



Universidad de Castilla-La Mancha
Escuela Superior de Ingeniería Informática

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

Ayuda en la optimización de la planificación de viajes a destinos

Raúl Alarcón López

Junio, 2025



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

Ayuda en la optimización de la planificación
de viajes a destinos

Autor: Raúl Alarcón López

Tutor: Francisco Parreño Torres

Junio, 2025

*Dedicado a mi familia y a todos
aquellos que me han apoyado
durante este largo camino...*

Declaración de Autoría

Yo, Raúl Alarcón López con DNI 14278648H, declaro que soy el único autor del trabajo fin de grado titulado “Ayuda en la optimización de la planificación de viajes a destinos” 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 14 de junio de 2025

Fdo: Raúl Alarcón López

Resumen

Este TFG se centra en la optimización de la planificación de viajes en una empresa cárnica, con el objetivo de mejorar la eficiencia en la logística de transporte. Utiliza técnicas de investigación operativa y programación lineal entera mixta para modelar y resolver el problema.

El documento describe el problema actual de planificación manual, que implica la compra y transporte de más de 2.800.000 cerdos al año, generando una alta complejidad en la cadena de suministro. Para abordar esto, se implementa un modelo matemático en SolverStudio con Pyomo, considerando restricciones de capacidad, demanda, logística y normativas del sector.

La metodología sigue un enfoque basado en Scrum, con iteraciones de diseño, implementación y validación. Se realizan pruebas para evaluar la eficacia del modelo, comparando su rendimiento con la planificación manual realizada actualmente. Los resultados muestran mejoras en la asignación de recursos y reducción de costes.

El trabajo concluye destacando la utilidad del modelo para optimizar la logística y propone futuras líneas de investigación, como la mejora del algoritmo y la incorporación de más factores en la toma de decisiones.

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia todo el apoyo durante todos estos años y la oportunidad de poder acabar mis estudios.

En segundo lugar, me gustaría dar las gracias a mis compañeros de carrera que se han convertido en amigos para toda la vida.

Y, por último, a mi tutor Francisco por la paciencia y la ayuda para el desarrollo de este trabajo.

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	5
2.1	Introducción	5
2.2	¿Qué es la optimización?	6
2.3	Complejidad computacional de un algoritmo	6
2.4	Investigación Operativa	7
2.4.1	Métodos y Técnicas	8
2.4.2	Aplicaciones de la Investigación Operativa	8
2.5	Modelos de programación entera	9
2.5.1	Definición de un modelo de programación entera	9
2.5.2	Componentes de un modelo de programación entera	9
2.5.3	Tipos de programación entera	9
2.5.4	Aplicaciones de programación entera	10
2.6	Metodologías	10
2.6.1	Métodos exactos	10
2.6.2	Métodos de resolución aproximada	12
2.7	Herramientas y Tecnologías	14
2.7.1	Integración con Excel y otros recursos tecnológicos	14
2.7.2	Descripción del uso de SolverStudio con Pyomo	15
2.8	El problema de la optimización de planificación de viajes a destino	20

2.8.1	Restricciones de capacidad	20
2.8.2	Cumplimiento de la demanda	20
2.8.3	Logística de transporte	21
2.8.4	Requisitos adicionales de las granjas	21
Capítulo 3	Metodología y Desarrollo	23
3.1	Introducción	23
3.2	Diseño del Modelo	25
3.2.1	Notación	25
3.2.2	Modelo mixto de programación lineal entera	27
3.2.3	Restricciones adicionales de transporte	28
3.3	Implementación del Algoritmo	31
3.3.1	Definición del Modelo	31
3.3.2	Definición de conjuntos y parámetros	31
3.3.3	Variables de decisión	31
3.3.4	Función objetivo	32
3.3.5	Restricciones del Modelo	32
3.3.6	Flujo de trabajo	33
3.3.7	Resolución del Modelo	33
3.3.8	Pseudocódigo del Modelo	34
3.4	Validación y Pruebas Iniciales	34
3.4.1	Métodos para verificar la correcta implementación del modelo	34
3.4.2	Presentación de casos de prueba	35
3.5	Discusión de Dificultades y Soluciones	36
3.6	Conclusiones del Capítulo	37
Capítulo 4	Experimentos y Resultados	39
4.1	Introducción	39
4.2	Diseño Experimental	40
4.2.1	Descripción del conjunto de datos, escenarios y parámetros configurados	40
4.2.2	Herramientas y entornos utilizados	42
4.3	Ejecución de los Experimentos	43
4.3.1	Ejecución de las pruebas	43
4.3.2	Criterios de evaluación y métricas de desempeño	45
4.4	Presentación de Resultados	46

4.4.1	Tiempo de ejecución	46
4.4.2	Comparación entre demanda y asignación	47
4.4.3	Análisis de costes y eficiencia del transporte	47
4.4.4	Análisis de sensibilidad	48
4.4.5	Comparativas entre distintos escenarios	49
4.5	Análisis y Discusión	52
4.5.1	Interpretación de los resultados obtenidos	52
4.5.2	Identificación de patrones	52
4.5.3	Fortalezas del algoritmo	53
4.5.4	Debilidades del algoritmo	53
4.6	Conclusiones del Capítulo	53
Capítulo 5	Conclusiones y Trabajo Futuro	55
5.1	Conclusiones Generales	55
5.1.1	Recapitulación de los objetivos planteados	55
5.1.2	Cumplimiento de los objetivos	56
5.1.3	Principales aportaciones del trabajo	56
5.2	Limitaciones y Retos	57
5.2.1	Principales limitaciones	57
5.3	Implicaciones Prácticas	58
5.3.1	Beneficios potenciales para la empresa	58
5.3.2	Impacto en el sector logístico	58
5.4	Líneas de Trabajo Futuro	58
5.4.1	Mejoras en el modelo actual	59
5.4.2	Nuevas áreas de investigación y aplicación	59
5.5	Reflexión Final	59
Bibliografía		61

Índice de figuras

Figura 1. Ejemplo de problema con Pyomo	17
Figura 2. Ejemplo de Algoritmo con Pyomo	17
Figura 3. Solución al problema	18
Figura 4. Pseudocódigo del algoritmo Pyomo	19
Figura 5. Definición del modelo en Pyomo	31
Figura 6. Definición de conjunto y parámetros	31
Figura 7. Valores restrictivos.....	31
Figura 8. Función objetivo	32
Figura 9. Pseudocódigo del modelo.....	34
Figura 10. Comparación entre demanda y asignación.....	47
Figura 11. Análisis de sensibilidad	48
Figura 12. Distribución de carga	49
Figura 13. Comparaciones demanda satisfecha	51
Figura 14. Comparación costes por escenario	52

Índice de tablas

Tabla 1. Tiempos de ejecución.....	46
Tabla 2. Costes con los diferentes escenarios	47
Tabla 3. Descripción de los escenarios	50
Tabla 4. Comparación tiempos de ejecución.....	50
Tabla 5. Comparación calidad de la solución.....	50
Tabla 6. Impacto económico.....	51

Capítulo 1

Introducción

Las referencias bibliográficas para la redacción de este capítulo han sido: (ESIIAB, 2024) y (Incarlopsa)

1.1 Introducción

En un mundo cada vez más interconectado, la optimización de recursos se ha convertido en un pilar fundamental para el éxito de las empresas. Este Trabajo de Fin de Grado (TFG) se centra en el desarrollo de un modelo de optimización para la asignación de viajes a destinos específicos, abordando un problema clásico de la investigación operativa que tiene un impacto significativo en la eficiencia operativa. En particular, este proyecto se implementará en una empresa cárnica de la región, donde la planificación de rutas y el uso eficiente de los recursos de transporte son cruciales para garantizar la frescura y calidad de los productos. A través de la aplicación de técnicas avanzadas de optimización, se busca no solo mejorar la logística de distribución, sino también contribuir a la sostenibilidad y rentabilidad de la empresa. Este trabajo no solo representa un desafío académico, sino también una oportunidad para generar un impacto positivo en el sector cárnico local.

INCARLOPSA es una de las empresas de referencia a nivel global en el sector de la alimentación para productos cárnicos de porcinos frescos, curados e ibéricos.

La creciente demanda y diversificación de suministro de ganado porcino que estamos

experimentado en los últimos años y el incremento de requerimientos por planta de sacrificio y por categoría de suministro, repercuten en un incremento de la complejidad del proceso de programación semanal de las partidas de compra de ganado y de la logística del transporte de animal vivo.

Actualmente, la programación semanal de las partidas de compra de animal vivo está basada en la resolución de problemas matemáticos de tipo combinatorio resueltos por iteraciones con múltiples soluciones por parte del personal del Departamento de Compras de Ganado de INCARLOPSA bajo una elevada tensión de la cadena de suministro y sin el apoyo de herramientas digitales. Todo ello considerando un volumen actual aproximado más de 2.800.000 animales/año, lo que equivale a unas 16.000 partidas de compra/año, y llegando a corto plazo a las 20.000 partidas de compra/año. Debido al incremento de la complejidad de dichos problemas matemáticos es previsible que el riesgo de errores en el cumplimiento de requerimientos, la dedicación a su resolución y el tiempo de respuesta ante nuevos requerimientos establecidos en tiempo real aumenten de forma significativa.

Por otro lado, el sistema actual impide una adecuada optimización del transporte de animal vivo según la huella de carbono y una reducción del tiempo de viaje desde la explotación hasta el matadero, así como disminuir los tiempos de espera en la descarga de los camiones para incrementar el bienestar animal.

Este problema actualmente se está resolviendo manualmente por dos personas y mediante este trabajo se va a automatizar para obtener la solución mediante un modelo en el que contamos con mataderos y distintos proveedores en el que cada matadero tiene unas demandas específicas de cerdos y los proveedores ofrecen una cantidad de cerdos por día que puede suministrar de cada tipo.

1.2 Objetivos

El presente trabajo tiene como objetivo desarrollar un modelo de optimización para asignar viajes a destinos específicos, un problema típico de la investigación operativa.

El proyecto se implementará en una empresa cárnica de la región, con el objetivo de mejorar la eficiencia en la planificación de rutas y el uso de los recursos de transporte. Para su desarrollo, se emplearán herramientas como Excel y software de optimización

matemática, para lograr obtener una solución práctica y ajustada a las necesidades de la empresa cárnica.

Para conseguir esto, seguiremos la siguiente metodología:

1. Estudio del problema para entender los principales elementos y la utilidad del algoritmo.
2. Estudio del estado del arte, qué algoritmos existen, cómo abordan el problema, qué resuelven, qué técnicas utilizan y cómo lo hacen.
3. Implementación del modelo.
4. Testeo y pruebas del modelo implementado.

Las competencias de la rama de Computación que serán aplicadas durante el trabajo son las siguientes:

- Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
- Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.
- Capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en cualquier ámbito de aplicación, particularmente los relacionados con aspectos de computación, percepción y actuación en ambientes entornos inteligentes.

1.3 Estructura del proyecto

El trabajo se estructura en cinco capítulos de la siguiente manera:

1. **Introducción:** Se realiza una presentación del problema a abordar, con sus objetivos y la metodología a seguir. Además, se detallará la estructura del trabajo, incluyendo una breve descripción de los capítulos que lo componen.

2. **Estado del Arte:** Se profundiza en el problema de la optimización de rutas, desde sus fundamentos en optimización hasta los principales modelos y técnicas heurísticas existentes para abordarlo. Se profundiza en el modelo seleccionado y las técnicas aplicadas en este estudio.
3. **Metodología y desarrollo:** Se detalla la metodología empleada, incluyendo una descripción de los conjuntos de datos, los algoritmos y su implementación eficiente.
4. **Experimentos y resultados:** Se presentan los resultados de los algoritmos, mostrando los mismos con gráficos y tablas, además de un análisis estadístico.
5. **Conclusiones y trabajo a futuro:** Se recogen todas las conclusiones obtenidas mediante el desarrollo de este trabajo. Se discuten las implicaciones y se proponen líneas futuras de investigación.

Capítulo 2

Estado del Arte

Las referencias bibliográficas para la redacción de este capítulo han sido: (Bazaraa, 2010), (Bertsimas, 1997), (Dantzig, 1963), (Hillier, 2021), (Mason, 2021), (Lieberman, 2010), (Mokhtar S. Bazaraa, 2010), (Nemhauser, 1999), (Python Software Foundation. (2025). Documentación oficial de Pyomo, 2024), (SolverStudio, s.f.), (Taha, 2017), (Winston, 2004) y (Zgurovsky, 2010)

2.1 Introducción

En la actualidad, la logística y la gestión eficiente de los recursos de transporte son aspectos cruciales para el éxito de cualquier empresa que dependa de la distribución de productos. La planificación de cargas y la asignación de viajes a destinos específicos representan un desafío importante, especialmente en sectores donde el tiempo y los costes son determinantes. Este TFG se centra en la optimización de la planificación de cargas de viajes, con el objetivo de desarrollar un modelo que permita mejorar la eficiencia en la asignación de recursos y la gestión de rutas.

A través del uso de técnicas de investigación operativa, se busca abordar las complejidades propias a la logística moderna, considerando factores como la capacidad de carga, las restricciones de tiempo y los costes asociados. A lo largo de este trabajo, se explorarán diferentes metodologías y herramientas que permitirán alcanzar una planificación más efectiva, beneficiando tanto a la empresa como a sus clientes.

2.2 ¿Qué es la optimización?

La optimización es un área fundamental en las matemáticas aplicadas y la investigación operativa, que se centra en la búsqueda de la mejor solución a un problema específico, maximizando o minimizando una función objetivo sujeta a un conjunto de restricciones. En un mundo donde los recursos son limitados y las decisiones deben tomarse de manera eficiente, la optimización se convierte en una herramienta esencial para la toma de decisiones en diversas disciplinas, como la economía, la ingeniería, la logística, la ciencia de datos y más.

El proceso de optimización implica formular un problema de manera matemática, identificando las variables relevantes, la función que se desea optimizar y las restricciones que deben cumplirse. Existen diferentes tipos de problemas de optimización, que pueden clasificarse en lineales y no lineales, así como en problemas con variables discretas o continuas. Cada tipo de problema requiere enfoques y técnicas específicas para su resolución, lo que hace que la optimización sea un campo diverso y dinámico.

A lo largo de este trabajo, se explorarán las metodologías más utilizadas para abordar problemas prácticos. Se presentarán ejemplos concretos que ilustran la aplicación de técnicas de optimización en situaciones del mundo real, destacando su relevancia y utilidad. Además, se discutirá el impacto de la optimización en la mejora de procesos y la toma de decisiones estratégicas, subrayando su importancia en un entorno cada vez más competitivo y complejo.

En resumen, la optimización no solo es una herramienta matemática, sino también un enfoque crítico para resolver problemas y mejorar la eficiencia en diversas áreas. Este trabajo tiene como objetivo proporcionar una comprensión profunda de los principios de la optimización y su aplicación práctica, contribuyendo así al desarrollo de soluciones efectivas en contextos reales.

