



Universidad de Castilla-La Mancha
Escuela Superior de Ingeniería Informática

Trabajo Fin de Grado
Grado en Ingeniería Informática
Tecnología Específica de
Computación

Optimización de la asignación diaria de pedidos a flota propia y mensajería externa

Enrique Álvarez Álvarez

Febrero, 2026



Trabajo Fin de Grado
Grado en Ingeniería Informática
Tecnología Específica de
Computación

Optimización de la asignación diaria de
pedidos a flota propia y mensajería externa

Autor: Enrique Álvarez Álvarez

Tutor: Francisco Parreño Torres

Febrero, 2026

Dedicado a mi familia.

Declaración de Autoría

Yo, Enrique Álvarez Álvarez, con DNI 76581301H, declaro que soy el único autor del trabajo fin de grado titulado “Optimización de la asignación diaria de pedidos a flota propia y mensajería externa” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 9 de febrero de 2026

Fdo: Enrique Álvarez Álvarez

Resumen

Este Trabajo Fin de Grado aborda el problema de la planificación diaria del transporte en una empresa logística, centrado en la asignación eficiente de pedidos entre flota propia de camiones y servicios de mensajería externa. Dicho proceso se realiza habitualmente de forma manual, lo que presenta importantes limitaciones en términos de reproducibilidad, eficiencia económica y escalabilidad.

Con el objetivo de mejorar esta situación, el problema se formula como un modelo de Programación Lineal Entera Mixta, capaz de representar decisiones discretas, costes fijos y variables, restricciones de capacidad, cantidad de mercancías peligrosas (ADR) y un mecanismo de incentivo para la anticipación de pedidos futuros. El modelo se implementa mediante Pyomo y se integra en un entorno accesible basado en Microsoft Excel utilizando SolverStudio, lo que facilita su utilización en contextos empresariales reales.

La validación del modelo se realiza a través de experimentos computacionales sobre instancias sintéticas de distinto tamaño, analizando tanto la coherencia operativa de las decisiones como el rendimiento computacional del *solver* de código abierto CBC. Los resultados muestran que el modelo resuelve adecuadamente el dilema *make-or-buy* y es capaz de proporcionar soluciones de alta calidad en tiempos compatibles con la operativa diaria, incluso cuando no se alcanza el óptimo matemático.

En conjunto, el trabajo demuestra el potencial de las técnicas de Investigación Operativa como herramienta de apoyo a la toma de decisiones en entornos logísticos reales, ofreciendo una solución flexible, extensible y fácilmente integrable en sistemas existentes.

Índice general

Capítulo 1	Introducción	1
1.1	Introducción	1
1.2	Objetivos	2
1.3	Estructura del proyecto	3
Capítulo 2	Estado del Arte	4
2.1	Introducción	4
2.2	Problemas logísticos relacionados	5
2.2.1	Rutas de vehículos (<i>Vehicle Routing Problem</i> , VRP)	5
2.2.2	Problemas de asignación	6
2.2.3	Problemas de transporte de mercancías peligrosas (ADR)	7
2.2.4	Decisiones <i>Make-or-Buy</i> / Flota propia vs externalización	8
2.3	Métodos y técnicas de resolución	9
2.3.1	Métodos Exactos	9
2.3.2	Heurísticas	10
2.3.3	Metaheurísticas	11
2.4	Herramientas <i>software</i>	12
2.5	Conclusiones	14
Capítulo 3	Metodología y Desarrollo	16
3.1	Introducción	16
3.2	Implementación del modelo de planificación de transporte	16
3.2.1	Contexto operativo y definición del problema	16
3.2.2	Formulación matemática del modelo base	17
3.2.3	Preparación de datos en Excel	21

3.2.4	Definir <i>Data Items</i> en SolverStudio	22
3.2.5	Escribir <i>script</i> Pyomo en SolverStudio	23
3.2.6	Ejecución del modelo	23
3.2.7	Verificación y validación	25
Capítulo 4	Experimentos y Resultados	27
4.1	Objetivo y diseño experimental	27
4.1.1	Diseño del experimento y robustez estadística	28
4.2	Generación de escenarios de prueba	29
4.2.1	Implementación del generador de instancias	29
4.2.2	Perfiles de escenarios definidos	30
4.2.3	Repositorio de datos y código abierto	31
4.3	Análisis del incentivo por anticipación de envío	31
4.3.1	Rendimiento computacional	32
4.3.2	Validación de la lógica operativa	32
4.3.3	Comportamiento del incentivo por anticipación	33
4.3.4	Síntesis de resultados	34
4.4	Análisis económico: decisión <i>Make-or-Buy</i>	34
4.4.1	Activación de flota y comparación de costes	34
4.4.2	Sensibilidad del modelo a la tipología de carga	35
4.4.3	Síntesis del análisis <i>Make-or-Buy</i>	38
4.5	Escalabilidad y punto de inflexión computacional	38
4.5.1	Escalabilidad con el número de pedidos y camiones	38
4.5.2	Punto de inflexión computacional	39
4.5.3	Análisis coste–tiempo y soluciones cuasi-óptimas	40
4.6	Resolución bajo límite de tiempo y soluciones cuasi-óptimas	42
4.6.1	Configuración experimental	42
4.6.2	Resultados y análisis de convergencia	43
4.6.3	Patrones de convergencia	44
4.6.4	Estabilidad estructural de las soluciones	45
4.7	Síntesis global de los resultados	45
Capítulo 5	Conclusiones y Trabajo Futuro	47
5.1	Conclusiones	47
5.2	Uso de herramientas de inteligencia artificial	48

5.3	Competencias adquiridas	48
5.4	Trabajo futuro	49
Bibliografía		51
Anexo I. Implementación del modelo de optimización en Pyomo		53
I.1	Visión general	53
I.2	Declaración del modelo abstracto	53
I.3	Conjuntos	53
I.4	Parámetros	54
I.5	Parámetros derivados	54
I.6	Variables	55
I.7	Función Objetivo	55
I.8	Restricciones	56

Índice de ilustraciones

Ilustración 1. Estructura del archivo Excel	21
Ilustración 2. Interfaz de SolverStudio en Excel	22
Ilustración 3. Hoja Resultados del archivo Excel con la asignación de pedidos a camiones y mensajería.	24
Ilustración 4. Distribución del Coste Operativo según Tipología de Carga.....	36
Ilustración 5. Análisis de Escalabilidad (400 pedidos): Distribución del Coste Operativo (Cajas) vs. Tiempo de Resolución (Línea)	41
Ilustración 6. Convergencia Coste vs Tiempo.....	44

Índice de tablas

Tabla 1. Resultados del modelo en instancias pequeñas y medias (n=30).....	32
Tabla 2. Impacto progresivo de la densidad de ADR (n=30).....	37
Tabla 3. Escalabilidad del modelo en escenarios grandes (Límite 300s, n=30)	39
Tabla 4. Evolución de la calidad de la solución según tiempo de cómputo (n=30).....	42

Capítulo 1

Introducción

1.1 Introducción

La planificación del transporte constituye una de las actividades clave en la gestión logística de cualquier empresa de distribución. Una asignación ineficiente de los pedidos a los recursos disponibles puede derivar en un incremento significativo de los costes operativos, un uso inadecuado de la flota propia y una dependencia excesiva de servicios de transporte externalizados. Estas decisiones adquieren una especial relevancia en entornos reales, donde coexisten múltiples restricciones operativas como capacidades de carga, limitaciones normativas, diversidad de tipologías de pedidos y plazos de entrega ajustados.

En la práctica, la planificación diaria del transporte suele realizarse de forma manual o semimanual, apoyándose en la experiencia del responsable logístico. Aunque este enfoque puede resultar válido en escenarios simples, presenta importantes limitaciones cuando el volumen de pedidos aumenta o cuando se introducen restricciones adicionales, como la gestión de mercancías peligrosas o la necesidad de equilibrar flota propia y mensajería externa. En estos casos, la toma de decisiones basada únicamente en criterios heurísticos puede conducir a soluciones subóptimas desde el punto de vista económico.

La Investigación Operativa ofrece un marco metodológico sólido para abordar este tipo de problemas mediante modelos matemáticos de optimización. En particular, la Programación Lineal Entera Mixta permite formular de manera rigurosa decisiones mediante el uso de variables discretas y continuas, incorporando restricciones reales del sistema y persiguiendo la minimización del coste total. No obstante, la aplicación práctica de estos modelos requiere un equilibrio entre la calidad de la solución y el tiempo de cómputo disponible, especialmente en contextos operativos diarios.

En este Trabajo Fin de Grado se aborda el problema de la planificación diaria del transporte mediante el desarrollo de un modelo de optimización que decide la asignación de pedidos a flota propia o a mensajería externa. El modelo tiene en cuenta restricciones reales de capacidad, compatibilidad de carga y costes, y se implementa utilizando herramientas de código abierto, con el objetivo de ofrecer una solución aplicable en entornos empresariales reales.

1.2 Objetivos

El objetivo principal de este trabajo es **desarrollar y validar un modelo de optimización para la planificación diaria del transporte**, que permita asignar pedidos de forma eficiente entre flota propia y mensajería externa, minimizando el coste operativo total bajo un conjunto de restricciones realistas.

De forma más específica, se persiguen los siguientes objetivos:

- Formular el problema de planificación del transporte como un modelo de Programación Lineal Entera Mixta, incorporando decisiones binarias de asignación y activación de flota.
- Integrar restricciones operativas reales, como capacidades de volumen y peso, limitaciones en el número de paradas y gestión de mercancías peligrosas (ADR).
- Incorporar un mecanismo que permita la anticipación controlada de pedidos futuros cuando exista capacidad ociosa, sin comprometer el envío de los pedidos obligatorios del día.
- Implementar el modelo utilizando herramientas de código abierto, combinando lenguajes de modelado y *solvers* accesibles.

- Evaluar el comportamiento del modelo mediante experimentos computacionales sobre datos sintéticos, analizando su coherencia operativa, rendimiento computacional y escalabilidad.
- Analizar el compromiso entre calidad de solución y tiempo de resolución, valorando la viabilidad de soluciones cuasi-óptimas en entornos operativos reales.

1.3 Estructura del proyecto

El resto del documento se organiza de la siguiente manera:

- En el **Capítulo 2** se presenta el estado del arte, revisando los principales problemas logísticos relacionados con la planificación del transporte, así como los métodos de resolución y herramientas software empleadas en la literatura.
- El **Capítulo 3** describe la metodología seguida y el desarrollo del modelo propuesto, incluyendo la formulación matemática, la preparación de los datos y la implementación computacional.
- En el **Capítulo 4** se analizan los experimentos computacionales realizados y se discuten los resultados obtenidos, evaluando el comportamiento del modelo en distintos escenarios y tamaños de instancia.
- Finalmente, el **Capítulo 5** recoge las conclusiones del trabajo, resume las principales aportaciones y propone posibles líneas de trabajo futuro.

Capítulo 2

Estado del Arte

2.1 Introducción

En este capítulo se revisan los principales enfoques, modelos y herramientas empleados en la optimización de sistemas logísticos y de transporte. El objetivo es establecer una base teórica que permita contextualizar el problema de planificación diaria del transporte abordado en este trabajo, identificando los tipos de problemas relacionados, las técnicas de resolución más habituales y las herramientas software utilizadas en la práctica.

La revisión se centra especialmente en problemas de asignación de pedidos, decisiones de utilización de flota propia frente a externalización y gestión de restricciones operativas reales, como las asociadas al transporte de mercancías peligrosas. Asimismo, se analizan los métodos de resolución exactos y aproximados propuestos en la literatura, así como las herramientas disponibles para su implementación.

Este estado del arte permite situar el modelo propuesto dentro del contexto de la investigación existente y justificar las decisiones metodológicas adoptadas en los capítulos posteriores.

2.2 Problemas logísticos relacionados

En este apartado se revisan los principales problemas logísticos relacionados con la planificación del transporte y la asignación de pedidos, con el objetivo de contextualizar el modelo propuesto dentro de la literatura existente. En particular, se analizan los problemas de rutas de vehículos, los problemas de asignación y la gestión del transporte de mercancías peligrosas, así como las decisiones de utilización de flota propia frente a externalización.

El propósito de esta revisión no es abordar exhaustivamente cada uno de estos problemas, sino identificar aquellos enfoques y conceptos que resultan relevantes para la planificación diaria del transporte en empresas logísticas, donde debe decidirse cómo asignar pedidos a recursos disponibles bajo restricciones operativas reales.

2.2.1 Rutas de vehículos (*Vehicle Routing Problem*, VRP)

El problema de rutas de vehículos (*Vehicle Routing Problem*, VRP) es uno de los problemas clásicos de la optimización combinatoria en logística y transporte. Su objetivo es determinar el conjunto óptimo de rutas que debe seguir una flota de vehículos para atender las demandas de un conjunto de clientes, minimizando un determinado coste operativo, como la distancia total recorrida, el tiempo o el coste económico, y cumpliendo diversas restricciones.

El VRP puede considerarse una generalización del conocido Problema del Viajante (*Travelling Salesman Problem*, TSP), y fue formulado por primera vez por Dantzig y Ramser en 1959 en el contexto de la planificación de la distribución de gasolina [1]. Desde entonces, se ha convertido en un problema ampliamente estudiado debido a su elevada complejidad computacional, ya que pertenece a la clase de problemas NP-hard. Como consecuencia, la resolución exacta de instancias de gran tamaño resulta computacionalmente costosa, lo que ha motivado el desarrollo de numerosos métodos aproximados.

No obstante, el problema abordado en este trabajo no pretende resolver un VRP completo con optimización explícita de rutas. En su lugar, se centra en una planificación de nivel táctico, donde el objetivo principal es la asignación de pedidos a vehículos disponibles o a mensajería externa, bajo restricciones de capacidad, compatibilidad de carga y número máximo de paradas. En este sentido, el VRP se considera como un marco

de referencia teórico que permite contextualizar la problemática, pero el modelo propuesto adopta una formulación más simple y adecuada para la planificación diaria del transporte.

