

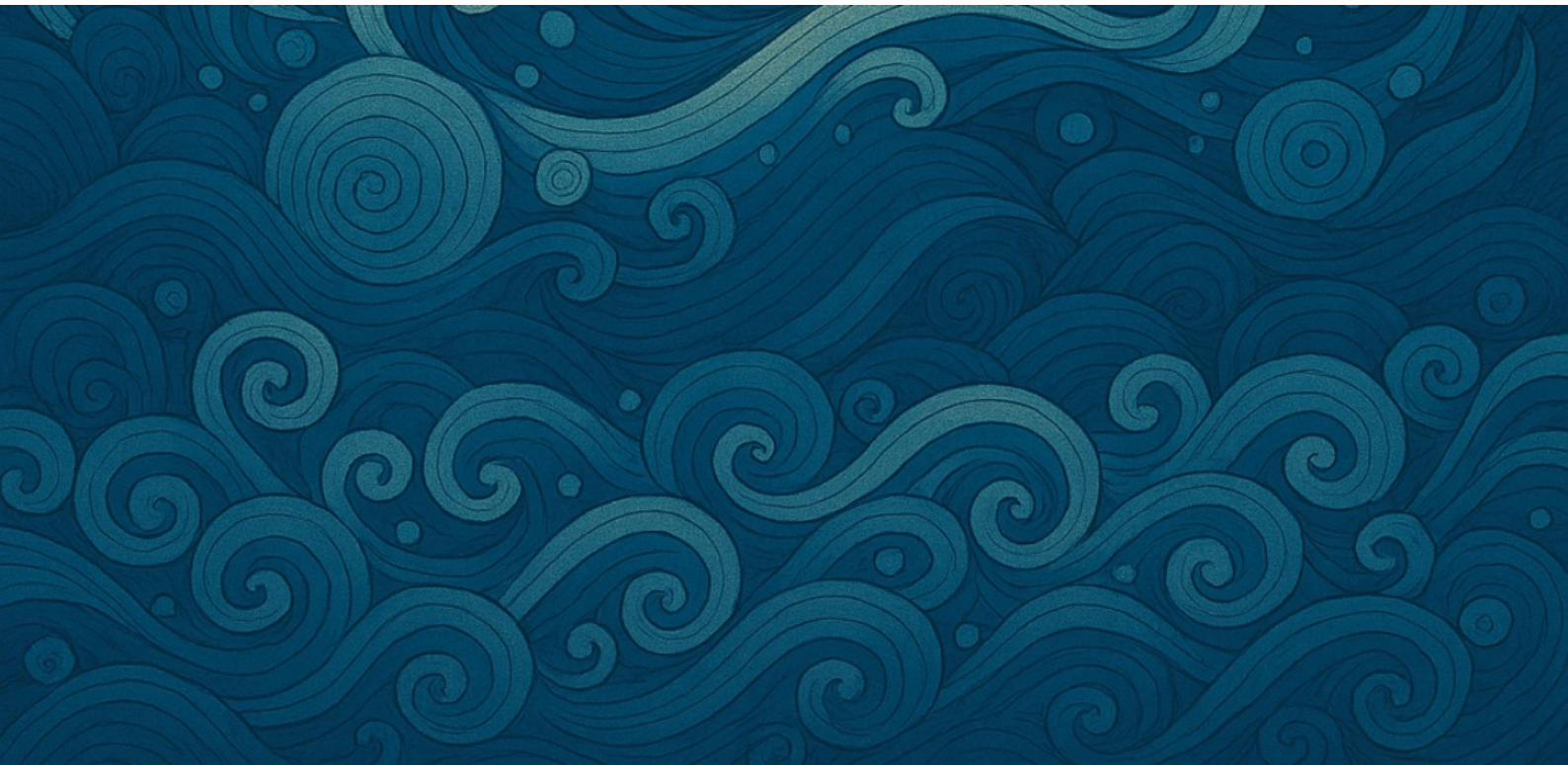
Investigación Operativa

Guía de estudio

Material docente en abierto de la Universidad Rey Juan Carlos
Grado En Ingeniería en Sistemas de Telecomunicación
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Telemática

2025 – 2026 v1.02.00

Javier Quintero & Almudena Buelta





Introducción

La Investigación Operativa es una rama de las matemáticas que tiene como objetivo mejorar u optimizar el rendimiento de sistemas considerando las restricciones impuestas sobre las variables del problema. Esta disciplina surge del estudi ode los problemas relacionados con la dirección y la gestión de sistemas industriales, económicos y sociales. La Investigación Operativa usa técnicas de optimización en procesos de toma de decisiones. Las decisiones a tomar tienen la finalidad de mejorar u optimizar el funcionamiento de estos sistemas suponiendo que los recursos disponibles sean limitados.

Esta guía de estudio está concebida como un recurso integral para los estudiantes matriculados en la asignatura **Investigación Operativa** del los grados en **Ingeniería en Sistemas de Telecomunicación**, **Ingeniería en Tecnologías de la Telecomunicación** y **Ingeniería en Telemática**.

La guía ofrece un **calendario detallado** para cada tema y actividad, una lista seleccionada de **recursos**, objetivos de aprendizaje claros, actividades relevantes y una **lista exhaustiva de contenidos** abordados a lo largo del curso.

Contents

1. Estructura del curso 3
2. Estructura de aprendizaje 4
3. Recursos 4
4. Calendario 5
5. Basic bibliography 6
6. Farewell 6

Este documento se encuentra disponible en el Repositorio Institucional de la Universidad Rey Juan Carlos (BURJC Digital) y se distribuye bajo la licencia "Creative Commons **Attribution** – **Share alike**" disponible en https://creativecommons.org/licenses/by-sa/4.0/deed.es.



1. Estructura del curso

El curso está dividido en seis bloques fundamentales, que se concentran en diferentes temas de la Investigación Operativa, cubriendo todo el temario propuesto en la guía docente de la asignatura. El contenido está dividido de la siguiente forma:

I. Introducción a la Investigación Operativa
   - Introducción y orígenes de la Investigación Operativa
   - Panorama del enfoque de modelado en investigación de operaciones
   - Modelos de optimización matemática

II. Programación Lineal
   - Introducción a la Programación Lineal
   - Solución de problemas de programación lineal
   - Programación Entera

III. Programación en Redes
   - Problemas de transporte y asignación
   - Problemas clásicos de optimización de redes
   - Administración de proyectos con PERT/CPM

IV. Programación no lineal
   - Optimización convexa
   - Optimización con o sin restricciones
   - Condiciones de Karush-Kuhn-Tucker
   - Solución numérica

V. Teoría de colas
   - Modelos de colas y aplicaciones
   - Procesos de nacimiento y muerte
   - Sistemas de colas con resolución analítica
   - Redes de colas
   - Sistemas de colas con distribuciones no exponenciales.

Al finalizar el curso los estudiantes serán capaces de :

- Contar con una base sólida en materias fundamentales y tecnologías que les permita incorporar nuevos métodos y herramientas, mostrando versatilidad y capacidad de adaptación a situaciones novedosas.
- Resolver problemas con iniciativa, tomar decisiones fundamentadas, aplicar creatividad y comunicar eficazmente conocimientos, habilidades y destrezas, reconociendo la responsabilidad ética y profesional del Ingeniero Técnico de Telecomunicación.



- Realizar mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos dentro del ámbito específico de la telecomunicación.
- Adquirir de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, desarrollo o explotación de sistemas y servicios de telecomunicación.
- Utilizar aplicaciones de comunicación e informáticas (ofimáticas, bases de datos, cálculo avanzado, gestión de proyectos, visualización, etc.) como apoyo al desarrollo y explotación de redes, servicios y aplicaciones de telecomunicación y electrónica.

2. Estructura de aprendizaje

Para los participantes en **modalidad remota**, se recomienda estudiar los temas en el orden indicado en el calendario propuesto. Posteriormente, se sugiere resolver los ejemplos incluidos en el *Libro de Apuntes* y los problemas correspondientes a cada tema, disponibles en la *Colección de Ejercicios*. Al finalizar la segunda y la cuarta parte, se recomienda evaluar el grado de comprensión revisando los objetivos de aprendizaje y trabajando con los exámenes resueltos de muestra.

3. Recursos

The following materials are used throughout the course:

- **Bibliografía básica:** Consulte la bibliografía básica del curso en Teaching Guide.
- **Libro de apuntes:** Una guía completa de los contenidos de la asignatura, organizada según la estructura de las clases, disponible en el Aula Virtual.
- **Diapositivas:** Las diapositivas utilizadas en las clases, disponibles en el Aula Virtual.
- **Colección de Problemas:** Un conjunto extenso de ejercicios que incluye los problemas a resolver en cada clase práctica, problemas adicionales para el aprendizaje individual y exámenes resueltos.
- **Solver**: Optimization toolbox de Matlab. Solver de pyomo para Windows.
- **Graficador**: Geogebra 2-d.
- **Tutorías**: Consulte Aula Virtual y/o contactar al Profesor en javier.quintero@urjc.es o

Start a chat on TEAMS
with Javier Quintero

- **Guía Docente**: Consulta la Guía Docente

4. Calendario

Table 1: Planificación semanal

Semana	Tema	Secs.
1	Introducción-Programación Lineal Continua. **Problemas:** 1.1 – 1.10	Cap. 0 y 1
2	Métodos de resolución. **Problemas:** 1.11 – 1.18	1.1 – 1.4
3	Programación Entera **Problemas:** 2.1 – 2.6	2.1
4	Modelación con variable binaria. **Problemas:** 2.7 – 2.23	2.2
5	Transporte y Asignación **Problemas:** 3.1 – 3.11	3.1 – 3.2
6	Introducción a Grafos y a la Optimización en Redes. **Problemas:** 4.2–4.7	4.1 – 4.3
7	PERT/CPM. **Problemas:** 5.1 – 5.2	5.1 – 5.7
8	Resumen de la primera parte Primer Parcial	8.1 – 8.2
9	Programación no lineal **Problemas:** 6.1–6.3	6.1– 6.3
10	Programación no lineal **Problemas:** 6.3 – 6.5	6.3 – 6.5
11	Introducción a la teoría de colas **Problemas:** 7.1 – 7.3	7.1 – 7.5
12	Variación de número de canales **Problemas:** 7.4 – 7.6	8.1 – 8.5
13	Modelos de colas con población finita. **Problemas:** 7.7 – 7.10	8.6

Continued on next page

Table 1: Planificación semanal (Continued)

14	Redes de colas **Problemas:** 7.10 – 7.12	8.7 – 8.8

5. Basic bibliography

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. https://web.stanford.edu/class/ee364a/.

[2] Frederick S. Hillier and Gerald J. Lieberman. *Introducción a la investigación de operaciones*. McGraw-Hill, 2015.

[3] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

[4] Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, 2009.

[5] Hamdy A. Taha. *Investigación de operaciones*. Pearson Educación, México, 9 edition, 2012.

[6] M. Miranda, *Teoría de colas*. Buenos Aires: EDUCA, 2005. ISBN 9871190395.

6. Farewell

Espero que esta guía de estudio resulte un recurso valioso para profundizar en la comprensión de la asignatura y desarrollar las competencias clave establecidas como objetivos del curso. Le deseo mucho éxito en su trayectoria académica.

Javier Quintero
javier.quintero@urjc.es

Febrero 2026



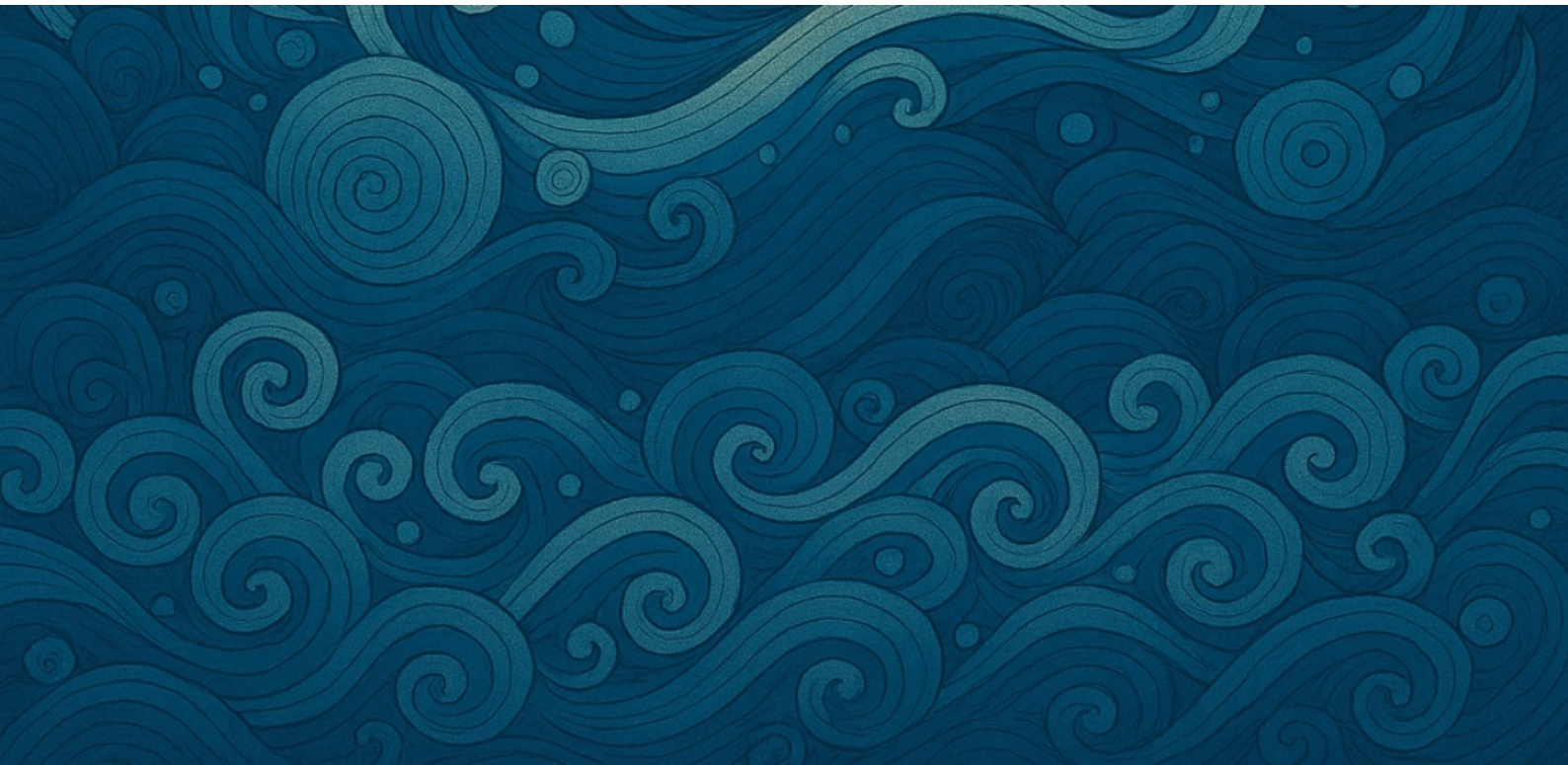
Investigación Operativa

Libro de Apuntes

Material docente en abierto de la Universidad Rey Juan Carlos
Grado En Ingeniería en Sistemas de Telecomunicación
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Telemática

2025-2026 – v1.02.01

Javier Quintero





Investigación Operativa

Javier Quintero



Disponible en Repositorio Institucional de la Universidad Rey Juan Carlos (BURJC Digital)
Copyright © 2026 Javier Quintero
Algunos derechos reservados
Este documento emplea la plantilla: The Legrand Orange Book LaTeXTemplate Version 2.1.1



Este trabajo se distribuye bajo la licencia "Creative Commons **Attribution – Share alike"**

Prefacio

Este texto es una recopilación de apuntes de clase derivados de mi experiencia enseñando Optimización e Investigación Operativa a estudiantes de ingeniería. Las referencias principales empleadas para la elaboración de este material son [2], [3], [4] y [5], ampliamente reconocidas como textos estándar en programas de ingeniería para asignaturas relacionadas.

El público principal de este texto son los estudiantes matriculados en los grados en Ingeniería en Sistemas de Telecomunicación, Ingeniería en Tecnologías de la Telecomunicación e Ingeniería en Telemática; en la Universidad Rey Juan Carlos. Para aprovechar plenamente el material, se espera que los estudiantes tengan conocimientos previos de Cálculo, Álgebra y Matlab.

El contenido está estructurado para alinearse con el programa del curso, dividido en varios temas:

1. Introducción a la Investigación Operativa
2. Panorama del enfoque de modelado en investigación de operaciones
3. Introducción a la Programación Lineal
4. Solución de problemas de programación lineal
5. Programación Entera
6. Problemas de transporte y asignación
7. Modelos de optimización de redes
8. Administración de proyectos con PERT/CPM
9. Programación no lineal
10. Teoría de colas

Para apoyar el proceso de aprendizaje, este texto se complementa con recursos adicionales:

I. **Guía de estudio**: Información importante sobre el curso y las diferentes sesiones. Programación temporal de actividades.
II. **Colección de problemas**: Amplia colección de problemas para usar durante las diferentes sesiones prácticas, incluyendo exámenes resueltos.
III. **Transparencias de clase**: Las diapositivas utilizadas durante el curso.
IV. **Pyomo Solver for Windows**: Un programa basado en Pyomo para Windows, que permite resolver numéricamente modelos de optimización.

Quisiera agradecer de modo especial a los siguientes profesores, María Ángeles Rodríguez de Cara, Víctor Manuel González Vicente, Sara Cornejo Bueno y Elena Aparicio Esteve. Sin su labor hubiese sido imposible la elaboración de estos materiales.

En caso de dudas, puedes contactarme en TEAMS.

