida del Escenario 1

A continuación, se presenta el Escenario 2, en donde se va a tener fija la variable $L = 25$, así como los tiempos antes mencionados y será ahora el valor de CT el que varíe

Escenario 2	Flujo (ff) (veh/h)	Capacidad (q) (veh/h)	L	CT
-------------	--------------------	-----------------------	---	----

2.1	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	88
2.2	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	90
2.3	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	92
2.4	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	94
2.5	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	96
2.6	1000; 950; 220; 200	3800; 3800; 3400; 3400	25	98

Tabla 3-4 Datos de partida del Escenario 2

Los siguientes dos escenarios que a continuación se relacionan, siguen la misma filosofía que los Escenarios 1 y 2. Los tiempos máximos y mínimos se mantienen, pero los valores de flujo aumentan. Asimismo, se seguirá la dinámica de alterar tanto valores de CT como de L.

En la Tabla 3-5, se expone el Escenario 3, con un aumento del flujo de vehículos y sin variar la capacidad del ciclo.

Escenario 3	Flujo (ff) (veh/h)	Capacidad (q) (veh/h)	L	CT
3.1	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	90
3.2	1500; 1450; 480; 420	3800; 3800; 3400; 3400	27	90
3.3	1500; 1450; 480; 420	3800; 3800; 3400; 3400	30	90
3.4	1500; 1450; 480; 420	3800; 3800; 3400; 3400	32	90
3.5	1500; 1450; 480; 420	3800; 3800; 3400; 3400	34	90

Tabla 3-5 Datos de partida del Escenario 3

El Escenario 4 (Tabla 3-6) muestra las variaciones en los valores de CT, manteniendo los datos de partida del Escenario 3.

OPTIMIZACIÓN DE TIEMPOS SEMAFORIZADOS EN UN CRUCE
URBANO CON AFLUENCIA MEDIA

Escenario 4	Flujo (ff) (veh/h)	Capacidad (q) (veh/h)	L	CT
4.1	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	88
4.2	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	90
4.3	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	92
4.4	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	94
4.5	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	96
4.6	1500; 1450; 480; 420	3800; 3800; 3400; 3400	25	98

Tabla 3-6 Datos de partida del Escenario 4

Los siguientes dos escenarios que a continuación se relacionan, siguen la misma filosofía que antes. Los tiempos máximos y mínimos se mantienen, pero los valores de flujo disminuyen y se establecen entre los utilizados entre el de los Escenarios 1-2 y 3-4. Asimismo, se seguirá la dinámica de alterar tanto valores de CT como de L.

En la Tabla 3-7, se expone el Escenario 5, con la adaptación ya mencionada del flujo.

Escenario 5	Flujo (ff) (veh/h)	Capacidad (q) (veh/h)	L	CT
5.1	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	90
5.2	1300; 1250; 380; 320	3800; 3800; 3400; 3400	27	90
5.3	1300; 1250; 380; 320	3800; 3800; 3400; 3400	30	90
5.4	1300; 1250; 380; 320	3800; 3800; 3400; 3400	32	90
5.5	1300; 1250; 380; 320	3800; 3800; 3400; 3400	34	90

Tabla 3-7 Datos de partida del Escenario 5

Se sigue por tanto con la metodología de siempre, presentándose a continuación los nuevos datos de CT del Escenario 6 (Tabla 3-8).

Escenario 6	Flujo (ff) (veh/h)	Capacidad (q) (veh/h)	L	CT
6.1	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	88
6.2	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	90
6.3	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	92
6.4	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	94
6.5	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	96
6.6	1300; 1250; 380; 320	3800; 3800; 3400; 3400	25	98

Tabla 3-8 Datos de partida del Escenario 6

Cabe destacar, que, para cada uno de los escenarios expuestos, se ha llevado a cabo cálculos con el código en los que se elimina la restricción (C) del Modelo P3. Los resultados de esta modificación del Modelo P3 se expondrán igualmente en el siguiente capítulo usando la numeración: (**Número escenario_MOD**).

Adicionalmente, para establecer relaciones o conclusiones entre las retenciones (ui) y el vector de flujos (f), se considera la siguiente ecuación:

$$X = \frac{\sum_{i=1}^4 \frac{f_i}{q_i}}{1 - \frac{L}{CT}} \quad (5)$$

Dependiendo de los valores de X se pueden establecer diferentes conclusiones: si $X < 1$ no existe sobresaturación del sistema, si $X = 1$ el sistema está en la capacidad máxima y si $X > 1$ el sistema está sobresaturado. Se llevarán a cabo las simulaciones con la expresión (5) para los escenarios 1,3 y 5.

4 RESULTADOS OBTENIDOS

En los presentes apartados se expondrán en formatos de tablas los resultados obtenidos para cada escenario planteado en el anterior capítulo, así como las conclusiones obtenidas. Los datos que a continuación se muestran hacen referencia a los tiempos de luz verde en cada fase; tiempos en luz roja en cada fase; valor de la función objetivo y la restricción de capacidad contabilizando los coches retenidos. La numeración del vector luz verde, luz roja y coches retenidos sigue la siguiente estructura: (valor movimiento N-S; valor movimiento E-O; valor movimiento giros N-S; valor movimiento giros E-O).

Cabe mencionar que en el presente trabajo también se realizaron simulaciones de todos los escenarios usando la función objetivo $Z_2(g)$. Sin embargo, los resultados obtenidos apenas distan de los calculados en el presente capítulo y no arrojan conclusiones distantes de las extraídas por lo que se ha decidido omitirlos.

4.1 Resultados del Escenario 1

En la siguiente tabla se muestran los resultados derivados de los datos de partida correspondientes, con un $CT = 90$ fijo y un valor de L en incremento. En el escenario 1.5 no se obtuvo solución factible por lo que se ha omitido.