En los apartados siguientes se revisan otros problemas logísticos relacionados, como los problemas de asignación y el transporte de mercancías peligrosas, que resultan más directamente alineados con la estructura y los objetivos del modelo desarrollado.

2.2.2 Problemas de asignación

El problema de asignación (PA) es un problema clásico de optimización combinatoria que consiste en asignar un conjunto de recursos a un conjunto de tareas de forma óptima, minimizando el coste total o maximizando una medida de eficiencia. En su formulación más simple, se busca emparejar n agentes con n tareas, de manera que cada agente realice exactamente una tarea y cada tarea sea asignada a un único agente. Tradicionalmente, el problema de asignación puede resolverse mediante Programación Lineal Entera o mediante algoritmos específicos, como el Algoritmo Húngaro propuesto por Kuhn en 1955, que permite obtener soluciones óptimas en tiempo polinómico $O(n^3)$ [2]. Desde el punto de vista estructural, los problemas de asignación presentan similitudes con los problemas de transporte, diferenciándose en que la oferta de cada origen y la demanda de cada destino son unitarias.

En contextos logísticos reales, el problema de asignación suele extenderse para incorporar costes heterogéneos y restricciones adicionales, como capacidades, compatibilidades o limitaciones operativas. En estos casos, resulta habitual formular el problema mediante modelos de Programación Lineal Entera o Programación Lineal Entera Mixta, que permiten representar de forma flexible dichas restricciones.

El problema abordado en este trabajo puede interpretarse como una extensión del problema de asignación, en la que se decide diariamente la asignación de pedidos a recursos logísticos alternativos (flota propia o mensajería externa), considerando costes, capacidades de los vehículos y restricciones específicas como la compatibilidad de mercancías peligrosas. Esta estructura de asignación constituye el núcleo del modelo propuesto, cuya formulación matemática detallada se presenta en el Capítulo 3.

2.2.3 Problemas de transporte de mercancías peligrosas (ADR)

El transporte de mercancías peligrosas, regulado por el Acuerdo Europeo sobre Transporte Internacional de Mercancías Peligrosas por Carretera (ADR), constituye un problema logístico de especial complejidad debido a las estrictas restricciones de seguridad y a la normativa legal que limita la cantidad de estos productos durante el transporte.

Desde el punto de vista de la optimización, este tipo de problemas requiere gestionar la asignación y el transporte de los envíos de manera que se minimice el riesgo para las personas y el medio ambiente, considerando simultáneamente factores económicos como el coste y, en algunos casos, el tiempo de distribución. En la literatura, esta problemática se conoce como *Hazardous Materials Transportation Problem* (HMTP) y suele enmarcarse dentro de la familia de problemas de ruteo y asignación [3].

En términos generales, el HMTP puede interpretarse como una extensión de los problemas clásicos de ruteo de vehículos, formulada habitualmente como un problema multiobjetivo que combina coste y riesgo, ya sea mediante funciones objetivo compuestas o mediante restricciones de seguridad explícitas [4]. Entre las restricciones más habituales consideradas en este tipo de modelos destacan:

- **Compatibilidad de la carga:** determinados productos peligrosos no pueden transportarse conjuntamente.
- **Límite de carga ADR:** cada vehículo tiene una cantidad máxima de mercancía peligrosa permitida.
- **Restricciones de infraestructura:** determinadas rutas o vías están prohibidas para vehículos que transportan mercancías peligrosas.

Para la resolución de estos problemas se han propuesto tanto modelos exactos basados en Programación Lineal Entera o Mixta, como enfoques heurísticos y metaheurísticos que permiten abordar instancias de mayor tamaño.

En el modelo desarrollado en este trabajo, la gestión de mercancías peligrosas se incorpora mediante restricciones de capacidad específicas por vehículo, lo que introduce una capa adicional de complejidad operativa sin necesidad de optimizar explícitamente las rutas. De este modo, se consigue representar de forma realista el impacto de la normativa ADR en la planificación diaria del transporte, equilibrando la

eficiencia económica con el cumplimiento de las condiciones de seguridad exigidas por la legislación vigente.

El siguiente apartado aborda otro aspecto clave en la toma de decisiones logísticas: la elección entre el uso de flota propia o la externalización del servicio de transporte.

2.2.4 Decisiones *Make-or-Buy* / Flota propia vs externalización

El modelo de decisión *make-or-Buy* (hacer o comprar) es una herramienta ampliamente utilizada en la planificación estratégica para determinar si una organización debe realizar internamente una determinada actividad o subcontratarla a un proveedor externo. Este tipo de análisis compara los costes, capacidades y limitaciones de cada alternativa, seleccionando la opción que mejor se ajusta a los objetivos económicos y operativos de la empresa [5].

En el ámbito logístico, la incorporación de esta decisión dentro de modelos de optimización permite determinar si una empresa debe utilizar su flota propia de vehículos o recurrir a transportistas externos para atender la demanda. El objetivo principal es minimizar el coste total de distribución, equilibrando el uso de recursos internos y externos bajo restricciones operativas reales, como capacidades de volumen y peso, costes fijos de activación o limitaciones normativas asociadas al transporte de mercancías peligrosas [6].

Desde el punto de vista de la literatura, esta problemática se conoce como *Vehicle Routing Problem with Private Fleet and Common Carriers* (VRPPC), una extensión del VRP clásico en la que una parte de los clientes es atendida mediante flota propia y el resto a través de transportistas externos. Diversos estudios abordan este problema mediante modelos de Programación Lineal Entera Mixta, así como mediante heurísticas y metaheurísticas, con el objetivo de encontrar un equilibrio entre coste, flexibilidad operativa y utilización eficiente de los recursos [7].

El modelo desarrollado en este trabajo se sitúa en esta línea, abordando la decisión *make-or-Buy* a nivel operativo diario. En particular, se decide para cada pedido si resulta más conveniente asignarlo a un camión propio o enviarlo mediante mensajería externa, considerando los costes asociados y las restricciones existentes. De este modo, el modelo combina una estructura de asignación con elementos de planificación de flota

mixta, permitiendo obtener soluciones económicamente eficientes y operativamente viables para cada jornada de reparto.

2.3 Métodos y técnicas de resolución

En el mundo de la optimización logística existen distintos enfoques para resolver los problemas de decisión: métodos exactos, heurísticos y metaheurísticos. La elección del método depende del tamaño del problema, el tiempo disponible y la necesidad de obtener una solución óptima o aproximada.

2.3.1 Métodos Exactos

Los métodos exactos son algoritmos de optimización que garantizan la obtención de la solución óptima para un problema. Estos métodos resultan especialmente adecuados cuando se requiere una precisión absoluta en la toma de decisiones y el tamaño del problema es manejable desde el punto de vista computacional.

En el ámbito de la optimización logística, estos métodos se basan principalmente en técnicas de Programación Lineal (PL), Programación Entera (IP) y Programación Lineal Entera Mixta (MILP), donde se optimiza una función objetivo sujeta a un conjunto de restricciones lineales mediante variables continuas y discretas.

Entre las principales estrategias empleadas para resolver modelos MILP destacan:

- **Ramificación y acotamiento (*Branch & Bound*)**: divide el problema original en subproblemas más pequeños y utiliza cotas para descartar regiones del espacio de búsqueda que no pueden contener la solución óptima.
- **Ramificación y poda (*Branch & Cut*)**: amplía el método anterior incorporando planos de corte, que permiten eliminar soluciones fraccionarias del problema relajado y acelerar la convergencia.
- **Descomposición de Benders**: técnica adecuada para problemas con una estructura separable, en la que el problema se divide en un problema maestro y uno o varios subproblemas, intercambiando información mediante cortes de factibilidad y optimalidad.

A pesar de su precisión, el principal inconveniente de los métodos exactos es su elevado coste computacional. En problemas de naturaleza combinatoria y NP-hard, como los problemas de enrutamiento de vehículos o de asignación con múltiples

restricciones, el tiempo de resolución crece de forma exponencial con el número de variables y restricciones.

En el contexto del problema abordado en este trabajo, los métodos exactos permiten formular de manera rigurosa la asignación de pedidos a flota propia o mensajería externa, incorporando restricciones reales de capacidad, compatibilidad de carga y costes. Este enfoque resulta especialmente útil tanto para la obtención de soluciones óptimas en escenarios de dimensión reducida como para servir de referencia en la evaluación de métodos aproximados aplicados a instancias de mayor tamaño.

2.3.2 Heurísticas

Las heurísticas de optimización combinatoria son métodos aproximados diseñados para obtener soluciones de buena calidad en problemas complejos cuya solución óptima resulta difícil o inviable de calcular en un tiempo razonable. Estos métodos se basan en reglas empíricas de decisión que permiten obtener soluciones factibles de manera rápida, aunque sin garantía de optimalidad.

A diferencia de los métodos exactos, que exploran sistemáticamente el espacio de soluciones, las heurísticas priorizan la eficiencia computacional, por lo que se utilizan ampliamente en ámbitos como la logística, el transporte y la planificación operativa.

Según su enfoque, las heurísticas pueden clasificarse en dos grandes grupos:

- **Heurísticas constructivas**, que generan una solución inicial desde cero, incorporando progresivamente los elementos del problema.
- **Heurísticas de mejora**, que parten de una solución existente y la optimizan mediante modificaciones locales.

Entre las heurísticas más utilizadas destacan:

- **Heurística del ahorro (Clarke & Wright)**: ampliamente empleada en problemas de transporte, parte de rutas individuales y las combina gradualmente en función del ahorro obtenido al unir puntos, con el objetivo de minimizar la distancia total recorrida.
- **Heurística de inserción**: selecciona iterativamente un elemento no asignado (como un cliente o un pedido) y lo incorpora en la posición que provoca el menor incremento de coste o tiempo, hasta construir una solución completa.

Las heurísticas presentan como principales ventajas su simplicidad, rapidez y facilidad de adaptación a distintos contextos. Sin embargo, su rendimiento depende en gran medida de la calidad de la solución inicial y de las reglas de decisión empleadas, y no garantizan la obtención de la solución óptima.

En el contexto del problema abordado en este trabajo, las heurísticas pueden utilizarse para generar soluciones iniciales factibles, por ejemplo, asignando pedidos a camiones o a mensajería externa mediante reglas sencillas. Estas soluciones pueden servir posteriormente como punto de partida para la aplicación de técnicas metaheurísticas.

2.3.3 Metaheurísticas

Las metaheurísticas son estrategias de optimización de alto nivel diseñadas para resolver problemas complejos en los que la obtención de la solución óptima resulta computacionalmente inviable. A diferencia de las heurísticas simples, que suelen quedar atrapadas en óptimos locales, las metaheurísticas incorporan mecanismos de exploración y explotación que permiten recorrer de forma más amplia el espacio de soluciones.

El objetivo de estas técnicas es obtener soluciones cercanas al óptimo global en tiempos razonables, optimizando una función objetivo bajo un conjunto de restricciones. Constituyen una herramienta fundamental en la resolución de problemas de gran complejidad y se aplican con éxito en ámbitos como el enrutamiento, la asignación y la planificación logística.

Existen numerosos enfoques metaheurísticos, entre los que destacan los métodos de búsqueda local mejorada y aquellos inspirados en procesos naturales, físicos o biológicos. Algunos de los más representativos son:

- ***Simulated Annealing (Templado Simulado)***: imita el proceso de recocido de la metalurgia. El algoritmo explora soluciones vecinas aceptando, con una probabilidad decreciente, soluciones peores que la actual, lo que permite escapar de óptimos locales.
- ***GRASP (Greedy Randomized Adaptive Search Procedure)***: combina una fase constructiva basada en criterios voraces y aleatorios con una fase posterior de búsqueda local, repitiendo el proceso de forma iterativa.

- **Tabu Search (Búsqueda Tabú):** amplía la búsqueda local incorporando una memoria que evita visitar soluciones o movimientos recientes, favoreciendo la diversificación de la búsqueda.
- **Genetic Algorithms (Algoritmos Genéticos):** se basan en principios de evolución biológica, utilizando operadores de selección, cruce y mutación para mejorar progresivamente una población de soluciones candidatas.
- **Ant Colony Optimization (ACO, Optimización por Colonia de Hormigas):** emula el comportamiento de las colonias de hormigas, guiando la construcción de soluciones mediante rastros de feromonas virtuales que refuerzan las rutas más prometedoras.

Las metaheurísticas ofrecen una gran flexibilidad y pueden adaptarse a una amplia variedad de problemas, proporcionando soluciones de alta calidad sin necesidad de una modelización matemática exhaustiva. No obstante, su rendimiento depende de la correcta calibración de sus parámetros, lo que puede influir significativamente en los resultados obtenidos.

En el marco del problema considerado, las metaheurísticas constituyen una alternativa eficaz para abordar instancias de mayor tamaño o complejidad, así como una herramienta útil para comparar y validar las soluciones obtenidas mediante métodos exactos.

2.4 Herramientas *software*

La resolución práctica de modelos de optimización combinatoria requiere herramientas informáticas que permitan formular los problemas de manera rigurosa, aplicar distintos métodos de resolución y analizar los resultados de forma eficiente. En este apartado se describen las principales herramientas empleadas tanto en el ámbito industrial como en la investigación académica, prestando especial atención a aquellas utilizadas en problemas de planificación y transporte.