Javier Quintero
javier.quintero@urjc.es
Enero 2026



Índice

I Introducción a la Investigación Operativa

0 Introducción a la Investigación Operativa 13
0.1 Orígenes de la Investigación Operativa 13
0.2 Naturaleza de la Investigación Operativa 14
0.3 Panorama del enfoque de modelado en investigación de operaciones 15
0.4 Algoritmos y paquetes de Investigación Operativa 16
0.4.1 Paquetes libres y de acceso abierto 16
0.5 La investigación Operativa dentro de la optimización matemática 17
0.6 Modelos de Optimización 19
0.7 Caracterización de la Solución Óptima de un Problema de Investigación Operativa 21
0.8 Clasificación de problemas de optimización 22

II Programación Lineal

1 Programación lineal con variable continua 27
1.1 Modelos de programación lineal. Elementos de un problema LP 27
1.2 Ejemplo prototipo de problema LP 28
1.3 Solución de problemas de programación lineal. Esencia del método Símplex 30
1.3.1 Método Analítico: Algoritmo Símplex 30
1.3.2 Método gráfico para problemas bidimensionales. El método del las curvas de nivel . 32

1.3.3 Matlab Optimization Toolbox: linprog . . . 33
1.3.4 Pyomo solver . . . 35
1.4 Análisis post-óptimo: Precios sombra 35
1.4.1 Precios sombra . . . 36

2 Programación Entera . . . 39
2.1 Modelo de Programación Lineal Entera. Modelos Puros y Mixtos 39
2.2 Modelado de Problemas con Variables Binarias 42
2.2.1 Condicionales y pre-requisitos . . . 43
2.2.2 Implicaciones lógicas . . . 44
2.2.3 Incluir K de entre N restricciones . . . 46
2.2.4 Acotación . . . 46
2.2.5 Costes fijos . . . 47

III Programación en Redes

3 Problemas de transporte y asignación . . . 53
3.1 Problema de Transporte 53
3.2 Problema de Asignación 56

4 Modelos de optimización de redes . . . 61
4.1 Teoría de grafos 61
4.2 El problema del camino más corto 63
4.3 Árbol de Expansión Mínima 66
4.4 Problema del flujo máximo 67
4.5 Problema de flujo de costo mínimo 71

5 Administración de proyectos con PERT/CPM . . . 77
5.1 Introducción a PERT y CPM 77
5.2 Ejemplo prototipo: proyecto de Reliable Construction Co 78
5.3 Uso de una red para visualizar un proyecto 78
5.4 Programación de un proyecto con PERT/CPM 78
5.4.1 Cálculo de tiempos mínimos y máximos . . . 79
5.5 Manejo de la incertidumbre en las duraciones de las actividades 81
5.6 Consideración del trueque entre tiempo y costo. Crashing. 85
5.7 Programación y control de los costos del proyecto. Evaluación de PERT/CPM 89

IV Programación no Lineal

6 Programación no lineal . . . 93
6.1 Optimización convexa. Condiciones de primer y segundo orden. 93
6.1.1 Condiciones de primer y segundo orden sin restricciones . . . 97

6.2 Aplicaciones de muestra. Ilustración gráfica de problemas de programación no lineal 98
6.3 Optimización no restringida. Resolución con Matlab 102
6.4 Optimización no lineal con restricciones. Solución con Matlab 104
6.5 Condiciones de Karush-Kuhn-Tucker 106
6.5.1 Solución con Matlab . . . 110
6.5.2 Software: Windows solver . . . 113

V Teoría de Colas

7 Teoría de colas . . . 117
7.1 Introducción 117
7.2 Estructura básica de los modelos de colas. Ejemplo prototipo 118
7.3 Ejemplos de sistemas de colas reales. Aplicaciones 119
7.4 Proceso de nacimiento y muerte 120
7.4.1 Ecuaciones de equilibrio . . . 120
7.5 Modelos de colas basados en procesos de nacimiento y muerte. Ejemplos 121

8 Resolución analítica de modelos de colas . . . 123
8.1 Modelos de colas con disciplina de prioridades 123
8.2 Papel de las distribuciones exponenciales 124
8.3 Nomenclatura a utilizar para los sistemas de colas 125
8.4 Sistemas de un solo canal 128
8.4.1 Modelo M/M/1 . . . 128
8.4.2 Modelo M/M/1/K . . . 131
8.5 Sistemas de varios canales en paralelo 134
8.5.1 Modelo M/M/m . . . 135
8.5.2 Modelo M/M/m/K . . . 138
8.6 Sistemas de población finita 142
8.6.1 Modelo M/M/1/-/N . . . 143
8.6.2 Modelo M/M/m/-/N . . . 146
8.7 Redes de colas 150
8.7.1 Redes en serie y en paralelo . . . 150
8.7.2 Redes con realimentación . . . 153
8.8 Modelos de colas con distribuciones no exponenciales 155
8.8.1 Casos particulares relevantes . . . 157
8.8.2 Implicaciones prácticas en el diseño de redes . . . 157

VI Apéndice. Bibliografía. Índice de contenidos

9 Apéndice . . . 161
A: Hessian Matrix 161

B: Pyomo solver para Windows. Ayuda. **163**
C: Guía de Geogebra **165**
D: Formulario Teoría de Colas **165**
E: Excel Guide **168**
F: Pyomo Guide **169**
G: Funciones de Matlab **174**

Referencias 178

Index 181

Introducción a la Investigación Operativa

0 Introducción a la Investigación Operativa 13

0.1 Orígenes de la Investigación Operativa
0.2 Naturaleza de la Investigación Operativa
0.3 Panorama del enfoque de modelado en investigación de operaciones
0.4 Algoritmos y paquetes de Investigación Operativa
0.5 La investigación Operativa dentro de la optimización matemática
0.6 Modelos de Optimización
0.7 Caracterización de la Solución Óptima de un Problema de Investigación Operativa
0.8 Clasificación de problemas de optimización

0. Introducción a la Investigación Operativa

Para esta parte se emplean como referencia principal los textos [2], [5] y [1].

0.1 Orígenes de la Investigación Operativa

La **Investigación Operativa**, conocida también como *Operational Research* o *Management Science*, es una disciplina científica orientada al apoyo de la toma de decisiones mediante la aplicación de métodos analíticos avanzados. Su objetivo fundamental es proporcionar soluciones racionales y óptimas a problemas complejos relacionados con la asignación eficiente de recursos escasos en sistemas organizativos.

El origen histórico de la Investigación Operativa se sitúa en el contexto de la Segunda Guerra Mundial, cuando las administraciones militares británica y estadounidense recurrieron a grupos interdisciplinarios de científicos –matemáticos, físicos, ingenieros y estadísticos– con el fin de mejorar la planificación y ejecución de operaciones militares. Estos equipos fueron encargados de analizar problemas estratégicos y tácticos tales como la asignación de radares, la optimización de convoyes, la defensa aérea o la gestión de inventarios de armamento, empleando por primera vez modelos matemáticos y métodos cuantitativos de manera sistemática.

Este surgimiento no fue un hecho aislado, sino que estuvo estrechamente ligado al aumento progresivo de la complejidad organizativa derivada de la Revolución Industrial. El crecimiento sostenido de empresas, instituciones y redes de transporte incrementó de forma notable la dificultad asociada a la gestión de recursos humanos, materiales y financieros, poniendo de manifiesto la necesidad de enfoques científicos para la toma de decisiones.

El notable éxito alcanzado por los equipos de Investigación Operativa durante la guerra impulsó su rápida expansión en el ámbito civil tras el conflicto. Durante la década de 1950 se produjo la consolidación formal de la disciplina con la creación de sociedades científicas especializadas, entre las que destaca la *Operational Research Society*, fundada en el Reino Unido y considerada la sociedad científica más antigua dedicada específicamente a este campo.

El desarrollo y la difusión de la Investigación Operativa se vieron además favorecidos por varios factores tecnológicos y metodológicos clave. En primer lugar, la disponibilidad de técnicas matemáticas adecuadas al nivel de desarrollo científico de la época permitió formular modelos cada

vez más realistas. En segundo lugar, la revolución informática facilitó la realización de cálculos complejos que, hasta entonces, resultaban inabordables. Finalmente, la aparición de los ordenadores personales contribuyó decisivamente a la generalización de estas técnicas, haciéndolas accesibles a empresas y organizaciones de todos los tamaños.

En este contexto resulta especialmente relevante la figura de George B. Dantzig (1913–2005), creador del método *Simplex*, uno de los algoritmos más influyentes para la resolución de problemas de programación lineal. Este método constituyó un hito fundamental en la consolidación de la Investigación Operativa como disciplina aplicada, al proporcionar una herramienta eficiente para resolver problemas reales de gran escala.

La estrecha relación entre la evolución de la Investigación Operativa y el desarrollo de la computación queda ilustrada por ejemplos históricos concretos, como la llegada en 1959 del IBM 560 Magnetic Drum Calculator a España, adquirido por RENFE para la gestión de líneas ferroviarias. Este hecho pone de manifiesto cómo los primeros sistemas informáticos fueron empleados de forma temprana para resolver problemas logísticos y de planificación, ámbitos naturales de aplicación de la Investigación Operativa.

En síntesis, la Investigación Operativa surge de la confluencia entre necesidades prácticas de decisión, avances científicos y progreso tecnológico. Desde sus orígenes militares hasta su consolidación como herramienta esencial en sectores como el transporte, la logística, la construcción o las telecomunicaciones, la disciplina ha mantenido como rasgo distintivo una visión global y organizativa, orientada a optimizar el funcionamiento del sistema en su conjunto y no únicamente de sus componentes individuales.

0.2 Naturaleza de la Investigación Operativa

La Investigación Operativa es una disciplina orientada al análisis, gestión y coordinación de actividades dentro de una organización, con el objetivo de mejorar su funcionamiento global. Sus ámbitos de aplicación son muy amplios e incluyen, entre otros, el transporte y la logística, la construcción, las telecomunicaciones, la producción industrial, la energía y los servicios. En todos estos contextos, la Investigación Operativa proporciona un marco metodológico para abordar problemas complejos de decisión en los que intervienen múltiples variables, restricciones y objetivos potencialmente contrapuestos.

Desde un punto de vista metodológico, la Investigación Operativa se caracteriza por un enfoque sistemático y estructurado, que permite transformar situaciones reales, a menudo poco definidas y de gran complejidad, en problemas susceptibles de ser analizados mediante modelos matemáticos. Este enfoque no persigue únicamente la obtención de soluciones factibles, sino la identificación de soluciones óptimas o cuasi óptimas que mejoren de forma cuantificable el desempeño del sistema considerado.

El desarrollo de un estudio de Investigación Operativa suele organizarse en una serie de fases bien definidas, que, aunque se presentan de forma secuencial, deben entenderse como parte de un proceso iterativo:

1. **Observación y formulación del problema.** Esta fase inicial consiste en el análisis detallado de la situación real, la identificación de los objetivos del sistema y la recopilación de los datos relevantes. Una correcta formulación del problema es esencial, ya que determina en gran medida la utilidad y validez de los resultados posteriores.
2. **Construcción del modelo matemático.** A partir de la formulación del problema, se desarrolla un modelo matemático que represente de forma simplificada pero suficientemente realista el sistema bajo estudio. Este modelo debe capturar los elementos esenciales del problema, como las variables de decisión, las restricciones y la función objetivo, de modo que las conclusiones obtenidas sean aplicables a la situación real.

3. **Fase experimental y validación.** El modelo propuesto se somete a pruebas mediante datos reales o simulados con el fin de verificar su coherencia y su capacidad para reproducir el comportamiento del sistema. En esta etapa pueden detectarse deficiencias que requieran modificar las hipótesis iniciales o ajustar el modelo, lo que pone de manifiesto el carácter iterativo del proceso.
4. **Obtención de la solución óptima.** Una vez validado el modelo, se aplican las técnicas analíticas o computacionales adecuadas para identificar la mejor solución entre las múltiples alternativas posibles. Esta solución debe interpretarse en el contexto del problema original y servir de apoyo a la toma de decisiones.

Un aspecto fundamental de la Investigación Operativa es la adopción de una *perspectiva organizativa*. A diferencia de enfoques parciales que optimizan únicamente componentes individuales de un sistema, la Investigación Operativa busca resolver los conflictos de interés entre las distintas partes de una organización, de modo que la decisión final sea la más conveniente para el sistema en su conjunto. Esta visión global es especialmente relevante en entornos complejos, donde las decisiones locales pueden tener efectos adversos a nivel global.

En consecuencia, la Investigación Operativa no debe entenderse únicamente como un conjunto de técnicas matemáticas, sino como una forma de abordar los problemas de decisión desde una perspectiva integradora, rigurosa y orientada a la mejora continua del desempeño organizativo.

0.3 Panorama del enfoque de modelado en investigación de operaciones

El enfoque de modelado constituye el núcleo metodológico de la Investigación Operativa y define su forma característica de abordar problemas complejos de toma de decisiones. Este enfoque se basa en la construcción y el análisis de modelos que representan, de manera simplificada pero rigurosa, sistemas reales en los que intervienen múltiples variables, restricciones y objetivos. El propósito fundamental del modelado es proporcionar una base racional que permita evaluar alternativas y apoyar decisiones de forma objetiva y coherente.

Un estudio de Investigación Operativa no debe entenderse como un ejercicio puramente matemático, sino como un proceso integral que combina análisis cuantitativo, conocimiento del sistema real y juicio profesional. En la práctica, el desarrollo de modelos matemáticos es solo una parte del trabajo total, y su utilidad depende en gran medida de la correcta formulación del problema, de la validez de los supuestos adoptados y de la adecuada interpretación de los resultados.

El proceso de modelado en Investigación Operativa suele estructurarse en una serie de etapas interrelacionadas, que no deben considerarse estrictamente secuenciales, sino como fases de un proceso iterativo y dinámico:

1. **Definición del problema:** El punto de partida es la identificación clara del problema que se desea resolver, así como de los objetivos que se persiguen. Esta fase requiere comprender el contexto organizativo en el que se toma la decisión, delimitar el alcance del estudio y determinar los criterios de desempeño relevantes.
2. **Recolección y análisis de datos:** Una vez definido el problema, se recopilan los datos necesarios para describir el sistema. Estos datos pueden proceder de registros históricos, mediciones directas, estimaciones o hipótesis razonables. Su calidad y fiabilidad son determinantes para la validez del modelo.
3. **Formulación del modelo:** En esta etapa se construye un modelo matemático que represente los aspectos esenciales del problema real. El modelo incluye variables de decisión, parámetros, restricciones y una función objetivo que refleja los objetivos del sistema. La formulación implica un compromiso entre realismo y simplicidad, ya que un modelo excesivamente complejo puede resultar intratable, mientras que uno demasiado simple puede carecer de utilidad práctica.
4. **Obtención de soluciones:** Una vez formulado el modelo, se emplean métodos analíticos o algoritmos computacionales para obtener soluciones. En muchos casos existen múltiples

soluciones factibles, por lo que el análisis se orienta a identificar aquellas que optimizan el criterio establecido en la función objetivo.

5. **Validación y análisis del modelo:** Las soluciones obtenidas deben evaluarse para comprobar que el modelo representa adecuadamente el sistema real. Esto puede implicar comparar los resultados con datos observados, realizar análisis de sensibilidad o modificar los supuestos iniciales. Si el modelo no resulta satisfactorio, es necesario reformularlo y repetir el proceso.
6. **Implementación y seguimiento:** La etapa final consiste en aplicar los resultados del estudio al problema real. La implementación puede requerir ajustes organizativos, tecnológicos o operativos, así como un seguimiento posterior para verificar que los beneficios esperados se materializan en la práctica.

Un rasgo esencial del enfoque de modelado en Investigación Operativa es su orientación global. Las decisiones se analizan desde la perspectiva del sistema completo, y no únicamente desde el punto de vista de sus componentes individuales. Este enfoque permite identificar y resolver conflictos entre objetivos parciales, garantizando que la solución adoptada sea la más adecuada para el conjunto de la organización.

En conclusión, el enfoque de modelado proporciona un marco sistemático para analizar problemas complejos de decisión, integrando modelos matemáticos, datos y criterios organizativos. Su correcta aplicación permite transformar situaciones reales difíciles de gestionar en problemas estructurados, cuyas soluciones ofrecen un apoyo sólido y cuantificable a la toma de decisiones en ingeniería y gestión.

Para una discusión más ampliada del tema de esta sección remitimos al lector al Capítulo 2 de [2].

0.4 Algoritmos y paquetes de Investigación Operativa

La aplicación práctica de la Investigación Operativa se apoya de manera fundamental en el desarrollo de algoritmos eficientes y en el uso de paquetes de software especializados. Los algoritmos constituyen procedimientos sistemáticos diseñados para resolver modelos matemáticos asociados a problemas de decisión, garantizando, cuando es posible, la obtención de soluciones óptimas en un tiempo computacional razonable.

A lo largo de la evolución de la disciplina se han desarrollado numerosos algoritmos específicos, como métodos para programación lineal, entera y no lineal, algoritmos para problemas de redes, modelos de colas y métodos de simulación. La elección del algoritmo adecuado depende tanto de la naturaleza del problema como de su tamaño y complejidad, siendo habitual recurrir a aproximaciones heurísticas o metaheurísticas cuando los métodos exactos resultan computacionalmente inviables.

El uso de paquetes de Investigación Operativa permite implementar estos algoritmos sin necesidad de desarrollarlos desde cero, facilitando su aplicación en entornos reales. Estos paquetes integran rutinas de optimización, herramientas de análisis y capacidades de gestión de datos, lo que los convierte en instrumentos esenciales tanto en el ámbito académico como en el profesional. No obstante, el uso efectivo de estas herramientas requiere una comprensión sólida de los modelos y algoritmos subyacentes, ya que el software debe entenderse como un apoyo al análisis y no como un sustituto del razonamiento científico propio de la disciplina.

Los programas que permiten el acceso e implementación de diferentes algoritmos para la resolución de modelos de optimización se conocen con **solvers**.

0.4.1 Paquetes libres y de acceso abierto

A continuación se presenta una breve lista de paquetes de software **no comerciales** y de **acceso libre** que son relevantes para la práctica de la Investigación Operativa:

- **OR-Tools** – Conjunto de herramientas de optimización de código abierto desarrollado por Google, que incluye solvers para programación lineal y entera, además de herramientas para problemas de rutas y redes.
- **GLPK (GNU Linear Programming Kit)** – Biblioteca libre para programación lineal y programación entera mixta, ampliamente utilizada en entornos académicos y compatible con muchos lenguajes y entornos de modelado.
- **CBC (COIN-OR Branch and Cut)** – Solver de programación lineal entera mixta parte del proyecto COIN-OR, adecuado para problemas de tamaño pequeño y mediano.
- **SCIP** – Sistema de optimización de código abierto capaz de resolver problemas combinatorios y de programación entera, con interfaces para múltiples lenguajes.
- **Pyomo** – Lenguaje de modelado basado en Python para definir y resolver aplicaciones de optimización estructurada, que soporta una amplia variedad de tipos de problemas, desde lineales hasta no lineales y estocásticos.
- **JuMP** – Lenguaje de modelado algebraico embebido en Julia que facilita la declaración de variables, restricciones y funciones objetivo para múltiples clases de problemas de optimización.
- **SimPy** – Marco de simulación de eventos discretos basado en Python que permite modelar sistemas dinámicos, colas y procesos estocásticos en tiempo discreto.
- **OpenModelica** – Entorno libre para modelado, simulación y análisis de sistemas dinámicos complejos, útil para problemas de simulación y control en Ingeniería.
- **PyORlib** – Biblioteca Python de acceso abierto diseñada para explorar conceptos de investigación operativa y probar distintos métodos de optimización de forma sencilla.
- **COIN-OR Optimization Suite** – Conjunto de herramientas interoperables que incluye múltiples solvers y utilidades para programación lineal, entera, no lineal y otros métodos de optimización.

Para este curso se emplearán de manera específica dos solvers:

- Optimization Toolbox: Módulo de optimización de Matlab, licenciado en los ordenadores de campus.
- Pyomo solver for Windows: Herramienta de código abierto desarrollada para Windows, disponible en Aula Virtual, bajo la pestaña de Recursos. Este solver está basado en Pyomo y fue desarrollado específicamente para los fines de este curso. Más información y ayuda en Apéndice B.

Además en los Apéndices F y E se ofrecen instrucciones para la implementación de modelos de optimización en Python y Excel, como alternativas a las herramientas oficiales del curso.