Escenario 1	Vector luz verde (seg)	Σg_i	Coches retenidos	Σ Coches retenidos	Valor función objetivo	X
1.1	29; 24; 6; 6	65	0; 0; 0; 0	0	-1358	0,8816
1.1_MOD	31; 23; 6; 5	65	0; 0; 0; 1	1	-1379,2	0,8816
1.2	29; 24; 6; 6	65	0; 0; 0; 0	0	-1358	0,9095
1.2_MOD	29; 23; 6; 5	63	0; 0; 0; 1	1	-1329,2	0,9095
1.3	25; 23; 6; 6	60	0; 0; 0; 0	0	-1234,2	0,955

1.3_MOD	26; 23; 6; 5	60	0; 0; 0; 1	1	-1254,2	0,955
1.4	24; 23; 6; 5	58	0; 0; 0; 0	0	-1104,2	0,988
1.4_MOD	24; 23; 6; 5	58	0; 0; 0; 1	1	-1204,2	0,988

Tabla 4-1 Resultados obtenidos del Escenario 1

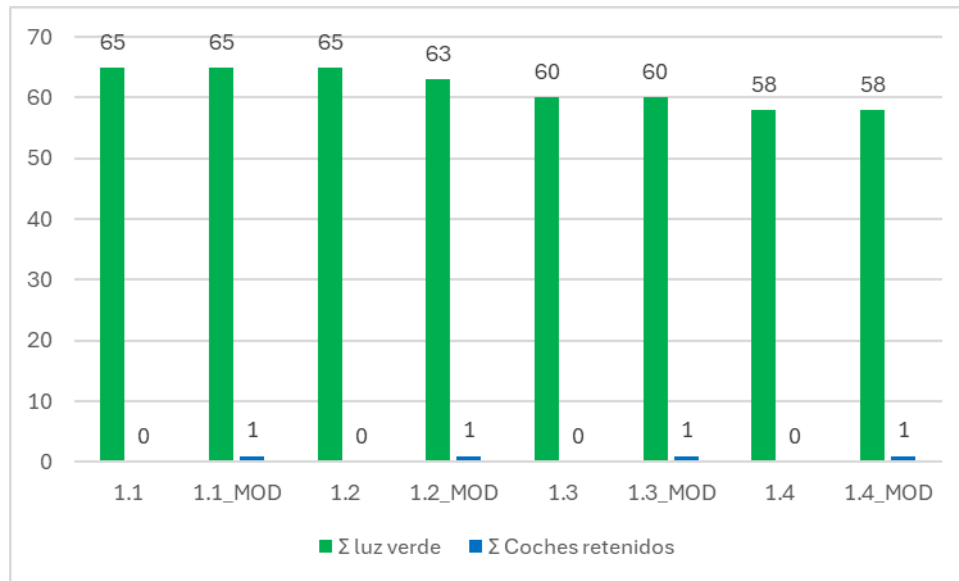


Figura 4-1 Representación resultados del Escenario 1

En la Figura 4-1 se puede observar que la suma de tiempos en verde (en **segundos**) disminuye progresivamente entre escenarios debido al aumento en el tiempo asignado al margen de seguridad (L). Sin embargo, el número de coches retenidos se mantiene prácticamente nulo.

En conclusión, podemos observar que las modificaciones asignadas no afectan de forma significativa al rendimiento del ciclo planteado en el presente apartado.

4.2 Resultados del Escenario 2

A continuación, se exponen los resultados obtenidos en la simulación del segundo escenario, esta vez manteniendo un valor fijo de $L = 25$ y aumentando CT .

Escenario 2	Vector luz verde (seg)	Σg_i	Coches retenidos	Σ Coches retenidos	Valor función objetivo
2.1	28; 23; 6; 6	63	0; 0; 0; 0	0	-1280,2
2.1_MOD	29; 23; 6; 5	63	0; 0; 0; 1	1	-1299,7
2.2	29; 24; 6; 6	65	0; 0; 0; 0	0	-1358

OPTIMIZACIÓN DE TIEMPOS SEMAFORIZADOS EN UN CRUCE
URBANO CON AFLUENCIA MEDIA

2.2_MOD	29; 23; 6; 5	63	0; 0; 0; 1	1	-1329,2
2.3	30; 25; 6; 6	67	0; 0; 0; 0	0	-1438
2.3_MOD	33; 23; 6; 5	67	0; 0; 0; 1	1	-1461
2.4	30; 26; 7; 6	69	0; 0; 0; 0	0	-1499,8
2.4_MOD	35; 23; 6; 5	69	0; 1; 1; 1	3	-1545
2.5	31; 27; 7; 6	71	0; 0; 0; 0	0	-1583,7
2.5_MOD	27; 23; 6; 5	71	0; 2; 1; 1	4	-1631,2
2.6	33; 29; 6; 5	73	0; 0; 1; 1	2	-1711,5
2.6_MOD	39; 23; 6; 5	75	0; 2; 1; 1	4	-1719,6

Tabla 4-2 Resultados obtenidos del Escenario 2

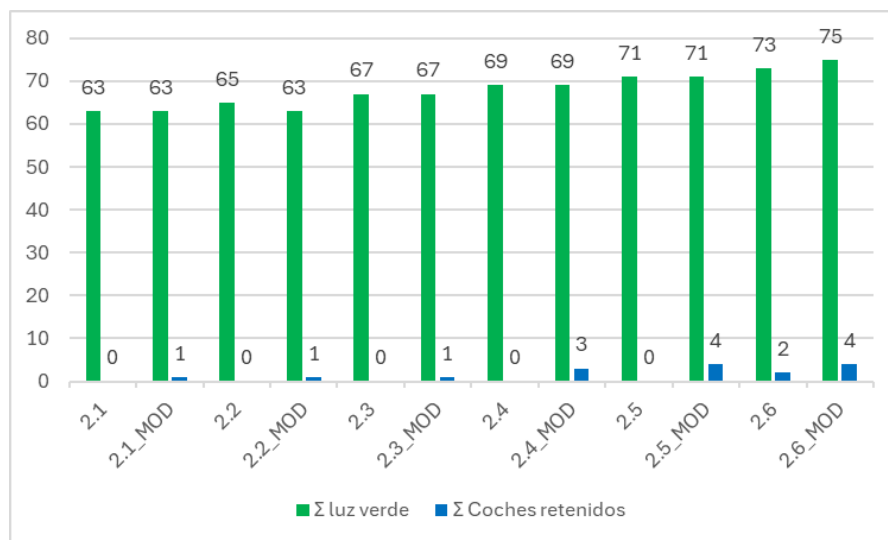


Figura 4-2 Resultados obtenidos del Escenario 2

En la Figura 4-2 se puede observar que a medida que aumenta la suma de tiempos en verde también tiende a incrementarse el número de coches retenidos, especialmente en los últimos casos.

Esto sugiere que ampliar el tiempo total de luz verde no siempre mejora el rendimiento del cruce y puede generar distribuciones menos eficientes del tiempo total del ciclo.

4.3 Resultados del Escenario 3

A continuación, se exponen los resultados obtenidos en la simulación del tercer escenario con flujos aumentados, valores de $CT = 90$ fijos y de L aumentando progresivamente. Cabe mencionar que en el escenario 3.5 no se obtuvo solución factible por lo que se ha omitido.