- **Solucionadores comerciales.** Los *solvers* comerciales destacan por su alto rendimiento computacional y su capacidad para resolver modelos de gran tamaño:
 - **Gurobi:** motor de optimización matemática de alto rendimiento para problemas de Programación Lineal, Programación Entera y Programación

Lineal Entera Mixta (MILP). Se integra fácilmente con lenguajes como Python, MATLAB o C++ y es ampliamente reconocido por su rapidez y estabilidad. Su principal limitación es el coste de la licencia, aunque dispone de versiones académicas gratuitas.

- **CPLEX**: desarrollado por IBM, constituye una de las soluciones más robustas del mercado para la resolución de modelos de optimización. Emplea técnicas exactas avanzadas y ofrece un rendimiento sobresaliente incluso en problemas de gran complejidad. También dispone de licencias académicas.
- **Solucionadores de código abierto**. Existen alternativas gratuitas que, si bien suelen ofrecer un rendimiento inferior al de los *solvers* comerciales, resultan adecuadas para entornos académicos y aplicaciones de tamaño medio:
 - **GLPK (GNU Linear Programming Kit)**: *solver* de código abierto para problemas de programación lineal y entera, ampliamente utilizado en docencia y prototipado.
 - **CBC (Coin-or Branch and Cut)**: *solver* de código abierto orientado a la resolución de problemas MILP, desarrollado dentro del proyecto COIN-OR. En general, ofrece un mejor rendimiento que GLPK en modelos con variables binarias.
 - **Google OR-Tools**: conjunto de herramientas de optimización desarrollado por Google Research, especialmente eficiente en problemas de ruteo (VRP, TSP), planificación y flujos de red. Proporciona implementaciones específicas de algoritmos combinatorios y se integra directamente con Python y C++.
- **Lenguajes de modelado**. Los lenguajes de modelado permiten definir de forma estructurada los modelos matemáticos, delegando la resolución en un *solver* externo:
 - **PuLP**: librería de Python orientada a la formulación sencilla de modelos de optimización. Resulta especialmente adecuada para proyectos educativos y prototipado rápido. Es importante destacar que PuLP no resuelve el modelo por sí mismo, sino que genera la formulación y la envía a un *solver* externo como CBC, GLPK o Gurobi).

- **Pyomo:** colección de paquetes de software en Python para la modelización de problemas lineales, no lineales y estocásticos. Al igual que PuLP, actúa como lenguaje de modelado y requiere un *solver* externo para la resolución. Su flexibilidad y compatibilidad con múltiples *solvers* lo convierten en una herramienta ampliamente utilizada en investigación y aplicaciones industriales.
- **Entornos integrados:**
 - **SolverStudio:** complemento gratuito para Microsoft Excel que permite integrar lenguajes de modelado como PuLP o Pyomo directamente en hojas de cálculo. Esta integración ofrece una ventaja clave desde el punto de vista práctico: los usuarios pueden definir datos, ejecutar el modelo y analizar los resultados utilizando un entorno ampliamente conocido como Excel, sin necesidad de instalar herramientas adicionales ni poseer conocimientos avanzados de programación. Esta característica facilita la adopción del modelo en entornos empresariales y mejora su accesibilidad para perfiles no técnicos.

En conjunto, los *solvers* comerciales como Gurobi o CPLEX ofrecen el mejor rendimiento computacional, mientras que las alternativas de código abierto proporcionan una mayor accesibilidad. Por su parte, los lenguajes de modelado como PuLP y Pyomo permiten formular los problemas de forma clara y flexible, independientemente del solver utilizado. Los entornos integrados como SolverStudio combinan estas ventajas, ofreciendo una solución equilibrada entre potencia de cálculo y facilidad de uso.

2.5 Conclusiones

Los problemas de optimización logística analizados en este capítulo se enmarcan en el ámbito de la **Investigación Operativa**, disciplina que proporciona herramientas matemáticas y computacionales para apoyar la toma de decisiones en sistemas complejos. En particular, muchos de los problemas de planificación del transporte y la distribución pertenecen a la **optimización combinatoria**, caracterizada por un elevado número de decisiones discretas y una complejidad computacional creciente con el tamaño del problema.

La **programación entera** y la **programación lineal entera mixta** constituyen una de las principales aproximaciones para modelar formalmente este tipo de problemas, permitiendo representar de manera rigurosa decisiones binarias, costes y restricciones operativas reales. Los métodos exactos asociados a estos modelos garantizan soluciones óptimas, aunque su aplicabilidad práctica se limita a instancias de tamaño moderado debido a su elevado coste computacional.

Frente a estas limitaciones, las **heurísticas** y **metaheurísticas** ofrecen alternativas eficientes para abordar problemas de mayor escala o con numerosas restricciones, proporcionando soluciones de alta calidad en tiempos razonables. Mientras que las heurísticas permiten obtener soluciones viables de forma rápida, las metaheurísticas amplían la exploración del espacio de búsqueda y reducen el riesgo de quedar atrapadas en óptimos locales. En la práctica, los enfoques más efectivos suelen combinar ambas aproximaciones, dando lugar a estrategias híbridas que equilibran precisión y eficiencia. Asimismo, la revisión de problemas logísticos clásicos como el enrutamiento de vehículos, los modelos de asignación, el transporte de mercancías peligrosas y las decisiones *make-or-Buy* proporciona una base teórica sólida sobre la que construir modelos de planificación del transporte realistas y aplicables.

Finalmente, el análisis de las herramientas *software* disponibles pone de manifiesto la importancia de combinar lenguajes de modelado y *solvers* de optimización adecuados con entornos que faciliten su aplicación práctica. Este enfoque permite cerrar el ciclo entre la formulación teórica y la implementación computacional, sentando las bases para el desarrollo del modelo propuesto en los capítulos siguientes.

Capítulo 3

Metodología y Desarrollo

3.1 Introducción

Este capítulo describe el desarrollo metodológico del modelo de planificación del transporte propuesto en este trabajo. En él se define formalmente el problema abordado, se justifica la necesidad de su automatización y optimización, y se presenta la formulación matemática empleada para su resolución mediante técnicas de Programación Lineal Entera Mixta.

El objetivo es transformar un proceso de planificación que actualmente se realiza de forma manual, basada en la experiencia del planificador, en un procedimiento sistemático, reproducible y capaz de generar soluciones de alta calidad en tiempos compatibles con la operativa diaria. Para ello, se propone un modelo que automatiza la asignación de pedidos a flota propia o mensajería externa y, simultáneamente, busca minimizar el coste operativo total bajo un conjunto de restricciones reales.

3.2 Implementación del modelo de planificación de transporte

3.2.1 Contexto operativo y definición del problema

El problema abordado en este trabajo corresponde a la planificación diaria del transporte de pedidos en una empresa logística, donde debe decidirse qué envíos se realizan mediante flota propia y cuáles se externalizan a servicios de mensajería. En la

práctica actual, esta planificación se realiza de forma manual por parte de un responsable logístico.

Este enfoque presenta dos limitaciones fundamentales. En primer lugar, el proceso no está automatizado, lo que lo hace dependiente del criterio individual del planificador y difícilmente reproducible. En segundo lugar, la solución obtenida no garantiza ser óptima desde el punto de vista económico, especialmente cuando el número de pedidos y las restricciones operativas aumentan.

Con el fin de mejorar esta situación, en este trabajo se proponen dos avances principales. Por un lado, se plantea la automatización del proceso de asignación de pedidos, permitiendo generar planes de transporte de forma sistemática y consistente. Por otro, se formula el problema como un modelo de optimización, cuyo objetivo es encontrar la mejor solución posible en términos de coste operativo, teniendo en cuenta las restricciones reales de capacidad, compatibilidad de carga y disponibilidad de flota. A partir de este contexto, en los apartados siguientes se presenta la formulación matemática del modelo base y su implementación mediante herramientas de modelado y resolución de optimización, siguiendo un desarrollo progresivo que permite validar el comportamiento del modelo a medida que se incorporan restricciones y elementos operativos adicionales.

3.2.2 Formulación matemática del modelo base

3.2.2.1 Conjuntos

I : conjunto de pedidos, $i = 1, 2, \dots, n$.

J : conjunto de camiones disponibles, $j = 1, 2, \dots, m$.

3.2.2.2 Parámetros

3.2.2.2.1 Parámetros del pedido

vol_i : volumen del pedido i , medido en pallets.

Cada pedido se modela como una unidad indivisible que ocupa un número entero de pallets, lo que permite representar de forma realista la ocupación del espacio en los camiones. Por tanto, $vol_i \in \mathbb{Z}^+$.

pes_i : peso del pedido i .

$fecha_i$: fecha asociada al pedido i .

$fecha_{ho y}$: fecha actual del día de la planificación.

$adr_i \in \{0,1\}$: parámetro binario que toma valor 1 si el pedido i corresponde a mercancía peligrosa (ADR), y 0 en caso contrario.

t_i : coste de envío del pedido i mediante mensajería externa.

$dist_i$: distancia (en kilómetros) desde la base logística hasta el destino del pedido i .

α : factor de conversión de kilómetros a coste monetario $\left(\frac{\text{€}}{\text{km}}\right)$.

u_i : coste variable asociado al transporte del pedido i en camión, definido como

$$u_i = \alpha * dist_i$$

s : coeficiente de incentivo temporal, que pondera la anticipación de pedidos futuros en la función objetivo.

d_i : número de días de antelación del pedido i , definido como

$$d_i = fecha_i - fecha_{ho y}, \quad d_i \geq 0$$

$FECHA_{max}$: fecha más lejana entre todos los pedidos considerados, definida como

$$FECHA_{max} = \max_{i \in I} \{fecha_i\}$$

$\Delta_i \in [0,1]$: coeficiente de prioridad temporal del pedido i , definido como

$$\Delta_i = \frac{FECHA_{max} - d_i}{FECHA_{max}}$$

Este parámetro toma valores más elevados cuanto más cercana es la fecha del pedido y valores menores para pedidos más lejanos.

3.2.2.2.2 Parámetros del camión

V_j : capacidad máxima de volumen del camión j (en pallets).

W_j : capacidad máxima de peso del camión j .

P_j : número máximo de paradas (clientes) permitidas en el camión j .

L_{ADR_j} : límite máximo de mercancía peligrosa (ADR) permitido en el camión j , expresado como número máximo de pallets ADR que puede transportar.

F_j : coste fijo asociado a la activación del camión j .

3.2.2.3 Variables de decisión

$x_{ij} \in \{0,1\}$:

toma valor 1 si el pedido i se asigna al camión j , y 0 en caso contrario.

$y_i \in \{0,1\}$:

toma valor 1 si el pedido i se envía mediante mensajería externa, y 0 en caso contrario.

$z_j \in \{0,1\}$:

toma valor 1 si el camión j es activado, y 0 en caso contrario.

3.2.2.4 Función objetivo

El objetivo del modelo es minimizar el coste total diario de transporte, incorporando un incentivo por la anticipación de pedidos futuros:

$$\min Z = \sum_{i \in I} \sum_{j \in J} u_i x_{ij} + \sum_{i \in I} t_i y_i + \sum_{j \in J} F_j z_j - s * \sum_i \Delta_i (\sum_{j \in J} x_{ij} + y_i)$$

3.2.2.4.1 Interpretación de la función objetivo

La función objetivo está compuesta por cuatro términos principales:

- [1] Coste variable de transporte en camión propio

$$\sum_{i \in I} \sum_{j \in J} u_i x_{ij}$$

Representa el coste proporcional a la distancia recorrida para los pedidos asignados a flota propia.

- [2] Coste de mensajería externa

$$\sum_{i \in I} t_i y_i$$

Contabiliza el coste de externalizar el envío de pedidos mediante servicios de mensajería.

- [3] Coste fijo de activación de camiones

$$\sum_{j \in J} F_j z_j$$

Refleja el coste fijo asociado a poner en operación un camión, independientemente de su nivel de carga.

- [4] Incentivo por anticipación de pedidos futuros

$$-s * \sum_i \Delta_i (\sum_{j \in J} x_{ij} + y_i)$$

Este término introduce un beneficio artificial que favorece el envío anticipado de pedidos cuya fecha es posterior al día actual. El parámetro s permite regular la

importancia relativa de la anticipación frente al coste económico, actuando como un ponderador de prioridad temporal sin imponer restricciones adicionales al modelo.

Cabe destacar que el coste variable u_i y el coste de mensajería t_i representan costes medios por envío. El volumen y peso influyen de forma indirecta en el coste total al restringir la asignación de pedidos a los camiones disponibles, de modo que pedidos de mayor tamaño pueden provocar la activación de camiones adicionales o la externalización del transporte, incrementando el coste global de la solución.

3.2.2.5 Restricciones

- Envío obligatorio de pedidos del día

$$\sum_{j \in J} x_{ij} + y_i = 1 \quad \forall i \in I, \text{ tal que } fecha_i = fecha_{hoy}$$

- Envío opcional de pedidos futuros

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in I, \text{ tal que } fecha_i > fecha_{hoy}$$

- Pedidos futuros no se envían por mensajería

$$y_i = 0 \quad \forall i \in I, \text{ tal que } fecha_i > fecha_{hoy}$$

- Un camión solo puede cargarse si está activado

$$x_{ij} \leq z_j \quad \forall i \in I, j \in J$$

- Restricción de capacidad de volumen

$$\sum_{i \in I} vol_i x_{ij} \leq V_j \quad \forall j \in J$$

- Restricción de capacidad de peso

$$\sum_{i \in I} pes_i x_{ij} \leq W_j \quad \forall j \in J$$

- Límite de carga ADR por camión

$$\sum_{i \in I} adr_i vol_i x_{ij} \leq L_{ADR_j} \quad \forall j \in J$$

- Número máximo de paradas por camión

$$\sum_{i \in I} x_{ij} \leq P_j \quad \forall j \in J$$

3.2.3 Preparación de datos en Excel

La implementación del modelo se apoya en un archivo Excel que actúa como interfaz de entrada y salida de datos. Esta elección permite separar claramente la definición del modelo matemático de los datos específicos de cada instancia, facilitando tanto la generación de escenarios como la modificación de parámetros sin necesidad de alterar el código del modelo.