0.5 La investigación Operativa dentro de la optimización matemática

La Investigación Operativa desempeña un papel esencial en la Ingeniería de Telecomunicaciones, donde el diseño de redes eficientes, la gestión de recursos y la planificación de servicios dependen, en gran medida, de la resolución de problemas matemáticos de decisión. En términos generales, la Investigación Operativa se define como el proceso de determinar la mejor solución dentro de un conjunto de alternativas factibles, de acuerdo con un objetivo específico, como minimizar la latencia en la transmisión de datos, maximizar el ancho de banda disponible o reducir los costes de despliegue de infraestructura.

La Investigación Operativa se enmarca dentro de la disciplina matemática de Optimización. Los problemas abordados, desde un punto de vista puramente matemático, pueden clasificarse en diversas categorías:

- **Programación Lineal:** Problemas en los que tanto la función objetivo como las restricciones son lineales. Ejemplos incluyen la asignación de recursos en redes de telecomunicaciones o la planificación de capacidad en sistemas de transmisión.

- **Programación No Lineal:** Problemas que involucran funciones objetivo o restricciones no lineales, frecuentes en el diseño de antenas, optimización de potencia en sistemas inalámbricos o control de interferencias en redes.
- **Programación Entera y Mixta:** Problemas en los que algunas o todas las variables deben tomar valores enteros. Son comunes en la planificación de rutas en redes IP, asignación de canales en sistemas móviles o ubicación óptima de estaciones base.
- **Optimización con Restricciones:** Problemas que incluyen limitaciones físicas, regulatorias o de calidad de servicio, por ejemplo, garantizar que la latencia no supere ciertos umbrales en redes críticas.
- **Optimización sin Restricciones:** Problemas sin restricciones explícitas, que suelen aparecer en estimación de parámetros o ajuste de modelos en algoritmos de aprendizaje aplicados a telecomunicaciones.
- **Optimización Dinámica:** Problemas donde las decisiones evolucionan en el tiempo, relevantes para la gestión de tráfico en redes, control de congestión o planificación de despliegue de infraestructura.
- **Optimización Estocástica:** Problemas que incorporan incertidumbre, como variabilidad en la demanda de tráfico, fluctuaciones en la calidad del canal o comportamiento impredecible de usuarios.

A lo largo de este curso estudiaremos los diferentes tipos de problemas abordados por la Investigación Operativa, aprenderemos las herramientas matemáticas necesarias para su formulación y resolución, y exploraremos aplicaciones que impactan directamente en la práctica profesional de la Ingeniería de Telecomunicaciones.

Cuando existe una decisión que requiere obtener el mejor resultado posible, decimos que estamos frente a un problema de Investigación Operativa. Surgen entonces varias preguntas: ¿Cómo definir qué significa "el mejor resultado"? ¿Dónde reside el poder de decisión? ¿Son válidas todas las soluciones en el contexto dado? Resolver un problema de Investigación Operativa implica seguir cuatro pasos fundamentales, que denominaremos a partir de ahora el protocolo:

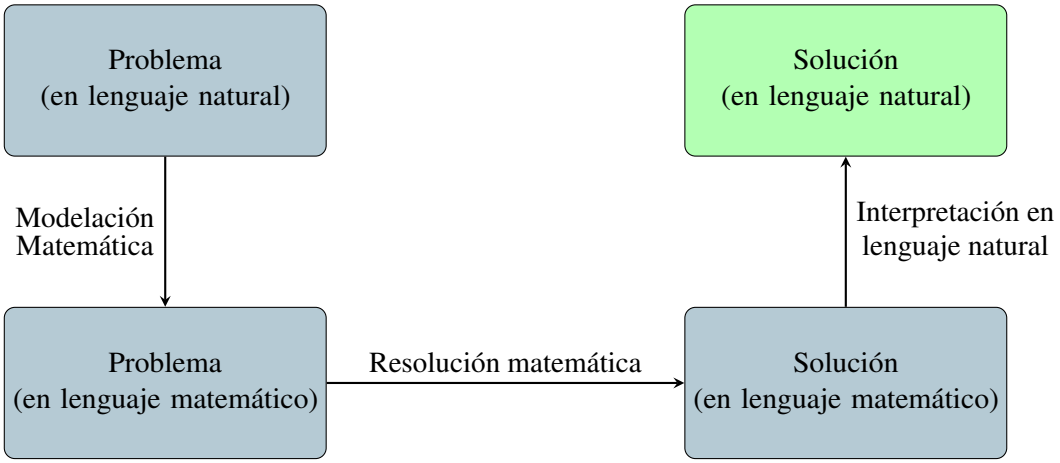


Figura 1: Protocolo de solución de problemas de Investigación Operativa

Como ingenieros, podemos encontrarnos con una pregunta (un problema expresado en lenguaje natural) y buscamos una respuesta (una solución igualmente en lenguaje natural). En este curso estudiaremos el proceso que permite obtener dicha respuesta mediante técnicas matemáticas avanzadas, concluyendo con una solución clara y directa al problema original.

El objetivo principal de esta asignatura es proporcionar al estudiante las herramientas necesarias para afrontar este proceso en un entorno profesional, sirviendo como apoyo al responsable de la

toma de decisiones para que pueda partir de un problema y llegar a una respuesta sin necesidad de involucrarse en los detalles matemáticos internos.

0.6 Modelos de Optimización

Por simplicidad, en este curso nos centraremos en problemas deterministas, es decir, aquellos en los que todos los parámetros del modelo son conocidos y no dependen de factores incontrolables. La formulación matemática de un problema determinista en Investigación Operativa se expresa de la siguiente manera:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p, \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{1}$$

Al conjunto de ecuaciones 1 le llamaremos **modelos de optimización**. Sus elementos son

- **Dirección:** `min` para problemas de minimización y `max` para problemas de maximización. Por defecto, asumimos un problema de minimización en la forma estándar.
- **Variables de decisión** $\mathbf{x} \in \mathbb{R}^n$: las cantidades desconocidas que deben determinarse. Cada componente x_k representa una variable controlable, como la asignación de ancho de banda, la potencia de transmisión o la inclusión de nodos en una red. La dimensión n corresponde al número de variables de decisión.
- **Función objetivo** $f(\mathbf{x})$: la función que se desea minimizar (o maximizar). Cuantifica el criterio de rendimiento, por ejemplo, minimizar la latencia, minimizar el consumo energético o maximizar la capacidad de la red.
- **Región factible** $\mathscr{F}$: el conjunto de todos los $\mathbf{x}$ que satisfacen las restricciones:

  $$\mathscr{F} = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \leq 0,\ h_j(\mathbf{x}) = 0\}.$$

  Solo las soluciones dentro de $\mathscr{F}$ son válidas; a estas soluciones válidas las denominamos **soluciones factibles**, y el conjunto de todas ellas se denomina **región factible**.
- **Solución factible:** Cualquier punto $\mathbf{x} \in \mathscr{F}$.
- **Solución no factible:** Cualquier punto $\mathbf{x} \notin \mathscr{F}$.
- **Solución óptima:** Un punto $\mathbf{x}^* \in \mathscr{F}$ que satisface $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para todo $x \in \mathbb{X}$. En caso de un problema de maximización, esta caracterización cambia a $f(\mathbf{x}^*) \geq f(\mathbf{x})$.
- **Restricción activa:** Una restricción activa es aquella que se cumple en igualdad. Por ejemplo, considere la desigualdad $f_1 : 2x_1 + 3x_2 \leq 5$; la solución $x^* = (1,1)$ satisface la desigualdad en igualdad, porque $2(1) + 3(1) = 5$. Decimos que la restricción f_1 está activa en x^*.

R Algunas observaciones.

- Por simplicidad, la forma estándar de un problema de Investigación Operativa se escribe como un problema de minimización. Un problema de maximización puede transformarse en uno de minimización considerando que

  $$\max_{\mathbf{x}} f(x) = -\min_{\mathbf{x}} -f(x).$$

- Para las restricciones de desigualdad asumimos como forma estándar $g_i(x) \leq 0$, dado que la condición $g(x) \geq 0$ es equivalente a $-g(x) \leq 0$.
- La región factible debe ser cerrada, es decir, las restricciones deben incluir las fronteras. Si una frontera de la región no está incluida, el problema puede no tener solución.
- Si la región factible es no acotada, es decir, no puede contenerse en ningún círculo de radio finito, entonces decimos que la región factible es **no acotada**. Véase el Ejemplo 0.1.

■ **Ejemplo 0.1 — Región No Acotada.** Considere la siguiente región factible determinada por las desigualdades,

$$0 \leq x_1 \leq 4,$$
$$0 \leq x_2$$

que corresponde a la Figura 2. Esta región factible es no acotada.

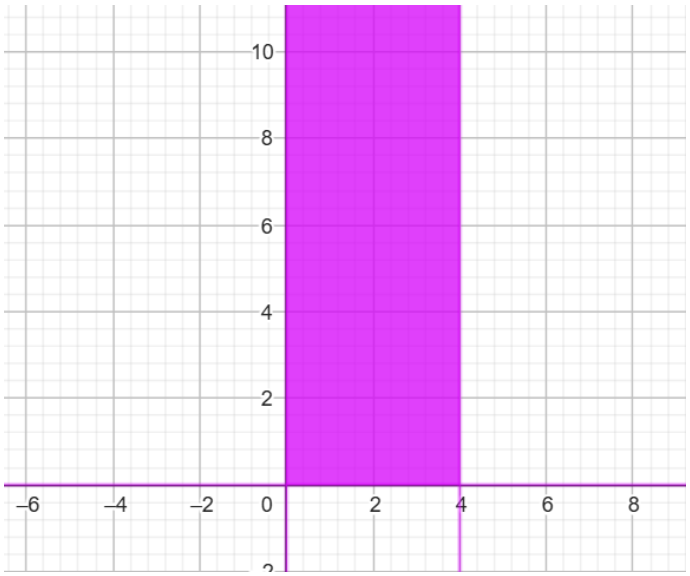


Figura 2: Región factible no acotada de un problema de optimización bidimensional

■

La "solución óptima" de un problema de optimización debe almacenar la siguiente información:

- Valor Óptimo $z^* = f(\mathbf{x}^*)$, el valor mínimo de la función objetivo, que se alcanza en el punto de la
- Solución Óptima $\mathbf{x}^* \in \mathscr{F}$, el cual es el vector que hace que el valor de la función objetivo sea mínimo y, al mismo tiempo, pertenece a la región factible.

■ **Ejemplo 0.2** Considere el siguiente **problema de optimización**

$$\begin{array}{lll} \max & 2x_1 + x_2 & \\ \text{s.t.} & x_1 + x_2 & \leq 2 \\ & x_1, x_2 \geq 0 & \end{array}$$

Este es un problema de programación lineal porque tanto la función objetivo como todas las restricciones son funciones lineales. Los elementos son:

- **Variables de decisión**: $x_1, x_2 \in \mathbb{R}^n$
- **Función Objetivo**: $z = f(x_1, x_2) = 2x_1 + x_2$
- **Restricciones de Desigualdad**: $x_1 + x_2 \leq 2$
- **Restricciones de dominio o de no-negatividad**: $x_1, x_2 \geq 0$
- **Región Factible**: La combinación en $\mathbb{R}^2$ de las restricciones anteriores:

$$\mathscr{F} = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 \leq 2, x_1 \geq 0, x_2 \geq 0\}$$

que corresponde con la Figura3

Observa que esta región factible está acotada. En casos con más de dos variables, una representación gráfica no siempre es útil ni posible. La **solución** de este problema es el par $\mathbf{x}^* = (x_1^*, x_2^*) = (2, 1)$ con valor óptimo $z^* = f(x_1^*, x_2^*) = 5$.

Más detalles sobre cómo resolver este tipo de problemas se presentan en el Capítulo 1. ■

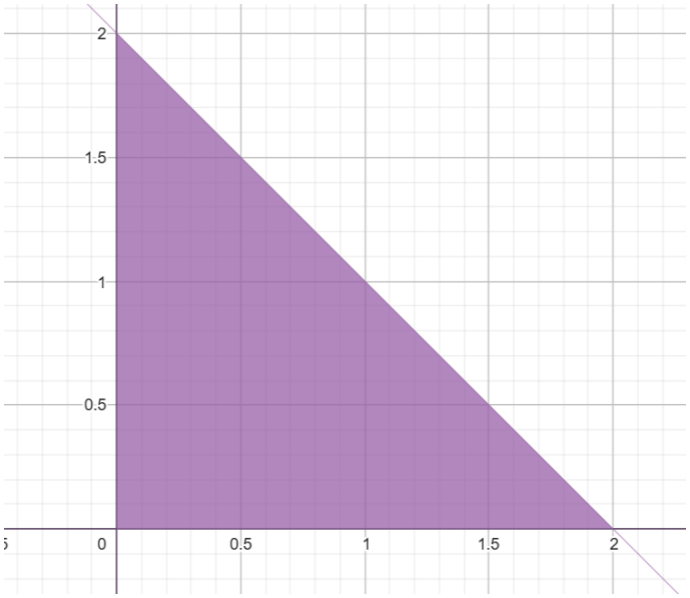


Figura 3: Región factible de un problema de optimización en dos dimensiones

0.7 Caracterización de la Solución Óptima de un Problema de Investigación Operativa

Considera el siguiente problema de optimización:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \le 0, \quad i = 1,\dots,m, \\ & h_j(\mathbf{x}) = 0, \quad j = 1,\dots,p. \end{aligned}$$

La solución óptima $\mathbf{x}^*$ se caracteriza por las siguientes propiedades:

1. **Factibilidad:** La solución debe pertenecer al conjunto factible

$$\mathscr{F} = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \le 0,\ h_j(\mathbf{x}) = 0\}.$$

   Si no existe tal punto, se dice que el problema es *infactible*.

2. **Optimalidad:** Una solución óptima $\mathbf{x}^*$ satisface

$$f(\mathbf{x}^*) \le f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathscr{F}.$$

   Si la desigualdad es estricta, la solución es *única*.

3. **Tipos de optimalidad:**
   - *Óptimo global:* la mejor solución en todo el conjunto factible.
   - *Óptimo local:* la mejor solución solo en una vecindad de $\mathbf{x}^*$.

   En problemas convexos, todo óptimo local también es global; en problemas no convexos pueden existir múltiples óptimos locales.

4. **Condiciones de optimalidad:** Para problemas suaves, las soluciones óptimas se caracterizan mediante las *condiciones de Karush–Kuhn–Tucker (KKT)*, que incluyen:
   - Estacionariedad: el gradiente del Lagrangiano se anula.
   - Factibilidad primal: todas las restricciones se satisfacen.
   - Factibilidad dual: los multiplicadores de Lagrange para desigualdades son no negativos.
   - Holgura complementaria: una restricción de desigualdad es activa solo si su multiplicador es positivo.

Más detalles en la Subsección 6.5.

5. **Calidad de la solución:** En la práctica, los algoritmos numéricos pueden devolver soluciones aproximadas.
   - ε-optimalidad: la solución está dentro de una tolerancia respecto al óptimo verdadero.
   - Tolerancia en restricciones: la factibilidad se satisface dentro de un pequeño error numérico.

0.8 Clasificación de problemas de optimización

En este curso vamos a estudiar problemas dentro de la taxonomía mostrada en la Figura 4. Los problemas se clasifican según la naturaleza matemática de los elementos del modelo involucrados.

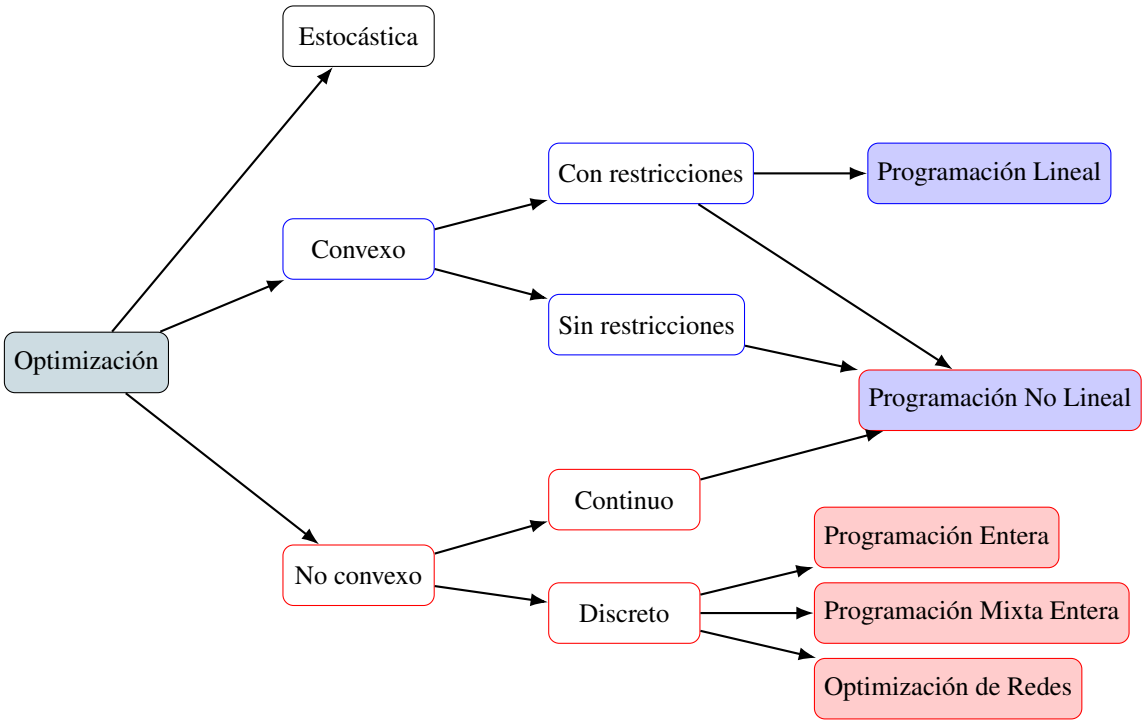


Figura 4: Taxonomía simplificada de los problemas de investigación operativa; para una taxonomía más detallada, consulte este sitio.

Los problemas de investigación operativa pueden clasificarse según las propiedades matemáticas de la función objetivo $f(\mathbf{x})$ y la región factible $\mathscr{F}$. Estas características determinan tanto la dificultad del problema como los tipos de algoritmos que pueden aplicarse para resolverlo.

1. Problemas lineales vs. no lineales

- **Programación Lineal (LP):** tanto la función objetivo como las restricciones son lineales. Estos problemas están bien estudiados y pueden resolverse de manera eficiente mediante métodos como el algoritmo Simplex o los métodos de Punto Interior. Véase el Capítulo 1.
- **Programación No Lineal (NLP):** al menos una de las funciones, ya sea la función objetivo o las restricciones, es no lineal. Estos problemas pueden ser significativamente más complejos, especialmente si son no convexos. Véase el Capítulo 6.

2. Problemas convexos vs. no convexos

- **Optimización convexa:** la función objetivo es convexa y el conjunto factible es un conjunto convexo. Cualquier óptimo local es también óptimo global, lo que hace que estos problemas sean tratables con sólidas garantías teóricas. De hecho, la solución de un problema convexo

está **garantizada** de existir y ser única.