Escenario 3	Vector luz verde (seg)	Σg_i	Coches retenidos	Σ Coches retenidos	Valor función objetivo	X
3.1	29; 25; 6; 5	65	7; 10; 7; 6	30	-2118,2	1,4433
3.1_MOD	31; 23; 6; 5	65	5; 12; 7; 6	30	-2120,8	1,4433
3.2	28; 24; 6; 5	63	8; 11; 7; 6	32	-2044,5	1,487
3.2_MOD	29; 23; 6; 5	63	7; 12; 7; 6	32	-2045,8	1,487
3.3	26; 23; 6; 6	61	11; 12; 7; 6	36	-1933,2	1,5614
3.3_MOD	26; 23; 6; 5	60	11; 12; 7; 6	36	-1933,2	1,5614
3.4	24; 23; 6; 5	58	13; 12; 7; 6	38	-1858,2	1,6152
3.4_MOD	24; 23; 6; 5	58	13; 12; 7; 6	38	-1858,2	1,6152

Tabla 4-3 Resultados obtenidos del Escenario 3

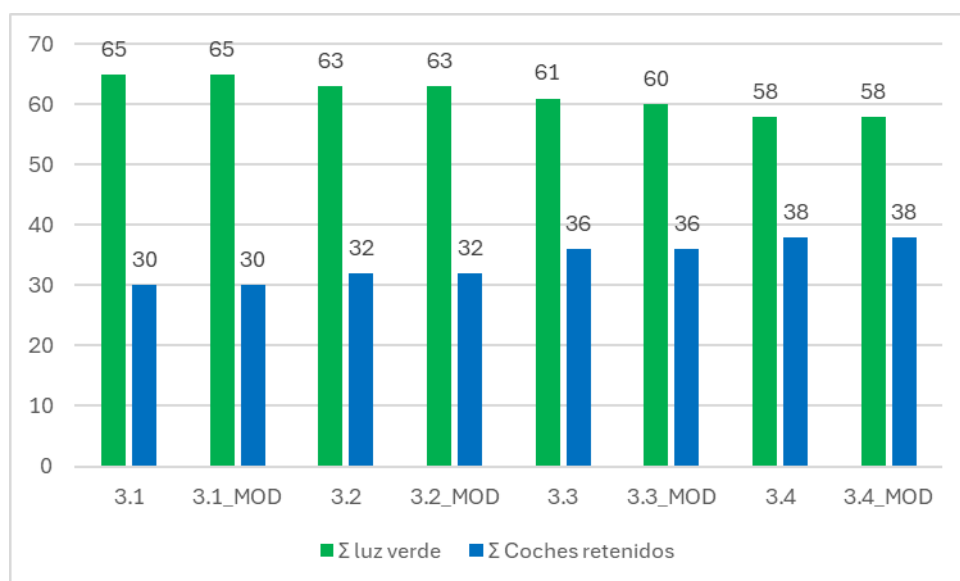


Figura 4-3 Resultados obtenidos del Escenario 3

En la Figura 4-3 se puede observar que a medida que disminuyen los tiempos en verde (debido al aumento en el tiempo de seguridad), aumenta progresivamente el valor de los vehículos retenidos.

Este comportamiento viene motivado por el aumento de la demanda de vehículos en un ciclo en que se mantiene la capacidad.

4.4 Resultados del Escenario 4

A continuación, se exponen los resultados obtenidos en la simulación del cuarto escenario, los cuales corresponden a un modelo de ciclo con los mismos flujos aumentados que en el Escenario 3, pero variando esta vez los valores de CT y manteniendo $L = 25$.

Escenario 4	Vector luz verde (seg)	Σg_i	Coches retenidos	Σ Coches retenidos	Valor función objetivo
4.1	28; 24; 6; 5	63	8; 11; 7; 6	32	-1999,1
4.1_MOD	29; 23; 6; 5	63	7; 12; 7; 6	32	-2000,3
4.2	29; 25; 6; 5	65	7; 10; 7; 6	30	-2118,2
4.2_MOD	31; 23; 6; 5	65	5; 12; 7; 6	30	-2120,8
4.3	30; 26; 6; 5	67	7; 10; 7; 7	31	-2240,7
4.3_MOD	33; 23; 6; 5	67	4; 13; 7; 7	31	-2244,5
4.4	31; 27; 6; 5	69	7; 10; 7; 7	31	-2366,5
4.4_MOD	35; 23; 6; 5	69	3; 14; 7; 7	31	-2371,7
4.5	32; 28; 6; 5	71	7;10; 8; 7	32	-2495,5
4.5_MOD	37; 23; 6; 5	71	1; 15; 8; 7	31	-2502,1
4.6	33; 29; 6; 5	73	6; 9; 8; 7	30	-2627,8
4.6_MOD	39; 23; 6; 5	73	0; 16; 8; 7	31	-2365,9

Tabla 4-4 Resultados obtenidos del Escenario 4

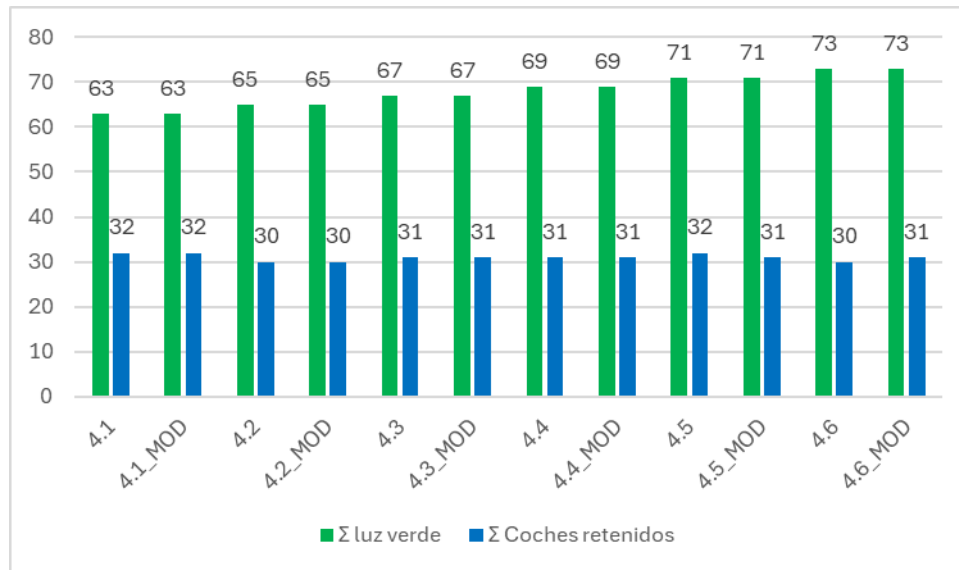


Figura 4-4 Resultados obtenidos del Escenario 4

En la Tabla 4-4 se puede observar que al aumentar el tiempo total en verde a distribuir en el ciclo y manteniendo fijo el tiempo de seguridad, con un aumento de la demanda de vehículos frente a la capacidad máxima, no se reduce de forma significativa los vehículos retenidos, por tanto, podemos concluir que el problema reside en que el ciclo se ve sobrepasado por el aumento del flujo, pese a asignar mayores tiempos de paso.