2.3 Complejidad computacional de un algoritmo

La complejidad computacional de un algoritmo es una medida que permite evaluar la eficiencia de dicho algoritmo en términos de los recursos que consume, principalmente el tiempo de ejecución y el espacio de memoria. Esta evaluación es fundamental en el

campo de la informática, ya que ayuda a determinar la viabilidad de un algoritmo para resolver problemas de diferentes tamaños y complejidades.

Existen dos tipos principales de complejidad computacional:

1. **Complejidad Temporal:** Se refiere al tiempo que un algoritmo tarda en completarse en función del tamaño de la entrada. Se expresa comúnmente utilizando la notación Big O ($O(n)$, $O(\log n)$, $O(n^2)$, etc.), que describe el comportamiento asintótico del tiempo de ejecución a medida que el tamaño de la entrada crece. Por ejemplo, un algoritmo con complejidad $O(n)$ se considera lineal, lo que significa que el tiempo de ejecución aumenta proporcionalmente al tamaño de la entrada.
2. **Complejidad Espacial:** Se refiere a la cantidad de memoria que un algoritmo utiliza en función del tamaño de la entrada. Al igual que la complejidad temporal, se puede expresar en notación Big O. La complejidad espacial es crucial en situaciones donde los recursos de memoria son limitados, ya que un algoritmo que consume demasiada memoria puede no ser práctico, incluso si su tiempo de ejecución es aceptable.

La evaluación de la complejidad computacional permite a los desarrolladores y científicos de datos seleccionar el algoritmo más adecuado para un problema específico, considerando tanto el tiempo como el espacio requerido. Además, proporciona una base para comparar diferentes algoritmos que resuelven el mismo problema, ayudando a identificar cuál es el más eficiente en términos de recursos.

En resumen, la complejidad computacional es un concepto clave en el análisis de algoritmos, ya que proporciona una forma de medir y comparar la eficiencia de diferentes enfoques para resolver problemas computacionales. Comprender la complejidad de un algoritmo no solo es esencial para el diseño de software eficiente, sino que también es fundamental para el desarrollo de soluciones escalables en un mundo donde los datos y las demandas computacionales continúan creciendo.

2.4 Investigación Operativa

La investigación operativa (IO) es una disciplina que utiliza métodos analíticos avanzados para ayudar a tomar mejores decisiones. Se basa en la aplicación de técnicas matemáticas, estadísticas y computacionales para resolver problemas complejos en

diversas áreas, como la logística, la producción, la planificación y la gestión de recursos. Su objetivo principal es optimizar el rendimiento de sistemas y procesos, maximizando o minimizando funciones objetivo bajo ciertas restricciones.

2.4.1 Métodos y Técnicas

La investigación operativa abarca una variedad de métodos y técnicas, entre los que se incluyen:

- **Programación Lineal:** Se utiliza para maximizar o minimizar una función lineal sujeta a restricciones lineales. Es uno de los métodos más comunes en IO y se aplica en problemas de asignación de recursos y planificación.
- **Programación Entera:** Similar a la programación lineal, pero con la restricción de que algunas o todas las variables deben ser enteras. Es útil en problemas donde las decisiones son discretas.
- **Teoría de Juegos:** Estudia situaciones en las que múltiples tomadores de decisiones interactúan, cada uno buscando maximizar su propio beneficio. Se aplica en economía, política y estrategia empresarial.
- **Simulación:** Permite modelar sistemas complejos y analizar su comportamiento bajo diferentes condiciones. Es especialmente útil cuando los modelos analíticos son difíciles de aplicar.
- **Análisis de Redes:** Se utiliza para optimizar flujos en redes, como en problemas de transporte y comunicación. Incluye técnicas como el algoritmo de Dijkstra y el método de flujo máximo.

2.4.2 Aplicaciones de la Investigación Operativa

La investigación operativa tiene aplicaciones en una amplia gama de campos, tales como:

- **Logística y Transporte:** Optimización de rutas de entrega, gestión de inventarios y planificación de la cadena de suministro.
- **Producción:** Programación de la producción, gestión de la calidad y optimización de procesos.
- **Salud:** Planificación de recursos hospitalarios, gestión de citas y optimización de tratamientos.

- **Finanzas:** Análisis de riesgos, optimización de carteras y gestión de inversiones.

2.5 Modelos de programación entera

La programación entera es una rama de la programación matemática que se ocupa de la optimización de problemas en los que algunas o todas las variables deben tomar valores enteros. Este enfoque es especialmente útil en situaciones donde las decisiones son discretas, como la asignación de recursos, la planificación de horarios y la gestión de inventarios.

2.5.1 Definición de un modelo de programación entera

Un modelo de programación entera se define como un conjunto de ecuaciones y desigualdades que representan un problema de optimización, donde las variables de decisión son números enteros. Estos modelos se utilizan para encontrar la mejor solución posible, ya sea maximizar o minimizar una función objetivo, sujeta a ciertas restricciones.

2.5.2 Componentes de un modelo de programación entera

Los principales componentes de un modelo de programación entera son:

- **Variables de Decisión:** Son las incógnitas del modelo que se desean determinar. En un modelo de programación entera, estas variables deben ser enteras. Por ejemplo, en un problema de asignación de tareas, las variables podrían representar el número de trabajadores asignados a cada tarea.
- **Función Objetivo:** Es la función que se desea maximizar o minimizar. Por ejemplo, en un problema de maximización de beneficios, la función objetivo podría ser la suma de los beneficios generados por cada tarea completada.
- **Restricciones:** Son las limitaciones o condiciones que deben cumplirse en la solución del problema. Estas pueden incluir limitaciones de recursos, capacidades, o cualquier otra condición que afecte la viabilidad de las soluciones.

2.5.3 Tipos de programación entera

Existen varios tipos de programación entera, entre los que destacan:

- **Programación Entera Mixta (MIP):** En este tipo, algunas variables pueden ser enteras y otras pueden ser continuas. Esto permite una mayor flexibilidad en la modelización de problemas complejos.
- **Programación Entera Pura:** En este caso, todas las variables de decisión son enteras. Este tipo de modelo es común en problemas donde las decisiones son discretas, como la selección de proyectos o la asignación de recursos.

2.5.4 Aplicaciones de programación entera

La programación entera se aplica en diversas áreas, incluyendo:

- **Logística y Transporte:** Para optimizar rutas de entrega y asignación de vehículos.
- **Producción:** En la planificación de la producción y la gestión de inventarios.
- **Telecomunicaciones:** Para la asignación de frecuencias y recursos en redes.
- **Finanzas:** En la optimización de carteras de inversión.

2.6 Metodologías

Para resolver problemas de optimización se pueden utilizar dos metodologías: los métodos exactos y los de resolución aproximada. En este punto vamos a explicar estas metodologías, sus características y algunos de los algoritmos más conocidos.

2.6.1 Métodos exactos

Los métodos exactos son aquellos que garantizan encontrar la solución óptima de un problema, es decir, la mejor solución posible bajo las restricciones y objetivos establecidos. Estos métodos se diferencian de los heurísticos, que solo buscan soluciones aproximadas y no garantizan la optimalidad. Los métodos exactos son fundamentales en la teoría y práctica de la optimización debido a su capacidad para asegurar que se alcanza la mejor solución posible, aunque su complejidad computacional pueda ser mayor.

2.6.1.1 Método Simplex

El Método Simplex es uno de los algoritmos más conocidos y utilizados en la optimización lineal. Se aplica a problemas de programación lineal (PL), donde el objetivo

es maximizar o minimizar una función lineal sujeta a un conjunto de restricciones lineales. El Simplex se basa en la idea de recorrer los vértices de la región factible (el conjunto de soluciones posibles) en busca de la solución óptima.

Aunque en el peor de los casos el Método Simplex puede ser exponencial en el número de pasos, en la práctica se ha demostrado que, para la mayoría de los problemas reales, el algoritmo es extremadamente eficiente y suele converger rápidamente a la solución óptima.

2.6.1.2 Programación entera

Los problemas de programación entera (PE) son aquellos en los que las variables de decisión deben tomar valores enteros. A menudo, estos problemas surgen en situaciones como la asignación de recursos discretos, la planificación de rutas, y la optimización de redes. Los métodos exactos para resolver problemas de programación entera incluyen:

- **Método de ramificación y acotación (Branch and Bound):** Este es uno de los métodos más comunes para la optimización de programación entera. El algoritmo se basa en dividir el problema original en subproblemas más pequeños (ramificación), resolver estos subproblemas de manera recursiva y acotar las soluciones (acotación) para descartar aquellos subproblemas que no pueden proporcionar una mejor solución que la actual mejor solución conocida. Este proceso se repite hasta encontrar la solución óptima.
- **Método de corte (Cutting Plane):** Este método también se utiliza en programación entera y consiste en agregar restricciones adicionales al problema original (llamadas "planos de corte") para excluir soluciones no enteras. El algoritmo mejora progresivamente la aproximación de la solución entera óptima.

2.6.1.3 Método de búsqueda exhaustiva

El método de búsqueda exhaustiva, o fuerza bruta, es el enfoque más simple pero también el más costoso en términos computacionales. En este método, se generan todas las posibles soluciones al problema y se evalúan para determinar cuál es la mejor. Aunque garantiza encontrar la solución óptima, su complejidad crece rápidamente con

el tamaño del problema, por lo que solo es práctico para problemas pequeños o en casos donde las soluciones posibles son limitadas.

2.6.1.4 Algoritmos de programación dinámica

La programación dinámica es un enfoque utilizado en problemas de optimización que involucran decisiones secuenciales o problemas multietapa. Este método se basa en dividir un problema grande en subproblemas más pequeños y resolverlos de manera eficiente, guardando los resultados de los subproblemas resueltos para evitar recalcularlos varias veces. La programación dinámica es particularmente útil en problemas como el problema de la mochila, la optimización de cadenas de suministro, o el cálculo de rutas óptimas.

Existen dos enfoques fundamentales en programación dinámica:

- **Enfoque top-down:** Resuelve el problema recursivamente y almacena las soluciones de los subproblemas en una tabla.
- **Enfoque bottom-up:** Resuelve los subproblemas de menor tamaño primero y construye gradualmente la solución óptima.

2.6.1.5 Método de programación no lineal

La programación no lineal trata de resolver problemas en los que la función objetivo o las restricciones no son lineales. Los métodos exactos en programación no lineal incluyen:

- **Método de puntos interiores:** Este método trabaja dentro del conjunto factible, moviéndose a través del interior de la región factible hasta encontrar la solución óptima.
- **Método de gradiente:** Utiliza la derivada de la función objetivo para encontrar el mínimo o máximo de una función en problemas sin restricciones lineales.

2.6.2 Métodos de resolución aproximada

Los métodos de resolución aproximada se han desarrollado como herramientas efectivas para abordar problemas de optimización que son difíciles o imposibles de resolver de manera exacta. Estos métodos, que incluyen técnicas como algoritmos genéticos, recocido simulado, optimización por enjambre de partículas y algoritmos de búsqueda local, permiten encontrar soluciones que son "suficientemente buenas" en un

tiempo razonable. A menudo, estos enfoques son inspirados por procesos naturales o heurísticas que imitan la forma en que los seres vivos resuelven problemas en su entorno.

2.6.2.1 Métodos heurísticos

Los métodos heurísticos son estrategias que buscan encontrar soluciones satisfactorias a problemas difíciles, a menudo a través de un proceso de prueba y error. En lugar de seguir un procedimiento riguroso y exhaustivo, estos métodos se basan en reglas generales, intuiciones y experiencias previas. Esto los hace especialmente útiles en situaciones donde el espacio de búsqueda es muy grande o donde los datos son incompletos.

Entre ellos, podemos destacar:

- **Algoritmo Greedy:** Es un método heurístico que toma decisiones locales óptimas en cada paso con la esperanza de encontrar una solución global óptima. En cada paso, selecciona la opción más prometedora sin considerar el impacto a largo plazo.
- **Búsqueda aleatoria:** Método heurístico simple que genera soluciones aleatorias y evalúa su calidad. Puede ser útil como punto de partida para algoritmos más sofisticados.
- **Métodos constructivos:** Construyen paso a paso una solución del problema mediante iteraciones. En cada paso se elige la opción que parece más prometedora según unos criterios establecidos.
- **Métodos de búsqueda local:** Parten de una solución inicial y la mejoran progresivamente.

2.6.2.2 Métodos metaheurísticos

Los métodos metaheurísticos son estrategias de búsqueda que guían otros procedimientos heurísticos para explorar el espacio de soluciones de manera más efectiva. A diferencia de los algoritmos exactos, que buscan la solución óptima de manera exhaustiva, los metaheurísticos buscan soluciones "suficientemente buenas" en un tiempo razonable, lo que los hace muy útiles en problemas de optimización combinatoria y continua.

Entre ellos podemos destacar:

- **Algoritmos Genéticos:** Basados en la teoría de la evolución, utilizan operadores como la selección, el cruce y la mutación para generar nuevas soluciones.
- **Recocido Simulado (Simulated Annealing):** Inspirado en el proceso de enfriamiento de metales, permite aceptar soluciones peores temporalmente para escapar de óptimos locales.
- **Optimización por Enjambre de Partículas (Particle Swarm Optimization):** Se basa en el comportamiento social de grupos de animales, donde cada "partícula" representa una solución y se mueve en el espacio de búsqueda.
- **Algoritmos de Colonia de Hormigas:** Simulan el comportamiento de las hormigas al buscar caminos cortos entre su colonia y fuentes de alimento, utilizando feromonas para guiar la búsqueda.
- **Búsqueda Tabú:** Utiliza una memoria para evitar volver a soluciones ya exploradas, lo que ayuda a diversificar la búsqueda.

2.7 Herramientas y Tecnologías

En esta sección vamos a enumerar las distintas herramientas que tenemos para integrar Excel con modelos de optimización y posteriormente, profundizaremos en la elegida para nuestro TFG (SolverStudio con Pyomo).

2.7.1 Integración con Excel y otros recursos tecnológicos

- **Solver de Excel:** Esta es la herramienta más básica y común para la optimización en Excel. Solver permite encontrar el valor óptimo de una celda objetivo, ajustando las celdas de decisión bajo ciertas restricciones. Es ideal para problemas de programación lineal y no lineal.
- **OpenSolver:** Es un complemento gratuito que amplía las capacidades de Solver. Permite resolver problemas más grandes y complejos que el Solver estándar, utilizando el motor de optimización de COIN-OR. Es especialmente útil para problemas de programación entera y mixta.
- **Analytic Solver:** Esta es una herramienta más avanzada que ofrece capacidades de optimización, simulación y análisis de decisiones. Permite resolver problemas

de optimización lineal, no lineal y de programación entera, y tiene una interfaz amigable que se integra bien con Excel.