- **Optimización no convexa:** la función objetivo o la región factible es no convexa. Estos problemas pueden tener múltiples mínimos locales, y encontrar el óptimo global puede ser computacionalmente difícil o incluso intratable.

■ **Ejemplo 0.3** La función objetivo en el siguiente problema es no convexa, véase la Figura 5

$$\begin{aligned} \min & f(x) = x^4 - 3x^2 + 2 \\ & x \in \mathbb{R} \end{aligned}$$

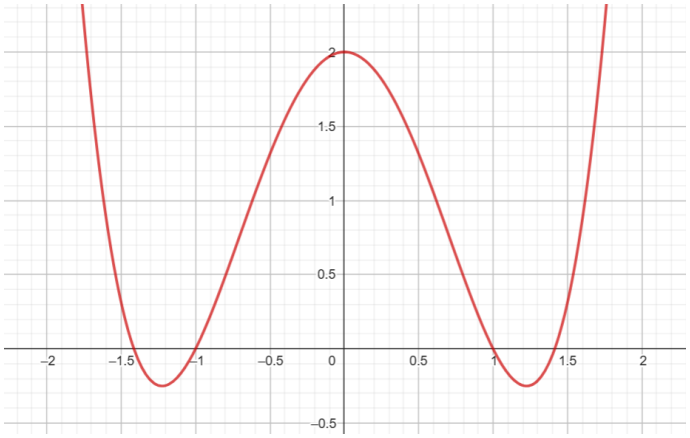


Figura 5: Función objetivo no convexa en dos dimensiones.

- La función objetivo $f(x)$ es un polinomio cuártico con múltiples mínimos locales y un mínimo global.
- Esta no convexidad implica que las técnicas estándar de optimización convexa no pueden garantizar encontrar el óptimo global.

■

R Una función no convexa es aquella cuya segunda derivada no mantiene signo constante. Esto conduce a la existencia de puntos de inflexión y, por tanto, a la presencia de múltiples extremos locales. En la literatura, esto también se conoce como funciones multimodales y representa un problema complejo para abordar con técnicas clásicas de optimización.

3. Problemas discretos vs. continuos

- **Optimización continua:** las variables de decisión **x** pueden tomar cualquier valor real dentro del conjunto factible.
- **Optimización discreta:** las variables de decisión están restringidas a valores discretos (por ejemplo, enteros o binarios). Estos problemas surgen con frecuencia en aplicaciones de planificación, asignación y diseño.
- **Optimización mixta-entera (MIP/MINLP):** algunas variables son continuas mientras que otras están restringidas a valores discretos, combinando la complejidad de ambos mundos. Estos son los problemas más comunes en aplicaciones reales.

4. Problemas con restricciones vs. sin restricciones

- **Optimización sin restricciones:** no existen restricciones explícitas; la región factible es todo el espacio $\mathbb{R}^n$. Estos problemas suelen ser más simples de analizar y resolver. Véase el Ejemplo 0.3.

- **Optimización con restricciones:** la región factible está limitada por igualdades y/o desigualdades. Estas restricciones representan limitaciones físicas, operativas o tecnológicas en aplicaciones reales. Véase la Figura 3.

De los cursos de cálculo multivariable tenemos las herramientas para tratar problemas convexos y continuos. Aplicando simplemente las pruebas de primera y segunda derivada, podemos encontrar fácilmente los valores óptimos de cualquier problema que sea dos veces derivable. Por lo tanto, vamos a comenzar con el caso más sencillo de optimización convexa, según la naturaleza de los elementos del modelo: el Problema de Programación Lineal (LP).

II Programación Lineal

1 Programación lineal con variable continua 27

1.1 Modelos de programación lineal. Elementos de un problema LP

1.2 Ejemplo prototipo de problema LP

1.3 Solución de problemas de programación lineal. Esencia del método Símplex

1.4 Análisis post-óptimo: Precios sombra

2 Programación Entera 39

2.1 Modelo de Programación Lineal Entera. Modelos Puros y Mixtos

2.2 Modelado de Problemas con Variables Binarias

1. Programación lineal con variable continua

La programación lineal (LP) constituye una de las clases más fundamentales y ampliamente estudiadas dentro de los problemas de optimización. Su relevancia radica tanto en su aplicabilidad práctica como en la sólida teoría matemática que sirve de base para métodos de optimización más generales. Los modelos de programación lineal se emplean de manera extensa en diversas áreas de la ingeniería, incluyendo las telecomunicaciones, así como en economía y ciencias de la gestión. Entre las aplicaciones típicas se incluyen:

- **Asignación de Recursos:** determinar la distribución óptima de recursos limitados entre actividades que compiten entre sí, como ancho de banda, potencia de transmisión o canales de comunicación.
- **Planificación de Redes:** optimizar costes, capacidad y rendimiento en el diseño y operación de redes de telecomunicaciones, sujetos a restricciones de tráfico y calidad de servicio.
- **Transporte y Logística de Datos:** minimizar costes de enrutamiento y transmisión cumpliendo requisitos de demanda y disponibilidad de enlaces.
- **Gestión de Infraestructura:** optimizar la asignación de equipos, frecuencias y recursos en sistemas de comunicación para garantizar eficiencia y fiabilidad.

En este capítulo se estudia el problema de LP, los elementos que conforman un modelo lineal, los métodos de solución, las interpretaciones geométricas y las particularidades que hacen que este modelo sea tan relevante en el ámbito de la optimización aplicada a las telecomunicaciones.

1.1 Modelos de programación lineal. Elementos de un problema LP

Definición 1.1.1 — Modelo de Programación Lineal. Un **problema de Programación Lineal (LP)** es un problema de optimización en el que tanto la función objetivo como las restricciones son lineales en las variables de decisión. Formalmente, la forma estándar (canónica o matricial) de un problema de programación lineal se expresa como:

$$
\begin{array}{lll}
\min\limits_{\mathbf{x}} z = & c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\
\text{s.a.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 & \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 & \\
& \vdots & \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m & \\
& x_1, x_2, \ldots, x_n \geq 0 &
\end{array}
\quad \Leftrightarrow \quad
\boxed{\begin{array}{ll} \min\limits_{\mathbf{x}} z = & \mathbf{c}^t\mathbf{x} \\ \text{s.a.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array}} \tag{1.1}
$$

donde:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ es el vector de variables de decisión.
- $\mathbf{c} = (c_1, c_2, \ldots, c_n)^t \in \mathbb{R}^n$ es el vector de coeficientes de la función objetivo (vector de costes).
- $(a_{ij}) = A \in \mathbb{R}^{m \times n}$ es la matriz de coeficientes de las m restricciones:

$$
A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \cdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}
$$

- $\mathbf{b} = (b_1, b_2, \ldots, b_m)^t \in \mathbb{R}^m$ es el vector de términos independientes en las restricciones (vector de recursos).
- El conjunto de restricciones $\mathbf{x} \geq 0 \Rightarrow x_1, x_2, \ldots, x_n \geq 0$ se conoce como restricciones de no negatividad.

R

- La función objetivo $\mathbf{c}^t\mathbf{x}$ es lineal en las variables de decisión.
- La región factible, definida por las desigualdades lineales y las restricciones de no negatividad, es un poliedro convexo.
- Cualquier solución óptima, si existe, se encuentra en un *vértice* de la región factible.

1.2 Ejemplo prototipo de problema LP

■ **Ejemplo 1.1 — Asignación de Producción.** Una empresa tecnológica ha decidido ampliar su catálogo de productos avanzados en el ámbito de las telecomunicaciones, incorporando dos nuevos dispositivos:

- **Producto 1**: un módulo de transmisión que requiere un bastidor metálico y ensamblaje final.
- **Producto 2**: un kit de diagnóstico de red que requiere una carcasa de polímero y ensamblaje final.

La empresa dispone de tres instalaciones especializadas organizadas de la siguiente manera:

- Instalación 1 fabrica **componentes y bastidores metálicos**,
- Instalación 2 produce **partes de polímero y compuestos**,
- Instalación 3 se encarga del **ensamblaje y pruebas finales de los dispositivos de telecomunicaciones**.

Los requisitos de producción son los siguientes:

- El Producto 1 consume capacidad de las Instalaciones 1 y 3,
- El Producto 2 consume capacidad de las Instalaciones 2 y 3.

Cada producto se fabricará en lotes de 20 unidades. La tasa de producción será el número de lotes producidos por semana que satisfacen las restricciones. Los lotes pueden producirse de manera fraccionada. Para formular cuidadosamente el modelo matemático, se han estimado, Tabla 1.1, los siguientes datos:

- el número de horas de producción disponibles por semana en cada instalación para estos dos nuevos productos,
- el número de horas de producción requeridas por lote de cada producto en cada instalación,
- el beneficio por lote de cada producto.

Tabla 1.1: Datos de Ejemplo 1.1

Instalación	**Tiempo de producción por lote (horas)**		**Tiempo disponible por semana (horas)**
	Producto		
	1	**2**	
1	1	0	4
2	0	2	12
3	3	2	18
Beneficio por lote	\$3,000	\$5,000	

El problema consiste en determinar los niveles óptimos de producción semanal de los dos productos, es decir, cuánto fabricar de cada uno para:

- **maximizar** el beneficio de la empresa,
- sujeto a las **restricciones** impuestas por las capacidades de producción disponibles en las tres instalaciones.

Para resolver este problema, activamos el protocolo de la Figura 1. Tenemos un problema en lenguaje natural, por lo que el primer paso es extraer los datos para elaborar un modelo como (1.1).

Primero, observamos que la decisión se centra en los niveles de producción, es decir, el número de dispositivos de cada tipo que la empresa fabrica. Por tanto, declaramos las siguientes variables x_i: número de lotes del producto i que la empresa produce cada semana, para $i = 1,2$. El problema indica que pueden producirse lotes fraccionados, pero no pueden ser negativos, por lo tanto, $x_1, x_2 \geq 0$.

Además, el beneficio depende directamente del número de lotes producidos, es decir:

$$\text{beneficio} = 3000x_1 + 5000x_2.$$

Dado que el objetivo es maximizar el beneficio, la función objetivo se expresa como:

$$z = 3x_1 + 5x_2$$

en miles de dólares.

Existen varias restricciones de producción asociadas a los recursos disponibles, en este caso el tiempo de producción en cada instalación. Por ejemplo, en la instalación 1 hay 4 horas disponibles por semana, y dado que cada lote del producto 1 consume una hora, la restricción correspondiente es $x_1 \leq 4$. Las otras instalaciones generan restricciones adicionales: $2x_2 \leq 12$ y $3x_1 + 2x_2 \leq 18$, respectivamente.

Una vez extraída toda la información, estamos listos para el segundo paso del protocolo 1. Escribimos esta información en lenguaje matemático utilizando la forma estándar proporcionada en (1.1):

$$\begin{array}{lrrrrl} \max\limits_{\mathbf{x}} & z & = 3x_1 & +5x_2 & & \\ \text{s.a.} & & x_1 & & \leq & 4 \\ & & & 2x_2 & \leq & 12 \\ & & 3x_1 & +2x_2 & \leq & 18 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

■

Existen varias suposiciones que hacen viable un modelo de programación lineal (LP):

1. **Proporcionalidad.** Esta suposición implica que:
   a. la contribución de cada actividad al valor de la función objetivo Z es proporcional al nivel de dicha actividad, y
   b. la contribución de cada actividad a cada restricción funcional es proporcional al nivel de dicha actividad.
   c. **Ejemplo:** El coste total de transmisión es proporcional al número de enlaces activados en una red. Si se duplican los enlaces, se duplica el coste.
2. **Aditividad.** Esta suposición implica que toda función en un modelo de programación lineal se expresa como la suma de las contribuciones individuales de las respectivas actividades. **Ejemplo:** El consumo total de ancho de banda en una red es la suma del ancho de banda utilizado por cada conexión individual.
3. **Divisibilidad.** Esta suposición implica que las variables de decisión en un modelo de programación lineal pueden tomar cualquier valor, incluyendo valores no enteros. **Ejemplo:**La asignación de potencia en un sistema de transmisión puede realizarse en valores fraccionarios, por ejemplo, asignar 2.5 W a un canal.
4. **Certeza.** Se asume que los valores asignados a cada parámetro en un modelo de programación lineal son conocidos y permanecen constantes. **Ejemplo:** Se asume que la demanda de tráfico y la capacidad de los enlaces son conocidas y constantes durante el periodo de planificación.

Estas suposiciones son fundamentales para la tratabilidad matemática de la programación lineal. Garantizan que las relaciones en el modelo permanezcan lineales, lo que permite aplicar métodos de solución eficientes como el algoritmo símplex. Sin embargo, en aplicaciones reales, una o más de estas suposiciones pueden no cumplirse estrictamente. Por ejemplo, las variables de decisión pueden necesitar tomar valores enteros, los parámetros pueden involucrar incertidumbre o las relaciones pueden ser no lineales. En tales casos, se requieren extensiones de la programación lineal, como la programación entera, la programación no lineal o la programación estocástica, para modelar el problema con mayor precisión.

1.3 Solución de problemas de programación lineal. Esencia del método Símplex

Un problema lineal es **por definición** un problema convexo. Un problema convexo garantiza la existencia de una solución óptima única. En el caso de problemas lineales, el principio de optimalidad en los vértices nos permite encontrar dicha solución de manera directa.

Definición 1.3.1 — Solución en Vértice. Se denomina solución en vértice a una solución del problema (1.1) que se obtiene como la intersección de dos restricciones. Es decir, $\mathbf{x}^+$ es una **solución en vértice** si y solo si

$$\mathbf{x}^+ \in \{g_i\} \cap \{g_j\} \cap \mathscr{F}$$

donde g_i, g_j son dos restricciones del problema (1).

Teorema 1.3.1 — Optimalidad en Vértices. Un problema de programación lineal (LP) con variables continuas y región factible acotada satisface:

- la solución óptima se encuentra en el vértice con mejor valor de la función objetivo, o
- si dos vértices tienen el mismo valor de la función objetivo, entonces el problema tiene infinitas soluciones situadas en el segmento que une ambos vértices.

1.3.1 Método Analítico: Algoritmo Símplex

■ **Ejemplo 1.2 — Soluciones en Vértices.** Retomemos el Ejemplo 1.1. Utilizando Geogebra podemos graficar la región factible como en la Figura 1.1:

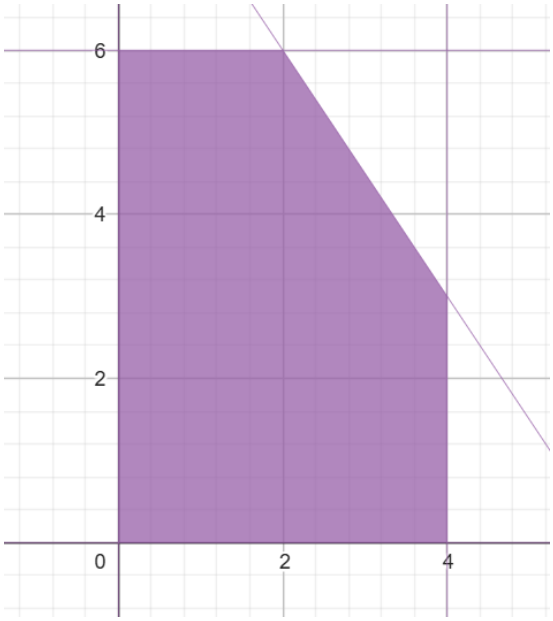


Figura 1.1: Región factible del Ejemplo 1.1

Dado que la región es acotada, según el Teorema 1.3.1, la solución óptima se encuentra en los vértices de la región. Calculamos los vértices mediante la intersección de las restricciones y evaluamos cada punto en la función objetivo:

x	z
$(0,0)$	0
$(4,0)$	12
$(4,3)$	27
$\boxed{(2,6)}$	$\boxed{36}$
$(0,6)$	30

Este problema es de maximización, por lo tanto, dado que $(2,6)$ devuelve el valor máximo de la función objetivo, concluimos que esta es la solución óptima.

Finalmente, cerramos el último paso del protocolo 1, expresando la solución en los mismos términos del problema, es decir, traducimos la solución matemática a lenguaje natural:

"El nivel óptimo de producción semanal es fabricar 2 *lotes del producto* 1 *y* 6 *lotes del producto* 2*, para maximizar el beneficio a $36 mil dólares."*

Obsérvese que en esta respuesta se resume el objetivo principal del problema: nivel de producción semanal para beneficio máximo, utilizando las unidades correspondientes. ∎

Este ejemplo ilustra cómo aplicar el método analítico a un problema bidimensional.

Algorithm 1 Método Analítico

1: Aplicar únicamente si la región factible es acotada.
2: Encontrar el conjunto V de vértices factibles mediante la intersección $\{g_i\} \cap \{g_j\}$ para todo $i = 1, \ldots, m+n$, donde g_i para $i = m+1, \ldots, n+m$ corresponde a las últimas n restricciones de no negatividad.
3: Para cada $v_i \in V$, calcular $f(v_i)$.
4: La solución óptima corresponde al menor (mayor) valor $z^* = f(v_i^*)$ en un problema de minimización (maximización).

Este principio está detrás del algoritmo Símplex. El **algoritmo Símplex** es uno de los métodos más famosos en optimización. Fue desarrollado en 1947 por **George Dantzig**, quien trabajaba en problemas de planificación para la Fuerza Aérea de los EE.UU. Desde entonces, se ha convertido en un pilar de la **investigación operativa** y se ha aplicado en innumerables campos, desde *ingeniería y economía* hasta *telecomunicaciones*.

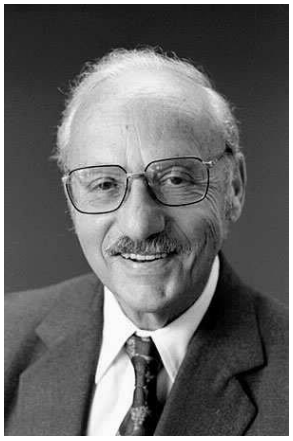


Figura 1.2: George B. Dantzig (1913–2005) Creador del método SÍMPLEX

La idea principal detrás del algoritmo es simple pero poderosa: en un problema de programación lineal, las soluciones factibles forman una figura geométrica llamada **politopo** (un polígono multidimensional). Dantzig demostró que la **solución óptima siempre se encuentra en uno de los vértices** de este politopo. En lugar de verificar todos los puntos posibles, el algoritmo Símplex se desplaza inteligentemente de vértice en vértice, mejorando la solución en cada paso, hasta alcanzar la mejor.

Algunas características importantes del método Símplex son:

- Proporciona no solo la **solución óptima**, sino también información sobre la **sensibilidad** del modelo (cómo afectan los cambios en recursos o coeficientes al resultado).
- En la práctica, es extremadamente **eficiente**, aunque en teoría puede requerir tiempo exponencial en el peor caso.
- Sentó las bases de la optimización moderna e inspiró muchas extensiones, como el **Símplex Revisado** y el **Símplex Dual**.

Hoy en día, incluso con algoritmos más recientes (como los **Métodos de Punto Interior**), el método Símplex sigue siendo ampliamente utilizado debido a su **eficiencia práctica, robustez e interpretabilidad**.