4.5 Resultados del Escenario 5

A continuación, se exponen los resultados obtenidos en la simulación del quinto escenario, ligado a unos valores de flujo intermedios al resto de escenarios anteriores. Además, se vuelve a mantener los valores de $CT = 90$ y se fueron aumentando los de L . El escenario 5.5 no se obtuvo solución factible por lo que se ha omitido.

Escenario 5	Vector luz verde (seg)	Σg_i	Coches retenidos	Σ Coches retenidos	Valor función objetivo	X
5.1	29; 25; 6; 5	65	2; 5; 4; 4	15	-1820,8	1,2141
5.1_MOD	31; 23; 6; 5	65	0; 7; 4; 4	15	-1823,2	1,2141
5.2	28; 24; 6; 5	63	3; 6; 4; 4	17	-1757	1,2527
5.2_MOD	29; 23; 6; 5	63	2; 7; 4; 4	17	-1758,2	1,2527
5.3	26; 23; 6; 5	60	6; 7; 4; 4	21	-1660,8	1,3153
5.3_MOD	26; 23; 6; 5	60	6; 7; 4; 4	21	-1660,8	1,3153

5.4	24; 23; 6; 5	58	8; 7; 4; 4	23	-1595,8	1,3607
5.4_MOD	24; 23; 6; 5	58	8; 7; 4; 4	23	-1595,8	1,3607

Tabla 4-5 Resultados obtenidos del Escenario 5

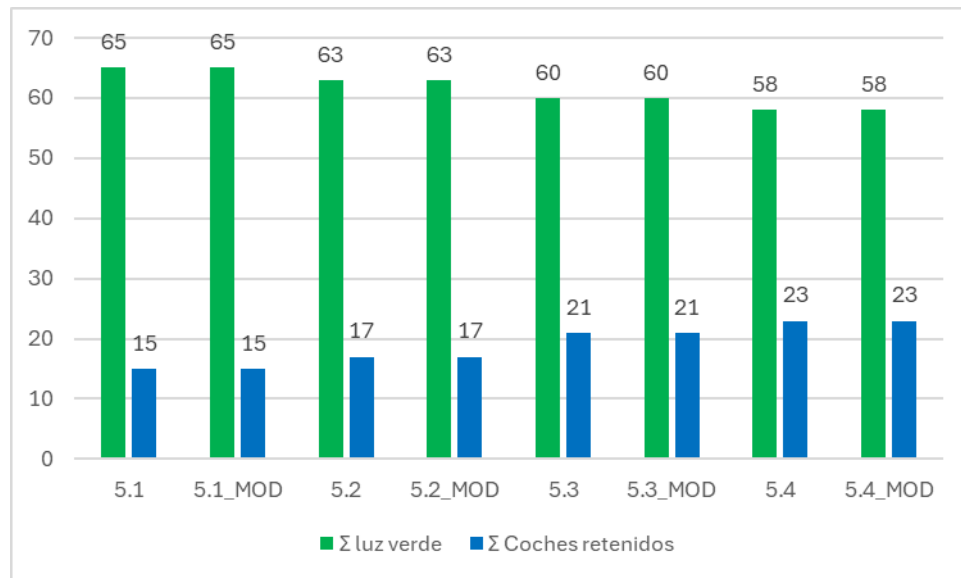


Figura 4-5 Resultados obtenidos del Escenario 5

En la Tabla 4-5 se puede observar que, al igual que ocurría en el Escenario 3, los coches retenidos aumentan ya que se reduce el tiempo de paso en verde para cada movimiento, aunque en menor medida ya que la demanda de partida es ligeramente inferior a al anterior escenario ya citado.

4.6 Resultados del Escenario 6

A continuación, se exponen los resultados obtenidos en la simulación del sexto y último escenario, donde se continúa con los mismos valores de flujo que el Escenario 5 y asignando el valor $L = 25$ a medida que CT aumenta.

Escenario 6	Vector luz verde (seg)	Σg_i	Coches retenidos	Σ Coches retenidos	Valor función objetivo
6.1	28; 24; 6; 5	63	3; 6; 4; 4	17	-1718
6.1_MOD	29; 23; 6; 5	63	2; 7; 4; 4	17	-1719,2
6.2	29; 25; 6; 5	65	2; 5; 4; 4	15	-1820,8
6.2_MOD	31; 23; 6; 5	65	0; 7; 4; 4	15	-1823,2

6.3	30; 26; 6; 5	67	2; 5; 5; 4	16	-1926,4
6.3_MOD	33; 23; 6; 5	67	0; 8; 5; 4	17	-1930,2
6.4	31; 27; 6; 5	69	2; 5; 5; 4	16	-2034,8
6.4_MOD	35; 23; 6; 5	69	0; 9; 5; 4	18	-2040,1
6.5	32; 28; 6; 5	71	1; 4; 5; 4	14	-2146,1
6.5_MOD	37; 23; 6; 5	71	0; 10; 5; 4	19	-2152,8
6.6	33; 29; 6; 5	73	1; 4; 5; 4	14	-2260,3
6.6_MOD	39; 23; 6; 5	73	0; 10; 5; 4	19	-2268,4