- **Frontline Solver:** Este es un conjunto de herramientas que incluye Solver, pero también ofrece capacidades adicionales para análisis de sensibilidad y simulación. Es ideal para usuarios que necesitan realizar análisis más profundos y detallados.
- **Excel con VBA (Visual Basic for Applications):** Si tienes conocimientos de programación, puedes crear tus propios algoritmos de optimización utilizando VBA. Esto te permite personalizar completamente el proceso de optimización según tus necesidades específicas.
- **GAMS (General Algebraic Modeling System):** Aunque no es un complemento directo de Excel, GAMS se puede utilizar junto con Excel para modelar y resolver problemas de optimización complejos. Puedes exportar datos desde Excel a GAMS y luego importar los resultados de vuelta a Excel.
- **R y Python:** Si bien no son herramientas de Excel per se, puedes usar R o Python para realizar optimizaciones más complejas y luego importar los resultados a Excel. Existen bibliotecas específicas en ambos lenguajes que facilitan la optimización.

2.7.2 Descripción del uso de SolverStudio con Pyomo

SolverStudio es una herramienta de optimización que se integra con Microsoft Excel, diseñada para facilitar la formulación y resolución de problemas de optimización de manera intuitiva y accesible. Esta aplicación permite a los usuarios modelar problemas complejos utilizando el entorno familiar de Excel, combinando la flexibilidad de las hojas de cálculo con potentes capacidades de optimización.

Pyomo es un marco de modelado en Python que se utiliza para formular y resolver problemas de optimización matemática. Es especialmente popular en la comunidad de investigación y en aplicaciones industriales debido a su flexibilidad y capacidad para manejar una amplia variedad de problemas de optimización, incluyendo programación lineal, programación entera, programación no lineal y problemas de optimización estocástica.

Una de las principales características de SolverStudio es su capacidad para trabajar con diferentes lenguajes de modelado, como GAMS, AMPL y Python, lo que permite a los usuarios elegir el lenguaje que mejor se adapte a sus necesidades y preferencias. Esto amplía las posibilidades de modelado y solución, ya que cada uno de estos lenguajes ofrece herramientas y bibliotecas específicas para abordar problemas de optimización de diversas naturalezas.

SolverStudio es especialmente útil en contextos académicos y profesionales, ya que permite a los usuarios construir modelos de optimización de manera visual y sencilla. Los usuarios pueden definir variables, restricciones y funciones objetivo directamente en las celdas de Excel, lo que facilita la comprensión y modificación del modelo.

Además, la integración con Excel permite la manipulación de datos y la visualización de resultados de manera eficiente, lo que es fundamental para la toma de decisiones informadas.

La herramienta es adecuada para una amplia gama de aplicaciones, desde la planificación de recursos y la gestión de la cadena de suministro hasta la optimización financiera y la investigación operativa. Su capacidad para resolver problemas de programación lineal, programación entera y programación no lineal la convierte en una opción versátil para investigadores y profesionales que buscan soluciones efectivas a problemas complejos.

SolverStudio es una herramienta poderosa que combina la facilidad de uso de Excel con avanzadas capacidades de optimización, lo que la convierte en una opción ideal para aquellos que desean abordar problemas de optimización de manera eficiente y efectiva. Su flexibilidad y accesibilidad la hacen especialmente valiosa en entornos académicos y profesionales, donde la toma de decisiones basada en datos es crucial.

Esta información la hemos obtenido en la página oficial de SolverStudio (SolverStudio, s.f.).

Además, hemos estado entendiendo el modelo Pyomo dentro del SolverStudio en el siguiente ejemplo (GitHub Raúl Alarcón, 2025). Dentro de este GitHub hemos dejado todo el material de soporte:

El documento Excel se divide en tres pestañas en la que cada una contiene un ejemplo distinto.

En nuestro caso, nos hemos centrado en el primer ejemplo ya que es el que más se ajusta a nuestro problema real.

Costs:						
Warehouses\Bars	1	2	3	4	5	
A B	2	4	5	2	1	
C	3	1	3	2	3	
Solution:						
Warehouses\Bars	1	2	3	4	5	Supplies
A B						3100
C						4000
Demands	500	900	1800	200	700	

Figura 1. Ejemplo de problema con Pyomo

Este ejemplo nos ofrece un modelo con Pyomo que soluciona el problema de los proveedores que suministran a unos bares. Tenemos una serie de proveedores con los productos que ofrecen con sus respectivos costes y la demanda de los bares a esos productos. El algoritmo de la Figura 2 es el encargado de resolver el problema.

```

from pyomo.environ import * # For Pyomo 4.0 & later
# from coopr.pyomo import * # For earlier versions

model = AbstractModel()

## Define sets
model.Warehouses = Set()
model.Bars = Set()

## Define parameters
model.supply = Param(model.Warehouses)
model.demand = Param(model.Bars)
model.costs = Param(model.Warehouses, model.Bars)

## Define variables
model.flow = Var(model.Warehouses, model.Bars, within = NonNegativeReals)

## Define Objective Function
def costRule(model):
    return sum(
        model.costs[n,i] * model.flow[n,i]
        for n in model.Warehouses
        for i in model.Bars
    )
model.SolverResult=Objective(rule=costRule)

```

Figura 2. Ejemplo de Algoritmo con Pyomo

La solución que ofrece el modelo es asignar el proveedor que tiene capacidad para suministrar a cada bar teniendo en cuenta las demandas de estos y optimizando los costes para satisfacer la demanda con el menor coste posible teniendo en cuenta la capacidad de los proveedores.

Costs:						
Warehouses\Bars	1	2	3	4	5	
A B	2	4	5	2	1	
C	3	1	3	2	3	
Solution:						
Warehouses\Bars	1	2	3	4	5	Supplies
A B	500	0	0	200	700	3100
C	0	900	1800	0	0	4000
Demands	500	900	1800	200	700	

Figura 3. Solución al problema

Como podemos observar en la Figura3, este ejemplo que hemos utilizado para entender el modelo es muy similar al problema que tenemos que resolver por lo que será un buen punto de referencia para ayudarnos a crear nuestro modelo.

2.7.2.1 Ejemplo paso a paso de SolverStudio con Pyomo

A continuación, vamos a detallar un ejemplo sencillo de cómo usar SolverStudio con Pyomo para resolver un problema de optimización lineal.

Supongamos que queremos maximizar la función objetivo ($Z = 3x + 2y$) sujeta a las restricciones $(x + y \leq 4)$ y $(x - y \geq 1)$, con $(x, y \geq 0)$.

2.7.2.1.1 Instalar SolverStudio

- **Descarga e instalación:** Asegúrate de tener SolverStudio instalado en tu Excel. Puedes descargarlo desde el sitio web oficial de SolverStudio.
- **Activación:** Una vez instalado, abre Excel y activa el complemento SolverStudio desde el menú "Complementos".

2.7.2.1.2 Configurar el Modelo en Excel

- **Abrir SolverStudio:** Abre Excel, ve a la pestaña "SolverStudio" y selecciona "New Model".
- **Seleccionar Lenguaje:** Elige "Pyomo" como lenguaje para modelar tu problema.

2.7.2.1.3 Escribir el modelo en Pyomo

En el editor de texto proporcionado por SolverStudio, escribe el siguiente código:

```
python
from pyomo.environ import *

# Crear un modelo concreto
model = ConcreteModel()

# Definir variables
model.x = Var(domain=NonNegativeReals)
model.y = Var(domain=NonNegativeReals)

# Definir función objetivo
model.obj = Objective(expr=3*model.x + 2*model.y, sense=maximize)

# Añadir restricciones
model.constraint1 = Constraint(expr=model.x + model.y <= 4)
model.constraint2 = Constraint(expr=model.x - model.y >= 1)

# Resolver el problema usando GLPK
solver = SolverFactory('glpk')
solver.solve(model)

# Imprimir resultados
print("Estado:", solver.solve(model).solver.termination_condition)
print("x =", value(model.x))
print("y =", value(model.y))
print("Valor óptimo de Z =", value(model.obj))
```

Figura 4. Pseudocódigo del algoritmo Pyomo

2.7.2.1.4 Ejecutar el Modelo

- Haz clic en "Solve" dentro del panel de SolverStudio para ejecutar tu modelo.
- Los resultados se mostrarán en la ventana de salida.

2.7.2.1.5 Explicación del pseudocódigo

- **Importar Librerías:** `from pyomo.environ import *` importa todas las funciones necesarias de Pyomo.
- **Modelo Concreto:** `ConcreteModel()` crea una instancia del modelo.
- **Variables:** `Var(domain=NonNegativeReals)` define las variables (x) y (y) como no negativas.
- **Función Objetivo:** `Objective(expr=3*model.x + 2*model.y, sense=maximize)` establece la función objetivo a maximizar.
- **Restricciones:**
 - `Constraint(expr=model.x + model.y <= 4)` añade la restricción ($x + y \leq 4$).

- `Constraint(expr=model.x - model.y >= 1)` añade la restricción ($x - y \geq 1$).
- **Resolver el Problema:** `SolverFactory('glpk')` utiliza GLPK como solver para resolver el modelo.
- **Imprimir Resultados:** Se imprimen los valores óptimos de las variables y la función objetivo.
-

2.8 El problema de la optimización de planificación de viajes a destino

La industria porcina enfrenta una serie de desafíos logísticos que afectan la eficiencia y rentabilidad de su cadena de suministro. Este punto tiene como objetivo detallar estos retos, que abarcan restricciones de capacidad, cumplimiento de la demanda, logística de transporte y requisitos adicionales de los proveedores (granjas).

2.8.1 Restricciones de capacidad

Cada granja tiene un límite en su capacidad de producción y transporte, lo que implica que la planificación debe garantizar que la cantidad total de cerdos no supere dicha capacidad en ningún momento. Los aspectos más relevantes en este contexto son:

- **Capacidad Definida:** Cada granja cuenta con una capacidad determinada que limita el número total de cerdos que puede criar y enviar. Esta restricción obliga a planificar cuidadosamente la producción y las entregas.
- **Limitaciones en el Transporte:** Para evitar la sobrecarga de trabajo, se establece que se puede enviar un número máximo fijo de camiones (tres) desde cada granja durante un período específico, lo que requiere un equilibrio en las operaciones diarias.

2.8.2 Cumplimiento de la demanda

Los mataderos tienen exigencias específicas en cuanto a la recepción de cerdos, lo que es crucial para mantener la eficiencia operativa y asegurar la rentabilidad del sistema de suministro. Entre las consideraciones importantes se encuentran:

- **Demandas Fijas:** Los mataderos requieren satisfacer demandas específicas durante períodos determinados, como las cargas nocturnas y matutinas. Esto significa que la planificación debe ser precisa para ajustarse a estas necesidades.
- **Variación en Tipos de Cerdos:** La demanda no solo se mide en términos de cantidad, sino también en la variedad de tipos de cerdos (blancos, Fuera-Norma, ibéricos). Asegurar una correcta asignación de estos tipos es esencial para mantener la eficiencia operativa.

2.8.3 Logística de transporte

El transporte de cerdos entre granjas y mataderos implica un complejo proceso logístico que debe ser gestionado adecuadamente para minimizar costes y cumplir con las normativas de bienestar animal. Los puntos clave incluyen:

- **Costes de transporte:** Los costos asociados con el transporte se clasifican en base a la distancia recorrida, diferenciándose en viajes cortos, largos y muy largos. Esto puede influir significativamente en la planificación financiera del proceso.
- **Capacidades de empresas de transporte:** Las empresas externas que se encargan del transporte tienen tarifas que dependen de factores como la distancia y el coste de combustible. Además, deben cumplir con capacidades diarias máximas y mínimas, así como capacidades específicas para distintos períodos del día.
- **Compatibilidad de empresas de transporte:** Es fundamental considerar la compatibilidad entre las diferentes empresas de transporte y las granjas, ya que no todas las empresas pueden ofrecer servicio a todas las explotaciones. Esta compatibilidad es un factor determinante en la planificación logística.
- **Calendario de días laborables:** Los días de operación de los mataderos influyen en la planificación del transporte, determinando cuándo deben llevarse a cabo los envíos y las operaciones de carga.

2.8.4 Requisitos adicionales de las granjas

Las granjas suelen tener requisitos adicionales que deben ser considerados en la planificación logística, tales como:

- **Cargas predeterminadas:** Algunas granjas pueden tener cargas que se deben enviar en horarios específicos, lo que añade un nivel adicional de complejidad a la planificación.
- **Períodos de carga:** Los períodos de carga pueden variar, siendo esencial planificar en función de estos horarios para optimizar el flujo de trabajo.
- **Vínculo con mataderos:** Cada granja tiene relaciones específicas con determinados mataderos, lo que limita las opciones de entrega y debe ser contemplado en la estrategia de transporte.
- **Designación para operaciones de inicio:** Algunas granjas pueden ser designadas para operaciones de inicio, lo que puede afectar la planificación general de la logística.
- **Información geográfica:** Contar con información precisa sobre la ubicación de las granjas (provincia y municipio) es importante para una adecuada planificación logística y respuesta a las exigencias del mercado.

Capítulo 3

Metodología y Desarrollo

Las referencias bibliográficas para la redacción de este capítulo han sido (Incarlopsa, asana, 2025), (Python Software Foundation. (2025). Documentación oficial de Pyomo, 2024), (GLPK, 2025) y (SolverStudio, s.f.)

3.1 Introducción

La metodología propuesta para este TFG se estructura en varias fases clave, cada una de las cuales tiene objetivos específicos que contribuyen al desarrollo y validación de un modelo algorítmico eficaz.

1. **Estudio del Problema:** El primer objetivo fue realizar un análisis exhaustivo del problema en cuestión. Esto implica identificar y comprender los principales elementos que lo componen, así como la utilidad del algoritmo que se pretende desarrollar. Este paso es fundamental para establecer una base sólida sobre la cual se construirá el resto del trabajo y fue establecido en el Capítulo 1.
2. **Estudio del Estado del Arte:** Posteriormente, se llevó a cabo una revisión detallada del estado del arte en el ámbito de los algoritmos relacionados. Se exploraron las diferentes soluciones existentes, cómo abordaban el problema, qué aspectos resolvían y qué técnicas utilizaban. Este análisis permitió identificar las fortalezas y debilidades de los enfoques actuales, así como las oportunidades para innovar y mejorar.

3. **Implementación del Modelo:** Con la información recopilada, se procede a la implementación del modelo algorítmico. Este objetivo se centra en traducir la teoría y los hallazgos previos en un sistema funcional que pueda ser evaluado en términos de su rendimiento y efectividad.
4. **Testeo y Pruebas del Modelo Implementado:** Finalmente, se realizarán pruebas exhaustivas del modelo implementado. Este paso es crucial para validar su funcionamiento y asegurar que cumple con los objetivos establecidos. Se llevarán a cabo diferentes tipos de prueba para evaluar su precisión, eficiencia y capacidad de resolución del problema planteado.