■ **Ejemplo 1.3 — Soluciones infinitas.** Consideremos el siguiente problema de optimización:

$$\begin{array}{lll} \max & x_1 + x_2 & \\ \text{s.a.} & x_1 + x_2 & \leq 2 \\ & x_1, x_2 \geq 0 & \end{array}$$

La región factible tiene tres vértices:

x	z
$(0,0)$	0
$(2,0)$	2
$(0,2)$	2

Obsérvese cómo el valor de la función objetivo coincide en dos vértices. Esto significa, según el Teorema 1.3.1, que las soluciones óptimas se encuentran en el segmento que une ambos vértices, es decir, el segmento:

$$\mathbf{x}^*(\lambda) = \lambda(2,0) + (1-\lambda)(0,2)$$

para $\lambda \in [0,1]$, y $z^* = 2$. Dado que para cada valor de λ existe una solución diferente, concluimos que este problema tiene infinitas soluciones. ■

1.3.2 Método gráfico para problemas bidimensionales. El método del las curvas de nivel

¿Qué significa geométricamente el principio de optimalidad (Teorema 1.3.1)? Añadamos una curva de nivel que represente la función objetivo a la región factible del Ejemplo 1.1. Para ello, utilizamos la calculadora gráfica Geogebra (ver más en el Apéndice C). Para esta sección, los gráficos pueden consultarse aquí.

Algorithm 2 Método Gráfico

1: Aplicar únicamente a problemas bidimensionales.
2: Graficar la región factible mediante la intersección de todas las restricciones.
3: **if** la región factible no está vacía **then**
4: Crear una curva de nivel correspondiente a la función objetivo.
5: Disminuir (problema de minimización) o aumentar (problema de maximización) el nivel de la curva hasta que toque el último punto factible posible.
6: La solución óptima corresponde al menor (mayor) valor de la curva de nivel en un problema de minimización (maximización). La solución $\mathbf{x}^*$ es el último punto factible que toca la curva de nivel.
7: **else**El problema es infactible
8: **end if**

■ **Ejemplo 1.4 — Encontrar la asignación óptima de producción mediante el método gráfico.** El gráfico izquierdo en la Figura 1.3 corresponde al paso inicial del método gráfico descrito anteriormente. La línea verde es la curva de nivel correspondiente a la función objetivo. En el gráfico derecho se observa cómo, al aumentar (es un problema de maximización) el valor de la curva de nivel, encontramos la solución óptima en $x^* = (2,6)$ (el último punto factible) para $z^* = 36$ (el valor de la curva de nivel). ■

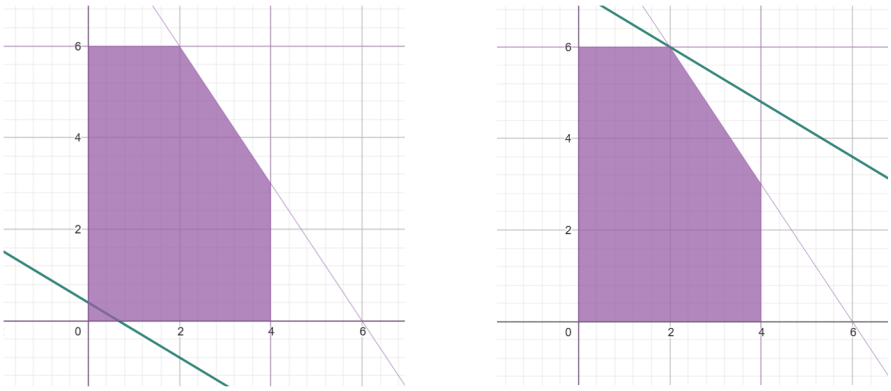


Figura 1.3: Método gráfico aplicado al Ejemplo 1.1

■ **Ejemplo 1.5 — Soluciones infinitas.** Existen casos en los que la función objetivo es paralela a alguna de las fronteras del politopo de la región factible, ver Figura 1.4. En tales casos, tendremos soluciones óptimas infinitas, como se muestra en el Ejemplo 1.3. Para visualizar la región y la implementación del método gráfico, consulte este gráfico en Geogebra. ■

1.3.3 Matlab Optimization Toolbox: linprog

Hasta ahora, el método presentado en esta sección permite resolver problemas pequeños y bidimensionales. Sabemos que, en la práctica, la mayoría de los problemas son mucho más grandes que eso. Por lo tanto, introducimos dos solucionadores computacionales para abordar esta limitación. Comenzaremos con el solver de Matlab para problemas lineales.

Importante — La plantilla de Matlab no estará disponible durante los exámenes. Durante los exámenes se empleará como optimizador única y exclusivamente los módulos de Matlab descritos en clase. La plantilla de Aula Virtual no estará disponible, por lo que un conocimiento de la sintaxis y los comandos básicos para la resolución del problema mediante un script de

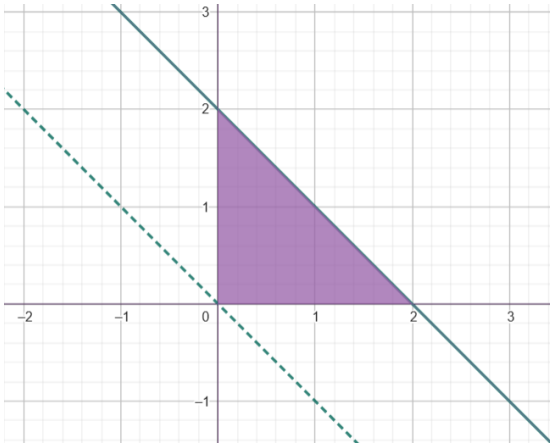


Figura 1.4: Método gráfico aplicado al Ejemplo 1.3

Matlab será requerido.

Para resolver un problema de programación lineal se empleará la función de Matlab conocida como `linprog`, se puede consultar su configuración en Mathworks, o escribiendo `help linprog` en la consola de Matlab. Para más información consultar Apéndice G.

La función `linprog` resuelve los problemas LP con variable continua, esto es, el problema tiene que ser **lineal**, y las variables tienen permitido asumir valores no-enteros. La lista de argumentos se resume en la Tabla 1.2.

Tabla 1.2: Entradas/salidas de la función `linprog` en MATLAB

Entrada / Salida	Descripción	Tipo / Valores
Entradas		
`f`	Vector de coeficientes de la función objetivo	Vector columna
`A`	Matriz de coeficientes de desigualdades lineales	Matriz $m \times n$ (opcional)
`b`	Vector de términos independientes de desigualdades	Vector columna (opcional)
`Aeq`	Matriz de coeficientes de ecuaciones lineales	Matriz $p \times n$ (opcional)
`beq`	Vector de términos independientes de ecuaciones lineales	Vector columna (opcional)
`lb`	Vector de límites inferiores de las variables	Vector columna (opcional)
`ub`	Vector de límites superiores de las variables	Vector columna (opcional)
`x0`	Vector inicial de la solución	Vector columna (opcional)
Salidas		
`x`	Vector solución óptima que satisface las restricciones	Vector columna
`fval`	Valor de la función objetivo en la solución óptima	Escalar
`exitflag`	Indica el resultado de la optimización	Escalar entero
`output`	Información detallada del solver (iteraciones, tiempo, etc.)	Estructura
`lambda`	Multiplicadores de Lagrange asociados a las restricciones lineales	Estructura (si aplica)

En el Código 1.1 se ejemplifica el uso de la función `linprog` para la resolución del Ejemplo 1.1. Nótese que `linprog` solamente acepta problemas de mínimo, con restricciones de menor o igual; problemas en forma canónica, por lo que se debe cambiar el sentido de la desigualdad y considerar el opuesto de la función objetivo. Ver código 1.1.

```
1 % vector de costes
2 f = -[3 5];
```

```
3
4  %vector de recursos
5  b =  [4 12 18];
6
7  %cotas
8  ub = []; %superior
9  lb = zeros(1,2); % inferior
10
11 %matriz de restricciones
12       %x1,x2
13 A = [ 1 0;
14        0 2;
15        3 2];
16
17 %igualdades
18 Aeq = []; beq= [];
19
20 %optimizar
21 [x,fval] = linprog(f,A,b,Aeq,beq,lb,ub)
22
23 %imprimir resultados
24 x
25 -fval
```

Código 1.1: Ejemplo de uso función `linprog`

Nótese que la solución coincide con las soluciones anteriores.

1.3.4 Pyomo solver

Este software se presenta como material complementario del curso para usuarios de Windows, y consiste simplemente en una codificación de `Pyomo` para ejecutarse en ordenadores personales con Windows. Está disponible en Aula Virtual.

Esta solución es una interfaz gráfica sencilla (GUI) que contiene varios paquetes y solucionadores de `Pyomo`. Su uso es intuitivo y sigue las mismas instrucciones que el script de `Pyomo`. Más información en el Apéndice B.

Importante — Pyomo solver no estará disponible para su uso en exámenes. El solver de Pyomo no estará disponible durante los exámenes. Durante los exámenes se empleará como optimizador única y exclusivamente los módulos de Matlab descritos en el curso.

■ **Ejemplo 1.6 — Resolviendo el problema de asignación de producción con el solucionador para Windows.** Recordemos el Ejemplo 1.1. La GUI de este solucionador solicita los elementos básicos del problema. Introduciendo cada uno en el área correspondiente, como se muestra en la Figura 1.5, obtenemos el resultado deseado.

■

1.4 Análisis post-óptimo: Precios sombra

El análisis post-óptimo es un análisis *what-if* que realizamos después de encontrar la solución óptima. Este análisis responde a preguntas sobre cuán sólida y/o robusta es la solución actual, y qué acciones debemos llevar a cabo para mejorarla. También es una herramienta para estimar el impacto de incluir nuevas variables o mejorar/limitar recursos en el modelo.

Para realizar el análisis post-óptimo, introducimos cambios en los parámetros del modelo uno por uno, manteniendo el resto constante. Este análisis nos permite estimar con alta precisión cómo

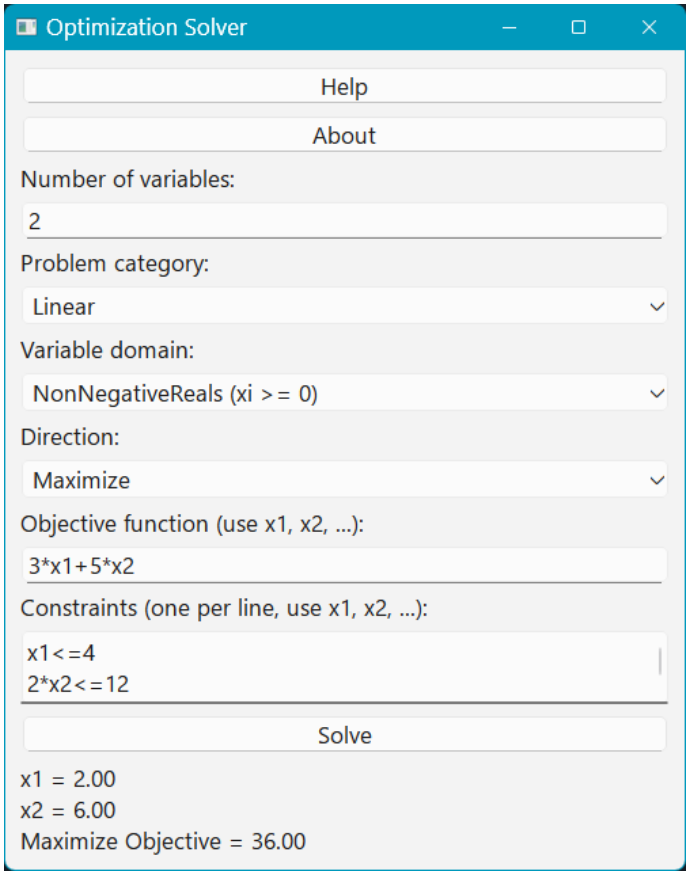


Figura 1.5: Resolución del Ejemplo 1.1 con el solucionador para Windows

afectan los cambios a la solución actual sin necesidad de resolver un nuevo modelo.

En este curso nos centraremos en la obtención y significado práctico de los precios sombra.

1.4.1 Precios sombra

Consideremos un problema como (1.1), con solución óptima $\mathbf{x}^*$, valor óptimo z^* y coeficiente del lado derecho (RHS) $\mathbf{b}$, al que denominamos LP. Consideremos un problema asociado $\hat{\text{LP}}_i$, con solución óptima $\hat{\mathbf{x}}^*$, valor óptimo $\hat{z}^*$ y vector RHS $\hat{\mathbf{b}}_i = (b_1, \ldots, b_i + 1, \ldots, b_n)$.

Definición 1.4.1 — Precio Sombra. El **precio sombra** s_i correspondiente a la i-ésima restricción cuantifica la mejora marginal en la función objetivo z^* resultante de un incremento unitario en b_i. Sea $\hat{z}^*$ la solución del problema donde $\hat{b}_i = b_i + 1$, entonces:

$$\hat{z}^* = z^* + s_i$$

Si $s_i^* = 0$, entonces los cambios en b_i no afectan el valor óptimo de la función objetivo.

R

- Los cambios en b_i se denotan como Δb_i, es decir, $\hat{b}_i - b_i = \Delta b_i$.
- El precio sombra s_i asociado al coeficiente b_i, de la restricción $i = 1, \ldots, m$ contiene la información acerca de cuánto cambia la función objetivo si el coeficiente b_i se incrementa en una unidad.
- Si $s_i = 0$, esto significa que aumentar el recurso i no mejora la solución, y por lo tanto el recurso no es crítico.

- Si $s_i \neq 0$, se tiene que el recurso i es crítico y que aumentarlo en una unidad cambia el valor óptimo del problema.

■ **Ejemplo 1.7 — Precios sombra de Ejemplo 1.1.** Para obtener los precios sombra se extrae la información dual que brinda `linprog` almacenda en la estructura `lambda`. Ver Código 1.2. ■

```
f = -[3 5]; % vector de costes
b =  [4 12 18]; %vector RHS de recursos
lb = zeros(1,2); % cota inferior

%matriz de restricciones
     %x1,x2
A = [ 1 0;
      0 2;
      3 2];

%optimizar
[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb)

%imprimir resultados
x
-fval
lambda.ineqlin
```

Código 1.2: Extracción de precios sombra

La información de `lambda` se extrae de la estructura. Como en este caso contamos con desigualdades de recursos (`b`), se emplea `lambda.ineqlin` para acceder a los precios sombra.

De acuerdo a la información obtenida, se tiene que

$$\mathbf{s} = (0, 1.5, 1)$$

Esto significa que:

- $s_1 = 0$, se refiere a la desigualdad $x_1 \leq 4$. Si se aumenta la disponibilidad de horas semanales en la planta 1, las ganancias no mejoran. Luego concluimos que el tiempo en disponible en la planta 1 **no** es un recurso crítico.
- $s_2 = 1.5$, se refiere a la desigualdad $x_2 \leq 12$. Si se aumenta la disponibilidad de horas semanales en la planta 2, de 12 a 13, las ganancias aumentarán en 1.5 unidades, de 36 a 37.5. Por lo tanto, las horas en la planta 2 son un recurso crítico.
- $s_3 = 1$, se refiere a la desigualdad $3x_1 + 2x_2 \leq 18$. Si se aumenta la disponibilidad de horas semanales en la planta 3, de 18 a 19, las ganancias aumentarán en 1 unidad, de 36 a 37. Por lo tanto, las horas en la planta 3 son un recurso crítico.

Cuando queramos calcular de forma gráfica los precios sombra, para problemas con dos variables de decisión, relajamos en una unidad cada una de las restricciones y vemos qué ocurre. Si no afecta en nada aumentar el recurso en una restricción, entonces son bienes libres (tenemos recursos disponibles en exceso y el precio sombra es cero). Si mejora Z al relajar en una unidad la restricción, entonces los recursos son limitados y vamos a tener precios sombra positivos.

2. Programación Entera

2.1 Modelo de Programación Lineal Entera. Modelos Puros y Mixtos

La **Programación Lineal Entera (ILP)** es una extensión fundamental de la Programación Lineal (LP) en la que algunas o todas las variables de decisión deben tomar valores enteros. Este tipo de modelado es esencial cuando se abordan problemas que implican decisiones discretas, como planificación de horarios, ubicación de instalaciones o diseño de redes. Particularmente en aplicaciones de telecomunicaciones, esto incluye determinar el número de enlaces, asignación de canales o planificación de nodos en una red.

Definición 2.1.1 — Problema de Programación Lineal Entera. La forma general de un problema de ILP puede escribirse como:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.a.} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{Z}^n \text{ (si todas las variables son enteras).} \end{aligned} \tag{2.1}$$

Donde:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ es el vector de variables de decisión.
- $\mathbf{c}^T\mathbf{x}$ es la función objetivo lineal.
- $A\mathbf{x} \leq \mathbf{b}$ representa el sistema de restricciones lineales.
- La condición de integralidad exige que algunas o todas las variables pertenezcan al conjunto de enteros.

R Nótese que $\mathbb{Z}_+ = \mathbb{Z} \cap \{x > 0\}$, es decir, x es un entero estrictamente positivo, excluyendo el cero. Sin embargo, para los fines de este curso, dado que las regiones factibles deben ser cerradas, cuando nos referimos a $\mathbb{Z}_+$, incluiremos el cero, es decir, redefinimos $\mathbb{Z}_+ = \{x \geq 0\} \cap \mathbb{Z}$.

En un modelo **entero puro**, *todas las variables de decisión* deben tomar valores enteros, mientras que en un modelo **mixto** (**MILP**), solo un subconjunto de las variables son enteras y las

demás permanecen continuas.

Los modelos ILP y MILP son más complejos de resolver que los modelos LP, ya que las restricciones de integralidad hacen que la región factible deje de ser convexa. Para resolver estos problemas de manera eficiente se emplean algoritmos especializados como *branch-and-bound*, *branch-and-cut* y *cutting planes*.

En la práctica, ¿cómo resolver un problema ILP? Primero, veamos qué ocurre si tratamos este problema como un problema LP, con el siguiente ejemplo:

■ **Ejemplo 2.1** Consideremos el siguiente problema de optimización:

$$\begin{aligned} \max \quad & z = 5x_1 + 3x_2 \\ \text{s.a.} \quad & -1.4x_1 + x_2 \leq 2, \\ & 3x_1 + x_2 \leq 12, \\ & x_1, x_2 \geq 0, \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$

Para resolver este problema, relajamos la condición que obliga a las variables a ser enteras. Trabajamos con el problema **relajado**:

$$\begin{aligned} \max \quad & z = 5x_1 + 3x_2 \\ \text{s.a.} \quad & -1.4x_1 + x_2 \leq 2, \\ & 3x_1 + x_2 \leq 12, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Aplicando el método analítico y encontrando los vértices de la región factible, obtenemos la siguiente tabla:

x	z
$(0,0)$	0
$(0,2)$	6
$(4,0)$	20
$(2.25,5.18)$	26.91

Concluimos que la solución óptima del problema relajado es $\mathbf{x}^* = (2.25, 5.18)$ para $\mathbf{z}^* = 26.91$. También podemos aplicar el método gráfico y obtener el mismo resultado, ver Figura 2.1 izquierda.