Tabla 4-6 Resultados obtenidos del Escenario 6

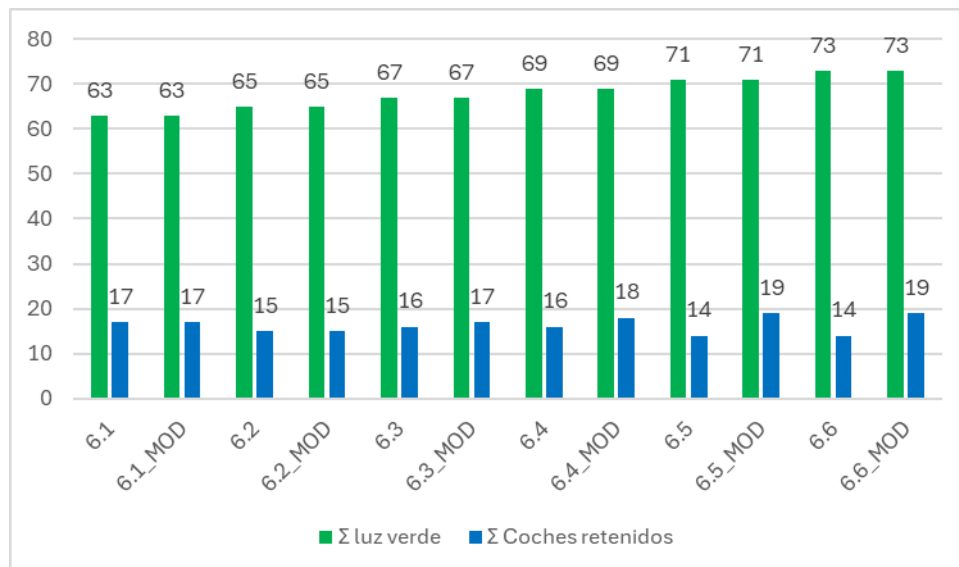


Figura 4-6 Resultados obtenidos del Escenario 6

En la Tabla 4-6 se puede observar que el hecho de que se asigne un mayor tiempo en verde no da lugar a una mayor fluidez del tráfico como podemos observar en las variaciones de vehículos retenidos.

Esto se debe a que el condicionante con más peso es el flujo o demanda asignada al ciclo y un mantenimiento en los niveles de capacidad de cada movimiento o fase.

Por todo lo expuesto en los escenarios, puede observarse que las modificaciones en los valores de los flujos de vehículos tienen una relación directamente proporcional en la cantidad de coches retenidos, ya que cuando el ciclo admite un mayor flujo, los coches retenidos aumentan, y apenas varían los resultados de tiempos en verde.

Por otro lado, se puede observar claramente que en aquellos escenarios en los que el valor de L aumenta y CT queda fijo, el resultado de la función objetivo aumenta de valor, ya que asignamos un mayor tiempo de seguridad. Por el contrario, mientras que CT aumenta dejando fijo L , el valor de la función disminuye debido a una mayor asignación de tiempos al ciclo.

A continuación, la Figura 4-7 muestra una comparación entre los tiempos muertos L versus X , para los escenarios 1,3 y 5.

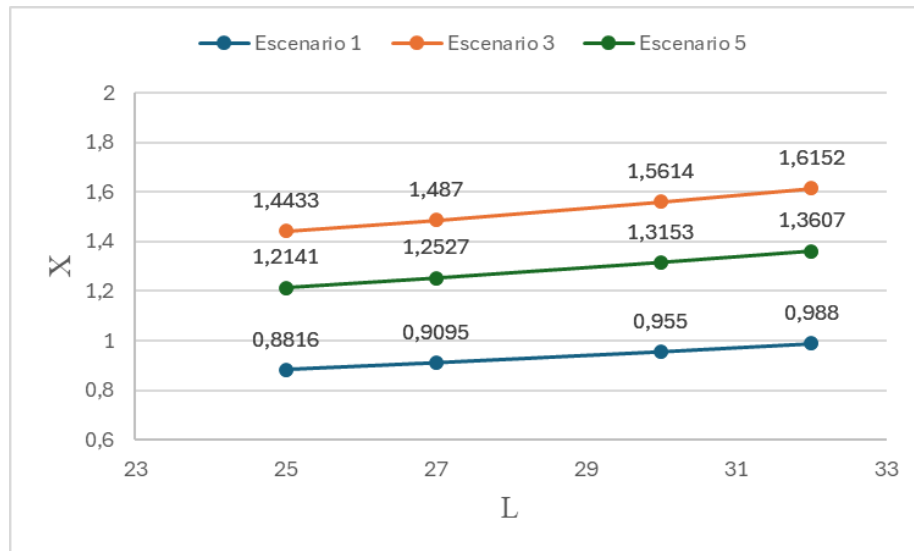


Figura 4-7 Comparativa de los valores X en función de los tiempos muertos aplicados a los distintos escenarios.

Como bien puede observarse, la aparición de variables de retención distintas a cero es consecuencia directa de la sobresaturación del sistema, especialmente las del escenario 3, que corresponde a la situación con mayor demanda de vehículos. Por el contrario, el escenario 1 al tener un menor flujo de vehículos tiene valores de $X < 1$ por lo que en dicho escenario no se experimenta sobresaturación.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

A partir del análisis conjunto de los escenarios planteados, en los que se han considerado tres variaciones del flujo manteniendo constante la capacidad del ciclo, modificaciones en los tiempos máximos del ciclo (CT) y en los tiempos muertos o de seguridad (L), así como la eliminación de la restricción C , se observa que la variación del flujo es el factor que mayor influencia ejerce sobre los resultados obtenidos, especialmente en lo relativo al número de vehículos retenidos. Asimismo, resulta interesante señalar que el aumento del tiempo máximo de ciclo, que en principio podría asociarse a una mayor asignación de tiempos en verde y, por tanto, a una mejora en la fluidez del tráfico, no ha producido ese efecto en los escenarios analizados. Este comportamiento refuerza la conclusión anterior: el flujo de entrada condiciona de forma más decisiva el funcionamiento del sistema que las modificaciones introducidas en el tiempo de ciclo o en los tiempos muertos.

Por otra parte, es muy reducida la diferencia de los resultados obtenidos entre cada escenario y su correspondiente modificado, lo cual no es más que un reflejo de la robustez del código.

El análisis del modelo creado ha demostrado satisfactoriamente que es capaz de distribuir, dentro de unos límites de tiempos máximos y mínimos, los tiempos en verde en función del movimiento del ciclo, con prioridad de unos sobre otros y manteniendo el resto de las restricciones.

En términos generales, se ha podido concluir que las técnicas de programación lineal son una herramienta útil para la optimización de tiempos semafORIZADOS en cruces urbanos de afluencia media. Asimismo, ha permitido analizar el comportamiento de los ciclos dependiendo de qué datos fueran variables o constantes, pudiendo extraer conclusiones acerca de los puntos débiles del programa.

5.2 Líneas futuras

El presente trabajo ha demostrado que mediante la optimización lineal es posible construir un modelo para la gestión de tiempos de un ciclo semafORIZADO y aplicarlo a la vida real con datos de flujos y capacidades estándares en la mayoría de las ciudades promedio.