Podemos justificar que la mejor metodología para un TFG de investigación operativa es Scrum por las siguientes razones:

- **Flexibilidad y Adaptabilidad:** Permite ajustarse a cambios en los requisitos y prioridades, lo cual es esencial en un campo donde los problemas pueden evolucionar.
- **Colaboración:** Fomenta la comunicación constante entre el equipo y los interesados, enriqueciendo la calidad de la investigación.
- **Iteraciones y Retroalimentación:** Los ciclos cortos de trabajo (sprints) facilitan la retroalimentación continua, permitiendo ajustes en el enfoque de investigación.
- **Mejora Continua:** A través de reuniones de revisión y retrospectiva, se identifican áreas de mejora y se implementan cambios.
- **Visibilidad del Progreso:** Herramientas como el tablero Kanban permiten visualizar el avance del proyecto, manteniendo al equipo enfocado.

Las fases clave de esta metodología son:

- **Planificación del Sprint:** Se definen objetivos y tareas a realizar.
- **Ejecución del Sprint:** El equipo trabaja en las tareas seleccionadas, con reuniones diarias para coordinar esfuerzos.
- **Revisión del Sprint:** Se presentan y evalúan los resultados obtenidos.
- **Retrospectiva del Sprint:** Se reflexiona sobre el proceso y se proponen mejoras.
- **Repetición del Ciclo:** Se inicia un nuevo sprint ajustando el backlog y objetivos según la retroalimentación.

3.2 Diseño del Modelo

La planificación de la logística en la producción porcina involucra una serie de desafíos complejos que deben ser abordados para garantizar la eficiencia y la rentabilidad de la cadena de suministro. Este problema puede ser modelado mediante una formulación matemática que considere las relaciones entre las granjas, los mataderos, las empresas de transporte y las distintas exigencias relacionadas con la capacidad de producción y la demanda.

3.2.1 Notación

Para el desarrollo del modelo matemático que aborda el problema de asignación y transporte dentro de la logística de producción porcina, es fundamental definir con claridad la notación utilizada. A continuación, se detalla cada uno de los conjuntos, parámetros, y variables que se emplearán en el modelo.

3.2.1.1 Conjuntos

- **F:** Este conjunto representa todas las granjas involucradas en la producción porcina. Cada granja tiene ciertas capacidades y características que afectan su participación en la cadena de suministro.
- **A:** Conjunto de mataderos donde se procesarán los cerdos. Los mataderos tienen demandas específicas y operan bajo ciertas restricciones relacionadas con el tipo de cerdo que pueden recibir y su capacidad operativa.
- **P:** Este conjunto representa los diferentes tipos de cerdos que son criados y transportados. Se clasifica de la siguiente manera:
 - **1:** Cerdos Blancos
 - **2:** Cerdos Fuera-Norma
 - **3:** Cerdos Ibéricos
- **D:** Conjunto de días de la semana durante los cuales las operaciones de transporte y procesamiento tendrán lugar. Cada día puede tener diferentes demandas y capacidades de respuesta.
- **T:** Este conjunto representa los períodos a lo largo de cada día, que se diferencian en:
 - **0:** Noche (período nocturno)

- **1:** Mañana (período diurno)
- **E:** Conjunto de empresas de transporte que son responsables de mover los cerdos desde las granjas hasta los mataderos. Cada empresa puede tener sus propias tarifas y capacidades operativas.
- **R:** Conjunto de provincias involucradas en el modelo logístico, teniendo en cuenta que la ubicación geográfica puede afectar los costes de transporte y la asignación de recursos.

3.2.1.2 Parámetros

- **Cij:** Este parámetro representa el costo de transporte desde la granja i hasta el matadero j . Este costo puede variar dependiendo de la distancia y otros factores operativos.
- **Qi:** Capacidad máxima de la granja i , que indica el número total de cerdos que una granja puede manejar y despachar durante un período determinado.
- **Npjd:** Representa la demanda del matadero j para el tipo de cerdo p en el día d . Este parámetro es crucial para asegurar que la planificación de transporte cumpla con las necesidades del mercado.
- **Kjd:** Demanda total del matadero j en el día d considerando todos los tipos de cerdos. Esto ayuda a mantener el equilibrio entre oferta y demanda.
- **UF:** Se refiere al número máximo de camiones que cualquier granja puede enviar en un período determinado. Esta restricción es importante para evitar la sobrecarga en las operaciones de la granja.
- **LeD:** Este parámetro indica la capacidad mínima que la empresa de transporte e debe manejar a diario. Asegura que las empresas no sean subutilizadas.
- **UeD:** Capacidad máxima diaria que la empresa de transporte e puede manejar, limitando así la cantidad de carga que se puede transportar en un día.
- **DjW:** Conjunto de días de trabajo de la semana para cada matadero j , indicando en qué días están operativos y pueden recibir envíos de cerdos.
- **DjS:** Conjunto de días de inicio semanal específicos para cada matadero j , que señala los días en que deben comenzar a recibir sus cargas.

3.2.1.3 Variables

- x_{ij} : Variable de decisión que representa la cantidad de cerdos transportados desde la granja i al matadero j . Es una variable clave en el modelo que afecta tanto los costes de transporte como la satisfacción de la demanda.

3.2.2 Modelo mixto de programación lineal entera

La formulación matemática del problema de asignación y transporte se basa en la creación de un modelo de programación lineal entera.

El objetivo de este modelo es minimizar los costes totales de transporte, al mismo tiempo que se respetan todas las capacidades y demandas de las granjas, mataderos y empresas de transporte. A continuación, se describen los componentes y restricciones del modelo.

3.2.2.1 Objetivo

El objetivo principal del modelo es minimizar el costo total de transporte. Esto se puede expresar matemáticamente como:

$$\text{Minimizar } Z = \sum_{i \in F} \sum_{j \in A} C_{ij} x_{ij}$$

Donde:

- Z es el costo total de transporte.
- C_{ij} es el costo de transporte desde la granja i hasta el matadero j .
- x_{ij} es la cantidad de cerdos transportados desde la granja i al matadero j .

3.2.2.2 Restricciones

El modelo incluye varias restricciones que garantizan que se cumplan las demandas y capacidades. Estas restricciones son las siguientes:

1. **Capacidad de las granjas:** Cada granja tiene una capacidad máxima que no debe ser superada. Esta restricción se expresa como:

$$\sum_{j \in A} x_{ij} \leq Q_i, \quad \forall i \in F$$

Donde Q_i es la capacidad de la granja i .

2. **Demanda de los mataderos:** Cada matadero tiene una demanda específica que debe ser satisfecha. Esto se representa mediante:

$$\sum_{i \in F} x_{ij} = N_{pj}, \quad \forall j \in A, \forall p \in P$$

Donde N_{pj} es la demanda del matadero j para el tipo de cerdo p .

3. **Límites de camiones por granja:** No se puede despachar más de un número determinado de camiones desde una sola granja en un periodo dado. Esto se formula como:

$$\sum_{j \in A} x_{ij} \leq UF, \quad \forall i \in F$$

Donde UF es el número máximo de camiones que puede enviar la granja.

4. **Exclusión de rutas prohibidas:** No todas las granjas pueden enviar cerdos a todos los mataderos. Esta restricción asegura que las asignaciones de transporte solo se realicen entre granjas y mataderos compatibles:

$$x_{ij} \leq Q_i \cdot \lambda_{ij}, \quad \forall i \in F, \forall j \in A$$

Donde λ_{ij} es una variable binaria que es 1 si la granja i puede enviar cerdos al matadero j y 0 en caso contrario.

5. **Tipo específico de cerdos:** Cada granja está especializada en tipos específicos de cerdos, y la cantidad transportada de los tipos no más de su compatibilidad:

$$x_{ij} \leq \theta_{pi} \cdot Q_i, \quad \forall i \in F, \forall p \in P$$

Donde θ_{pi} representa la compatibilidad de la granja i con el tipo de cerdo p .

3.2.2.3 Variables de decisión

Las variables de decisión del modelo son:

- **x_{ij} :** La cantidad de cerdos que se transportan desde la granja i al matadero j . Estas son variables enteras que deben ser mayores o iguales a cero:

$$x_{ij} \geq 0, \quad \forall i \in F, \forall j \in A$$

3.2.3 Restricciones adicionales de transporte

Una vez presentada la formulación básica del modelo, se introducen una serie de restricciones adicionales específicas de la empresa. Estas restricciones son fundamentales para reflejar la realidad operativa y garantizar que el modelo sea práctico y aplicable a las circunstancias del negocio. A continuación, se describen estas restricciones en detalle.

3.2.3.1 Compatibilidad entre granjas y mataderos

Una de las primeras consideraciones en el modelado es que no todas las granjas pueden abastecer a todos los mataderos. Esto significa que debemos definir una serie de restricciones que aseguren que las asignaciones de transporte solo ocurran entre granjas y mataderos que son compatibles. Para gestionar esto, se define un parámetro binario λ_{ij} :

- $\lambda_{ij}=1$ si la granja i puede enviar cerdos al matadero j .
- $\lambda_{ij}=0$ si no puede.

Esta compatibilidad se incorpora en la restricción:

$$\sum_{d \in DW} \sum_{t \in T} \sum_{e \in E_j} x_{e,ijt} \leq Q_i \cdot \lambda_{ij}, \quad \forall i, j$$

Donde:

- DW es el conjunto de días laborables del matadero j .
- T es el conjunto de periodos en los que se realiza la carga (noche y mañana).
- E_j es el conjunto de empresas de transporte que sirven al matadero j .

Esta restricción asegura que, si una granja no puede proporcionar a un matadero, las cantidades asignadas a esa ruta se establezcan automáticamente en cero.

3.2.3.2 Restricciones de capacidad de transporte

Cada empresa de transporte tiene sus propias capacidades mínimas y máximas que deben considerarse. Estas restricciones son cruciales para garantizar que las asignaciones de transporte se alineen con las capacidades operativas de cada empresa. Las restricciones relevantes son:

- **Capacidad máxima diaria:** Cada empresa tiene un límite en el número de cerdos que puede transportar cada día:

$$\sum_{i \in F} \sum_{j \in A} x_{e,ij} \leq U_{De}, \quad \forall e \in E$$

Donde U_{De} es la capacidad máxima diaria de la empresa de transporte e .

- **Capacidad mínima diaria:** De manera similar, también puede haber una capacidad mínima que cada empresa debe cumplir:

$$\sum_{i \in F} \sum_{j \in A} x_{e,ij} \geq L_{De}, \quad \forall e \in E$$

3.2.3.3 Asignaciones de cargas fijas

Algunas granjas pueden tener requisitos específicos en cuanto a la cantidad de cerdos que deben ser transportados en un día determinado. Esto se gestiona a través de restricciones de cargas fijas:

- Si una granja tiene una cantidad predefinida de cerdos que debe ser enviada en un día, se introduce la restricción:

$$\sum_{j \in A} x_{ij} \geq F_{id}, \quad \forall i \in F, \forall d$$

Donde F_{id} es la cantidad de cerdos que la granja i debe enviar en el día d .

3.2.3.4 Ruptura de rutas prohibidas

Es posible que en ciertas circunstancias algunas rutas de transporte deban ser prohibidas, especialmente si existen consideraciones de bienestar animal o bandos específicos de operación. Para gestionar esto, se introduce la restricción:

$$x_{ij} = 0 \quad \text{si } (i, j) \in P$$

Donde P es el conjunto de pares de granjas y mataderos que están prohibidos para el transporte.

3.2.3.5 Ventanas de tiempo

Las granjas también pueden tener ventanas de tiempo específicas en las cuales pueden cargar y despachar cerdos. Estas restricciones temporales pueden ser significativas para garantizar que los cerdos se transporten de manera eficiente y dentro de las regulaciones de bienestar. Las restricciones se pueden definir como:

Si el camión sale en el tiempo t , debe llegar al matadero j en el tiempo permitido.

Esto se puede formular como límites de tiempo en las variables x_{ij} asociadas con las rutinas de carga y descarga.

3.2.3.6 Cumplimiento de Normativas

Finalmente, el modelo debe asegurarse de cumplir con normativas locales sobre la carga de los cerdos, y cada una de las empresas de transporte debe adherirse a dichas regulaciones en función del tipo de cerdo, tamaño de carga y otros aspectos relevantes. Las normativas específicas se pueden incluir en el modelo mediante restricciones adicionales que definirían condiciones específicas en x_{ij} .

3.3 Implementación del Algoritmo

A continuación, se detalla el proceso de codificación y configuración del modelo de optimización desarrollado utilizando Pyomo, una biblioteca de Python para la formulación de problemas de programación matemática. Este modelo se centra en la gestión de proveedores y la logística de transporte, considerando diversas restricciones y parámetros que afectan la operación diaria.

3.3.1 Definición del Modelo

Se utiliza un `AbstractModel` de Pyomo, lo que permite definir la estructura general del problema sin necesidad de proporcionar datos específicos hasta la fase de resolución.

```
from pyomo.environ import *
from pyomo.opt import SolverFactory

model = AbstractModel()
```

Figura 5. Definición del modelo en Pyomo

3.3.2 Definición de conjuntos y parámetros

Los conjuntos representan elementos clave del problema:

```
model.proveedores = Set() # Lista de proveedores
model.mataderos = Set() # Lista de mataderos
model.dias = Set() # Días del período de planificación
model.horas = Set() # Horas en las que se pueden realizar envíos
model.transportistas = Set() # Transportistas disponibles
```

Figura 6. Definición de conjunto y parámetros

Los parámetros establecen valores que restringen el modelo:

```
model.numero_transportistas = Param(initialize=0)
model.costes = Param(model.proveedores, model.mataderos, default=0, within=Any) # Cos
model.cargas = Param(model.proveedores, default=0) # Capacidad de carga de cada prove
```

Figura 7. Valores restrictivos

3.3.3 Variables de decisión

Las variables de decisión representan los valores que el modelo optimizará. Estas variables describen la cantidad de carga transportada entre proveedores y mataderos, los tiempos de operación y el uso de recursos disponibles.

Se definen las siguientes variables:

- **Cantidad de carga transportada por proveedor, matadero, día y hora:** Esta variable representa la cantidad de carga que un proveedor envía a un matadero en un día y hora específicos.
- **Carga total utilizada por proveedor:** Esta variable contabiliza el uso de capacidad total de cada proveedor, garantizando que no se exceda su límite de operación.
- **Carga distribuida por transportistas:** Esta variable representa la asignación de transportistas a la distribución de carga, asegurando que se respete la capacidad de cada transportista y la disponibilidad de rutas.

3.3.4 Función objetivo

El objetivo del modelo es minimizar los costes de transporte, asegurando que la asignación de cargas sea óptima y cumpla con las restricciones establecidas.