El archivo Excel se estructura en cuatro hojas principales: **Pedidos**, **Camiones**, **Clientes** y **Resultados**. Las tres primeras contienen los datos de entrada necesarios para la resolución del modelo, mientras que la hoja de resultados recoge la solución obtenida tras la ejecución. La **Ilustración 1** muestra un ejemplo de la estructura del archivo Excel utilizado, así como la organización general de las hojas y tablas de datos.

	A	B	C	D	E	F	G	H	I	J	K
1	ID	Cliente	Volumen	Peso	Fecha	ADR (0/1)	coste_mensajería		α	s	Fecha_hoy
2	P1	C1	2	200	23/11/2025	0	350		0,50	100,00	23/11/2025
3	P2	C2	2	200	23/11/2025	0	350				
4	P3	C3	2	200	23/11/2025	0	350		PyomoOptions		
5	P4	C4	2	200	23/11/2025	0	350		solver	cbc	
6	P5	C5	2	200	23/11/2025	0	350		solver-options	sec=20	
7	P6	C6	2	200	23/11/2025	0	350				
8	P7	C7	2	200	23/11/2025	0	350				
9	P8	C8	2	200	23/11/2025	0	350				
10	P9	C9	4	400	23/11/2025	1	350				
11	P10	C10	4	400	23/11/2025	1	350				
12	P11	C1	2	150	23/11/2025	0	350				
13	P12	C2	2	150	23/11/2025	0	350				
14	P13	C3	2	150	23/11/2025	0	350				
15	P14	C4	2	150	23/11/2025	0	350				
16	P15	C5	2	150	23/11/2025	0	350				
17	P16	C6	2	150	23/11/2025	0	350				
18	P17	C7	2	150	23/11/2025	0	350				
19	P18	C8	2	150	23/11/2025	0	350				
20	P19	C9	2	150	23/11/2025	0	350				
21	P20	C10	2	150	23/11/2025	0	350				

Ilustración 1. Estructura del archivo Excel

Cada hoja recoge la información asociada a una de las entidades principales del problema. La hoja **Pedidos** contiene los atributos de cada envío, incluyendo volumen, peso, fecha asociada, indicador de mercancía peligrosa (ADR) y coste de mensajería. Asimismo, incorpora parámetros globales del modelo, como el factor de coste por kilómetro (α), el coeficiente de incentivo temporal (s) y la fecha actual de planificación. La hoja **Clientes** almacena la información geográfica necesaria para el cálculo de los costes de transporte, en particular la distancia desde la base logística a cada cliente. Por último, la hoja **Camiones** define las características de la flota disponible, incluyendo capacidades de volumen y peso, número máximo de paradas, límites de carga ADR y costes fijos de activación.

Esta organización permite una correspondencia directa entre los datos del archivo Excel y los parámetros definidos en la formulación matemática, facilitando tanto la generación de nuevas instancias como la trazabilidad de los resultados.

3.2.4 Definir *Data Items* en SolverStudio

Para la implementación del modelo en Excel se ha utilizado el complemento **SolverStudio**, que permite integrar modelos de optimización escritos en Pyomo directamente en hojas de cálculo [3]. Una vez instalado y habilitado, SolverStudio aparece como un apartado adicional dentro de la pestaña “Datos” del entorno de Microsoft Excel (véase **Ilustración 2**).

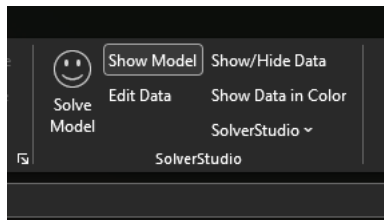


Ilustración 2. Interfaz de SolverStudio en Excel

Con el fin de que SolverStudio pueda acceder a los datos definidos en el archivo Excel, es necesario declarar explícitamente los parámetros y conjuntos del modelo mediante la opción *Edit Data*. Esta funcionalidad abre una ventana de configuración en la que se definen los denominados *Data Items*, que actúan como enlace entre las celdas de Excel y los elementos utilizados posteriormente en el script Pyomo.

Cada *Data Item* debe tener exactamente el mismo nombre que el parámetro o conjunto definido en la formulación matemática del modelo, garantizando así la correspondencia directa entre los datos y el modelo abstracto. A modo de ejemplo:

- El conjunto de pedidos I se asocia a la columna ID de la hoja Pedidos.
- Los parámetros vol_i y pes_i se vinculan a las columnas *volumen* y *peso* de dicha hoja.
- El parámetro adr_i se asocia a la columna ADR .
- El conjunto de pedidos J se define a partir de la columna *Camión (J)* de la hoja Camiones.

Una vez definidos todos los *Data Items*, SolverStudio puede cargar automáticamente los datos desde Excel y generar las estructuras necesarias para que el *script* Pyomo construya y resuelva el modelo de optimización sin intervención manual adicional.

3.2.5 Escribir *script* Pyomo en SolverStudio

Dentro de SolverStudio, la opción *Show Model* despliega un panel lateral que integra un editor de texto y una consola de ejecución. El editor proporciona un entorno básico de desarrollo, con menús para la gestión de archivos, edición de código y selección del lenguaje de modelado.

En este trabajo se ha seleccionado Pyomo como lenguaje de modelado, lo que permite definir el problema de optimización de forma declarativa y estructurada. El modelo se implementa como un *AbstractModel*, de modo que la definición matemática queda desacoplada de los datos concretos de cada instancia.

Es fundamental que los nombres utilizados en el *script* Pyomo (por ejemplo, `model.I`, `model.vol`, etc.) coincidan exactamente con los *Data Items* definidos previamente en SolverStudio. Esta correspondencia garantiza que los datos almacenados en el archivo Excel sean inyectados automáticamente en el modelo en el momento de la ejecución, sin necesidad de manipulación manual adicional.

El *script* Pyomo implementa la formulación matemática descrita en el apartado anterior, incluyendo la definición de conjuntos, parámetros, variables de decisión, función objetivo y restricciones. Esta estructura permite que el modelo sea fácilmente modificable y reutilizable dentro de SolverStudio, manteniendo separada la lógica de optimización de los datos gestionados en Excel.

3.2.6 Ejecución del modelo

La ejecución del modelo se realiza directamente desde SolverStudio mediante la opción *Solve Model*. Una vez finalizado el proceso de resolución, el complemento vuelca automáticamente los resultados en las celdas del archivo Excel previamente designadas, correspondientes a la hoja Resultados (véase **Ilustración 3**).

	A	B	C	D	E	F	G	H
1	ID	Cliente	Camión T1	Camión T2	Camión T3	Camión T4	Camión T5	Mensajería
2	P1	C1	0	0	0	0	0	1
3	P2	C2	0	1	0	0	0	0
4	P3	C3	1	0	0	0	0	0
5	P4	C4	0	0	0	0	0	1
6	P5	C5	0	0	1	0	0	0
7	P6	C6	1	0	0	0	0	0
8	P7	C7	1	0	0	0	0	0
9	P8	C8	0	1	0	0	0	0
10	P9	C9	0	0	0	0	0	1
11	P10	C10	0	0	1	0	0	0
12	P11	C1	0	1	0	0	0	0
13	P12	C2	0	0	1	0	0	0
14	P13	C3	0	1	0	0	0	0
15	P14	C4	0	0	1	0	0	0
16	P15	C5	0	1	0	0	0	0

Ilustración 3. Hoja Resultados del archivo Excel con la asignación de pedidos a camiones y mensajería.

Por defecto, SolverStudio utiliza el *solver* **CBC**, aunque es posible modificar el motor de resolución en función de las necesidades del experimento. Existen diferentes mecanismos para seleccionar un *solver* alternativo:

- **Configuración mediante tabla en Excel:** se define una tabla denominada `PyomoOptions`, compuesta por dos columnas, donde la primera recoge la opción de línea de comandos y la segunda su valor. Para seleccionar un *solver* concreto, se introduce la opción `soLver` junto con el nombre del *solver* deseado (por ejemplo, GLPK o Gurobi).
- **Configuración desde el entorno SolverStudio:** una vez seleccionado Pyomo como lenguaje de modelado, es posible elegir el *solver* directamente desde el menú correspondiente del editor del complemento.
- **Configuración desde el código:** el *solver* puede definirse explícitamente en el *script* Pyomo mediante la función `SoLverFactory`, especificando el motor de resolución a utilizar.

Durante la ejecución, SolverStudio muestra en la consola los mensajes informativos, advertencias y posibles errores generados por el *solver*, lo que facilita la identificación de problemas y la depuración del modelo.

3.2.7 Verificación y validación

3.2.7.1 Verificación estática y funcional

La verificación del modelo de Programación Lineal Entera Mixta se llevó a cabo en dos fases complementarias. En primer lugar, se realizó una **verificación estática**, destinada a comprobar que la implementación del modelo en Pyomo reproducía fielmente la formulación matemática propuesta, sin errores de definición ni inconsistencias en conjuntos, parámetros o restricciones.

En segundo lugar, se efectuó una **verificación funcional** mediante la resolución de una instancia base compuesta por 40 pedidos (I-40) y 5 camiones (J-5). Dicha instancia resultó factible y permitió alcanzar una solución óptima, con un valor de la función objetivo $Z = -87,44$ € en un tiempo de cómputo inferior a un segundo. Este resultado confirma el correcto funcionamiento del modelo en escenarios de tamaño reducido.

3.2.7.2 Validación de la lógica operacional

La validación del modelo se centró en comprobar que las decisiones obtenidas eran coherentes desde el punto de vista operativo y logístico, especialmente en situaciones de restricción de capacidad. En particular, se verificaron los siguientes comportamientos.

- **Restricción ADR:** al saturar la capacidad ADR del camión T3, el modelo optó por externalizar los pedidos ADR obligatorios excedentes en lugar de violar la restricción, confirmando el correcto funcionamiento de las limitaciones de seguridad.
- **Incentivo por anticipación:** se observó que el modelo utiliza el espacio residual de los camiones activados para adelantar pedidos futuros con un alto valor de Δ_i , validando el efecto del término de incentivo $s * \Delta_i$ en la función objetivo.
- **Activación de flota (coste fijo vs coste variable):** el modelo únicamente activa camiones adicionales cuando el coste total asociado a su uso resulta inferior al coste de externalizar los pedidos correspondientes mediante mensajería.
- **Envío obligatorio de pedidos del día:** se confirmó que todos los pedidos con fecha igual al día de planificación son siempre enviados, recurriendo a la

mensajería únicamente cuando no existe una alternativa factible mediante flota propia.

- **Restricciones físicas:** en todas las soluciones obtenidas se respetaron las restricciones de volumen, peso y número máximo de paradas por camión.

Una vez verificada la correcta implementación y validada la lógica operativa del modelo, en el siguiente capítulo se analizan de forma sistemática los resultados de distintos experimentos computacionales sobre instancias de mayor tamaño, evaluando su comportamiento en términos de calidad de la solución y tiempo de cómputo.

Capítulo 4

Experimentos y Resultados

4.1 Objetivo y diseño experimental

El objetivo de este capítulo es analizar el comportamiento del modelo de Programación Lineal Entera Mixta (MILP) propuesto mediante una validación empírica rigurosa basada en experimentos computacionales. Dichos experimentos no se limitan a verificar la corrección matemática de la formulación, sino que están orientados a evaluar su aplicabilidad práctica en un contexto empresarial real, caracterizado por restricciones de tiempo de cálculo y de recursos económicos.

En este marco, la experimentación persigue cuatro objetivos específicos:

1. **Validación funcional y solvencia del modelo.** Verificar que el procedimiento es capaz de encontrar soluciones factibles y óptimas para el problema de planificación del transporte planteado, gestionando adecuadamente el conjunto de restricciones operativas consideradas.
2. **Análisis de sensibilidad.** Evaluar el comportamiento del modelo ante variaciones en los parámetros de entrada, tales como distintas distribuciones geográficas de los clientes o cambios en la tipología de la carga, con el fin de comprobar que las decisiones obtenidas son coherentes y no dependen de una configuración de datos particular.

3. **Estudio de escalabilidad.** Determinar el tamaño máximo del problema, en términos de número de pedidos y vehículos disponibles, que puede resolverse en un tiempo computacional razonable, identificando el punto a partir del cual se produce la saturación del proceso de resolución.
4. **Análisis del compromiso entre tiempo de cómputo y calidad de la solución.** Analizar el comportamiento del modelo bajo límites estrictos de tiempo de resolución. Dado que en la operativa diaria no siempre resulta viable esperar a la convergencia matemática completa, se estudia la calidad de las soluciones obtenidas cuando el *solver* se interrumpe tras 20, 60 y 300 segundos de ejecución.

Todos los experimentos se han realizado utilizando CBC, un *solver* de optimización de código abierto. Esta elección responde a una decisión estratégica: aunque existen *solvers* comerciales de alto rendimiento, como Gurobi o CPLEX, su elevado coste de licencia puede suponer una barrera de entrada para pequeñas y medianas empresas. Demostrar la viabilidad del modelo empleando herramientas gratuitas refuerza el potencial de transferencia práctica de los resultados obtenidos.

4.1.1 Diseño del experimento y robustez estadística

Con el objetivo de evitar sesgos derivados de la resolución de una única instancia, se ha adoptado un diseño experimental basado en la generación sistemática de múltiples instancias independientes. Para cada escenario analizado (por ejemplo, 400 pedidos y 30 camiones disponibles), se han generado y resuelto 30 instancias distintas, utilizando semillas aleatorias diferentes que afectan tanto a la localización de los clientes como a las características de los pedidos.