Sin embargo, las situaciones reales de tráfico urbano nunca son constantes o algo que pueda estandarizarse, lo que matemáticamente se traduce a que muchas de las relaciones entre variables no son estrictamente lineales. Por ello, una posible línea futura de investigación consistiría en abordar el problema mediante **modelos de programación no lineal**, que permitan representar con mayor precisión el comportamiento del tráfico y las interacciones entre los distintos movimientos.

En este contexto, podría incorporarse el uso de **modelos clásicos de ingeniería del tráfico**, como la **fórmula de Webster** (2). La integración de este tipo de formulaciones dentro de un modelo de optimización permitiría comparar los resultados obtenidos mediante programación lineal con los derivados de modelos no lineales más próximos a la realidad del tráfico urbano.

Este cambio de tipo de programación permitiría avanzar hacia un modelo más preciso en términos de retardo, saturación y reparto de tiempos, aunque también implicaría una mayor complejidad matemática y computacional.

Otra posible línea futura de desarrollo consistiría en la aplicación de técnicas de **inteligencia artificial** al control de intersecciones semaforizadas, mediante el empleo de algoritmos heurísticos que permitan, mediante el procesamiento de imágenes y datos en tiempo real, adecuar los tiempos de paso a cada semáforo dependiendo de la densidad vehicular y ajustar por consiguiente los tiempos de espera [29] , representando estos últimos hasta el 10% de los retrasos a nivel mundial. [30]

A diferencia de los modelos de optimización estática, estos enfoques permiten que el sistema aprenda de la evolución del tráfico y adapte continuamente los tiempos de verde en función de variables como la longitud de las colas o el tiempo de espera de los vehículos.

La incorporación de técnicas de inteligencia artificial podría, por tanto, constituir una extensión interesante del modelo desarrollado en este trabajo.

6 BIBLIOGRAFÍA

- [1] ITF, «Descongesting our Cities: Summary and Conclusions,» ITF Roundtable Reports, N° 183, Paris, 2021.
- [2] M. Prados Martín, «Urbanismo y tráfico: La incidencia del automóvil en el desarrollo urbano de Madrid, 1929-1985,» Instituto de Historia (CSIC).
- [3] N. Rodríguez-Martín, «La aparición del "problema de la circulación" y los orígenes de la gestión del tráfico urbano en España, 1900-1936,» Universidad del País Vasco, 2016.
- [4] G. B. Dantzing, «Reminiscences about the origins of linear programming,» United States Government, 1981.
- [5] G. B. Dantzig, «Linear Programming and Extensions,» USAF Project RAND, 1963.
- [6] M. Prados Martín, «La modernización durante el franquismo: urbanismo, tráfico y descontento social en Madrid 1957-1973,» Instituto de Historia (CSIC), 2023.
- [7] J. Moral-Carcedo, «Predicting traffic intensity in the urban area of Madrid: Integrating route network topology into a machine-learning model,» ScienceDirect, Madrid, 2024.
- [8] P. Christidis, M. Vega Gonzalo, G. Ulpiani y N. Vettters, «Post-pandemic trends in urban mobility,» Publications Office of the European Union, Luxembourg, 2023.
- [9] «RISK LEVEL; Nationwide power blackouts and associated disruptions,» 2025. [En línea]. Available: <https://riskline.com/>. [Último acceso: 14 Enero 2026].
- [10] «Factor Energético,» Redacción Factor Energético, 28 Abril 2025. [En línea]. Available: <https://factorenergetico.mx/apagon-electrico-en-espana-portugal-y-parte-de-francia/>. [Último acceso: 2026 Enero 14].
- [11] G. Vilches Mendoza, «Modelos matemáticos basados en ecuaciones diferenciales y su aplicación a la economía,» Universidad de Jaén, Jaén, España, 2019.
- [12] A. Ramos y B. Vitoriano, «Modelos matemáticos de simulación,» Universidad Pontificia de Comillas, Madrid.

- [13] P. Nicole Roldán, «Economipedia,» 1 Agosto 2025. [En línea]. Available: <https://economipedia.com/definiciones/modelo-matematico.html>. [Último acceso: 28 Enero 2026].
- [14] «Tema 4. Modelos y optimización. Universidad de Murcia.,» [En línea]. Available: <https://www.um.es/documents/4874468/10375216/msa-cap-04.pdf/7e53a0b7-fd27-4b11-ae0b-c26edbbf8c1e>.
- [15] D. H. Arsham, «Modelos Deterministas: Optimización Lineal,» 2020.
- [16] «Wolfram MathWorld,» Linear Programming, [En línea]. Available: <https://mathworld.wolfram.com/LinearProgramming.html>. [Último acceso: 16 Enero 2026].
- [17] R. Cottle, E. Johnson y R. Wets, George B. Dantzig (1914–2005), American Mathematical Society.
- [18] M. d. C. Hernández Ayuso, Introducción a la Programación Lineal, Facultad de Ciencias, UNAM, 2007.
- [19] M. S. Bazaraa y J. J. Jarvis, «Linear Programming and Network Flows,» 1977.
- [20] «Mathworks,» [En línea]. Available: https://es.mathworks.com/help/optim/ug/linprog.html#buus4rk-1_1. [Último acceso: 2 Febrero 2026].
- [21] «Informs. Smarter Decisions for a Better World.,» 13 Mayo 2005. [En línea]. Available: <https://www.informs.org/Explore/History-of-O.R.-Excellence/Biographical-Profiles/Dantzig-George-B.> [Último acceso: 6 Febrero 2026].
- [22] F. S. Hillier y G. J. Lieberman, «Introduction to operations research,» 1957.
- [23] «Tema 2. El método Simplex,» Universidad del País Vasco, País Vasco.
- [24] «Capítulo 3. Método del Simplex para la programación lineal,» Universidad de La Rioja.
- [25] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Simplex_algorithm. [Último acceso: 6 Febrero 2026].
- [26] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/MATLAB>. [Último acceso: 6 Febrero 2026].
- [27] «Simulink Getting Started Guide,» MathWorks.
- [28] «Mathworks,» [En línea]. Available: <https://es.mathworks.com/help/optim/ug/linprog.html>. [Último acceso: 6 Febrero 2026].
- [29] Y. I. Apestegui Florentino y H. E. Caselli Gismondi, «Sistema Inteligente de Gestión de Semáforos en Tiempo Real,» Revista Ibérica de Sistemas y Tecnologías de Información, Perú, 2024.
- [30] A. Aghahari, M. Dhabu, P. Deshpande, A. Tiwari, M. Baig y A. Sawarkar, «Artificial Intelligence-Based Adaptive Traffic Signal Control System: A Comprehensive Review,» Multidisciplinary Digital Publishing Institute, Basel, 2024.
- [31] OpenAI, «ChatGPT (modelo GPT-5.3),» OpenAI, 2026. [En línea]. Available: <https://chat.openai.com/>. [Último acceso: 24 Marzo 2026].