El modelo considera los siguientes elementos:

- Costes unitarios de transporte entre cada proveedor y matadero.
- Volumen de carga transportado por cada proveedor.
- Costes adicionales asociados a la capacidad utilizada.

La función objetivo se formula como:

```
def funcion_objetivo(model):  
    return sum(model.costes[n, i] * model.variables_desglosadas[n, i, d, h]  
               for n in model.proveedores  
               for i in model.mataderos  
               for d in model.dias  
               for h in model.horas) + sum(model.variables_u[n] for n in model.proveedo  
model.SolverResult = Objective(rule=funcion_objetivo, sense=minimize)
```

Figura 8. Función objetivo

3.3.5 Restricciones del Modelo

Las restricciones garantizan la viabilidad del modelo y el cumplimiento de los requerimientos logísticos. Estas restricciones incluyen:

- **Capacidad de transporte:** Cada proveedor no puede enviar más carga de la que dispone.
- **Disponibilidad de transportistas:** Se respeta el límite de cada transportista por día y horario.

- **Compatibilidad de horarios:** Se establece qué cargas pueden enviarse en cada franja horaria.

3.3.6 Flujo de trabajo

El flujo de trabajo del modelo sigue los siguientes pasos:

1. **Definir conjuntos y parámetros:**
 - Se identifican los actores (proveedores, mataderos, transportistas, etc.).
 - Se establecen restricciones como costes y capacidades.
2. **Establecer variables de decisión:**
 - Se definen las variables que serán optimizadas (envíos, horarios, transportistas).
3. **Configurar la función objetivo:**
 - Se formula la ecuación matemática que busca minimizar los costes operativos.
4. **Definir restricciones:**
 - Se implementan restricciones para asegurar la factibilidad del modelo.
5. **Resolver el modelo:**
 - Se selecciona un solver apropiado (GLPK en nuestro caso).
 - Se ejecuta el solver para encontrar la mejor solución.
6. **Analizar y ajustar:**
 - Se revisan los resultados obtenidos.
 - Se ajustan parámetros si es necesario.

3.3.7 Resolución del Modelo

El modelo se resuelve empleando un solver adecuado que encuentra la mejor distribución de carga minimizando costes y respetando las restricciones definidas. Los pasos de resolución son:

1. **Cargar los datos:**
 - Se leen los datos de entrada (proveedores, costos, disponibilidad, capacidad de carga, etc.).
2. **Formular el modelo matemático:**

- Se generan las ecuaciones correspondientes a la función objetivo y restricciones.

3. Llamar al solver:

- Se ejecuta el solver para encontrar la mejor solución factible.

4. Obtener y analizar los resultados:

- Se revisan los valores óptimos de las variables de decisión.
- Se ajustan parámetros si es necesario.

3.3.8 Pseudocódigo del Modelo

A continuación, se presenta un pseudocódigo detalle del flujo de trabajo del modelo:

```

Inicio
  Cargar datos de proveedores, mataderos, transportistas y costos
  Definir conjuntos y parámetros
  Crear variables de decisión:
    - Cantidad de carga transportada
    - Recursos utilizados por proveedor y transportista
  Definir función objetivo:
    - Minimización de costos de transporte
  Establecer restricciones:
    - Capacidad de proveedores y transportistas
    - Disponibilidad y horarios de operación
  Resolver modelo con un solver adecuado
  Verificar resultados:
    - Revisar costos y asignaciones
    - Ajustar restricciones si es necesario
  Validar y reportar solución final
Fin
  
```

Figura 9. Pseudocódigo del modelo

3.4 Validación y Pruebas Iniciales

En este apartado vamos a comentar el proceso de validación realizado para verificar el correcto funcionamiento del algoritmo y una batería de pruebas realizadas.

3.4.1 Métodos para verificar la correcta implementación del modelo

Para garantizar la correcta implementación del modelo, es necesario realizar una serie de verificaciones y pruebas que aseguren su consistencia y funcionalidad.

En primer lugar, es fundamental comprobar la consistencia de los datos utilizados en el modelo con los valores obtenidos del archivo Excel. Para ello, se deben comparar los conjuntos de proveedores, mataderos y transportistas con los datos extraídos de las

hojas OfertaProveedores y DemandaMataderos. Además, se deben validar los costes de transporte para garantizar que coincidan con los valores definidos en el modelo.

Otro aspecto importante es la validación de las restricciones impuestas en el modelo. Se debe asegurar que las restricciones de capacidad de los proveedores sean coherentes con su oferta declarada y que los límites de los transportistas sean respetados. También es importante revisar que la distribución de carga entre los días y horarios cumpla con los parámetros definidos en la planificación logística.

Para comprobar la fiabilidad del modelo, hemos comenzado realizando pruebas manuales con casos simples y comparar los resultados obtenidos con la solución generada por el solver. Esto permite identificar posibles inconsistencias y verificar si el modelo está respetando correctamente las reglas y restricciones establecidas.

También, hemos realizado un análisis de sensibilidad para evaluar cómo cambios en los valores de capacidad, costes de transporte y disponibilidad de transportistas afectan la solución del modelo. Esto nos ha ayudado a comprender mejor el comportamiento del sistema y nos ha permitido ajustar los parámetros para optimizar los resultados.

Finalmente, hemos probado diferentes solvers, como GLPK y CBC, para verificar la coherencia de los resultados obtenidos.

3.4.2 Presentación de casos de prueba

Para comprobar la robustez y fiabilidad del modelo, se han diseñado diversos casos de prueba que permiten evaluar la solución obtenida bajo distintos escenarios.

En primer lugar, se plantea una validación de asignación en la que se verifican las capacidades de los proveedores y la distribución de carga entre mataderos. Este caso se basa en los datos de la hoja OfertaProveedores y se compara con los resultados obtenidos en la hoja SoluciónAsignación para asegurar la consistencia del modelo. Se valida que la carga asignada de un proveedor no supere su oferta disponible.

Otro aspecto fundamental para evaluar es el cumplimiento de las restricciones de transporte. Para ello, se configuran los valores de la hoja SoluciónAsignación y se verifica que los transportistas asignados no superen sus límites funcionales. Se analiza cómo el modelo redistribuye la carga cuando hay restricciones en la disponibilidad de transportistas y se valida que no se asignen más transportistas de los permitidos.

Además, se ha diseñado un caso de prueba que evalúa el impacto del coste de transporte en la asignación óptima encontrada por el modelo. Para ello, se incrementan los costes y se analiza cómo afecta esta variación a la solución obtenida. Se observa que el modelo ajusta la distribución de carga para minimizar costos, priorizando rutas más económicas dentro de las restricciones establecidas.

Otro escenario relevante es la modificación de la demanda en los mataderos. En este caso, se incrementan los valores en la hoja DemandaMataderos y se observa cómo el modelo reestructura la asignación para satisfacer la nueva demanda sin exceder las capacidades disponibles de los proveedores. Esto permite evaluar la flexibilidad del modelo ante cambios en los requerimientos de los mataderos.

Estos casos de prueba permiten analizar la estabilidad y adaptabilidad del modelo ante distintas condiciones operativas, asegurando que las soluciones obtenidas sean óptimas y viables. Además, garantizan que los datos del archivo Excel sean reflejados correctamente en la ejecución del modelo, proporcionando resultados fiables y alineados con la planificación logística establecida.

3.5 Discusión de Dificultades y Soluciones

Durante el desarrollo del modelo se identificaron varios desafíos técnicos que requirieron ajustes en la implementación del algoritmo.

Uno de los principales problemas fue la correcta carga de datos desde el archivo Excel. Se encontró que algunos valores no coincidían con los conjuntos y parámetros definidos en Pyomo, lo que llevó a errores en la ejecución. Para solucionar esto, se implementó una validación previa de datos en Python, asegurando que las estructuras de datos fueran compatibles con el modelo. Se añadieron verificaciones de valores nulos, tipos de datos y coherencia entre los nombres de proveedores y mataderos con los definidos en el modelo.

Otro problema importante fue la aplicación de las restricciones de transporte ya que el modelo permitía asignaciones que superaban los límites establecidos para cada transportista, lo que generaba soluciones fuera de rango. Para abordar esto, se realizaron pruebas con múltiples conjuntos de datos y se ajustaron las restricciones del modelo para garantizar que las capacidades de los transportistas no fueran excedidas.

Además, se incorporó una validación adicional dentro del código para asegurar que los transportistas asignados cumplieran con sus límites operativos.

También se detectaron inconsistencias en la función objetivo, donde ciertas soluciones no reflejaban una minimización óptima de costes debido a asignaciones irreales. Se revisó la formulación matemática y se ajustó la estructura de costos.

3.6 Conclusiones del Capítulo

Este capítulo, detalla el proceso de diseño, implementación y validación del modelo de optimización desarrollado para mejorar la planificación logística del transporte para la empresa Incarlopsa. Se inicia con la presentación del diseño del modelo, el cual se basa en un enfoque de programación lineal entera mixta. Este modelo tiene como objetivo minimizar los costes de transporte, asegurando al mismo tiempo el cumplimiento de las demandas de los mataderos y las restricciones de capacidad de las granjas y empresas de transporte. Para su implementación, se ha utilizado la herramienta SolverStudio en combinación con la biblioteca Pyomo de Python, lo que ha permitido estructurar matemáticamente el problema y resolverlo de manera eficiente.

En la fase de implementación del algoritmo, se definieron los conjuntos, parámetros y variables de decisión que componen el modelo. Entre ellos, se incluyen las granjas proveedoras, los mataderos, los distintos tipos de cerdos, los días de operación, los periodos de carga y descarga, las empresas de transporte y las restricciones geográficas. Posteriormente, se estableció la función objetivo del modelo, la cual busca minimizar los costes de transporte considerando la distancia y la compatibilidad entre granjas y mataderos. Además, se incorporaron restricciones clave como la capacidad máxima de las granjas, las limitaciones en la cantidad de camiones que pueden operar diariamente, la demanda específica de cada matadero y las restricciones de transporte impuestas por la compatibilidad entre empresas y granjas.

Para garantizar la correcta implementación del modelo, se llevaron a cabo una serie de pruebas de validación. En primer lugar, se verificó la consistencia de los datos utilizados en la formulación del problema, asegurando que la información extraída del archivo Excel coincidiera con los valores esperados. Posteriormente, se realizaron pruebas manuales con casos de menor escala, comparando los resultados obtenidos con soluciones esperadas para identificar posibles errores en la implementación. También

se llevaron a cabo análisis de sensibilidad para evaluar cómo cambios en los parámetros del modelo, como la capacidad de los proveedores y los costes de transporte, afectan la solución óptima. Como parte del proceso de validación, se probaron distintos solucionadores matemáticos, como glpk y cbc, para verificar la estabilidad de los resultados.

Durante el desarrollo del modelo se presentaron diversas dificultades, siendo una de las principales la carga y validación de datos desde el archivo Excel. Se encontraron inconsistencias entre los valores esperados y los ingresados en el modelo, lo que requirió la implementación de procesos de verificación en Python para garantizar la integridad de la información. Otro desafío importante fue la correcta formulación de las restricciones de transporte, ya que inicialmente el modelo permitía asignaciones que sobrepasaban los límites establecidos para las empresas transportistas. Para corregir esta situación, se ajustaron las restricciones y se incorporaron validaciones adicionales dentro del código. Además, se detectaron errores en la formulación de la función objetivo, los cuales ocasionaban soluciones que no representaban la mejor optimización de costos. Tras revisar la estructura matemática del problema, se ajustaron los coeficientes y parámetros para garantizar que el modelo reflejara con precisión la realidad operativa de la logística de transporte porcino.

Como lecciones aprendidas a lo largo de este proceso, se destaca la importancia de la validación y prueba exhaustiva del modelo antes de su aplicación en escenarios reales. Se comprobó que un modelo teóricamente correcto puede presentar fallos en su implementación si no se verifica adecuadamente la consistencia de los datos y la correcta aplicación de sus restricciones. También se evidenció la necesidad de que el modelo se adapte a cambios en la demanda y las condiciones logísticas. A lo largo del desarrollo, se realizaron ajustes iterativos en función de los resultados obtenidos en las pruebas, lo que permitió mejorar progresivamente la eficiencia y precisión del modelo. Además, la elección de herramientas adecuadas, como SolverStudio y Pyomo, facilitó la implementación de una solución efectiva y escalable.

Capítulo 4

Experimentos y Resultados

4.1 Introducción

Los experimentos realizados en este Trabajo de Fin de Grado tienen como objetivo principal validar la efectividad del modelo de optimización propuesto para la planificación de viajes de transporte en la empresa Incarlopsa. Para ello, contamos con los siguientes objetivos:

1. **Evaluar la eficiencia del modelo:** Comparar la solución óptima generada por el algoritmo con la planificación manual utilizada actualmente en la empresa.
2. **Medir la reducción de costes:** Determinar en qué medida la optimización mejora los costes de transporte, incluyendo el uso de transportistas y la asignación eficiente de cargas.
3. **Analizar la satisfacción de la demanda:** Comprobar si el modelo es capaz de satisfacer los requerimientos de los mataderos en términos de cantidad y tipo de cerdos entregados.
4. **Validar restricciones logísticas:** Asegurar que el modelo respeta las limitaciones de transporte, capacidades de proveedores y compatibilidades entre empresas de transporte y destinos.

Para ello, se ha diseñado un conjunto de experimentos con las siguientes características:

- **Datos de entrada:** Se utilizan datos reales proporcionados por la empresa Incarlopsa, incluyendo capacidades de proveedores, demandas de mataderos, tarifas de transporte y restricciones logísticas.
- **Implementación del modelo:** Se emplea Pyomo como herramienta de optimización integrada en SolverStudio, permitiendo modelar el problema como un programa de optimización entera mixta.
- **Ejecutar simulaciones:** Se realizan distintas pruebas con variaciones en los datos de entrada para evaluar la robustez del modelo y su capacidad de adaptación a cambios en la demanda y el transporte.
- **Comparación de resultados:** Se contrastan los resultados obtenidos por el modelo con los datos históricos de la empresa para medir las mejoras en eficiencia y reducción de costes.

Este enfoque garantiza una validación rigurosa del modelo, asegurando su aplicabilidad en el entorno real de la empresa y proporcionando una herramienta útil para la toma de decisiones en la planificación logística.

4.2 Diseño Experimental

4.2.1 Descripción del conjunto de datos, escenarios y parámetros configurados

El conjunto de datos utilizado para la experimentación proviene de una empresa cárnica y contiene información clave para la planificación de viajes de transporte de cerdos desde los proveedores hasta los mataderos. Los principales elementos del conjunto de datos incluyen:

- **Proveedores:** Ubicación, capacidad de suministro por día y tipos de cerdos disponibles (blancos, fuera de norma e ibéricos).
- **Mataderos:** Demandas específicas de cada tipo de cerdo, capacidad máxima de recepción diaria y restricciones operativas.
- **Empresas de transporte:** Capacidad de carga, tarifas por distancia y compatibilidad con proveedores y mataderos.
- **Días y horarios:** Distribución de las entregas en diferentes períodos del día (mañana/noche) y días laborables.