Para facilitar este proceso, se ha desarrollado un entorno de ejecución automatizado mediante *scripts* en Python apoyados en la librería Pyomo, que permite:

- Generar, para cada escenario, lotes de 30 ficheros de datos sintéticos pero realistas.
- Resolver de forma secuencial dichas instancias mediante procesamiento por lotes, aplicando los límites de tiempo predefinidos.

- Agregar y analizar los resultados obtenidos, calculando métricas estadísticas relevantes como la media, la desviación típica, los valores mínimo y máximo, así como la tasa de éxito en la obtención de soluciones óptimas.

Este enfoque permite formular conclusiones basadas en estadísticas agregadas, proporcionando una caracterización más robusta y representativa del rendimiento esperado del modelo que la resolución de casos individuales.

4.2 Generación de escenarios de prueba

Ante la ausencia de un histórico amplio de datos reales y con el objetivo de garantizar la reproducibilidad científica del estudio, se ha optado por la generación de datos sintéticos. Esta aproximación, ampliamente utilizada en la literatura de investigación operativa, permite controlar de forma explícita las variables críticas del problema, lo cual resulta difícil de lograr a partir de datos históricos fijos.

4.2.1 Implementación del generador de instancias

Se ha desarrollado un *script* en lenguaje Python, denominado `generador_masivo.py`, que actúa como un motor de simulación estocástica. Para cada escenario se generan 30 instancias independientes, manteniendo fija la estructura del problema (número de pedidos, camiones y perfil operativo), pero introduciendo variabilidad controlada en los datos de entrada.

En particular, entre una instancia y otra se modifican los siguientes elementos:

- **Clientes.** A cada cliente se le asigna inicialmente una distancia base respecto a la base logística. En cada instancia, esta distancia se perturba ligeramente mediante una variación aleatoria acotada, de modo que las distintas instancias comparten una geografía común, pero no idéntica. Este enfoque permite introducir variabilidad espacial sin generar configuraciones irreales o incoherentes entre ejecuciones.
- **Pedidos.** En cada instancia, los pedidos se asignan de forma aleatoria a los distintos clientes y se generan sus atributos principales (peso, volumen, fecha y condición ADR). La probabilidad de que un pedido sea ADR depende exclusivamente del perfil del escenario considerado, mientras que el resto de los atributos se generan de forma independiente en cada ejecución.

- **Camiones.** La flota disponible se configura como heterogénea en cada instancia. A cada camión se le asigna de forma aleatoria uno de los tipos predefinidos (pequeño, mediano o grande), lo que determina sus capacidades de peso y volumen, su coste fijo de activación y la posibilidad de transportar mercancía peligrosa. Adicionalmente, el número máximo de paradas permitidas por camión se genera de forma aleatoria e independiente para cada instancia.

Este procedimiento garantiza que todas las instancias de un mismo escenario compartan las mismas características estructurales, pero difieran en la distribución concreta de pedidos, capacidades y restricciones. De este modo, se elimina el efecto de casos particulares excepcionalmente favorables o desfavorables, proporcionando una evaluación más representativa del comportamiento del modelo.

4.2.2 Perfiles de escenarios definidos

Con el objetivo de analizar el comportamiento del modelo bajo distintas condiciones operativas, se han definido cuatro tipologías de escenarios, cada una orientada a estresar un conjunto específico de restricciones:

- **Escenario base (normal).** Configurado con pesos y volúmenes equilibrados y una baja proporción de mercancía peligrosa. Actúa como escenario de referencia para establecer la línea base de costes y tiempos de resolución.
- **Escenario de carga pesada.** Se incrementa el peso medio de los pedidos manteniendo constante su volumen. En este caso, la restricción de capacidad por peso se convierte en el principal cuello de botella, permitiendo evaluar el comportamiento del modelo cuando los camiones se saturan por masa antes que por volumen.
- **Escenarios de sensibilidad ADR.** Para estudiar el impacto de la mercancía peligrosa, se ha diseñado un experimento específico en el que se varía la proporción de pedidos ADR en tres niveles controlados: 25%, 50% y 75%. Este planteamiento permite aislar el efecto de las incompatibilidades de carga sobre el coste operativo y la flota activada.
- **Escenarios de escalabilidad.** Se generan instancias de tamaño creciente, desde 40 pedidos y 5 camiones hasta 400 pedidos y 40 camiones, manteniendo las proporciones del escenario base. El objetivo es analizar el crecimiento del tiempo

de cómputo y caracterizar el comportamiento del modelo a medida que aumenta el tamaño del problema.

4.2.3 Repositorio de datos y código abierto

Siguiendo los principios de la ciencia abierta y con el fin de facilitar la validación externa de los resultados, tanto el código fuente del modelo y los scripts de generación de instancias como el conjunto completo de datos de prueba y los ficheros de resultados se han puesto a disposición de la comunidad académica en un repositorio público.

- **Enlace al repositorio:**
- <https://github.com/EnriqueBio/asignacion-pedidos-camiones-mensajeria.git>

El repositorio incluye instrucciones detalladas para la reproducción de los experimentos, lo que permite a otros investigadores utilizar estas instancias como referencia o *benchmark* para la evaluación de nuevos enfoques de resolución.

4.3 Análisis del incentivo por anticipación de envío

En este apartado se analizan los resultados obtenidos en instancias de tamaño pequeño y medio, comprendidas entre 40 y 100 pedidos. El objetivo es doble: por un lado, validar la coherencia lógica de las decisiones del modelo, con especial atención al comportamiento del incentivo por anticipación de envío; por otro, evaluar el rendimiento computacional del enfoque en escenarios controlados, previos al análisis de instancias de mayor escala.

Siguiendo el diseño experimental descrito en la Sección 4.2, para cada configuración de tamaño se han generado y resuelto múltiples instancias independientes, con distribuciones aleatorias de clientes y características de los pedidos. Las pruebas se han ejecutado con un límite de tiempo holgado de 300 segundos, con el fin de garantizar la convergencia a la solución óptima siempre que fuese computacionalmente viable.

Los experimentos se han realizado utilizando el *solver* CBC sobre un equipo con las siguientes características: procesador Intel Core i7-12800HX (2.00 GHz), 32 GB de memoria RAM y sistema operativo Windows 11. La tarjeta gráfica dedicada del equipo no interviene en el proceso de resolución, dado que el *solver* empleado es puramente *CPU-bound*. La **Tabla 1** recoge las métricas medias obtenidas a partir de las 30 iteraciones realizadas para cada escenario.

Tabla 1. Resultados del modelo en instancias pequeñas y medias (n=30)

Escenarios (Pedidos/ Camiones)	Coste Medio (€)	Desviación Típica (€)	Rango Coste [Min,Max]	Tiempo Medio (s)	Tasa de Éxito (% Óptimos)
40p / 5c	1.462,20 €	1.605,63 €	[-791,0; 6.894,0]	4,21 s	100,00%
60p / 5c	2.184,58 €	1.247,00 €	[221,0; 4.664,5]	10,41 s	96,67%
80p / 5c	5.715,07 €	1.653,55 €	[2.013,5; 9.180,5]	10,21 s	96,67%
100p / 5c	8.651,82 €	1.972,69 €	[4.800,0; 12.728,0]	10,15 s	96,67%
100p / 10c	1.350,95 €	1.257,04 €	[-507,99; 4.312,0]	11,47 s	96,67%

4.3.1 Rendimiento computacional

Los resultados estadísticos confirman la eficiencia del modelo en este rango de complejidad. La tasa de éxito en la obtención del óptimo es elevada, alcanzando un 97,3% de las instancias agregadas (146 de 150). En términos de tiempo de cómputo, aunque se estableció un límite máximo de 300 segundos, el *solver* converge a la solución óptima en un tiempo medio comprendido entre 4 y 12 segundos.

Para estos tamaños de problema no se observa un crecimiento exponencial del tiempo de resolución, lo que valida el uso de CBC como herramienta viable para instancias de dimensión pequeña y media. La variabilidad observada en los tiempos de cómputo es inherente al método de *Branch and Bound* empleado por el *solver*, cuyo rendimiento depende de la estructura combinatoria específica de cada instancia.

4.3.2 Validación de la lógica operativa

El análisis de los valores medios obtenidos revela patrones de comportamiento plenamente coherentes con la lógica operativa del problema logístico planteado.

En primer lugar, la comparación entre los escenarios de 100 pedidos con 5 y 10 camiones pone de manifiesto el impacto crítico de la disponibilidad de flota. En el caso de 5 camiones, la situación de saturación conduce a un coste medio elevado (8.651 €), consecuencia del recurso forzado a servicios de mensajería externa. Al ampliar la flota a 10 camiones, el coste medio se reduce hasta 1.351 €, lo que supone una disminución aproximada del 84%. Este resultado confirma que el modelo prioriza correctamente el uso de flota propia cuando existe capacidad disponible.

En segundo lugar, las desviaciones típicas relativamente elevadas observadas en el coste objetivo, incluso para tamaños de problema similares, reflejan la influencia significativa de factores como la localización geográfica de los clientes y la tipología de los pedidos en términos de peso y volumen. Esto indica que el coste no depende únicamente del número de pedidos, sino de la configuración espacial y logística concreta de cada instancia.

4.3.3 Comportamiento del incentivo por anticipación

El efecto del incentivo por anticipación (Δ_i) se aprecia con claridad al analizar los rangos mínimo y máximo de la función objetivo obtenidos en las distintas configuraciones.

En escenarios con holgura de capacidad, como 40 pedidos con 5 camiones o 100 pedidos con 10 camiones, se observan valores mínimos negativos de la función objetivo (-791 € y -508 €, respectivamente). Estos resultados indican que, en instancias con geometría favorable de las rutas, el beneficio acumulado asociado a la anticipación de pedidos futuros puede superar los costes operativos de transporte. En estos casos, el modelo utiliza de forma eficiente la capacidad residual de los vehículos para adelantar envíos, maximizando el incentivo temporal asociado al servicio anticipado.

Por el contrario, a medida que la flota disponible se aproxima a la saturación, como en los escenarios de 60, 80 y 100 pedidos con 5 camiones, los valores mínimos del coste se vuelven positivos y crecientes. Este comportamiento confirma que el modelo inhibe de forma automática la anticipación cuando la capacidad es escasa, priorizando estrictamente el cumplimiento de los pedidos obligatorios del día, en línea con una gestión racional de los recursos disponibles.

4.3.4 Síntesis de resultados

A partir del análisis estadístico de las 30 iteraciones realizadas para cada escenario, se pueden extraer las siguientes conclusiones:

- El modelo presenta un comportamiento robusto, resolviendo satisfactoriamente más del 96% de las instancias en tiempos de cómputo reducidos.
- El incentivo por anticipación actúa de forma adaptativa, generando beneficios únicamente cuando existe capacidad disponible y desactivándose de manera natural en situaciones de saturación.
- La disponibilidad de flota propia resulta ser el factor más determinante en el coste final, con un impacto superior al del número total de pedidos procesados.

4.4 Análisis económico: decisión *Make-or-Buy*

En este apartado se analiza el comportamiento del modelo ante el clásico dilema logístico *make-or-buy*, esto es, la decisión entre utilizar flota propia o recurrir a la mensajería externa cuando la demanda supera la capacidad inicialmente disponible. El objetivo es evaluar si el modelo adopta decisiones económicamente coherentes al comparar los costes fijos de activación de vehículos con los costes unitarios asociados a la externalización de pedidos.

Para ello, se estudian distintos escenarios en los que se amplía progresivamente la flota disponible y se modifican las características de la demanda, prestando especial atención a situaciones en las que el modelo debe decidir entre activar camiones con costes fijos elevados o derivar pedidos a servicios de mensajería.

4.4.1 Activación de flota y comparación de costes

Un caso especialmente ilustrativo de esta dinámica se observa al comparar las instancias de 100 pedidos bajo dos configuraciones de flota distintas (5 y 10 camiones), cuyos resultados medios se recogen en la **Tabla 1**.

En el escenario con 5 camiones disponibles, la flota propia resulta insuficiente para absorber la demanda obligatoria, lo que conduce a un coste medio elevado (8.651 €) debido al recurso intensivo a la mensajería externa. En cambio, al ampliar la flota a 10 camiones, el coste medio desciende hasta 1.351 €, reflejando que el modelo opta por activar vehículos adicionales cuando esta decisión resulta económicamente ventajosa.

Cabe destacar que el modelo no actúa de forma binaria ni activa indiscriminadamente todos los recursos disponibles. La activación de un camión adicional solo se produce cuando la suma de su coste fijo y el coste variable asociado a la ruta es inferior al coste de externalizar los pedidos que dicho vehículo puede absorber. De este modo, el modelo establece implícitamente un umbral de rentabilidad, utilizando la flota propia como herramienta principal de reducción del coste medio por pedido y relegando la mensajería a un papel complementario para picos de demanda o cargas incompatibles.

4.4.2 Sensibilidad del modelo a la tipología de carga

Uno de los resultados más relevantes del estudio es la cuantificación del impacto económico asociado a la naturaleza de la mercancía transportada. Para ello, se ha seguido una estrategia de análisis en dos niveles: una comparación general entre perfiles operativos y un análisis específico de sensibilidad para la mercancía peligrosa (ADR).

4.4.2.1 Comparativa General: Normal vs. Pesado vs. ADR

Utilizando como referencia un escenario de 200 pedidos y 15 camiones, se han comparado los costes operativos medios obtenidos tras 30 iteraciones para tres perfiles operativos distintos. Los resultados se muestran en la **Ilustración 4**, donde se emplean diagramas de cajas para representar la distribución de los costes obtenidos en cada escenario.