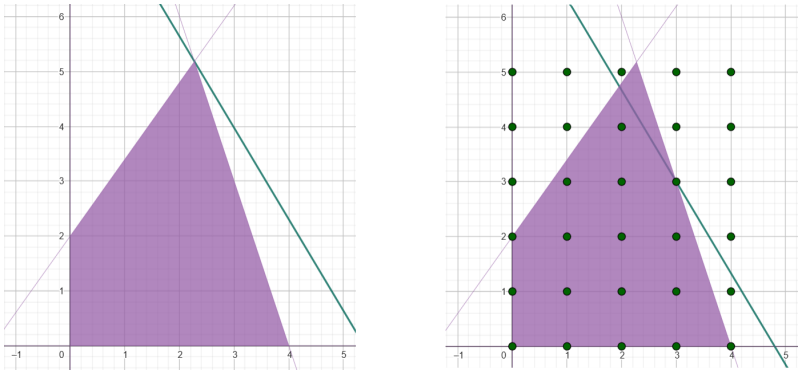


Figura 2.1: Problema relajado vs. problema entero

Sin embargo, las soluciones deseadas deben ser enteras, por lo que aplicar los métodos anteriores ya no es válido. Volviendo al problema original, debemos considerar únicamente los puntos dentro

de la región factible con coordenadas enteras, ver Figura 2.1 derecha. Ahora, aumentamos el valor de las curvas de nivel de la función objetivo hasta alcanzar el último punto factible con coordenadas enteras. De esta manera, concluimos que la solución óptima del problema entero es $\mathbf{x}^* = (3,3)$ con $z^* = 24$. ∎

Definición 2.1.2 — Problema Relajado. Consideremos el problema entero (ILP) (2.1). Llamamos **Problema Relajado (RLP)** al problema asociado:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.a.} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0. \end{aligned} \tag{2.2}$$

En otras palabras, relajamos la condición que restringe las variables a ser enteras.

Proposición 2.1.1 — ILP versus RLP.

Relación entre soluciones:

- En general, la solución del ILP y la del RLP no son iguales.
- No es correcto calcular la solución del ILP como una aproximación al punto entero más cercano de la solución del RLP.
- El único caso en el que ambas soluciones coinciden es cuando la matriz de restricciones es unimodular.

Relación entre regiones factibles:

- La región factible del ILP siempre es más pequeña que la del RLP, es decir, contiene menos soluciones.
- La región factible del ILP es un subconjunto de la región factible del RLP; todas las soluciones del ILP son también soluciones del RLP.
- La solución del ILP no está obligada a encontrarse en la frontera de la región factible (no existe una frontera continua).

Relación entre soluciones óptimas:

- Dado que la región factible del ILP es un subconjunto de la del RLP, la solución del RLP, z^*_{LP}, siempre es mejor que la solución del ILP, z^*_{IP}, proporcionando así una cota:
- Minimización: $z^*_{IP} \geq z^*_{LP}$,
- Maximización: $z^*_{IP} \leq z^*_{LP}$.

Para algunos problemas puede ser útil considerar solo un subconjunto de variables como enteras y permitir que el resto sean continuas. A estos problemas los denominamos **Problemas de Programación Entera Mixta**.

Definición 2.1.3 — Problema de Programación Entera Mixta. Decimos que un problema LP es un problema de programación entera mixta si un subconjunto de las variables está restringido a ser entero, es decir:

$$\begin{aligned} \min \quad & \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{c}_2^T\mathbf{x}_2 \\ \text{s.a.} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x}_2 \in \mathbb{Z}. \end{aligned} \tag{2.3}$$

En otras palabras, relajamos la condición que restringe todas las variables a ser enteras.

¿Cómo resolver un problema entero? Para resolver un problema entero en este curso, utilizamos software:

- **Matlab `intlinprog`:** de manera análoga a la solución de problemas lineales de variable continua, Matlab implementa la función `intlinprog` para la resolución de ILP y MILP. Más información acerca de estas funciones puede ser consultada en Mathworks o escribiendo

`help intlinprog` en la consola de Matlab. Para más información consultar Apéndice G.

Tabla 2.1: Entradas y salidas de la función `intlinprog` en MATLAB

Entrada / Salida	Descripción	Tipo / Valores
Entradas		
`f`	Vector de coeficientes de la función objetivo	Vector columna
`intcon`	Índices de las variables que deben ser enteras	Vector de enteros
`A`	Matriz de coeficientes de desigualdades lineales	Matriz m×n (opcional)
`b`	Vector de términos independientes de desigualdades	Vector columna (opcional)
`Aeq`	Matriz de coeficientes de ecuaciones lineales	Matriz p×n (opcional)
`beq`	Vector de términos independientes de ecuaciones lineales	Vector columna (opcional)
`lb`	Vector de límites inferiores de las variables	Vector columna (opcional)
`ub`	Vector de límites superiores de las variables	Vector columna (opcional)
`x0`	Vector inicial de la solución	Vector columna (opcional)
Salidas		
`x`	Vector solución óptima que satisface las restricciones	Vector columna
`fval`	Valor de la función objetivo en la solución óptima	Escalar
`exitflag`	Indica el resultado de la optimización	Escalar entero
`output`	Información detallada del solver (iteraciones, tiempo, etc.)	Estructura
`lambda`	Multiplicadores de Lagrange asociados a las restricciones lineales	Estructura (si aplica)

■ **Ejemplo 2.2** Para resolver el ejemplo anterior empleamos el código 2.1.

```
%intlinprog example
f = [-5; -3];                % Cambiado de signo (max -> min)
A = [-1.4  1;
      3    1];
b = [2;
     12];
intcon = [1 2];             % Ambas variables son enteras
lb = [0; 0];                % x1, x2 >= 0
ub = [];                    % Sin cotas superiores explicitas
[x_opt, fval] = intlinprog(f, intcon, A, b, [], [], lb, ub);
z_opt = -fval              % Volvemos a cambiar el signo
x_opt
```

Código 2.1: Código para Ejemplo 2.1

■

- **Pyomo Solver:** podemos especificar el dominio de las variables eligiendo la opción `NonNegativeIntegers`/`NonPositiveIntegers`/`Integers` en el menú contextual "Domain", más detalles en el Apéndice B. Al establecer esta opción, todas las variables se consideran enteras, por lo que no es posible resolver MILP.

2.2 Modelado de Problemas con Variables Binarias

En ocasiones necesitamos modelar una decisión de tipo sí/no; en tales casos utilizamos variables binarias.

Definición 2.2.1 — Variable Binaria. Se denomina variable binaria o variable 0-1 a una variable entera cuyos únicos valores posibles son 0 o 1, es decir:

$$x_j = \begin{cases} 1, & \text{si la condición } j \text{ se cumple,} \\ 0, & \text{en caso contrario.} \end{cases}$$

■ **Ejemplo 2.3 — Uso de variables binarias.** Una empresa tecnológica está considerando la activación de la producción de cuatro tipos de dispositivos de telecomunicaciones: A, B, C y D, para el próximo año. La producción de los dispositivos A, B, C y D implica:

- un coste fijo asociado a la puesta en marcha de la línea de producción (en EUR), y
- un requerimiento de horas de trabajo por unidad producida.

La Tabla 2.2 resume el coste fijo, la mano de obra requerida por unidad y el beneficio esperado por unidad:

Tabla 2.2: Coste fijo de la mano de obra requerida por unidad y beneficio esperado por la unidad

Dispositivo	Coste Fijo (EUR)	Mano de obra por unidad (horas)	Beneficio por unidad (EUR)
A	14 500	10	50
B	10 000	15	60
C	8 000	5	55
D	9 000	14	80

La empresa dispone de 20 000 horas de mano de obra en esta temporada y se requiere que:

I. si se activa la producción de B, no se puede activar la producción de D,
II. si se activa la producción de A, entonces al menos C o D también deben activarse,
III. al menos una línea de producción debe ser activada,
IV. se debe producir C o D,

Para cada tipo de producción, si se activa, la empresa se compromete a producir un mínimo de 1000, 1500, 3000 y 2000 unidades de los dispositivos A, B, C y D, respectivamente. Formule el modelo de programación matemática que permita a la empresa planificar su producción anual, con el objetivo de **maximizar** su beneficio total.

En este problema, debemos tomar dos tipos de decisiones: decidir si se activa una línea de producción y el nivel de producción de cada tipo de producto. La primera decisión es binaria (sí/no), y la segunda representa simplemente cuántas unidades de cada producto se producen. Por lo tanto, podemos utilizar como variables de decisión:

- x_j: unidades del producto j, donde $j \in I = A,B,C,D$, y $x_j \in \mathbb{Z}_+$ para todo $j \in I$; y
- $y_i = \begin{cases} 1, & \text{si la línea de producción } i \text{ está activada,} \\ 0, & \text{en caso contrario} \end{cases}$ para $i \in I$.

Obsérvese que usamos I como nuestro conjunto índice y declaramos la activación de las líneas de producción como una variable binaria. ■

2.2.1 Condicionales y pre-requisitos

Proposición 2.2.1 Condicionales y Prerrequisitos

1. j es condicional respecto a i. Para completar j, i debe completarse:

$$x_j - x_i \leq 0 \quad (x_j = 1 \Rightarrow x_i = 1)$$

2. i es un pre-requisito para j:

$$x_i - x_j = 0 \quad (x_i \Leftrightarrow x_j)$$

3. i y j son mutuamente excluyentes (no pueden ser ambos iguales a 1 simultáneamente):

$$x_i + x_j \leq 1$$

■ **Ejemplo 2.4**

1. Para llevar a cabo el proyecto 5, el proyecto 2 debe completarse primero: $x_5 \leq x_2$.
2. Los materiales se compran (actividad i) solo si la tarea correspondiente es seleccionada: $x_j - x_i = 0$.
3. Los ítems 4 y 2 no pueden seleccionarse al mismo tiempo: $x_4 + x_2 \leq 1$.

■

Esto significa que, volviendo al Ejemplo 2.3, la condición "si se activa la producción de B, no se puede activar la producción de D" se escribe como

$$y_B + y_D \leq 1$$

porque ambos no pueden ser iguales a 1 simultáneamente. Esto conduce a un conjunto más general de selección de k actividades de entre n.

Proposición 2.2.2 — Número de Actividades a Completar.

1. Completar como máximo k actividades de entre n:

$$\sum_{i \in I} x_i \leq k$$

2. Completar al menos k actividades de entre n:

$$\sum_{i \in I} x_i \geq k$$

3. Completar exactamente k actividades de entre n:

$$\sum_{i \in I} x_i = k$$

■ **Ejemplo 2.5**

1. De 5 miembros, seleccionar como máximo 2: $x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$.
2. De 5 ítems, tomar al menos 3: $x_1 + x_2 + x_3 + x_4 + x_5 \geq 3$.
3. De 5 proyectos, completar exactamente 2: $x_1 + x_2 + x_3 + x_4 + x_5 = 2$.

■

Esto significa que, volviendo al Ejemplo 2.3, la condición "al menos una línea de producción debe ser activada" se escribe como

$$y_A + y_B + y_C + y_D \geq 1$$

porque al menos una de ellas debe ser igual a 1.

2.2.2 Implicaciones lógicas

Proposición 2.2.3 — Inclusión de Restricciones OR. El modelo debe incluir una de dos restricciones, $g_1(x) \leq b_1$ o $g_2(x) \leq b_2$. Se introduce una nueva variable binaria δ para representar la no activación de una restricción, y las restricciones se incluyen como

$$g_1(x) \leq b_1 + M_1 \delta$$
$$g_2(x) \leq b_2 + M_2(1 - \delta)$$

Las constantes M_1, M_2 deben elegirse lo suficientemente grandes para trivializar la restricción no activada (típicamente una constante positiva muy grande).

■ **Ejemplo 2.6** Suponga que para viajar de A a B, se debe tardar menos de 10 minutos o pagar menos de 2 € por transporte. Sea $g_1(x)$ el tiempo de viaje y $g_2(x)$ el coste de viaje. Entonces el modelo incluiría:

$$g_1(x) \leq 10 + M_1\delta$$
$$g_2(x) \leq 2 + M_2(1-\delta)$$

con la elección adecuada de M. Nótese que si $\delta = 0$, la restricción de tiempo de viaje está activa; si $\delta = 1$, la restricción de coste está activa. ■

Esto es particularmente útil cuando se enfrenta al tipo de restricción $P \Rightarrow Q$, donde P y Q son proposiciones. Volviendo al Ejemplo 2.3, la condición «si se activa la producción de A, entonces al menos C o D también deben activarse» significa P: producción de A activada, $y_A = 1$ implica Q: al menos C o D activadas, $y_C = 1$ o $y_D = 1$.

Para modelar este tipo de implicaciones, las reducimos a implicaciones OR usando la siguiente equivalencia lógica

$$(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q) \tag{2.4}$$

donde $\neg$ significa la negación de P y $\vee$ significa OR (al menos una de las condiciones se cumple). Una vez obtenida la proposición OR, procedemos a usar el mecanismo OR como antes.

¿Cómo aplicar esto a variables binarias?

■ **Ejemplo 2.7 — Implicación lógica.** La condición "si se activa la producción de A, entonces al menos C o D también deben activarse" significa P: producción de A activada, $y_A = 1$ implica Q: al menos C o D activadas, $y_C = 1$ o $y_D = 1$. Esto es $y_A = 1 \Rightarrow y_C = 1$ o $y_D = 1$.

Como trabajamos con variables binarias, la condición $y_A = 1$ es equivalente a $y_A \geq 1$.

Por otro lado, la condición $y_C = 1$ o $y_D = 1$ es equivalente a la restricción $y_C + y_D \geq 1$. Así, el esquema $P \Rightarrow Q$ se convierte en $y_A \geq 1 \Rightarrow y_C + y_D \geq 1$.

La negación de P sería $\neg P = (y_A \leq 0)$, por lo que según la ecuación (2.4) obtenemos la relación OR

$$y_A \leq 0 \text{ o } y_C + y_D \geq 1$$

Use la Proposición 2.2.3 para concluir que la relación OR anterior puede incluirse en nuestro modelo como dos restricciones relacionadas mediante una nueva variable binaria

$$\delta = \begin{cases} 1, & \text{si la primera restricción está activada} \\ 0, & \text{si la segunda restricción está activada} \end{cases}$$

como

$$y_A \leq M_1\delta$$
$$y_C + y_D \geq 1 + M_2(1-\delta)$$

eligiendo el valor adecuado de M_1 y M_2. Nótese que si $\delta = 0$, activamos la primera restricción, lo que significa que la segunda debe ser trivial. Trivializar restricciones que involucran variables binarias es simple: basta elegir una condición que siempre se cumpla. Por ejemplo, $y_C + y_D \geq 0$, o cualquier número negativo, así que sea $1 + M_2 \leq 0$, lo que implica que $M_2 \leq -1$, por lo tanto cualquier valor menor que -1 es adecuado para M_2.

Repita el mismo razonamiento para M_1. Sea $\delta = 1$, entonces la segunda restricción está activada y la primera debe ser trivial. Para ello, elija $M_1 \geq 1$. Así, las restricciones resultan

$$y_A \leq \delta$$
$$y_C + y_D \geq \delta$$

■

R Del último ejemplo surge una pregunta natural: ¿Se pueden combinar las dos restricciones anteriores en una sola? En este caso la respuesta es sí. Sin embargo, en general no se recomienda. Considere el contraejemplo "si $y_A = 1$ y $y_B = 1$ implica $y_C = 1$". Usando el mecanismo anterior, obtenemos las restricciones

$$y_A + y_B \leq 1 + \delta$$
$$y_C \geq \delta$$

Si unimos estas dos restricciones obtenemos $y_A + y_B \leq 1 + y_C$, lo que permite el caso $y_A = 1$, $y_B = 0$ y $y_C = 1$, que no es factible.

2.2.3 Incluir K de entre N restricciones

Proposición 2.2.4 — Inclusión de K de N restricciones. El modelo debe incluir K de N restricciones del tipo $g_j(x) \leq b_j$, para $j = 1, \ldots, N$. Se introduce una nueva variable binaria δ_j para representar la no activación de cada restricción, y las restricciones se incluyen como

$$g_1(x) \leq b_1 + M_1\delta_1$$
$$g_2(x) \leq b_2 + M_2\delta_2$$
$$\vdots$$
$$g_N(x) \leq b_N + M_N\delta_N$$

con

$$\sum_{j=1}^{N} \delta_j = N - K$$

Las constantes M_j deben elegirse adecuadamente de forma que trivialicen las restricciones no activadas.

■ **Ejemplo 2.8** Para los nuevos laboratorios biomédicos que la empresa abrirá en Madrid, al menos dos de tres protocolos medioambientales deben respetarse: emisiones $g_1(x) \leq 10$, consumo energético $g_2(x) \leq 20$, o residuos $g_3(x) \leq 30$. En el modelo, estos se incluirían como

$$g_1(x) \leq 10 + M_1\delta_1$$
$$g_2(x) \leq 20 + M_2\delta_2$$
$$g_3(x) \leq 30 + M_3\delta_3$$

con

$$\sum_{j=1}^{3} \delta_j = 3 - 2 = 1$$

■

2.2.4 Acotación

En el caso de un problema que involucra variables binarias y enteras, en el cual el valor de la variable entera depende de la activación de algunas variables binarias, podemos utilizar la técnica de acotación. Por ejemplo, el nivel de producción de un bien x depende de realizar un pedido de material y, siendo la variable binaria y la que indica si el envío se solicita o no. Esta restricción se introduce en el modelo como

$$x \leq My$$

donde M es una constante que excede los límites de los valores de x.

Retomando el ejemplo 2.3, la condición «Para cada tipo de producción, si está activada, la empresa se compromete a producir un mínimo de 1000, 1500, 3000 y 2000 unidades de los dispositivos A, B, C y D, respectivamente», se escribe como un conjunto de cuatro restricciones:

$$M_1 y_A \leq x_A \leq M y_1$$
$$M_2 y_B \leq x_B \leq M y_2$$
$$M_3 y_C \leq x_C \leq M y_3$$
$$M_4 y_D \leq x_D \leq M y_4$$

donde los valores de las constantes son 1000, 1500, 3000 y 2000, respectivamente.

El valor de M y su inclusión obligan a que las variables x_i sean cero si y_i es cero, lo cual constituye el caso útil. La constante M se elige lo suficientemente grande para evitar una restricción artificial de acotación superior que no está en el modelo.

2.2.5 Costes fijos

Proposición 2.2.5 — Funciones con N valores posibles. Sea f una función que puede tomar los valores $d_1, d_2, \ldots, d_N$. Entonces f puede expresarse como

$$f(x) = \sum_{i=1}^{N} d_i y_i$$
$$\sum_{i=1}^{N} y_i = 1$$

donde y_i son variables binarias que indican qué valor asume f.