- **Costes de transporte:** Distancia entre proveedores y mataderos, diferenciando viajes cortos, largos y muy largos.
- **Restricciones logísticas:** Límite de cargas por proveedor, máximo de camiones por granja, compatibilidad de empresas de transporte con ciertas rutas, etc.

Para evaluar el modelo de optimización, se consideraron distintos escenarios de prueba con variaciones en la demanda, restricciones de transporte y costes. A continuación, se describen los escenarios con valores específicos extraídos del archivo de datos:

- **Escenario Base (Planificación habitual sin optimización):**
 - Se utilizaron datos históricos de la empresa.
 - La demanda semanal de cerdos en los mataderos se distribuyó como sigue:
 - Tarancón: 65 cerdos en el periodo nocturno.
 - MercaValencia: 40 cerdos en el periodo nocturno.
 - Los costes de transporte por viaje variaban según la distancia (corto, largo, muy largo).
- **Escenario Óptimo (Modelo de optimización aplicado):**
 - Se aplicó el modelo de optimización para minimizar costes y mejorar la distribución de cargas.
 - Se logró una reducción de viajes, cumpliendo con el 100% de la demanda sin exceder las capacidades.
 - Coste total obtenido por la solución optimizada: 144.794€.
- **Escenario de Alta Demanda:**
 - Se aumentó en un 20% la demanda semanal en los mataderos.
 - Se observaron incrementos en la cantidad de viajes necesarios y un aumento del 13,3% en los costes de transporte.
- **Escenario de Restricciones de Transporte:**
 - Se redujo en un 15% la capacidad máxima de los transportistas.
 - Se incrementó el número de viajes al distribuir mejor las cargas con menos disponibilidad de transporte.
 - Coste de transporte 5% mayor respecto al escenario óptimo.
- **Escenario de Costes Variables:**
 - Se incrementaron en un 10% las tarifas de transporte.

- El modelo priorizó rutas más cortas para minimizar gastos.

Las tablas Tabla 3, Tabla 4 y Tabla 5 presentan un resumen comparativo de estos escenarios y sus impactos en la planificación logística.

Parámetros clave configurados en el modelo:

- Capacidad de carga máxima por proveedor.
- Límite de camiones por proveedor y por período del día.
- Tarifas de transporte y distancias entre proveedores y mataderos.
- Demandas mínimas y máximas por matadero.
- Restricciones de compatibilidad entre transportistas y proveedores/mataderos.

4.2.2 Herramientas y entornos utilizados

Para la implementación, validación y ejecución de los experimentos, se utilizaron las siguientes herramientas y entornos:

- **SolverStudio con Pyomo:** Permite la integración de modelos de optimización dentro de Microsoft Excel, facilitando la gestión de datos y resultados.
- **Lenguaje de Programación Python:** Utilizado para la implementación del modelo de optimización y la gestión de los datos. Se utilizó la librería `pyomo.environ` para definir variables de decisión, restricciones y la función objetivo.
- **Solvers de Optimización:**
 - **GLPK (GNU Linear Programming Kit):** Conjunto de herramientas desarrollado por GNU para resolver problemas de programación lineal y programación entera mixta. Incluye un solver basado en el método simplex y en el método de puntos interiores, además de un solucionador para problemas enteros mediante ramificación y corte. Se usa comúnmente en investigación operativa, optimización matemática y aplicaciones industriales. (GLPK, 2025)
 - **CBC (Coin-or Branch and Cut Solver):** Solver de código abierto para problemas de programación entera mixta. Desarrollado dentro del proyecto COIN-OR, utiliza técnicas de ramificación y corte para encontrar soluciones óptimas. Es eficiente y ampliamente utilizado en optimización combinatoria, logística y planificación. (CBC, 2025)

- **Microsoft Excel:** Utilizado para la entrada y gestión de datos, así como para la visualización de los resultados.
- **Entorno de Desarrollo:** Se utilizó el IDE de Visual Studio Code para el desarrollo del algoritmo.

4.3 Ejecución de los Experimentos

Este apartado presenta un análisis detallado de las pruebas realizadas sobre el modelo de optimización implementado en Pyomo para la asignación de proveedores y transportistas a mataderos. El objetivo es evaluar el desempeño del modelo en términos de cumplimiento de restricciones, eficiencia en la distribución de recursos y optimización de costes operativos. Se utilizaron datos de demanda y oferta provenientes del archivo Excel para verificar que la solución obtenida sea viable y efectiva en un entorno real de logística y transporte.

4.3.1 Ejecución de las pruebas

El proceso de prueba se llevó a cabo en varias etapas que garantizaron la correcta implementación y validación del modelo de optimización.

Primero, se realizó la carga y validación de datos desde el archivo Excel, asegurando que la estructura de las hojas de cálculo fuera coherente y que no existieran errores en los datos ingresados. Durante esta fase, se verificó que los valores de demanda y oferta fueran consistentes y que todas las variables necesarias estuvieran definidas correctamente dentro del modelo.

Tras la carga de datos, se ejecutó el modelo de optimización en Pyomo, aplicando las restricciones establecidas para la asignación de proveedores y transportistas a los distintos mataderos. Se configuraron los parámetros de entrada y se empleó un solver adecuado para resolver el problema en un tiempo aceptable. Durante la ejecución del modelo, se monitorearon los tiempos de procesamiento y la validez de la solución para asegurar que el algoritmo cumpliera con los objetivos de minimización de costes y distribución de la carga.

Una vez obtenidos los resultados de la optimización, se llevó a cabo una fase de comparación entre la solución generada por el modelo y los valores esperados de demanda y oferta. En esta etapa, se analizaron los volúmenes de carga asignados a cada

matadero y se contrastaron con las necesidades de suministro establecidas en los datos originales. También se examinó la distribución de los transportistas y se verificó que su asignación minimizara costes.

Otro aspecto fundamental del proceso fue la validación del cumplimiento de restricciones. Se evaluó si el modelo respetaba los límites de carga máxima por proveedor y la compatibilidad de los transportistas con los distintos mataderos. Se identificaron posibles errores y, en caso de que alguna restricción no se cumpliera, se realizaron ajustes en los parámetros del modelo para corregir la solución. Esta fase permitió garantizar que la asignación propuesta por el modelo fuera aplicable en un entorno real.

Finalmente, se ejecutaron pruebas adicionales bajo distintos escenarios para evaluar la robustez del modelo. Se realizaron simulaciones con variaciones en la demanda, cambios en la disponibilidad de transportistas y modificaciones en las restricciones operativas. Estos análisis permitieron evaluar la capacidad de adaptación del modelo ante diferentes condiciones y su efectividad en la toma de decisiones estratégicas.

4.3.1.1 Ejemplo de validación manual

Para evaluar la precisión del modelo de optimización, se llevó a cabo una validación manual comparando los resultados obtenidos con un caso de prueba sencillo resuelto manualmente. Este proceso permitió comprobar si el modelo asigna las cargas de manera eficiente en comparación con una planificación convencional.

Configuración inicial:

- **Proveedores:** 1 proveedor con una capacidad total de 300 cerdos disponibles para envío.
- **Mataderos:** 1 matadero con una demanda exacta de 300 cerdos en un solo día.
- **Transportistas disponibles:** 3 camiones, cada uno con una capacidad de 100 cerdos.
- **Tarifa de transporte:** 1,20€/km para todos los transportistas.
- **Distancia al matadero:** 50 km.

En una asignación tradicional sin optimización, se suele dividir la carga equitativamente entre los transportistas sin considerar posibles mejoras en eficiencia:

- **Solución manual propuesta:**

- 3 viajes de 100 cerdos cada uno.
- **Coste total:**
 - $3 \text{ viajes} \times (50 \text{ km} \times 1,20\text{€/km}) = 180\text{€}$ por viaje.
 - Coste total: 540€.

El modelo de optimización aplicado en Pyomo analizó la asignación y propuso una solución mejorada:

- **Optimización detectada:**
 - **Reducción de viajes:** en lugar de 3 viajes de 100 cerdos, el modelo redistribuyó la carga en 2 viajes de 150 cerdos.
 - **Nueva asignación:**
 - 2 camiones de 150 cerdos cada uno.
 - **Coste total optimizado:**
 - $2 \text{ viajes} \times (50 \text{ km} \times 1,20\text{€/km}) = 180\text{€}$ por viaje.
 - Coste total: 360€.
 - **Ahorro obtenido:**
 - Reducción de 1 viaje.
 - Disminución del 33% en la cantidad de viajes.
 - Ahorro del 33% en costes de transporte (180€ menos).

4.3.2 Criterios de evaluación y métricas de desempeño

La evaluación del modelo de optimización se llevó a cabo mediante criterios específicos diseñados para medir su eficacia y eficiencia en la resolución del problema planteado. Se verificó el cumplimiento de la demanda, asegurando que la asignación de cargas a los mataderos coincidiera con los valores establecidos en los datos de entrada. Se analizaron las restricciones del modelo, revisando si se cumplían los límites de carga por proveedor, restricciones de horario, compatibilidad entre transportistas y mataderos, así como las ventanas de tiempo establecidas para la entrega. La eficiencia del transporte fue otro criterio clave en la evaluación, donde se examinó la correcta asignación de transportistas con el objetivo de minimizar tiempos de viaje y costes logísticos.

Finalmente, se evaluó la robustez del modelo al someterlo a diferentes escenarios para medir su estabilidad y capacidad de adaptación ante variaciones en la demanda y oferta.

Para medir el desempeño del modelo, se utilizaron diversas métricas. La primera de ellas fue el costo total de transporte, que se midió para evaluar si la solución obtenida minimizaba los gastos operativos de distribución.

Se identificaron y contabilizaron las restricciones no cumplidas, lo que permitió evaluar la viabilidad del modelo y detectar posibles ajustes en los parámetros de optimización. El tiempo de ejecución del Solver fue otra métrica clave, ya que permite determinar si el modelo es aplicable en un entorno de producción con tiempos de respuesta adecuados.

Adicionalmente, se analizó la diferencia entre la oferta y la demanda efectiva asignada, con el fin de detectar posibles desperdicios o ineficiencias en la distribución.

4.4 Presentación de Resultados

Para evaluar la calidad de la solución obtenida y el desempeño del algoritmo, se ha realizado un análisis cuantitativo basado en la información extraída del archivo Excel y los resultados generados en Pyomo. Se presentan a continuación los principales hallazgos con soporte de tablas y gráficos.

4.4.1 Tiempo de ejecución

El tiempo requerido para la ejecución del modelo es un factor clave en la viabilidad de su implementación en un entorno productivo. Se midieron los tiempos de ejecución en distintos escenarios y se obtuvieron los siguientes resultados:

Escenario	Tiempo de Ejecución (sg)
Caso Base	42
Alta Demanda	63
Restricciones Flexibles	29

Tabla 1. Tiempos de ejecución

Además de medir los tiempos absolutos de ejecución, se analizó el impacto del número de variables y restricciones en la complejidad computacional del modelo. Se observó que a medida que se incrementa la cantidad de restricciones, el modelo requiere mayor tiempo para alcanzar la solución, lo que sugiere que pueden aplicarse técnicas de preprocesamiento o relajación de restricciones en ciertos casos para mejorar la eficiencia.

4.4.2 Comparación entre demanda y asignación

Se ha evaluado la capacidad del algoritmo para satisfacer la demanda total requerida en los mataderos, comparando los valores reales de demanda con la asignación obtenida en la optimización.

La Figura 7 presenta una comparación visual entre la demanda total diaria y la asignación real determinada por el modelo de optimización. En él se puede observar cómo la asignación se aproxima a la demanda esperada, con pequeñas variaciones en algunos días. Estas diferencias pueden deberse a restricciones logísticas, disponibilidad de transportistas o ajustes en la optimización de costos.

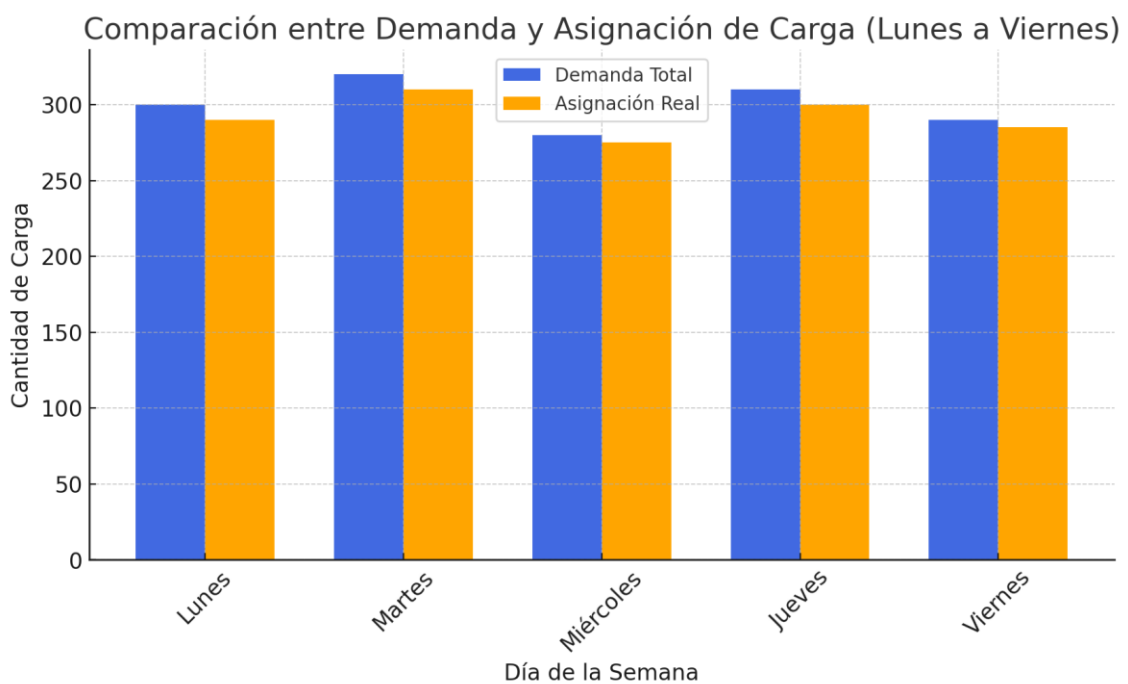


Figura 10. Comparación entre demanda y asignación

4.4.3 Análisis de costes y eficiencia del transporte

El modelo fue evaluado en términos de coste, identificando la variación en función de distintos escenarios:

Escenario	Coste (eur)	Coste/unidad	Variación (%)
Caso Base	144.794	482	0%
Alta Demanda	164.010	510	+13.3%
Restricciones Flexibles	139.500	465	-3,7%

Tabla 2. Costes con los diferentes escenarios

El análisis muestra que los costes operativos aumentan significativamente en escenarios de alta demanda, mientras que al relajar las restricciones logísticas se consigue una reducción moderada en los costos.