Los resultados empíricos ponen de manifiesto diferencias estructurales significativas entre los escenarios analizados:

- **Escenario base (normal).** Presenta un coste medio de 8.829,40 €, correspondiente a una operativa estándar con cargas mixtas y un equilibrio razonable entre peso y volumen. El reducido rango de variación observado indica un comportamiento estable del modelo en condiciones normales.
- **Escenario de carga pesada.** Registra el mayor coste medio, 26.359,77 €, lo que supone un incremento del 198,5% respecto al escenario base. Al aumentar el peso unitario de los pedidos, los vehículos alcanzan su capacidad máxima por peso antes de aprovechar completamente su volumen, lo que reduce la eficiencia de la consolidación y obliga a activar recursos adicionales.

- **Escenario ADR extremo.** El coste medio asciende a 15.204,15 €, un 72,2% superior al escenario base. Al igual que en el escenario de carga pesada, el incremento del coste responde a una pérdida de eficiencia en la consolidación de pedidos en flota propia. En este caso, dicha ineficiencia no viene causada por la saturación por peso, sino por las restricciones de incompatibilidad entre mercancías peligrosas, que obligan al modelo a recurrir con mayor frecuencia a la mensajería externa.

El uso de diagramas de cajas permite observar, además del valor medio, la dispersión de los resultados obtenidos en cada escenario, confirmando que las diferencias de coste no responden a ejecuciones puntuales, sino a efectos sistemáticos derivados de las restricciones operativas consideradas.

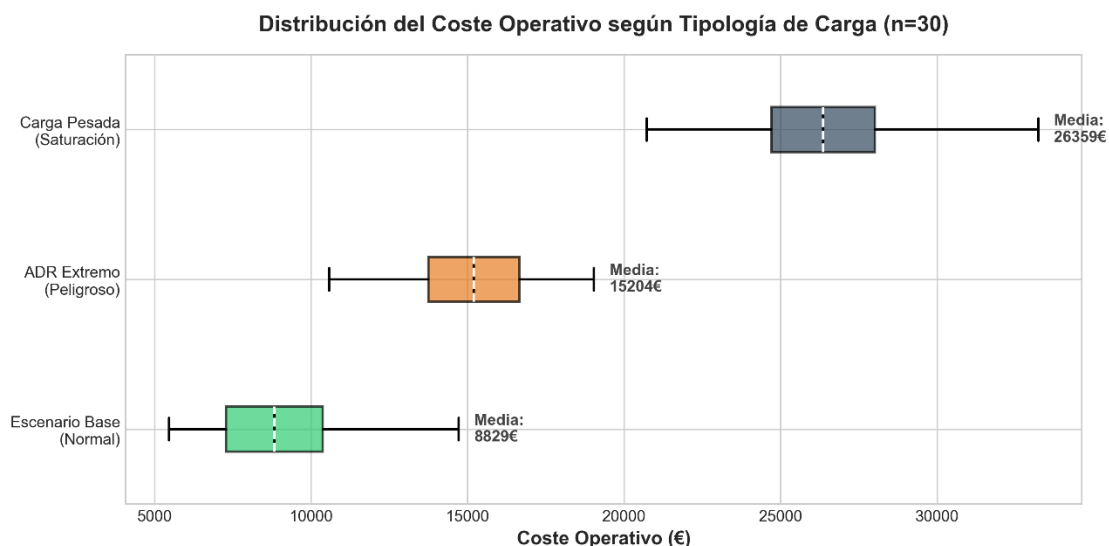


Ilustración 4. Distribución del Coste Operativo según Tipología de Carga.

4.4.2.2 Análisis de Gradiente de Peligrosidad (ADR)

Dada la complejidad observada en el escenario ADR, se ha realizado un experimento específico para analizar cómo evoluciona el coste operativo a medida que aumenta la proporción de mercancía peligrosa en la cartera de pedidos. Para ello, se han considerado tres niveles de densidad ADR: 25%, 50% y 75%.

Los resultados agregados de este experimento, recogidos en la , ponen de manifiesto una relación positiva y no lineal entre la densidad de mercancía peligrosa y el coste operativo. Al pasar del 25% al 50% de pedidos ADR, el coste medio aumenta de forma moderada (+24%), lo que indica que el modelo aún encuentra margen para consolidar

cargas compatibles. Sin embargo, al alcanzar una densidad del 75%, el coste prácticamente se duplica respecto al escenario base (+98,6%), reflejando que las restricciones de seguridad saturan la capacidad de los vehículos mucho antes de agotar su volumen físico.

Tabla 2, ponen de manifiesto una relación positiva y no lineal entre la densidad de mercancía peligrosa y el coste operativo. Al pasar del 25% al 50% de pedidos ADR, el coste medio aumenta de forma moderada (+24%), lo que indica que el modelo aún encuentra margen para consolidar cargas compatibles. Sin embargo, al alcanzar una densidad del 75%, el coste prácticamente se duplica respecto al escenario base (+98,6%), reflejando que las restricciones de seguridad saturan la capacidad de los vehículos mucho antes de agotar su volumen físico.

Tabla 2. Impacto progresivo de la densidad de ADR (n=30)

Escenarios (% ADR)	Coste Medio (€)	Incremento Coste	Tiempo Medio (s)	Tasa de Éxito (% Óptimos)
ADR 25%	10.207,63 €	Base	186,87 s	50,00%
ADR 50%	12.682,01 €	+24,2%	89,81 s	83,33%
ADR 75%	20.277,85 €	+98,6%	16,61 s	100,00%

Desde el punto de vista computacional, se observa un fenómeno inverso entre complejidad operativa y dificultad de resolución. En escenarios con baja densidad ADR, el *solver* dispone de una mayor libertad combinatoria, lo que amplía el espacio de búsqueda y se traduce en mayores tiempos medios de resolución y tasas de éxito más reducidas. Por el contrario, en escenarios con alta densidad de mercancía peligrosa, las restricciones severas reducen drásticamente el número de combinaciones factibles, permitiendo al *solver* podar rápidamente ramas inviables y alcanzar el óptimo matemático en tiempos significativamente menores.

Este resultado pone de manifiesto que restricciones operativas más estrictas, aunque encarecen la solución logística, simplifican el problema desde el punto de vista de la optimización matemática.

4.4.3 Síntesis del análisis *Make-or-Buy*

En conjunto, los experimentos realizados confirman de forma consistente que el modelo resuelve adecuadamente el dilema *make-or-buy*. En particular, se observa que el algoritmo:

- Prioriza el uso de flota propia cuando esta resulta económicamente ventajosa, recurriendo a la mensajería externa únicamente como mecanismo de ajuste ante situaciones de saturación.
- Activa los recursos de forma selectiva, discriminando entre camiones en función de su coste fijo y de la demanda efectiva a absorber.
- Presenta una elevada sensibilidad a la tipología de carga, mostrando incrementos no lineales del coste operativo ante un aumento de las restricciones de seguridad asociadas a la mercancía peligrosa.

Este análisis proporciona una base cuantitativa sólida para el estudio de escenarios de mayor escala y para la evaluación de los límites computacionales del modelo, que se aborda en el apartado siguiente.

4.5 Escalabilidad y punto de inflexión computacional

En este apartado se somete al modelo a un análisis de escalabilidad, incrementando de forma significativa el tamaño del problema con el objetivo de identificar el límite práctico de resolución del *solver* CBC. En particular, se pretende caracterizar el punto a partir del cual el crecimiento del espacio combinatorio provoca un aumento no lineal del tiempo de cómputo, dificultando o impidiendo la convergencia al óptimo matemático dentro de un tiempo razonable.

A diferencia de los apartados anteriores, el análisis se centra en escenarios de gran volumen, con 200 y 400 pedidos diarios y flotas de tamaño creciente.

4.5.1 Escalabilidad con el número de pedidos y camiones

La **Tabla 3** resume el comportamiento medio del modelo en estos escenarios de alta densidad. Los resultados muestran una degradación progresiva del rendimiento computacional a medida que aumenta el tamaño del espacio de búsqueda.

En escenarios con una flota muy reducida, como el caso de 400 pedidos y 10 camiones, el modelo alcanza el óptimo en la mayoría de las instancias (90% de éxito) y en tiempos

relativamente contenidos. Este comportamiento se explica por la fuerte restricción impuesta por la escasez de recursos, que limita el número de combinaciones factibles: los camiones se saturan rápidamente y la mayor parte de la demanda se deriva a mensajería externa, simplificando la decisión de asignación.

Sin embargo, al incrementar progresivamente el número de camiones disponibles, el coste operativo medio desciende de forma significativa debido a una mayor utilización de flota propia, pero a costa de un incremento acusado del esfuerzo computacional. En particular, en los escenarios de 400 pedidos con 20, 25 y 30 camiones, el tiempo medio de resolución se aproxima rápidamente al límite de 300 segundos, mientras que la tasa de éxito en la obtención del óptimo cae de forma pronunciada. En el caso de 30 camiones, ninguna de las 30 instancias logra cerrar el gap antes de agotar el tiempo disponible.

Tabla 3. Escalabilidad del modelo en escenarios grandes (Límite 300s, n=30)

Escenarios (Pedidos/ Camiones)	Coste Medio (€)	Tiempo Medio (s)	Tasa de Éxito (% Óptimos)	Gap Medio Final
200p / 10c	18.458,88 €	77,97 s	82,76%	0,01
400p / 10c	51.521,52 €	76,09 s	90,00%	0,00
400p / 20c	36.990,95 €	204,66 s	50,00%	0,03
400p / 25c	29.785,18 €	287,94 s	16,67%	0,10
400p / 30c	21.477,65 €	299,41 s	0,00%	0,24
400p / 40c	6.921,54 €	298,23 s	0,00%	1,30

Este comportamiento pone de manifiesto que el principal factor que impulsa la complejidad del problema no es únicamente el número de pedidos, sino el número de alternativas de asignación generadas por una flota amplia, que multiplica las combinaciones posibles entre pedidos y vehículos.

4.5.2 Punto de inflexión computacional

A partir de los resultados obtenidos, es posible situar el punto de inflexión computacional del *so/ver* CBC en el entorno de las 10.000–12.000 variables binarias de

asignación. Por debajo de este umbral, el modelo se resuelve de forma robusta y eficiente; una vez superado, el tiempo de cómputo crece de manera abrupta y la probabilidad de alcanzar el óptimo dentro del límite temporal disminuye notablemente.

De forma esquemática, pueden distinguirse dos regiones operativas:

- **Zona de confort.** Incluye escenarios con hasta 200 pedidos, así como escenarios de 400 pedidos con flotas muy reducidas. En esta región, el *solver* encuentra soluciones óptimas o cuasi-óptimas en tiempos moderados.
- **Zona de saturación.** A partir de aproximadamente 25 camiones en escenarios de 400 pedidos, el tiempo de resolución se aproxima de forma sistemática al límite de 300 segundos, evidenciando la saturación del proceso de búsqueda.

Este resultado confirma que la escalabilidad del modelo depende de forma crítica del tamaño de la flota, más que del número de pedidos en sí, al ser el número de vehículos el principal determinante del tamaño del espacio combinatorio.

4.5.3 Análisis coste–tiempo y soluciones cuasi-óptimas

La **Ilustración 5** representa gráficamente el compromiso entre coste operativo y tiempo de cómputo para el caso de 400 pedidos, mediante un gráfico de doble eje. En el eje izquierdo se muestra la distribución del coste operativo a través de diagramas de cajas correspondientes a las 30 iteraciones de cada configuración de flota, mientras que en el eje derecho se representa el tiempo medio de resolución.

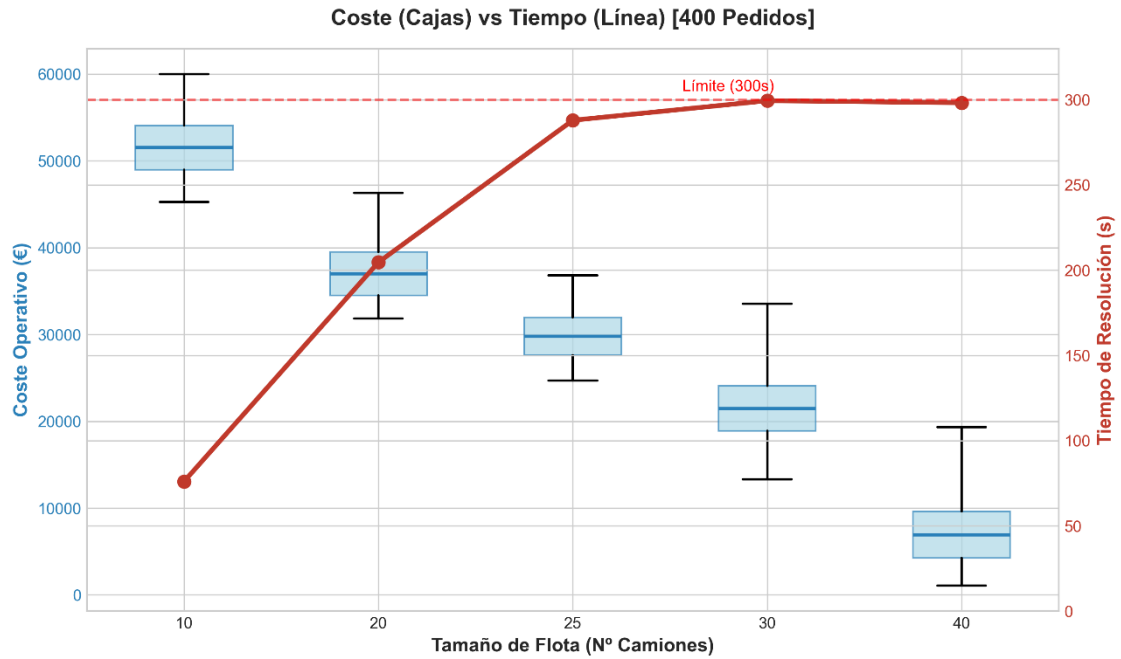


Ilustración 5. Análisis de Escalabilidad (400 pedidos): Distribución del Coste Operativo (Cajas) vs. Tiempo de Resolución (Línea)

A medida que aumenta el tamaño de la flota, el coste operativo medio disminuye, reflejando una menor dependencia de la mensajería externa. Sin embargo, este ahorro económico viene acompañado de un aumento pronunciado del tiempo de cómputo, que alcanza el límite impuesto de 300 segundos a partir de flotas de tamaño medio-alto.