ANEXO I: IMPLICACIONES SOCIALES, ECONÓMICAS Y AMBIENTALES

El desarrollo de modelos matemáticos aplicados a la gestión del tráfico urbano tiene implicaciones relevantes en distintos ámbitos sociales, económicos y ambientales. En primer lugar, desde el punto de vista social, la optimización de los tiempos semaforizados contribuye a mejorar la movilidad en las ciudades, reduciendo los tiempos de espera en los cruces y favoreciendo una circulación más fluida. Esto repercute directamente en la calidad de vida de los ciudadanos, al disminuir el estrés asociado a la congestión del tráfico y facilitar desplazamientos más eficientes tanto para vehículos privados como para servicios públicos o de emergencia.

En el ámbito económico, una gestión más eficiente del tráfico urbano puede traducirse en una reducción de costes asociados al transporte. La disminución de los tiempos de espera en intersecciones reduce el consumo de combustible de los vehículos, así como el tiempo empleado en desplazamientos diarios. Esto puede tener un impacto positivo en la productividad y en el funcionamiento general de las actividades económicas dentro de las ciudades, especialmente en aquellas relacionadas con la logística, el comercio y los servicios.

Desde una perspectiva ambiental, la optimización de los ciclos semaforicos puede contribuir a disminuir las emisiones contaminantes generadas por los vehículos. La reducción de paradas innecesarias y de aceleraciones repetidas permite limitar el consumo de combustible y, en consecuencia, las emisiones de dióxido de carbono (CO_2), óxidos de nitrógeno (NO_x) y partículas contaminantes. En este sentido, la aplicación de modelos matemáticos para mejorar la regulación del tráfico se alinea con las políticas actuales orientadas hacia una movilidad urbana más sostenible.

Por último, desde el punto de vista de la seguridad vial, una adecuada planificación de los tiempos semaforicos contribuye a ordenar los flujos de tráfico y reducir la probabilidad de conflictos entre vehículos, peatones y otros usuarios de la vía. De este modo, el uso de herramientas de optimización en la gestión del tráfico urbano no solo mejora la eficiencia del sistema, sino que también contribuye al desarrollo de entornos urbanos más seguros y sostenibles.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

El desarrollo de herramientas de optimización aplicadas a la gestión del tráfico urbano plantea también diversas consideraciones de carácter ético y social que deben ser tenidas en cuenta en el ámbito de la ingeniería. En primer lugar, es importante reconocer que las decisiones relacionadas con la planificación y regulación del tráfico afectan directamente a la vida cotidiana de los ciudadanos. Por ello, los modelos matemáticos utilizados deben ser diseñados con el objetivo de mejorar el bienestar colectivo, priorizando soluciones que favorezcan la movilidad de manera equitativa y segura para todos los usuarios de la vía.

Desde una perspectiva ética, el ingeniero tiene la responsabilidad de desarrollar soluciones que no solo sean técnicamente eficientes, sino también socialmente responsables. En el caso de la optimización de tiempos semafóricos, esto implica considerar no únicamente la eficiencia del tráfico vehicular, sino también la seguridad de peatones, ciclistas y otros usuarios vulnerables. La ingeniería del transporte debe buscar un equilibrio entre la fluidez del tráfico y la protección de los distintos actores que comparten el espacio urbano.

Asimismo, la utilización de modelos matemáticos y herramientas computacionales implica asumir ciertas limitaciones inherentes a la simplificación de la realidad. Los modelos desarrollados, como el presentado en este trabajo, representan aproximaciones que permiten analizar el comportamiento del sistema bajo determinadas hipótesis. Por ello, resulta fundamental interpretar los resultados con espíritu crítico y comprender que su aplicación práctica requiere siempre un análisis complementario basado en la observación real del entorno urbano.

Por último, el uso de métodos de optimización en la gestión del tráfico puede contribuir a avanzar hacia modelos de movilidad más sostenibles y eficientes. En este sentido, el trabajo desarrollado se enmarca dentro de una tendencia creciente en la ingeniería contemporánea que busca aplicar herramientas matemáticas y tecnológicas para afrontar problemas complejos de la sociedad actual, como la congestión urbana, la contaminación o la gestión eficiente de recursos limitados.

ANEXO III: CÓDIGO DEL PROGRAMA

```
% MODELOS P1 Y P2
%Definimos los flujos (ff) y capacidad (q)
% % ff=[1200; 1000; 350; 150];
% % q=[3900; 3500; 1100; 1000];
ff=[1000;950;220;200];
%ff=[1500;1450;480;420];
%ff=[1300;1250;380;320];
q=[3800;3800;3400;3400];
%Definimos duracion total del ciclo y tiempos muertos
CT=90;
L =25; %tiempo rojos y amarillos del ciclo
k=1;
alpa=1;
% bet1=0.9;
% bet2=0.7;
M=100;
N=4; %Número de fases del semáforo
n=1*ones(4,1);
%Definimos las cotas de tiempos máximos y mínimos para el ciclo
Tmax=[40; 40; 20; 20];
Tmin =[24; 23; 6; 5 ];
%tmax=sum(Tmax);
tmin=min(Tmin);
if tmin > CT-L
    display('reduzca un poco las cotas inferiores, ya que su suma es mayor que CT-L')
end
ff=(1/3600)*ff;
qq=(1/3600)*q;
A1=[-qq(1) 0 0 0; 0 -qq(2) 0 0; 0 0 -qq(3) 0; 0 0 0 -qq(4)];
b1=-alpa*CT*ff;
%A2=[-bet1 1 0 0 0 0 0 0; 0 0 -bet2 1 0 0 0 0];
%b2=[0;0];
%A=[A1;A2];
%b=[b1;b2];
A=A1;
b=b1;
%ub=Tmax;
ub=Tmax;
lb=Tmin
Aeq=[1 1 1 1];
beq=CT-L;
%Definimos función objetivo
f=-k*CT*ff;
%f=-diag(n)*ff;
for i=1:N
    f1(i)=(sum(ff)-ff(i))-M;
end
%f=f1';
f=f
incont=1:4;
%[g,fval,exitflag,output,lambda]=linprog(f,A,b,Aeq,beq,lb,ub)
[g,fval,exitflag,output]=intlinprog(f,incont,A,b,Aeq,beq,lb,ub)
%[g,fval,exitflag,output,lambda]=linprog(f,A,b,Aeq,beq,lb,ub,options)
display('El tiempo en segundos en verde del mov. N-S')
g(1)
display('El tiempo en segundos en verde del mov. E-O')
```

```
g(2)
display('El tiempo en segundos en verde del giro izq. N-S')
g(3)
display('El tiempo en segundos en verde del giro izq. E-O')
g(4)
for i=1:N
    s(i)=qq(i)*g(i)-alpa*ff(i)*g(i);
    s1(i)=qq(i)*g(i)-ff(i)*g(i);
end
display('el vetor de holguras de la capacidad con la relajación (alpha <1)')
s
display('el vetor de holguras de la capacidad sin la relajación (alpha =1)')
s1
if s1 >= zeros(4,1)
    display('Las restricciones de capacidad se satisfacen sin la relajación')
end

for i=1:N
    rojo(i)=sum(g)-g(i);
end
display('El vector de tiempos en rojo de las cuatro fases, viene dado por')
rojo

display('la suma de los tiempos en verde es')
sum(g)
display('el valor máximo de luz verde es')
CT-L
display('El valor de la función objetivo')
fval
F=diag(ff)
for i=1:N
    Q(i)=1/qq(i);
end
FQ=F*Q'
X=(sum(FQ))/(1-(L/CT));
display('el valor del grado de saturación es')
X
```