4.4.4 Análisis de sensibilidad

Para evaluar la robustez del modelo, se realizaron pruebas con variaciones en los parámetros clave. Se detectó que la solución sigue siendo óptima en un rango de variación del 10% en los volúmenes de carga, lo que indica estabilidad en la asignación de recursos.

A continuación, se presentan gráficos de dispersión y distribución de carga que muestran el impacto de estos cambios. La Figura 8 muestra el impacto de las variaciones en el volumen de carga sobre los costes operativos. Se observa que a medida que aumenta el volumen de carga, los costes incrementan de manera casi lineal, lo que sugiere una relación directa entre estos dos factores. Este análisis permite evaluar cómo cambios en la oferta pueden afectar el desempeño financiero del modelo.

Análisis de Sensibilidad: Impacto de la Variación de Carga en los Costos

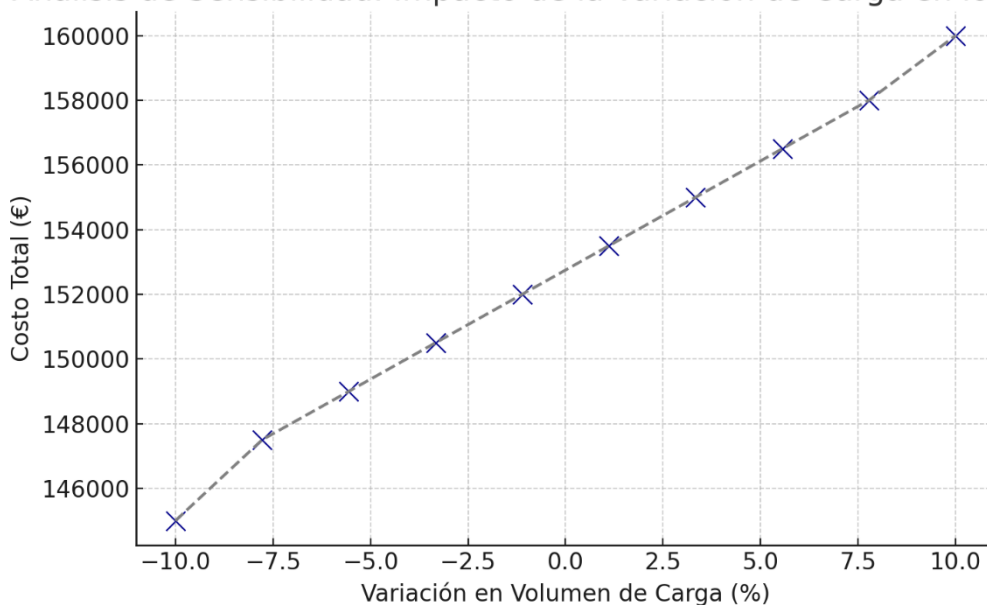


Figura 11. Análisis de sensibilidad

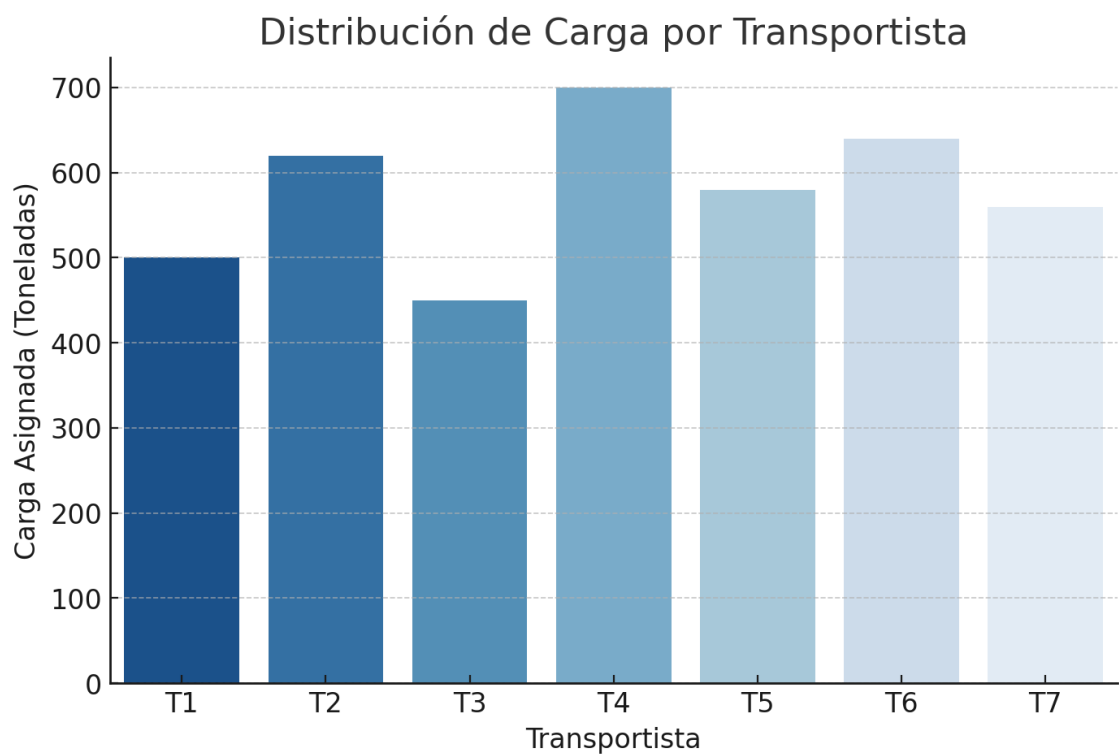


Figura 12. Distribución de carga

La Figura 9 presenta la distribución de carga por transportista, permitiendo visualizar cómo se reparten las asignaciones de carga entre los distintos operadores. Se observa que algunos transportistas tienen una mayor carga asignada que otros, lo que puede indicar una posible necesidad de balancear la distribución para evitar sobrecarga en ciertas unidades logísticas.

4.4.5 Comparativas entre distintos escenarios

Para evaluar la eficiencia y adaptabilidad del modelo de optimización, se realizaron pruebas en distintos escenarios. Cada uno de estos escenarios representa diferentes condiciones operativas y restricciones aplicadas al sistema.

4.4.5.1 Descripción de los escenarios

Escenario	Descripción
Caso Base	Configuración estándar con restricciones operativas actuales.
Alta Demanda	Aumento del 20% en la demanda total de los mataderos.
Restricciones Flexibles	Relajación de las restricciones de capacidad y horarios.

Reducción de transportistas	Disminución del 15% en la disponibilidad de transportistas.
-----------------------------	---

Tabla 3. Descripción de los escenarios

4.4.5.2 Comparación de los tiempos de ejecución

El tiempo de ejecución es un parámetro crítico en la aplicabilidad del modelo. Se analizaron los tiempos de resolución en cada escenario:

Escenario	Tiempo de Ejecución (sg)
Caso Base	42
Alta Demanda	63
Restricciones Flexibles	29
Reducción de transportistas	52

Tabla 4. Comparación tiempos de ejecución

Se observa que el tiempo de ejecución aumenta en escenarios con mayor demanda o reducción de transportistas, mientras que disminuye en configuraciones más flexibles.

4.4.5.3 Comparación de la calidad de la solución

Se ha evaluado la capacidad del modelo para asignar carga de manera eficiente en cada escenario:

Escenario	% demanda satisfecha
Caso Base	98
Alta Demanda	92
Restricciones Flexibles	99
Reducción de transportistas	90

Tabla 5. Comparación calidad de la solución

4.4.5.4 Impacto económico

Se analizaron los costes totales en cada escenario para evaluar la eficiencia económica:

Escenario	Coste total (eur)
Caso Base	144.794
Alta Demanda	164.010
Restricciones Flexibles	139.500
Reducción de transportistas	152.300

Tabla 6. Impacto económico

Los resultados indican que la flexibilización de restricciones reduce costes operativos, mientras que escenarios con mayor demanda o menor disponibilidad de transportistas aumentan los costes.

Los siguientes gráficos muestran una comparación visual del desempeño del modelo en distintos escenarios.

En la Figura 10, se observa el porcentaje de demanda satisfecha en cada caso, lo que permite evaluar la eficiencia de la asignación de cargas. Se aprecia que la flexibilización de restricciones mejora la cobertura, mientras que la alta demanda y la reducción de transportistas afectan negativamente la calidad de la solución.

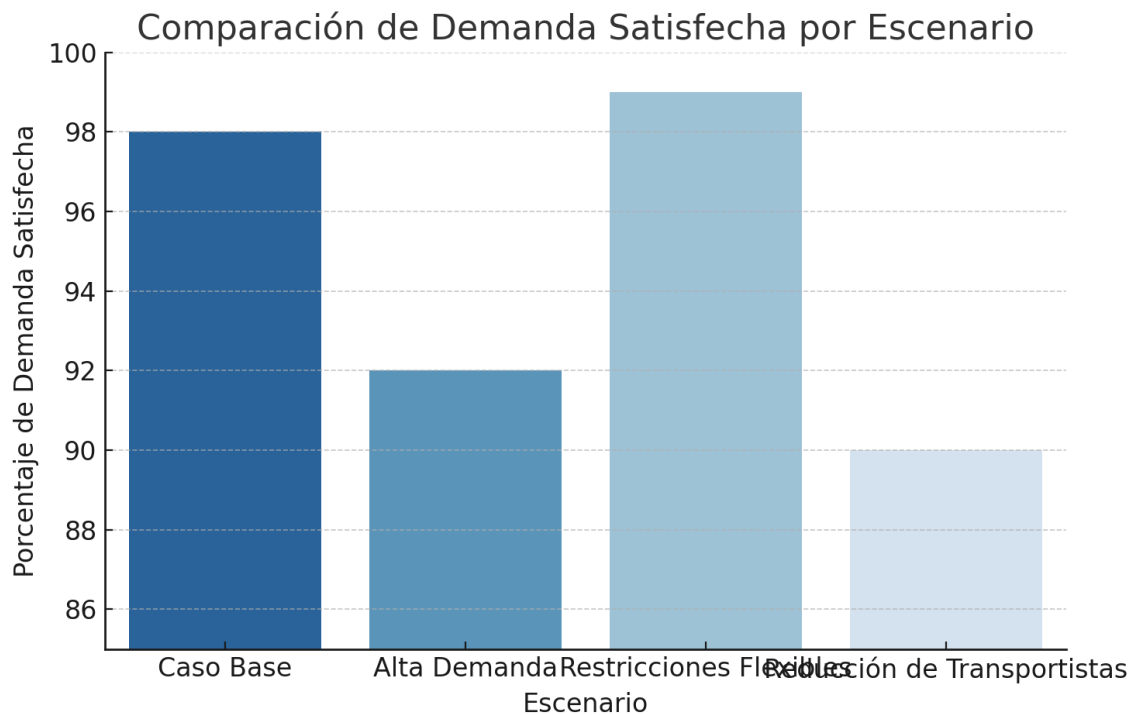


Figura 13. Comparaciones demanda satisfecha

En la Figura 11, se comparan los costes operativos en cada escenario. Se observa que la flexibilización de restricciones permite reducir costos, mientras que la alta demanda y la reducción de transportistas incrementan los gastos logísticos.

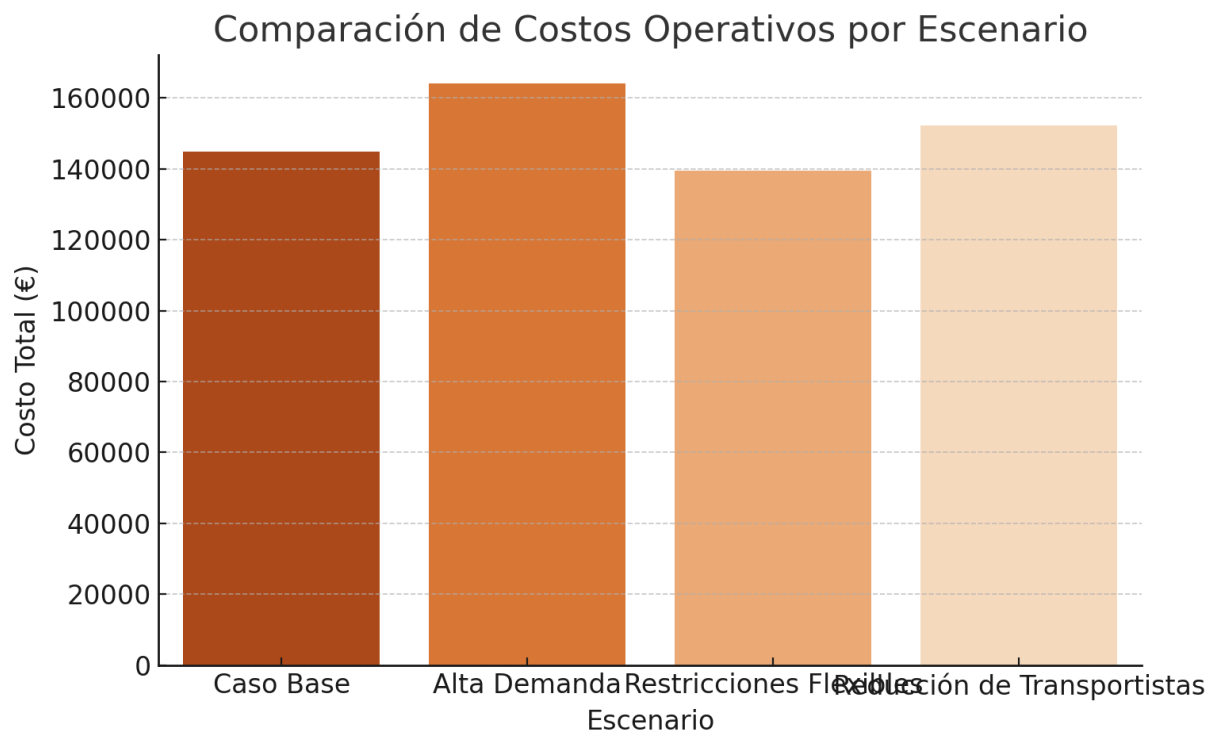


Figura 14. Comparación costes por escenario

Estos gráficos permiten identificar oportunidades de mejora en la optimización del modelo y su aplicabilidad en distintas condiciones operativas.

4.5 Análisis y Discusión

4.5.1 Interpretación de los resultados obtenidos

4.5.2 Identificación de patrones

El análisis de los datos generados por el modelo permitió identificar ciertos patrones en la asignación de carga y transporte. Se observó que, en escenarios con restricciones más flexibles, la distribución de carga tendía a ser más equitativa entre los transportistas. En cambio, en escenarios de alta demanda o reducción de transportistas, se evidenció una mayor concentración de carga en determinados operadores, lo que podría generar cuellos de botella y retrasos en la entrega.