Asimismo, en los escenarios de 30 y 40 camiones se observa una mayor dispersión en los valores del coste operativo. Esta variabilidad se debe a que, al agotarse el tiempo de resolución, el *solver* devuelve la mejor solución encontrada hasta ese momento, cuya calidad depende de la trayectoria seguida en el árbol de búsqueda y, por tanto, de la instancia aleatoria concreta.

En términos operativos, estos resultados indican que, para volúmenes elevados de demanda (en torno a 400 pedidos diarios), el uso de CBC resulta viable siempre que se acepten soluciones cuasi-óptimas, con *gaps* moderados, o se introduzcan restricciones adicionales que reduzcan el tamaño del espacio de búsqueda. La obtención sistemática del óptimo matemático en estos escenarios requeriría el uso de *solvers* comerciales de alto rendimiento o el desarrollo de enfoques heurísticos o metaheurísticos específicos.

4.6 Resolución bajo límite de tiempo y soluciones cuasi-óptimas

En la operativa logística diaria, la rapidez en la obtención de una solución suele ser más relevante que la optimalidad matemática estricta. En este contexto, esperar varios minutos de cómputo para obtener mejoras marginales en el coste operativo no siempre resulta justificable desde el punto de vista empresarial. Por este motivo, en este apartado se analiza el comportamiento del modelo bajo restricciones estrictas de tiempo, evaluando su capacidad para proporcionar soluciones cuasi-óptimas de calidad suficiente para la toma de decisiones en entornos reales.

4.6.1 Configuración experimental

Se han seleccionado los escenarios más exigentes desde el punto de vista computacional, correspondientes a instancias de 200 y 400 pedidos, y se han resuelto bajo tres límites de tiempo distintos: 20 segundos (respuesta rápida), 60 segundos (operativa estándar) y 300 segundos (búsqueda profunda).

Tabla 4. Evolución de la calidad de la solución según tiempo de cómputo (n=30)

Escenario	Límite	Coste Medio (€)	Gap Medio	Tasa de Éxito (% Óptimos)
200p / 10c	20s	18.681 €	0,06	46,4%
	60s	18.473 €	0,01	75,9%
	300s	18.458 €	0,01	82,8%
400p / 10c	20s	51.552 €	0,03	20,0%
	60s	51.521 €	0,01	70,0%
	300s	51.521 €	0,00	90,0%
400p / 25c	20s	49.554 €	0,66	0,0%
	60s	35.377 €	0,26	0,0%
	300s	29.785 €	0,10	16,7%
400p / 30c	20s	53.340 €	0,90	0,0%
	60s	48.249 €	0,67	0,0%
	300s	21.477 €	0,24	0,0%
400p / 40c	300s	6.921 €	1,30	0,0%

Para cada configuración se han ejecutado 30 iteraciones independientes. La **Tabla 4** recoge la evolución de las principales métricas de interés: coste operativo medio, *gap* medio de optimalidad y tasa de éxito, definida como el porcentaje de instancias en las que se certifica el óptimo matemático dentro del tiempo disponible.

4.6.2 Resultados y análisis de convergencia

Los resultados muestran dos comportamientos claramente diferenciados en función de la complejidad combinatoria del escenario analizado.

En los escenarios de tamaño medio, como el caso de 200 pedidos, el modelo presenta una convergencia rápida. A los 60 segundos de ejecución, el *solver* alcanza el óptimo matemático en el 75,9% de las instancias, con un *gap* residual del 1%. Extender el tiempo de resolución hasta 300 segundos apenas produce una mejora adicional en el coste operativo (de 18.473 € a 18.458 € de media), lo que sugiere que, para este tamaño de problema, un límite de tiempo del orden de un minuto resulta suficiente desde una perspectiva práctica.

En contraste, en los escenarios de gran tamaño, como el caso de 400 pedidos con 25 camiones, la dinámica de convergencia es sustancialmente distinta. A los 20 segundos, el *solver* se encuentra en una fase inicial de exploración, con costes elevados y *gaps* superiores al 60%. Al ampliar el tiempo a 60 segundos, se produce una mejora significativa del coste operativo (reducción aproximada del 28%), aunque sin lograr certificar el óptimo en ninguna iteración. Finalmente, a los 300 segundos, el coste medio continúa descendiendo hasta aproximadamente 29.785 €, lo que supone una mejora acumulada cercana al 40%, si bien la tasa de éxito en la certificación del óptimo permanece limitada (16,7%).

Estos resultados ponen de manifiesto que, en escenarios complejos, la ausencia de certificación del óptimo no implica necesariamente soluciones de baja calidad. De hecho, incluso en configuraciones donde la tasa de éxito es nula, como en el caso de 400 pedidos con 30 camiones, el modelo es capaz de reducir de forma sustancial el coste operativo, pasando de valores iniciales superiores a 53.000 € a soluciones en el entorno de 21.000 €.

4.6.3 Patrones de convergencia

El análisis de la evolución temporal del coste revela que no existe un patrón de convergencia único, sino que este depende de la estructura combinatoria del problema. La **Ilustración 6** muestra la evolución del coste operativo medio y su dispersión estadística (± 1 desviación típica) para las 30 iteraciones de los escenarios más exigentes.

En el escenario de 400 pedidos con 25 camiones se observa un patrón de rendimientos decrecientes. El coste se reduce de forma abrupta durante los primeros 60 segundos y, a partir de ese punto, la curva se aplana, de modo que el esfuerzo computacional adicional genera únicamente mejoras marginales. Desde una perspectiva operativa, este comportamiento sugiere que resulta eficiente interrumpir el proceso de resolución de forma temprana.

Por el contrario, el escenario de 400 pedidos con 30 camiones presenta un patrón de convergencia tardía. Durante el primer minuto de ejecución, el coste apenas desciende y la dispersión entre instancias es elevada. Sin embargo, al extender el tiempo de cómputo hasta 300 segundos, se produce una mejora sustancial del coste operativo, reflejando que el *solver* necesita un tiempo prolongado para podar el árbol de búsqueda y acceder a configuraciones logísticas de mayor calidad.

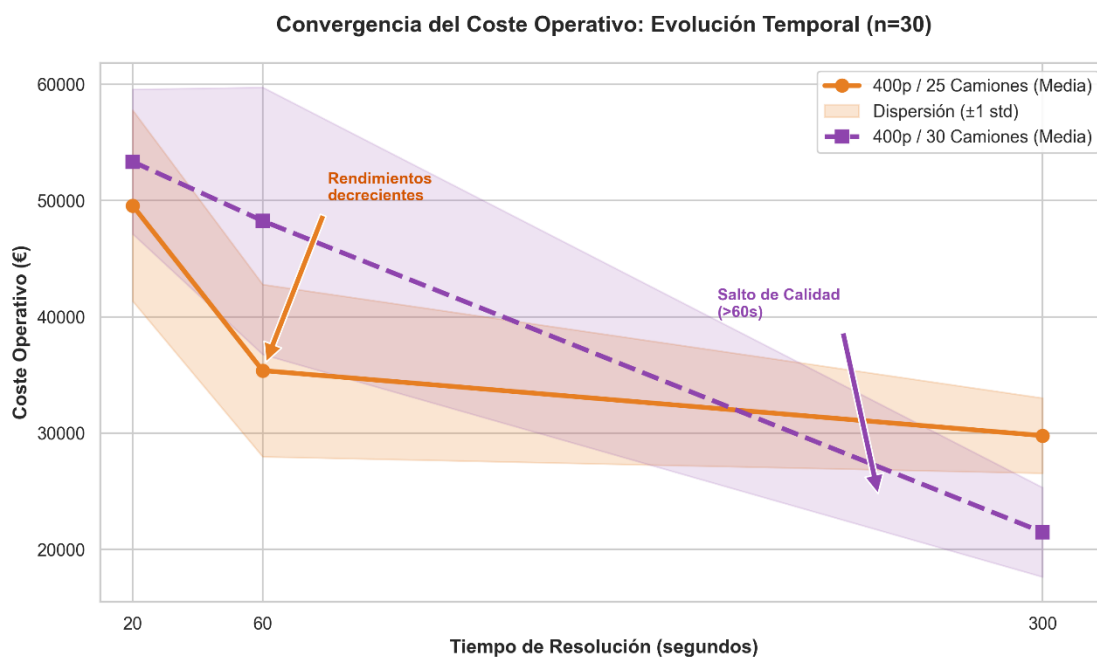


Ilustración 6. Convergencia Coste vs Tiempo

4.6.4 Estabilidad estructural de las soluciones

A pesar de que los gaps de optimalidad permanecen elevados en algunos escenarios complejos (por ejemplo, un 24% en el caso de 400 pedidos con 30 camiones a los 300 segundos), el análisis cualitativo de las soluciones obtenidas muestra una elevada estabilidad estructural en las decisiones estratégicas.

En particular, la selección de la flota activada y la decisión principal de externalización se determinan en fases tempranas del proceso de optimización. El tiempo adicional de cómputo se emplea fundamentalmente en ajustes de segundo orden, como reasignaciones puntuales de pedidos, que tienen un impacto limitado sobre la estructura global de la solución.

Este resultado valida que, incluso en presencia de gaps relativamente elevados, las soluciones generadas por el modelo son operativamente robustas y adecuadas para su implementación en contextos reales, donde la rapidez de respuesta y la coherencia de las decisiones son prioritarias frente a la optimalidad matemática estricta.

4.7 Síntesis global de los resultados

Los resultados obtenidos a lo largo de este capítulo permiten extraer una visión global y coherente del comportamiento del modelo propuesto. La experimentación realizada confirma que el enfoque es sólido desde el punto de vista operativo, económico y computacional, incluso bajo escenarios de alta exigencia.

En primer lugar, el modelo muestra un comportamiento operativo consistente, cumpliendo de forma estricta el conjunto de restricciones físicas y lógicas consideradas. La sensibilidad observada frente a la tipología de carga pone de manifiesto que el modelo responde de manera no lineal al aumento de restricciones, particularmente en escenarios con elevada densidad de mercancía peligrosa, donde el coste operativo se incrementa de forma significativa como consecuencia de la menor capacidad de consolidación.

Desde el punto de vista económico, los experimentos confirman que la decisión *make-or-buy* emerge de forma endógena a partir de la estructura de costes del problema. La activación de flota propia no sigue reglas predefinidas, sino que responde a comparaciones marginales entre costes fijos, costes variables y costes de

externalización, reproduciendo un comportamiento alineado con la lógica de gestión de un sistema de transporte real.

Asimismo, el mecanismo de incentivo por anticipación se ha revelado como una herramienta eficaz para gestionar la demanda de forma dinámica. Su activación selectiva en situaciones de capacidad disponible permite adelantar pedidos futuros sin comprometer el servicio diario, mientras que se inhibe automáticamente en escenarios de saturación, evitando interferencias con los envíos obligatorios.

Por último, el análisis de escalabilidad y resolución bajo límites de tiempo muestra que el modelo es computacionalmente viable para tamaños de problema representativos de entornos empresariales reales. Incluso cuando la certificación del óptimo matemático no es alcanzable dentro de tiempos estrictos, las soluciones obtenidas presentan una elevada estabilidad estructural, lo que las hace adecuadas para su uso operativo.

En conjunto, los resultados del capítulo confirman que el modelo no solo es matemáticamente consistente, sino que captura de forma realista los compromisos económicos y operativos inherentes a la planificación del transporte, proporcionando una base sólida para el análisis de conclusiones y líneas futuras de trabajo.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1 Conclusiones

En este Trabajo Fin de Grado se ha desarrollado y evaluado un modelo de optimización para la planificación diaria del transporte, orientado a la asignación eficiente de pedidos entre flota propia y mensajería externa. El problema se ha abordado desde el marco de la **Investigación Operativa**, formulándolo como un modelo de **Programación Lineal Entera Mixta**, capaz de representar de forma rigurosa decisiones discretas, costes operativos y restricciones reales del entorno logístico.

Los resultados obtenidos a lo largo de los experimentos computacionales permiten extraer varias conclusiones relevantes. En primer lugar, el modelo ha demostrado ser **operativamente coherente**, cumpliendo todas las restricciones impuestas incluso en escenarios exigentes. Las decisiones de asignación responden de manera lógica a la disponibilidad de flota, a la tipología de carga y a las limitaciones normativas, reproduciendo un comportamiento alineado con el razonamiento esperado de un planificador logístico experimentado.

En segundo lugar, el análisis económico confirma que el modelo resuelve de forma consistente el dilema **make-or-buy**, activando la flota propia únicamente cuando resulta económicamente ventajoso y recurriendo a la mensajería externa como mecanismo de

apoyo ante situaciones de saturación o incompatibilidad de carga. Esta activación selectiva permite reducir el coste medio por pedido y evita el uso ineficiente de recursos. Asimismo, la introducción de un **incentivo por anticipación** ha demostrado ser un mecanismo eficaz para aprovechar la capacidad ociosa de los camiones, adelantando pedidos futuros sin interferir con los envíos obligatorios del día. Este comportamiento dinámico permite suavizar la carga operativa y mejorar el aprovechamiento de la flota sin imponer restricciones adicionales al modelo.