% MODELO P3

```
%Definimos los flujos (ff) y capacidad (q)
% % ff=[1200; 1000; 350; 150];
% % q=[3900; 3500; 1100; 1000];
ff=[1300;1250;380;320];
%ff=[1500;1450;480;420];
%ff=[1300;1250;380;320];
q=[3800;3800;3400;3400];
%Definimos duración total del ciclo y tiempos muertos
CT=90;
%L=25; %tiempos rojos y amarillos muertos
L=25;
dif1 = 5;
dif2 = 2;
k=1;
alpa=1;
% el vector delta debe contener valores menores que 2 mayores e iguales que
% 1
N=4; %Número de fases del semáforo
n=1*ones(4,1);
```

```

M=100;
%Definimos las cotas de tiempos máximos y mínimos para el ciclo
% Tmax=[25; 25; 13; 10];
% Tmin =[10; 10; 6; 5];
Tmax=[40; 40; 20; 20];
Tmin =[24; 23; 6; 5];
% tmax=sum(Tmax);
tmin=min(Tmin);
if tmin > CT-L
    display('reduzca un poco las cotas inferiores, ya que su suma es mayor que CT-L')
end
ff=(1/3600)*ff;
qq=(1/3600)*q;
A1=[-qq(1) 0 0 0 -1 0 0 0 ; 0 -qq(2) 0 0 0 -1 0 0; 0 0 -qq(3) 0 0 0 -1 0; 0 0 0 -qq(4) 0 0
0 -1];
b1=-alpha*CT*ff;
A2=[1 -1 0 0 0 0 0 0; 0 0 1 -1 0 0 0 0; -1 1 0 0 0 0 0 0; 0 0 -1 1 0 0 0 0];
b2=[dif1; dif2;0;0];
A=[A1;A2];
b=[b1;b2];
A=A1;
b=b1;
%ub=Tmax;
ub=[Tmax; inf; inf; inf; inf];
lb=[Tmin; zeros(N,1)];
%Aeq=[1 1 1 1 zeros(1,N)];
%beq=CT-L;
%incont=1:4;
incont=1:2*N;
%Definimos funciones objetivo
z1=[-k*CT*ff;M*ones(N,1)]; %función demoras
z1_mod=[-k*CT*ff; zeros(N,1)]; %función demoras
for i=1:N
    f1(i)=(sum(ff)-ff(i))-M;
end
z2=[f1'; zeros(N,1)]; %función luces rojas y verdes
f=z1_mod;
%f=z1
[g,fval,exitflag,output,lambda]=linprog(f,A,b,Aeq,beq,lb,ub)
[g,fval,exitflag,output]=intlinprog(f,incont,A,b,Aeq,beq,lb,ub)
display('El tiempo en segundos en verde del mov. N-S')
g(1)
display('El tiempo en segundos en verde del mov. E-O')
g(2)
display('El tiempo en segundos en verde del giro izq. N-S')
g(3)
display('El tiempo en segundos en verde del giro izq. E-O')
g(4)
for i=1:N
    s(i)=qq(i)*g(i)+g(i+N)-ff(i)*CT;
end
display('Las restricciones de capacidad contabilizando coches retenidos')
s
for i=1:N
    rojo(i)=(sum(g(1:4,1))-g(i));
end
display('El vector de tiempos en rojo de las cuatro fases, viene dado por')
rojo
rela(1)=g(1)-g(2);
rela(2)=g(3)-g(4);

```

```
display('El vector resta de las relaciones entre variables ')
rela
if rela < zeros(2,1)
    display('las relaciones entre la duración de los tiempos en verde de las fases
no es la deseada')
else
    display('la relación de los tiempos verdes es la deseada')
end
% fnew= f(1:4,1);
% display('las demoras para este ciclo son')
% demora= -fnew'*g(1:4,1)
display('El valor de la función objetivo')
fval
% display('el valor de la penalización es')
% fval+demora
display('la suma de los tiempos en verde es')
sum(g(1:4,1))
F=diag(ff)
for i=1:N
    Q(i)=1/qq(i);
    UQ(i)=g(4+i)/(CT*q(i));
end
UQ
FQ=F*Q'
X=(sum(FQ))/(1-(L/CT));
X1=(sum(FQ)-sum(UQ))/(1-(L/CT));
display('el valor del grado de saturación es')
X
display('el valor de X1')
X1
```

ANEXO IV: DECLARACIÓN DE USO DE INTELIGENCIA ARTIFICIAL

A lo largo de la realización del presente Trabajo de Fin de Grado, se ha hecho uso puntual y ético de herramientas de inteligencia artificial generativa, en concreto ChatGPT, como apoyo en tareas de corrección y mejora de la claridad en la redacción.

El uso de esta herramienta ha tenido un carácter estrictamente complementario, con el objetivo de mejorar la coherencia y cohesión de determinados párrafos en distintos apartados, así como de servir de apoyo en la representación visual de uno de los escenarios del trabajo (Figura 3-1). En ningún caso ha supuesto una sustitución del trabajo intelectual del autor, ni ha intervenido en la realización de los experimentos, la interpretación de los resultados o la elaboración de las conclusiones.