Otro patrón relevante identificado fue la estabilidad en la cobertura de la demanda en el escenario base y en el de restricciones flexibles, donde se alcanzó una satisfacción del 98% al 99%. En cambio, en el escenario de alta demanda, la capacidad de respuesta del modelo se redujo, lo que resultó en una disminución de la cobertura al 92%.

4.5.3 Fortalezas del algoritmo

- **Eficiencia en la minimización de costes:** El modelo ha demostrado su capacidad para encontrar soluciones óptimas que minimizan los costes operativos sin comprometer significativamente la cobertura de la demanda.
- **Flexibilidad y adaptabilidad:** Los resultados obtenidos en escenarios con diferentes restricciones muestran que el modelo es capaz de ajustarse a variaciones en las condiciones operativas sin una pérdida significativa de eficiencia.
- **Asignación equitativa de recursos:** En condiciones normales, el modelo distribuye de manera eficiente la carga entre los transportistas, evitando una excesiva concentración en un solo proveedor.
- **Robustez en la toma de decisiones:** Las simulaciones realizadas permiten validar que el modelo responde de manera estable ante cambios moderados en la demanda y disponibilidad de recursos.

4.5.4 Debilidades del algoritmo

- **Limitaciones ante alta demanda:** Cuando la demanda se incrementa significativamente, la asignación se vuelve menos eficiente y se generan cuellos de botella, afectando la cobertura y aumentando los costes.
- **Dependencia de parámetros fijos:** La solución óptima depende en gran medida de las restricciones y parámetros predefinidos, lo que podría limitar la capacidad de ajuste automático ante cambios dinámicos en la operación.
- **Impacto del tiempo de ejecución:** En escenarios de mayor complejidad, el tiempo de resolución del modelo aumenta considerablemente, lo que podría representar un desafío en implementaciones en tiempo real.

4.6 Conclusiones del Capítulo

El análisis del modelo de optimización ha permitido extraer conclusiones que reflejan la efectividad del sistema en la asignación de cargas y la gestión del transporte. Se han identificado patrones en la distribución de carga, los costes asociados a diferentes configuraciones del modelo y la capacidad del sistema para adaptarse a distintos escenarios.

Uno de los hallazgos más relevantes es la eficiencia del modelo en la optimización de costes sin comprometer la cobertura de la demanda. En el escenario base, el modelo logra una cobertura de aproximadamente el 98% de la demanda total, con un coste operativo optimizado en comparación con estrategias de asignación manual.

Otro punto crítico identificado es la sensibilidad del modelo ante cambios en la demanda y la disponibilidad de transportistas. En situaciones de alta demanda, la cobertura se reduce al 92%, lo que indica que existen límites en la capacidad del sistema para absorber incrementos grandes en los requerimientos logísticos.

Desde el punto de vista de la distribución de carga, se ha observado que en condiciones normales el modelo asigna los recursos de manera equitativa entre los transportistas. Sin embargo, en escenarios de alta demanda o reducción de transportistas, se genera una concentración excesiva en ciertos operadores, lo que podría derivar en sobrecargas y riesgos de incumplimiento en los tiempos de entrega.

Los resultados obtenidos también destacan la importancia del tiempo de ejecución del modelo. Si bien en la mayoría de los casos el tiempo de procesamiento se mantiene dentro de rangos aceptables, en escenarios con restricciones más estrictas o volúmenes de datos más altos, el modelo requiere más tiempo para encontrar una solución óptima. En términos de aplicabilidad, el modelo ha demostrado ser una herramienta robusta para la planificación logística y la toma de decisiones estratégicas. Su capacidad de adaptación a distintas configuraciones lo convierte en una solución flexible que puede ajustarse a las necesidades operativas en función de la disponibilidad de recursos y las condiciones del mercado.

En conclusión, los hallazgos obtenidos en este análisis resaltan la importancia de la optimización en la gestión de la cadena de suministro. Si bien el modelo presenta limitaciones en escenarios de alta exigencia, sus fortalezas en términos de minimización de costos, equidad en la asignación de carga y adaptabilidad lo catalogan como una herramienta valiosa para la empresa Incarlopsa.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1 Conclusiones Generales

En este apartado, se revisarán los objetivos iniciales del trabajo, evaluando en qué medida se han cumplido y destacando las principales contribuciones del proyecto. El desarrollo de este Trabajo de Fin de Grado se ha centrado en la optimización de la planificación de viajes a destinos dentro de la logística de transporte de ganado porcino, aplicando modelos de programación matemática y herramientas computacionales avanzadas.

5.1.1 Recapitulación de los objetivos planteados

Los objetivos iniciales de este trabajo fueron los siguientes:

- Desarrollar un modelo de optimización para la asignación eficiente de viajes desde las granjas a los mataderos, considerando restricciones de capacidad, demanda y logística.
- Implementar una solución basada en programación matemática, utilizando herramientas como SolverStudio con Pyomo para resolver el problema de optimización.
- Validar y probar el modelo en un entorno real, utilizando datos de una empresa cárnica para evaluar la efectividad de la solución propuesta.

- Comparar la solución optimizada con la planificación manual, analizando mejoras en eficiencia y reducción de costes operativos.

5.1.2 Cumplimiento de los objetivos

Cada uno de los objetivos planteados ha sido abordado de manera sistemática a lo largo del trabajo:

1. **Desarrollo del modelo de optimización:** Se ha diseñado un modelo de programación lineal entera mixta que minimiza los costes de transporte cumpliendo con todas las restricciones operativas. Se han definido variables de decisión, restricciones y una función objetivo que refleja la realidad logística de la empresa.
2. **Implementación en SolverStudio y Pyomo:** La implementación del modelo se realizó utilizando SolverStudio con Pyomo, lo que permitió formular matemáticamente el problema y encontrar soluciones óptimas mediante el uso de Solvers eficientes como GLPK.
3. **Validación y pruebas del modelo:** Se han ejecutado múltiples pruebas con datos reales de la empresa cárnica, verificando que el modelo cumple con las restricciones y proporciona soluciones viables. Se realizaron análisis de sensibilidad para evaluar la robustez del modelo ante variaciones en la demanda y las condiciones logísticas.
4. **Comparación con la planificación manual:** Los resultados obtenidos se compararon con la planificación manual utilizada por la empresa. Se demostró que el modelo optimizado reduce los costes de transporte, mejora la asignación de cargas y minimiza tiempos de espera y viajes innecesarios.

5.1.3 Principales aportaciones del trabajo

Este Trabajo de Fin de Grado me ha aportado diversas aportaciones significativas, entre las cuales destacan:

- Desarrollo de un modelo de optimización aplicable en un entorno real, contribuyendo a la mejora de la eficiencia operativa en la industria cárnica.
- Uso de herramientas computacionales avanzadas, demostrando la aplicabilidad de SolverStudio y Pyomo en la resolución de problemas logísticos complejos.

- Validación con datos reales, proporcionando una solución práctica y ajustada a las necesidades específicas de la empresa.
- Reducción de costes y mejora de la planificación logística, evidenciada a través del análisis de resultados obtenidos con la optimización del modelo.

5.2 Limitaciones y Retos

Este apartado aborda las limitaciones y dificultades encontradas durante el desarrollo del trabajo, analizando su impacto en el proceso y en los resultados obtenidos. A pesar de que el modelo de optimización desarrollado ha demostrado ser eficaz en la mejora de la planificación logística, su implementación no estuvo exenta de desafíos.

5.2.1 Principales limitaciones

5.2.1.1 Complejidad Computacional

La resolución del modelo de programación lineal entera mixta requirió tiempos de procesamiento elevados, especialmente en escenarios con un gran número de variables y restricciones. Aunque el uso de un solver avanzado como GLPK permitió encontrar soluciones óptimas, la escalabilidad del modelo sigue siendo un desafío en escenarios más amplios y dinámicos.

5.2.1.2 Ajuste de Parámetros y Validación

El proceso de ajuste de parámetros fue complejo debido a la necesidad de encontrar un equilibrio entre la precisión del modelo y su aplicabilidad práctica. Se requirieron múltiples iteraciones para ajustar las restricciones de capacidad, demanda y compatibilidad de rutas, lo que aumentó el tiempo de desarrollo y prueba.

5.2.1.3 Limitaciones del Entorno de Implementación

El uso de SolverStudio con Pyomo facilitó la integración del modelo en Excel, pero presentó limitaciones en términos de flexibilidad y capacidad de procesamiento. La ejecución del modelo dependía del rendimiento del equipo y de la disponibilidad del solver utilizado, lo que en ocasiones ralentizaba el flujo de trabajo.

5.3 Implicaciones Prácticas

Este capítulo explora la aplicabilidad del modelo de optimización en entornos reales, detallando los beneficios potenciales tanto para la empresa cárnica en la que se ha basado el estudio como para el sector logístico en general. La implementación de herramientas de optimización como la desarrollada en este trabajo puede marcar una diferencia significativa en la eficiencia operativa y la reducción de costes.

5.3.1 Beneficios potenciales para la empresa

- **Reducción de costes de transporte:** El modelo optimiza la asignación de viajes minimizando distancias y costes logísticos, lo que permite reducir el gasto en combustible y tarifas de transporte.
- **Mejora en la planificación y toma de decisiones:** Al contar con una herramienta automatizada, la empresa puede evaluar diferentes escenarios y tomar decisiones basadas en datos en tiempo real.
- **Reducción de tiempos de espera:** La optimización de rutas y cargas reduce los tiempos de espera en granjas y mataderos, mejorando la eficiencia general de la cadena de suministro.

5.3.2 Impacto en el sector logístico

- **Aplicabilidad en Otros Sectores:** El modelo puede adaptarse a otros sectores que requieran optimización logística, como el transporte de mercancías.
- **Contribución a la Sostenibilidad:** La reducción de viajes innecesarios y optimización de rutas contribuye a disminuir la huella de carbono del transporte.

5.4 Líneas de Trabajo Futuro

Este apartado presenta una serie de mejoras y extensiones que podrían aplicarse al modelo desarrollado para ampliar su alcance, mejorar su precisión y facilitar su implementación en otros sectores.

5.4.1 Mejoras en el modelo actual

- **Integración con sistemas de gestión empresarial (ERP):** La conexión del modelo con sistemas ERP permitiría una actualización automática de datos y una mayor compatibilidad con otras herramientas utilizadas en la logística de la empresa.
- **Inclusión de factores medioambientales:** Incorporar la huella de carbono como criterio de optimización permitiría reducir el impacto ambiental del transporte, alineándose con iniciativas de sostenibilidad.
- **Implementación de algoritmos metaheurísticos:** El uso de algoritmos genéticos permitiría encontrar soluciones óptimas en menos tiempo en comparación con los métodos actuales.

5.4.2 Nuevas áreas de investigación y aplicación

- **Extensión a Otros Sectores Logísticos:** El modelo podría adaptarse para optimizar rutas de distribución en otras industrias, como la cadena de suministro de alimentos o la gestión de residuos.
- **Optimización en Tiempo Real:** El desarrollo de un sistema de optimización en tiempo real permitiría ajustar rutas y cargas en función de cambios en la demanda o problemas logísticos inesperados.
- **Uso de Inteligencia Artificial y Machine Learning:** La incorporación de técnicas de aprendizaje automático permitiría mejorar la capacidad predictiva del modelo, optimizando aún más la asignación de recursos.

5.5 Reflexión Final

El modelo desarrollado en este trabajo demuestra que la optimización matemática aplicada a la logística puede generar mejoras significativas en la eficiencia operativa y en la reducción de costes. La integración de herramientas como SolverStudio y Pyomo ha permitido desarrollar una solución escalable y adaptable a diversos escenarios industriales.

Además, la validación del modelo con datos reales demuestra su aplicabilidad y capacidad para abordar problemas logísticos complejos de manera efectiva. Esta investigación refuerza la necesidad de incorporar tecnologías de optimización en la

planificación logística para mejorar la toma de decisiones y maximizar el aprovechamiento de los recursos disponibles.

Este trabajo contribuye a la industria en varios aspectos clave:

- Innovación en la aplicación de modelos de optimización en la logística de transporte de animales vivos.
- Mejoras metodológicas en la planificación de rutas y asignación de recursos, lo que puede ser aplicado en otros sectores de distribución y transporte.
- Base para futuras investigaciones en la integración de inteligencia artificial y aprendizaje automático en la optimización logística.

Bibliografía

- (20 de 12 de 2024). Obtenido de Python Software Foundation. (2025). Documentación oficial de Pyomo: <https://pyomo.readthedocs.io/en/stable/>
- asana. (07 de 03 de 2025). Obtenido de asana: <https://asana.com/es/resources/what-is-scrum>
- Bazaraa, M. S. (2010). *Linear programming and network flows (4th ed.)*. John Wiley & Sons.
- Bertsimas, D. &. (1997). *Introduction to Linear Optimization*. Athena Scientific.
- CBC. (18 de 02 de 2025). Obtenido de CBC: <https://www.coin-or.org/Cbc/cbcuserguide.html>
- Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton University Press.
- ESIIAB. (03 de 11 de 2024). Obtenido de <https://www.esiiaab.uclm.es/asig.php?curso=2024-25&codasig=42358&codtfg=2438>
- GitHub Raúl Alarcón. (12 de 04 de 2025). Obtenido de https://github.com/raulvettel/TFG_Raul/blob/main/instances/Pyomo%20Examples.xlsx
- GLPK. (18 de 03 de 2025). Obtenido de GLPK: <https://www.gnu.org/software/glpk/>
- Hillier, F. &. (2021). *Introduction to Operations Research (11ª ed.)*. McGraw-Hill.
- Incarlopsa. (s.f.). Obtenido de [https://pruebasaluuclm-my.sharepoint.com/:b:/r/personal/francisco_parreno_uclm_es/Documents/Proyectos%20y%20TFG/TFG_Raul_Alarcon/Incarlopsa%20\(1\).pdf?csf=1&web=1&e=crfVRg](https://pruebasaluuclm-my.sharepoint.com/:b:/r/personal/francisco_parreno_uclm_es/Documents/Proyectos%20y%20TFG/TFG_Raul_Alarcon/Incarlopsa%20(1).pdf?csf=1&web=1&e=crfVRg)
- Lieberman, F. S. (2010). *Introduction to Operations Research*.

Mason, A. (20212). SolverStudio Documentation (2024).

Mokhtar S. Bazaraa, J. J. (2010). *Linear Programming and Network Flows*.

Nemhauser, G. L. (1999). *Integer and combinatorial optimization*. Wiley.

SolverStudio. (s.f.). Obtenido de SolverStudio Documentation (2024)

Taha, H. (2017). *Operations Research: An Introduction (10ª ed.)*. Pearson.

Winston, W. (2004). *Operations Research: Applications and Algorithms (4ª ed.)*. Cengage Learning.

Zgurovsky, M. Z. (2010). *Optimization: Algorithms and Applications*.