Desde el punto de vista computacional, los experimentos han puesto de manifiesto que el uso de un *solver* de código abierto como **CBC** permite obtener soluciones de alta calidad en tiempos compatibles con la operativa diaria para instancias de tamaño pequeño y medio. En escenarios de mayor escala, aunque la certificación del óptimo matemático puede resultar costosa, las soluciones cuasi-óptimas obtenidas en tiempos reducidos presentan una **estabilidad estructural** suficiente para su aplicación práctica, lo que refuerza la viabilidad del enfoque propuesto en entornos reales.

Finalmente, el uso de herramientas accesibles y ampliamente conocidas, como **Excel**, combinado con lenguajes de modelado y *solvers* abiertos, facilita la transferencia del modelo a contextos empresariales, permitiendo su utilización por perfiles no técnicos y reduciendo las barreras de adopción.

5.2 Uso de herramientas de inteligencia artificial

Durante el desarrollo del trabajo se han utilizado herramientas de inteligencia artificial como apoyo en tareas auxiliares. En particular, se emplearon asistentes basados en modelos de lenguaje para la revisión estilística y mejora del texto, así como herramientas de apoyo para la depuración y mejora de scripts en Python.

En ningún caso estas herramientas sustituyeron el razonamiento propio ni el diseño del modelo matemático, que fueron desarrollados íntegramente por el autor. El uso de la inteligencia artificial se limitó a funciones de apoyo, alineándose con las buenas prácticas actuales en el uso responsable de estas tecnologías en entornos académicos.

5.3 Competencias adquiridas

El desarrollo de este Trabajo Fin de Grado ha permitido adquirir y aplicar diversas competencias asociadas a la especialización en **Tecnología específica: Computación**.

En particular, el trabajo se alinea con la competencia **CM1**, al abordar el diseño y aplicación de modelos formales de computación para representar y resolver un problema real de planificación logística. La formulación del problema como un modelo de Programación Lineal Entera Mixta ha requerido un conocimiento profundo de los principios fundamentales de la computación y su aplicación práctica para la creación de una solución tecnológica viable.

Asimismo, se ha desarrollado la competencia **CM3**, relacionada con la evaluación de la complejidad computacional y el análisis de estrategias algorítmicas. A lo largo del trabajo se ha estudiado el comportamiento del modelo ante instancias de distinto tamaño, analizando tiempos de cómputo, escalabilidad y el compromiso entre calidad de solución y rendimiento computacional, lo que ha permitido valorar de forma crítica las limitaciones y ventajas de los métodos exactos utilizados.

Por último, el trabajo contribuye al desarrollo de la competencia **CM5**, al formalizar el conocimiento humano implícito en la planificación logística diaria y representarlo de manera computable mediante variables, restricciones y funciones objetivo. Este proceso de abstracción ha permitido transformar una toma de decisiones tradicionalmente manual en un procedimiento sistemático y reproducible, apto para su resolución mediante herramientas informáticas.

En conjunto, el proyecto ha permitido integrar conocimientos teóricos y prácticos de la computación en la resolución de un problema realista, reforzando las competencias fundamentales asociadas a la especialización cursada.

5.4 Trabajo futuro

El modelo propuesto ofrece múltiples posibilidades de ampliación y mejora que podrían abordarse en trabajos futuros. Entre las principales líneas de desarrollo destacan:

- **Extensión del horizonte temporal**, incorporando una planificación multidía que permita gestionar de forma más explícita la anticipación de pedidos y la carga futura de trabajo.
- **Integración explícita del enrutamiento**, combinando la asignación de pedidos con la optimización de rutas, lo que permitiría capturar de forma más precisa los costes reales de transporte.

- **Consideración de incertidumbre**, incorporando elementos estocásticos en la demanda, los tiempos de viaje o la disponibilidad de recursos.
- **Comparación con métodos heurísticos y metaheurísticos**, evaluando su rendimiento frente al modelo exacto en instancias de gran tamaño.
- **Uso de *solvers* comerciales de alto rendimiento**, como Gurobi o CPLEX, con el fin de analizar el impacto del rendimiento computacional en escenarios de alta complejidad. Dado que el modelo ha sido implementado mediante Pyomo, este cambio podría realizarse de forma inmediata simplemente modificando el *solver* empleado en la fase de resolución, sin necesidad de alterar la formulación matemática ni la estructura del modelo.
- **Despliegue en entornos** reales, integrando el modelo con sistemas de información logística y validándolo con datos históricos reales.

En conjunto, este trabajo ha permitido comprobar de forma práctica cómo herramientas matemáticas propias de la Investigación Operativa y la optimización combinatoria pueden aplicarse con éxito a la resolución de problemas complejos reales. La formalización de un proceso de planificación tradicionalmente manual en un modelo computable ha puesto de manifiesto el valor de estas técnicas para mejorar la eficiencia, la trazabilidad y la calidad de la toma de decisiones en entornos industriales. Asimismo, el enfoque desarrollado es fácilmente extensible a otros problemas de planificación y asignación existentes en distintos sectores, lo que refuerza el potencial de estas herramientas como apoyo fundamental en la transformación digital de la industria.

Bibliografía

- [1] Dantzig, G. and Ramser, J. (1959) *The Truck Dispatching Problem*. *Management Science*, 6, 80-91. <http://dx.doi.org/10.1287/mnsc.6.1.80>
- [2] Kuhn, H. W. (1955). *The Hungarian method for the assignment problem*. *Naval Research Logistics Quarterly*, 2(1-2), pp. 83-97. <http://dx.doi.org/10.1002/nav.3800020109>
- [3] Tarantilis, C.D. and Kiranoudis, C.T. (2001). *Using the vehicle routing problem for the transportation of hazardous materials*. *Operational Research International Journal*, 1, pp. 67-78. <https://doi.org/10.1007/BF02936400>
- [4] Keshavarzfard, R. and Naderi, A. (2024). *A sustainable HazMat logistic network design considering scale of economy and route sensitivity and solving with Hybrid GA-SA*. *International Journal of Research in Industrial Engineering*, 13(3), pp. 257-273. <https://doi.org/10.22105/riej.2024.420988.1397>
- [5] Redmer, A. (2022). *Strategic vehicle fleet management – a joint solution of make-or-buy, composition and replacement problems*. *Journal of Quality in Maintenance Engineering*, 28(2), pp. 327-341. <https://doi.org/10.1108/JQME-04-2020-0026>
- [6] Baeza, E., Montt, C. and Quezada, L. (2019). *Methodological proposal to evaluate the alternative of outsourcing the transportation fleet of a company*. *Procedia Manufacturing*, 39, pp. 1545-1551. <https://doi.org/10.1016/j.promfg.2020.01.292>
- [7] Bertoluci, R., Ramos, A. G., Lopes, M. and Bastos, J. (2019). *Capacitated Vehicle Routing Problem with Heterogeneous Fixed Proprietary Fleet and Outsourcing Delivery—A Clustering-Based Approach*. In: Alves, M., Almeida, J., Oliveira, J. and

Pinto, A. (eds) Operational Research. IO 2018. Springer Proceedings in Mathematics & Statistics, vol. 278. Springer, Cham.
https://doi.org/10.1007/978-3-030-10731-4_4

[8] SolverStudio (2024). *SolverStudio User Guide*. Available at:
<https://solverstudio.org>

Anexo I. Implementación del modelo de optimización en Pyomo

I.1 Visión general

El modelo ha sido implementado en Python utilizando la librería Pyomo, integrada en el entorno SolverStudio de Microsoft Excel, lo que permite ejecutar el modelo directamente desde hojas de cálculo sin necesidad de herramientas externas adicionales.

I.2 Declaración del modelo abstracto

```
from pyomo.environ import *  
model = AbstractModel()
```

Se declara un modelo abstracto de Pyomo, lo que permite separar la formulación matemática de los datos específicos de cada instancia, facilitando la reutilización del código para distintos escenarios de prueba.

I.3 Conjuntos

```
model.I = Set()  
model.J = Set()  
model.C = Set()
```

Se definen tres conjuntos fundamentales:

- **I**, conjunto de pedidos a planificar.
- **J**, conjunto de camiones disponibles en la flota.
- **C**, conjunto de clientes a los que se realizan los envíos.

I.4 Parámetros

```
model.vol = Param(model.I)
model.pes = Param(model.I)
model.fecha = Param(model.I, within=Reals)
model.adr = Param(model.I)
model.t = Param(model.I)
model.cli = Param(model.I, within=model.C)
```

Estos parámetros caracterizan los pedidos, incluyendo su volumen, peso, fecha asociada, condición de mercancía peligrosa (ADR), coste de envío mediante mensajería externa y cliente al que pertenece el pedido.

```
model.dist_c = Param(model.C, within=Reals)
```

Este parámetro indica la distancia a la que se encuentra el cliente de la base.

```
model.V = Param(model.J)
model.W = Param(model.J)
model.ADRmax = Param(model.J)
model.Pmax = Param(model.J)
model.F = Param(model.J, within=Reals)
```

Estos parámetros describen las características de los camiones, tales como capacidades de volumen y peso, límite ADR, número máximo de paradas y coste fijo de activación.

```
model.alpha = Param(within=Reals)
model.s = Param(within=Reals)
model.fecha_hoy = Param(within=Reals)
```

Parámetros globales que indican el factor de €/km (alpha), el coeficiente de incentivo temporal (s) y la fecha del día actual.

Todos estos valores son proporcionados externamente a través de los ficheros de datos en el Excel o generados de forma automática en los experimentos computacionales, lo que permite modificar los escenarios sin alterar el código del modelo.

I.5 Parámetros derivados

```
def dist_i_rule(m, i):
    return m.dist_c[m.cli[i]]
model.dist = Param(model.I, initialize=dist_i_rule)
```

Se calcula la distancia asociada a cada pedido a partir del cliente al que pertenece.

```
def u_rule(m, i):
    return m.alpha * m.dist[i]
model.u = Param(model.I, initialize=u_rule)
```


Se define el coste variable de transporte como proporcional a la distancia recorrida.

```
def d_rule(m, i):
    return m.fecha[i] - m.fecha_hoy
model.d = Param(model.I, initialize=d_rule)
```

Se calcula el número de días de antelación respecto a la fecha actual, utilizado posteriormente para incentivar el envío anticipado de pedidos futuros.

```
def fmax_rule(m):
    return max(m.fecha[i] for i in m.I)
model.Fmax = Param(initialize=fmax_rule)
```

Se recupera la fecha del envío más lejano.

```
def delta_rule(m, i):
    if m.Fmax > 0:
        return (m.Fmax - m.d[i])/m.Fmax
    else:
        return 1.0
model.delta = Param(model.I, initialize=delta_rule)
```

Se calcula el beneficio por anticipación, normalizado entre 0 y 1, siendo 1 un beneficio máximo (envíos de hoy).

I.6 Variables

```
model.x = Var(model.I, model.J, domain=Binary)
model.y = Var(model.I, domain=Binary)
model.z = Var(model.J, domain=Binary)
```

Las variables binarias representan la asignación de pedidos a camiones, la externalización mediante mensajería y la activación de camiones.

I.7 Función Objetivo

```
def obj_rule(m):
    return (
        sum(m.u[i] * m.x[i, j] for i in m.I for j in m.J)
        + sum(m.t[i] * m.y[i] for i in m.I)
        + sum(m.F[j] * m.z[j] for j in m.J)
        - sum(m.s * m.delta[i] * (sum(m.x[i, j] for j in m.J) +
m.y[i]))
        for i in m.I)
    )
model.OBJ = Objective(rule=obj_rule, sense=minimize)
```

La función objetivo minimiza el coste total diario, incorporando costes variables, costes fijos, costes de mensajería y el término de incentivo por anticipación. Este

planteamiento permite analizar el compromiso entre coste económico y aprovechamiento de capacidad.

I.8 Restricciones

```
def hoy_rule(m, i):
    if m.fecha[i] == m.fecha_hoy:
        return sum(m.x[i, j] for j in m.J) + m.y[i] == 1
    return Constraint.Skip
model.envio_hoy = Constraint(model.I, rule=hoy_rule)
```

Garantiza que todos los pedidos con fecha actual sean enviados.

```
def futuro_rule(m, i):
    if m.fecha[i] > m.fecha_hoy:
        return sum(m.x[i, j] for j in m.J) <= 1
    return Constraint.Skip
model.envio_fut = Constraint(model.I, rule=futuro_rule)
```

Permite el envío de pedidos futuros.

```
def futuro_mens_rule(m, i):
    if m.fecha[i] > m.fecha_hoy:
        return m.y[i] == 0
    return Constraint.Skip
model.futuro_mens = Constraint(model.I, rule=futuro_mens_rule)
```

Evita que los pedidos futuros puedan ser enviados por mensajería externa.

```
def linking_rule(m, i, j):
    return m.x[i, j] <= m.z[j]
model.link = Constraint(model.I, model.J, rule=linking_rule)
```

Impide que se asignen pedidos a camiones no activados.

```
def volumen_rule(m, j):
    return sum(m.vol[i] * m.x[i, j] for i in m.I) <= m.V[j]
model.volumen = Constraint(model.J, rule=volumen_rule)
```

Asegura que no se supere la capacidad de volumen del camión.

```
def peso_rule(m, j):
    return sum(m.pes[i] * m.x[i, j] for i in m.I) <= m.W[j]
model.peso = Constraint(model.J, rule=peso_rule)
```

Asegura que no se supere la capacidad de peso del camión.

```
def adr_rule(m, j):
    return sum(m.adr[i] * m.vol[i] * m.x[i, j] for i in m.I) <=
m.ADRmax[j]
model.adr_limit = Constraint(model.J, rule=adr_rule)
```

Asegura que no se supere la capacidad de ADR del camión.

```
def paradas_rule(m, j):  
    return sum(m.x[i, j] for i in m.I) <= m.Pmax[j]  
model.paradas = Constraint(model.J, rule=paradas_rule)
```

Asegura que no se supere el número de paradas del camión.