■ **Ejemplo 2.9** Las horas de operación de una planta farmacéutica están dadas por $g(x)$. La actividad de la planta no debe exceder las 18 horas, pero solo puede operar en bloques de 6 horas. Por lo tanto, $g(x)$ puede tomar los valores 6, 12 o 18. Esto puede representarse como

$$g(x) = 6y_1 + 12y_2 + 18y_3$$
$$y_1 + y_2 + y_3 = 1$$

■

Esto significa, retomando el Ejemplo 2.3, que la función objetivo se escribe como

$$\max_{\mathbf{x}} z = 50x_A + 60x_B + 55x_C + 80x_D - (14500y_A + 10000y_B + 8000y_C + 9000y_D)$$

Por lo tanto, el modelo del Ejemplo 2.3 queda

$$\begin{aligned}
\max_{\mathbf{x}} \quad & z = 50x_A + 60x_B + 55x_C + 80x_D - (14500y_A + 10000y_B + 8000y_C + 9000y_D) \\
s.t \quad & 10x_A + 15x_B + 5x_C + 14x_D \leq 20000 \\
& y_B + y_D \leq 1 \\
& y_A + y_B + y_C + y_D \geq 1 \\
& y_A \leq \delta \\
& y_C + y_D \geq \delta \\
& y_C + y_D \geq 1 \\
& 1000y_A \leq x_A \leq M y_A \\
& 1500y_B \leq x_B \leq M y_2 \\
& 3000y_C \leq x_C \leq M y_C \\
& 2000y_D \leq x_D \leq M y_D
\end{aligned}$$

para $x_j, \geq 0, j \in I$, enteras y $\delta, y_i \in \{0,1\}$, para $i \in I$. M suficientemente grande, por ejemplo $M = 10^6$.

La solución es

$$\mathbf{x}^* = (0,0,4000,0,0,0,0,1,0) \qquad z^* = 212000$$

Así, para maximizar el beneficio la empresa debe activar únicamente la línea C y producir 4000 unidades del producto C para obtener un beneficio de 212000 euros.

En el Código 2.2 se muestra el código empleado para la resolución, utilizando `intlinprog`.

```
% vector de costes
% xA    xB  xC  xD  yA  yB  yC  yD  delta
f = -[50 60 55 80 -14500 -10000 -8000 -9000 0];
%M grande
M = 90000;

%vector de recursos
b =  [20000 1 -1 0 0 -1 zeros(1,8)];

%cotas
ub = [M M M M 1 1 1 1 1];
lb = zeros(1,9);
intcon = 1:9;

%matriz de restricciones
     % xA    xB  xC  xD  yA  yB  yC  yD  delta
A = [ 10 15 5 14 0 0 0 0 0;
       0 0 0 0 0 1 0 1 0;
       0 0 0 0 -1 -1 -1 -1 0;
       0 0 0 0 1 0 0 0 -1;
       0 0 0 0 0 0 -1 -1 1;
       0 0 0 0 0 0 -1 -1 0;
       1 0 0 0 1000 0 0 0 0;
       0 -1 0 0 0 1500 0 0 0;
       0 0 -1 0 0 0 3000 0 0;
       0 0 0 -1 0 0 0 2000 0;
       1 0 0 0 -M 0 0 0 0;
       0 1 0 0 0 -M 0 0 0;
       0 0 1 0 0 0 -M 0 0;
       0 0 0 1 0 0 0 -M 0;
       ];

%igualdades
Aeq = []; beq= [];

%optimizar
[x,fval] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub);
x
-fval
```

Código 2.2: Código para Ejemplo 2.3

Para incluir variables binarias, en la definición de los parámetros `lb, ub`, se incluyen como $0,1$, respectivamente.

Notemos que, para Matlab, el modelo debe ser escrito en forma canónica, esto es, dirección de mínimo y desigualdades en forma de menor o igual. Luego, las restricciones de tipo

$$1500y_B \leq x_B \leq My_2$$

se codifican en el modelo como dos restricciones

$$-x_B + 1500y_B \leq 0$$
$$x_B - My_2 \leq 0$$

Nótese, además, que se debe adicionar una nueva variable δ, binaria, como una entrada extra a la función objetivo y con una columna en la matriz de restricciones.

Se define la constante $M = 9000$, para emplearla como cota superior, tanto del problema como en las restricciones de acotación de producción.

III Programación en Redes

3 Problemas de transporte y asignación 53

3.1 Problema de Transporte

3.2 Problema de Asignación

4 Modelos de optimización de redes 61

4.1 Teoría de grafos

4.2 El problema del camino más corto

4.3 Árbol de Expansión Mínima

4.4 Problema del flujo máximo

4.5 Problema de flujo de costo mínimo

5 Administración de proyectos con PERT/CPM 77

5.1 Introducción a PERT y CPM

5.2 Ejemplo prototipo: proyecto de Reliable Construction Co

5.3 Uso de una red para visualizar un proyecto

5.4 Programación de un proyecto con PERT/CPM

5.5 Manejo de la incertidumbre en las duraciones de las actividades

5.6 Consideración del trueque entre tiempo y costo. Crashing.

5.7 Programación y control de los costos del proyecto. Evaluación de PERT/CPM

3. Problemas de transporte y asignación

Para ampliar el contenido de este capítulo, consultar [2, Cap. 8].

3.1 Problema de Transporte

El *problema de transporte* es un modelo clásico en investigación operativa que aborda la distribución de un producto desde varias fuentes (o nodos de oferta) hacia varios destinos (o nodos de demanda) de manera que se minimice el coste total de transporte. Fue introducido por primera vez en la década de 1940 y sigue siendo una de las aplicaciones más estudiadas de la programación lineal debido a su elegancia teórica y relevancia práctica.

Consideramos la siguiente situación:

- Existen m puntos de oferta (fábricas, almacenes, plantas de producción), donde cada punto de oferta i tiene una capacidad total disponible $a_i \geq 0$.
- Existen n puntos de demanda (mercados, hospitales, zonas de clientes), donde cada punto de demanda j requiere una demanda total $b_j \geq 0$.
- El coste unitario de transporte desde el punto de oferta i al punto de demanda j es c_{ij}, donde $I = \{1,\dots,m\}$ y $J = \{1,\dots,n\}$.

Esta situación se representa como un grafo en la Figura 3.1.

En este problema, la decisión consiste en cuántas unidades enviar desde el origen i al destino j, por lo que definimos las variables de decisión:

$$x_{ij} = \text{número de unidades enviadas desde el punto de oferta } i \text{ al punto de demanda } j.$$

donde $I = \{1,\dots,m\}$ y $J = \{1,\dots,n\}$.

El objetivo es minimizar el coste total de transporte:

$$\min_{\mathbf{x}} z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

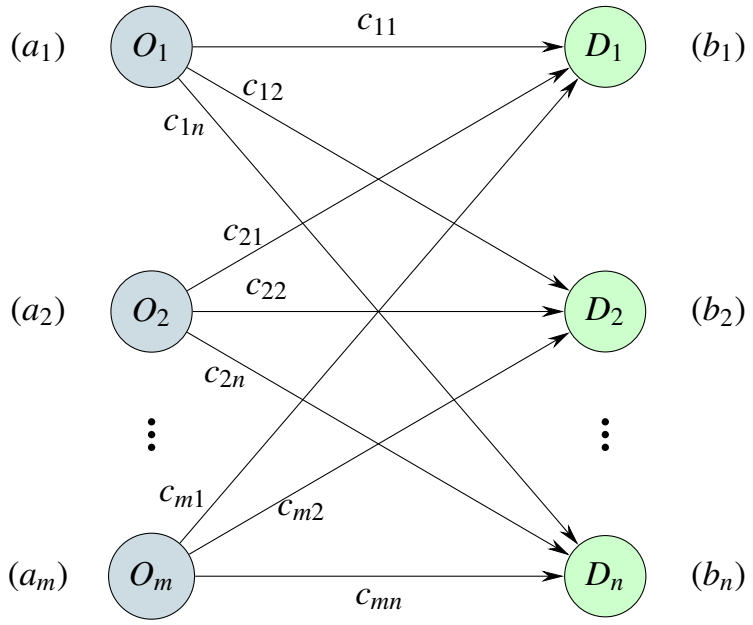


Figura 3.1: Representación como grafo del problema de transporte

sujeto a las restricciones (una para cada punto de oferta y destino):

$$\sum_{j=1}^{n} x_{ij} \leq a_i, \qquad i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} \geq b_j, \qquad j = 1, \ldots, n,$$

$$x_{ij} \geq 0, \qquad \forall i, j \in I \times J.$$

Estas restricciones garantizan que los envíos desde cada nodo de oferta no excedan su disponibilidad y que la demanda en cada destino sea satisfecha. En el caso especial en que $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, el problema se denomina *problema de transporte balanceado*, y la solución está garantizada. En caso contrario, se pueden introducir nodos ficticios de oferta o demanda para equilibrar el sistema.

■ **Ejemplo 3.1** Una empresa necesita enviar materia prima desde sus 2 almacenes a 3 fábricas. El inventario en unidades de materia prima de cada almacén, las demandas en unidades de materia prima de cada fábrica y los costes unitarios de transporte (euros por unidad de materia prima enviada entre cada origen y destino) se muestran en la Tabla 3.1. Formule y resuelva el modelo de optimización que minimiza los costes de transporte.

	Fábrica 1	Fábrica 2	Fábrica 3	Oferta
Almacén 1	4	6	8	20
Almacén 2	5	4	3	30
Demanda	10	25	15	

Tabla 3.1: Matriz de costes de transporte con ofertas y demandas.

El problema de transporte también puede visualizarse como un *grafo bipartito*, donde los nodos de la izquierda representan los almacenes (ofertas), los nodos de la derecha representan las fábricas (demandas) y las aristas representan los posibles envíos con sus costes asociados. La situación del problema se representa en el grafo de la Figura 3.2

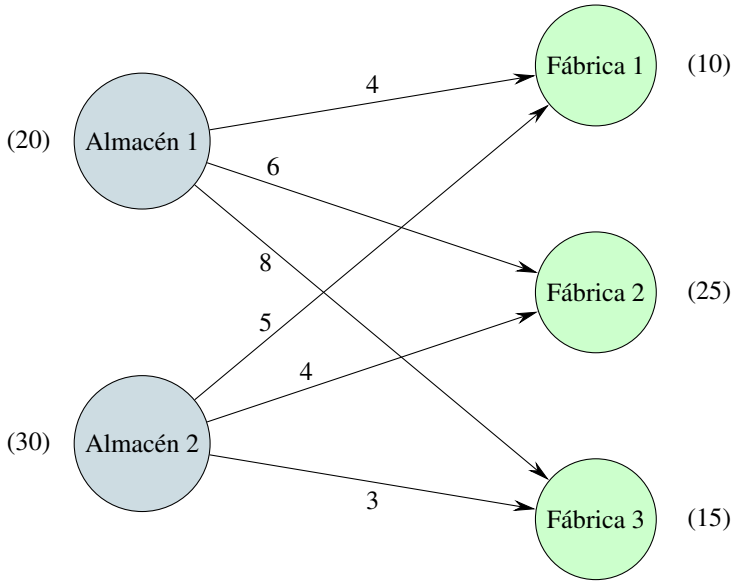


Figura 3.2: Grafo del Ejemplo 3.1

Dado que tenemos 2 almacenes y 3 fábricas, se tiene que $m = 2$ y $n = 3$. Así, $I = \{1,2\}$ y $J = \{1,2,3\}$, por lo tanto las variables de decisión son:

x_{ij} : unidades enviadas desde el almacén i a la fábrica j para $(i, j) \in I \times J, x_{ij} \geq 0$

Minimizar el coste de transporte implica minimizar la función de coste $\sum_{(i,j)\in I\times J} c_{ij}x_{ij}$, es decir,

$$\min_{\mathbf{x}} z = 4x_{11} + 6x_{12} + 8x_{13} + 5x_{21} + 4x_{22} + 3x_{23}$$

Este problema implica dos tipos de restricciones: un conjunto de demandas y otro conjunto de ofertas. Las restricciones de demanda significan que todas las fábricas deben recibir al menos las unidades demandadas, esto es

$$\sum_{i=1}^{2} x_{ij} \geq b_j \qquad j = 1,2,3$$

De igual forma, las restricciones de oferta significan que cada almacén puede enviar como máximo su inventario, lo cual es

$$\sum_{j=1}^{3} x_{ij} \leq a_i \qquad i = 1,2$$

Por lo tanto, el modelo queda

$$\min_{\mathbf{x}} z = 4x_{11} + 6x_{12} + 8x_{13} + 5x_{21} + 4x_{22} + 3x_{23}$$

s.a. $x_{11} + x_{12} + x_{13} \leq 20$ (El almacén 1 puede suministrar como máximo 20 unidades)

$x_{21} + x_{22} + x_{23} \leq 30$ (El almacén 2 puede suministrar como máximo 30 unidades)

$x_{11} + x_{21} \geq 10$ (La fábrica 1 demanda al menos 10 unidades)

$x_{12} + x_{22} \geq 25$ (La fábrica 2 demanda al menos 25 unidades)

$x_{13} + x_{23} \geq 15$ (La fábrica 3 demanda al menos 15 unidades)

$x_{ij} \geq 0, \quad \forall (i, j) \in I \times J.$

Para obtener la solución podemos usar `intlinprog` (Matlab) o el solver de Windows.

$$x^* = (10,10,0,0,15,15) \qquad z^* = 205$$

Por lo tanto, para minimizar el coste de transporte, debemos enviar 10 unidades desde el almacén 1 a la fábrica 1, 10 unidades desde el almacén 1 a la fábrica 2, 15 unidades desde el almacén 2 a la fábrica 2 y 15 unidades desde el almacén 2 a la fábrica 3, para un coste mínimo de 205 euros. ■



El problema de transporte es un caso especial de programación lineal con una matriz de restricciones altamente estructurada. Esta estructura especial permite el desarrollo de algoritmos eficientes como la *Northwest Corner Rule*, el *Método de Coste Mínimo* y el *Stepping Stone Method* para encontrar soluciones óptimas o casi óptimas.

3.2 Problema de Asignación

El *problema de asignación* es un modelo fundamental en investigación operativa que aborda la asignación óptima de recursos a tareas. Su objetivo es asignar un conjunto de agentes (por ejemplo, trabajadores, máquinas o proveedores de servicios) a un conjunto de tareas (por ejemplo, trabajos, ubicaciones o responsabilidades) de manera que se minimice el coste total, el tiempo o la distancia de las asignaciones. Este problema es un caso particular de programación lineal y optimización combinatoria, y tiene aplicaciones ampliamente extendidas en planificación, logística y sistemas de toma de decisiones.

Consideramos el siguiente escenario:

- Existen m agentes, cada uno capaz de realizar exactamente una tarea.
- Existen n tareas, cada una requiere exactamente un agente.
- El coste de asignar el agente i a la tarea j se denota por c_{ij}, donde $I = \{1,\dots,m\}$ y $J = \{1,\dots,n\}$.

Si $m = n$, entonces el problema está balanceado y cada tarea tendrá asignado su respectivo agente. Si $m > n$, existe un exceso de agentes y habrá agentes sin asignar; por otro lado, si $n > m$, existe un exceso de tareas y habrá tareas sin asignar. Esta situación se representa típicamente como un *grafo bipartito completo*, véase la Figura 3.3, donde un conjunto de nodos corresponde a los agentes y el otro a las tareas, y cada arista representa una posible asignación con un coste asociado.

Definimos las variables de decisión:

$$x_{ij} = \begin{cases} 1, & \text{si el agente } i \text{ se asigna a la tarea } j, \\ 0, & \text{en caso contrario.} \end{cases}$$

donde $I = \{1,\dots,m\}$ y $J = \{1,\dots,n\}$.

El objetivo es minimizar el coste total de asignación:

$$\min_{\mathbf{x}} z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

sujeto a las restricciones:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \text{para todo } i = 1,\dots,m \quad \text{(cada agente se asigna a una tarea)}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \text{para todo } j = 1,\dots,n \quad \text{(cada tarea se asigna a un agente)}$$

$$x_{ij} \in \{0,1\}, \quad \text{para todo } i,j \in I \times J.$$

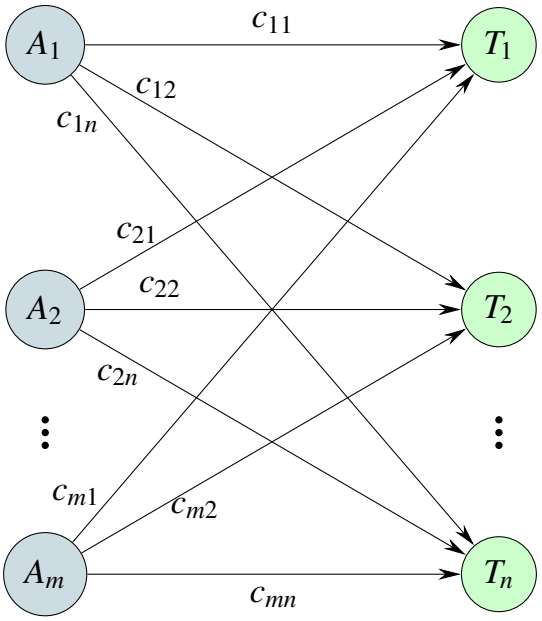


Figura 3.3: Representación empleando un grafo del problema de asignación

R El problema de asignación es un caso particular del problema de transporte donde todas las ofertas y demandas son iguales a uno. Su naturaleza binaria y estructura balanceada permiten algoritmos eficientes y lo convierten en un pilar de la optimización combinatoria.

■ **Ejemplo 3.2** Una empresa necesita asignar 4 empleados a 4 tareas diferentes. La matriz de costes (en horas requeridas para completar cada tarea) se muestra en la Tabla 3.2. Formule y resuelva el problema de asignación para minimizar el tiempo total empleado.

	Tarea 1	Tarea 2	Tarea 3	Tarea 4
Empleado 1	9	2	7	8
Empleado 2	6	4	3	7
Empleado 3	5	8	1	8
Empleado 4	7	6	9	4

Tabla 3.2: Matriz de costes para el problema de asignación (en horas).

Sea x_{ij} una variable binaria que indica si el empleado i se asigna a la tarea j. Definimos las variables de decisión:

$$x_{ij} = \begin{cases} 1, & \text{si el agente } i \text{ se asigna a la tarea } j, \\ 0, & \text{en caso contrario.} \end{cases}$$

donde $I = \{1,\ldots,m\}$ y $J = \{1,\ldots,n\}$.

La función objetivo es:

$$\begin{aligned} \min_{\mathbf{x}} z = & 9x_{11} + 2x_{12} + 7x_{13} + 8x_{14} + 6x_{21} + 4x_{22} + 3x_{23} + 7x_{24} \\ & + 5x_{31} + 8x_{32} + 1x_{33} + 8x_{34} + 7x_{41} + 6x_{42} + 9x_{43} + 4x_{44} \end{aligned}$$

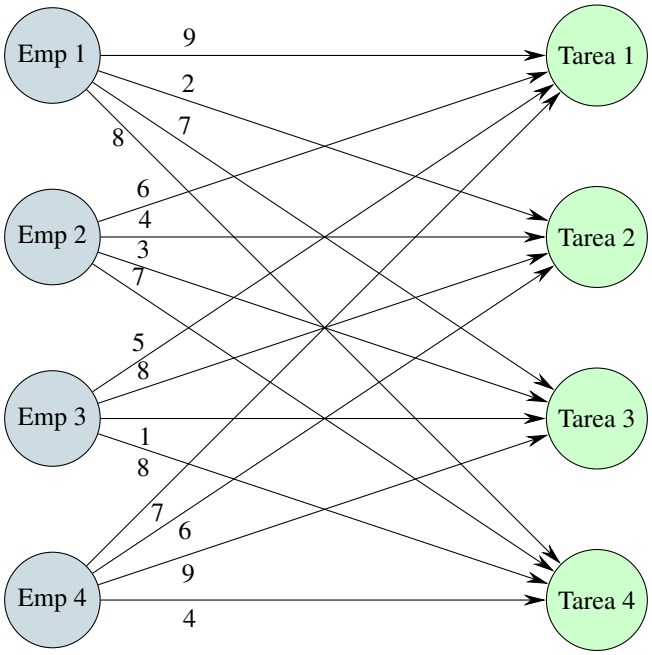


Figura 3.4: Representación empleando un grafo bipartito del problema de asignación

Sujeto a:

$$
\begin{aligned}
x_{11}+x_{12}+x_{13}+x_{14} &= 1 \quad \text{(Empleado 1 asignado a una tarea)} \\
x_{21}+x_{22}+x_{23}+x_{24} &= 1 \quad \text{(Empleado 2 asignado a una tarea)} \\
x_{31}+x_{32}+x_{33}+x_{34} &= 1 \quad \text{(Empleado 3 asignado a una tarea)} \\
x_{41}+x_{42}+x_{43}+x_{44} &= 1 \quad \text{(Empleado 4 asignado a una tarea)} \\
x_{11}+x_{21}+x_{31}+x_{41} &= 1 \quad \text{(Tarea 1 asignada a un empleado)} \\
x_{12}+x_{22}+x_{32}+x_{42} &= 1 \quad \text{(Tarea 2 asignada a un empleado)} \\
x_{13}+x_{23}+x_{33}+x_{43} &= 1 \quad \text{(Tarea 3 asignada a un empleado)} \\
x_{14}+x_{24}+x_{34}+x_{44} &= 1 \quad \text{(Tarea 4 asignada a un empleado)} \\
x_{ij} &\in \{0,1\} \quad \forall i,j \in I \times J.
\end{aligned}
$$

Para resovler el problema se emplea el Código 3.1.

```
% Number of employees and tasks
n = 4;
% Cost matrix (same as in Pyomo)
C = [ 9  2  7  8;
      6  4  3  7;
      5  8  1  8;
      7  6  9  4 ];

% Vectorize cost matrix (column-wise)
f = C(:);

% Number of variables
numVars = n^2;

% Integer variables (all binary)
intcon = 1:numVars;

```

```
% Bounds
lb = zeros(numVars,1);
ub = ones(numVars,1);

%Equality constraints

% Each employee is assigned exactly one task
Aeq1 = zeros(n, numVars);
for i = 1:n
    Aeq1(i, (i-1)*n+1:i*n) = 1;
end

% Each task is assigned exactly one employee
Aeq2 = zeros(n, numVars);
for j = 1:n
    Aeq2(j, j:n:end) = 1;
end

Aeq = [Aeq1; Aeq2];
beq = ones(2*n,1);

%Solve the problem
[x, fval] = intlinprog(f, intcon, [], [], Aeq, beq, lb, ub);

% Reshape solution
X = reshape(x, n, n);

% Output
disp('*** Solution ***')
for i = 1:n
    for j = 1:n
        fprintf('x_%d%d: %d\n', i, j, X(i,j));
    end
end
fprintf('Objective: %f\n', fval);
```

Código 3.1: Código para Ejemplo 3.2

$$x^* = (0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1), \quad z^* = 13$$

Así, la asignación óptima es:

- Empleado 1 $\rightarrow$ Tarea 2
- Empleado 2 $\rightarrow$ Tarea 1
- Empleado 3 $\rightarrow$ Tarea 3
- Empleado 4 $\rightarrow$ Tarea 4

Para un mínimo de 13 horas de trabajo. ■

4. Modelos de optimización de redes

La referencia principal para este capítulo es [2, Cap. 9].

4.1 Teoría de grafos

La Teoría de Grafos es una rama de la matemática discreta que estudia las relaciones entre objetos mediante estructuras denominadas *grafos*. Un grafo proporciona una forma natural de representar y analizar la conectividad, el flujo y las estructuras de red, lo que lo convierte en una herramienta fundamental en matemáticas, informática, ingeniería y numerosas ciencias aplicadas.

Definición 4.1.1 — Grafo. Un *grafo* $G = (V, E)$ consiste en un conjunto no vacío de vértices V y un conjunto de aristas E, donde cada arista es un par no ordenado de vértices distintos de V.

■ **Ejemplo 4.1** Considere el grafo representado en la Figura 4.1. Los vértices son los puntos etiquetados con letras y las aristas son las líneas que los conectan. Así, tenemos el conjunto de vértices $V = \{A, B, C, D\}$ y el conjunto de aristas $E = \{AB, AC, BC, BD, CD\}$. Para representar los vértices se suelen utilizar números y/o letras, y para las aristas, el par de vértices conectados.

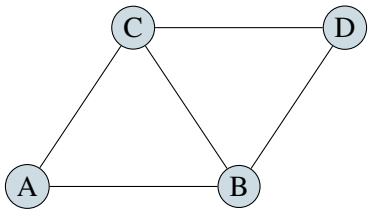


Figura 4.1: Un grafo con cuatro vértices y cinco aristas

■

Definición 4.1.2 — Digrafo. Un *grafo dirigido*, o *digrafo*, es un grafo en el que cada arista tiene una dirección y se representa como un par ordenado de vértices. Al escribir el conjunto de aristas de un digrafo, el **orden** es importante.

■ **Ejemplo 4.2** Los digrafo son adecuados para representar relaciones asimétricas, como calles de sentido único, flujo de información o relaciones de precedencia. Considere la Figura 4.2. Aquí $V = \{A, B, C\}$ y $E = \{AB, BC, CA\}$; observe que ahora el orden de los vértices que etiquetan una arista indica la dirección de dicha arista.

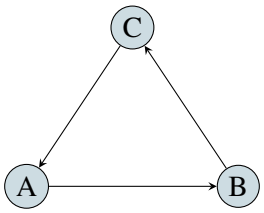


Figura 4.2: Digraph

■

Definición 4.1.3 — Cadena / Camino. Un *camino* es una secuencia de vértices tal que cada par consecutivo está conectado por una arista. Los caminos pueden ser abiertos (primer y último vértice diferentes) o cerrados (cuando el primer y el último vértice coinciden).

R En un grafo conectado, los caminos deben respetar la dirección de las aristas.

Definición 4.1.4 — Loop (Ciclo). Un *ciclo* es un camino en el que el primer y el último vértice son iguales. Si un grafo no contiene ciclos se denomina grafo *acíclico*.

■ **Ejemplo 4.3** Considere el grafo en la Figura 4.1

- Existen varios caminos que conectan A con D, por ejemplo, $P_1 = \{AC, CD\}$, $P_2 = \{AC, CB, BD\}$.
- El camino $\{AC, CB, AB\}$ es un ciclo, porque comienza y termina en el mismo vértice A.

Los caminos son fundamentales para analizar la alcanzabilidad y la conectividad dentro de un grafo.

■

Definición 4.1.5 — Grafo conexo. Un grafo se dice *conexo* si siempre existe un camino entre cada par de vértices.

■ **Ejemplo 4.4** Los grafos en las Figuras 4.1, 4.2 y 4.4 son conexos; sin embargo, el grafo en la Figura 4.3 no es conexo. Observe que no existe ningún camino que comience en A y termine en E.

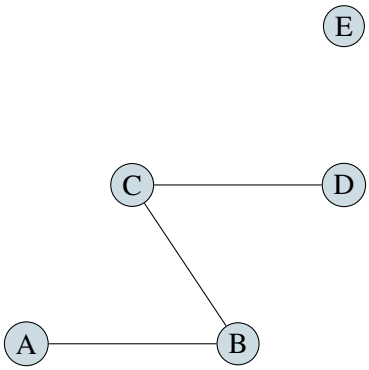


Figura 4.3: Grafo no conexo

■

Definición 4.1.6 — Árbol. Un *árbol* es un grafo conexo y acíclico.

Definición 4.1.7 — Subgrafo. Sean dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$. Decimos que G_1 es un subgrafo de G_2, y lo denotamos como $G_1 \subset G_2$, si $V_1 \subset V_2$ y $E_1 \subset E_2$.

Definición 4.1.8 — Árbol Soporte. Dado un grafo conexo G, un *árbol soporte* es un subgrafo de G que incluye todos los vértices de G y que es, a su vez, un árbol.

■ Ejemplo 4.5

- El grafo en la Figura 4.4 es un subgrafo de la Figura 4.1.
- El grafo en la Figura 4.4 es un grafo conexo y acíclico, por lo tanto, un árbol.
- Además, la Figura 4.4 es el árbol soporte de la Figura 4.1.

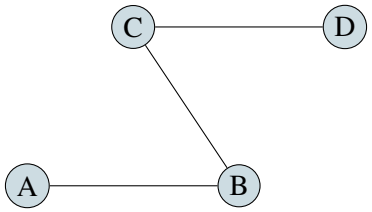


Figura 4.4: Spanning Tree (Árbol Soporte)

■

Los árboles son estructuras mínimas que conectan todos sus vértices sin ciclos y desempeñan un papel esencial en modelos jerárquicos y estructuras de datos. Los árboles soporte son útiles en problemas de optimización, como la minimización del coste de conexión de una red.

4.2 El problema del camino más corto

El *problema del camino más corto* es un concepto fundamental en teoría de grafos y optimización, ampliamente utilizado en logística, diseño de redes y aplicaciones biomédicas como el análisis de imágenes médicas, el mapeo de conectividad neuronal y los sistemas de enrutamiento hospitalario. El objetivo es determinar el camino entre dos nodos en un grafo de manera que el coste total (o distancia) sea mínimo.

Consideramos un grafo dirigido $G = (V, E)$ donde:

- V es el conjunto de vértices (nodos), que representan ubicaciones, estados o entidades.
- E es el conjunto de aristas (arcos), cada una con un coste no negativo asociado c_{ij} y una dirección para la arista $(i, j) \in E$.

Sea $s \in V$ el nodo origen y $t \in V$ el nodo destino. El objetivo es encontrar un camino desde s hasta t tal que la suma de los costes a lo largo del camino sea mínima.

Modelo Matemático

El objetivo de este problema es encontrar el camino más corto entre dos nodos predefinidos en un grafo. Dado que un camino es una secuencia de aristas, debemos decidir qué arista se incluye en el camino. Para ello, definimos las variables de decisión binarias:

$$x_{ij} = \begin{cases} 1 & \text{si la arista } (i, j) \text{ se incluye en el camino,} \\ 0 & \text{si no.} \end{cases}$$

para $(i, j) \in E$.

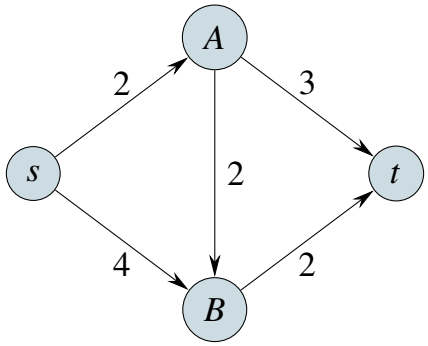


Figura 4.5: Ejemplo de grafo del problema del camino más corto

Así, dado que el coste de usar la arista (i, j) es $c_{ij}x_{ij}$, la función objetivo se escribe como

$$\min_{\mathbf{x}} z = \sum_{(i,j)\in E} c_{ij}x_{ij}$$

Finalmente, si el viajero toca un vértice, eso significa que utiliza una arista que entra en el vértice (una arista que termina en el vértice), y dado que debe continuar viajando, debe salir del vértice utilizando una arista que sale del vértice (una arista que comienza en el vértice). Esta restricción se denomina, por razones que estudiaremos más adelante, restricción de conservación de flujo.

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} 1 & \text{si } i = s, \\ -1 & \text{si } i = t, \\ 0 & \text{en otro caso.} \end{cases}$$

El protocolo establece que, si una arista entra en un vértice, se utiliza signo negativo, y si sale, se utiliza signo positivo. Existen dos excepciones a este comportamiento: s (origen) y t (destino). Desde el origen, el viajero solo puede salir, y de esta forma debemos usar únicamente una arista; lo mismo ocurre con el vértice destino, el viajero solo puede entrar, utilizando únicamente una arista, por lo que consideramos únicamente las aristas que entran.

Existen varios algoritmos para resolver el problema del camino más corto de manera eficiente:

- **Algoritmo de Dijkstra**: Adecuado para grafos con pesos no negativos en las aristas.
- **Algoritmo de Bellman-Ford**: Maneja grafos con pesos negativos y detecta ciclos negativos.
- **Búsqueda A***: Utiliza heurísticas para guiar la búsqueda, aplicada frecuentemente en imágenes biomédicas y robótica.

■ **Ejemplo 4.6** Considere el grafo en la Figura 4.5. Vamos a encontrar el camino más corto desde s hasta t. Primero, escriba el conjunto de aristas y vértices.

$$V = \{s, A, B, t\}$$
$$E = \{sA, sB, AB, At, Bt\}$$

Declare las variables binarias

$$x_{ij} = \begin{cases} 1 & \text{si la arista } (i, j) \text{ está incluida en el camino,} \\ 0 & \text{en caso contrario.} \end{cases}$$

para $(i, j) \in E$.

La función objetivo resulta

$$\min_{\mathbf{x}} z = \sum_{(i,j)\in E} c_{ij}x_{ij} = 2x_{sA} + 4x_{sB} + 2x_{AB} + 3x_{At} + 2x_{Bt}$$

Las restricciones de conservación de flujo son

$$\begin{aligned} x_{sA} + x_{sB} &= 1 && \text{el origen envía al viajero} \\ -x_{sA} + x_{AB} + x_{At} &= 0 && \text{vertex } A \text{ está en equilibrio} \\ -x_{sB} - x_{AB} + x_{Bt} &= 0 && \text{vertex } B \text{ está en equilibrio} \\ -x_{At} + x_{Bt} &= -1 && \text{el destino recibe al viajero} \end{aligned}$$

El modelo resulta

$$\begin{aligned} \min \quad z \quad & = 2x_{sA} + 4x_{sB} + 2x_{AB} + 3x_{At} + 2x_{Bt} \\ \text{s.t.} \quad & x_{sA} + x_{sB} = 1 \\ & -x_{sA} + x_{AB} + x_{At} = 0 \\ & -x_{sB} - x_{AB} + x_{Bt} = 0 \\ & -x_{At} + x_{Bt} = -1 \\ & x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E \end{aligned}$$

Para resolver el problema se emplea el Código 4.1.

```
% Number of variables
numVars = 5;

% Cost vector (objective function)
% [x_sA, x_sB, x_AB, x_At, x_Bt]
f = [2; 4; 2; 3; 2];

% Integer (binary) variables
intcon = 1:numVars;

% Bounds
lb = zeros(numVars,1);
ub = ones(numVars,1);

% ========================
% Equality constraints
% ========================

% Constraint 1: x_sA + x_sB = 1
Aeq1 = [1  1  0  0  0];
beq1 = 1;

% Constraint 2: -x_sA + x_AB + x_At = 0
Aeq2 = [-1  0  1  1  0];
beq2 = 0;

% Constraint 3: -x_sB - x_AB + x_Bt = 0
Aeq3 = [0 -1 -1  0  1];
beq3 = 0;

% Constraint 4: -x_At + x_Bt = -1
```

```
Aeq4 = [0  0  0 -1  1];
beq4 = -1;

% Assemble equality constraints
Aeq = [Aeq1;
       Aeq2;
       Aeq3;
       Aeq4];

beq = [beq1;
       beq2;
       beq3;
       beq4];

% ========================
% Solve the problem
% ========================
[x, fval] = intlinprog(f, intcon, [], [], Aeq, beq, lb, ub);

% ========================
% Output
% ========================
disp('*** Solution ***')
fprintf('x_sA = %d\n', x(1));
fprintf('x_sB = %d\n', x(2));
fprintf('x_AB = %d\n', x(3));
fprintf('x_At = %d\n', x(4));
fprintf('x_Bt = %d\n', x(5));
fprintf('Objective value: %f\n', fval);
```

Código 4.1: Camino más corto

Por lo tanto, el camino más corto desde s hasta t es $s \to A \to t$ con un coste total de 5. ∎

R En ingeniería en telecomunicaciones, los algoritmos de camino más corto se emplean en aplicaciones como el enrutamiento óptimo de paquetes en redes, el trazado eficiente de rutas en topologías de fibra óptica y la planificación de respuesta ante fallos en infraestructuras de comunicación.

4.3 Árbol de Expansión Mínima

El Problema del Árbol de Expansión Mínima es un concepto fundamental en teoría de grafos y optimización combinatoria. Consiste en encontrar un subconjunto de aristas en un grafo conexo y ponderado que conecte todos los vértices con el peso total mínimo posible y sin formar ciclos. En ingeniería en telecomunicaciones, los árboles de expansión mínima pueden utilizarse para diseñar redes de transmisión eficientes, optimizar la topología de redes de sensores en sistemas IoT o planificar infraestructuras de comunicación con coste mínimo.

Considere el grafo G en la Figura 4.6

El problema aquí consiste en extraer un subgrafo conexo de G que no contenga ciclos (un árbol) con los menores pesos en las aristas. Por lo tanto, se requiere que G sea un grafo conexo, no dirigido, con pesos positivos. El objetivo es obtener un conjunto de aristas que conecte **todos** los nodos cuyo peso total sea mínimo. Nótese que si una red contiene n nodos, entonces se requieren exactamente $n-1$ aristas para conectarlos a todos. Dichos arcos deben seleccionarse de manera

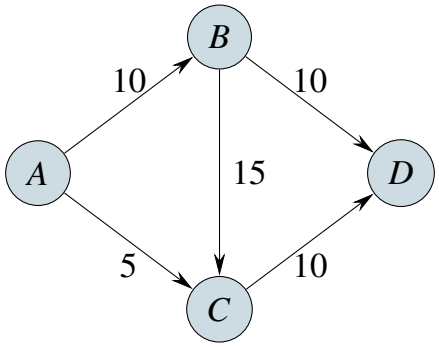


Figura 4.6: A graph with loops

que el peso total se minimice. Este problema puede aplicarse en el despliegue de redes, transporte y distribución.

Para resolver este problema aplicamos el algoritmo de Prim.

Algorithm 3 Algoritmo de Prim

1: Sea $T = (V_T, E_T)$, donde $V_T = E_T = \emptyset$. Elija un nodo arbitrario $v_0 \in V = \{v_0, \ldots, v_{n-1}\}$ y añádalo al subgrafo T, así, $V_T = \{v_0\}$
2: **while** existan nodos que aún no estén conectados a T, es decir, $V_T \neq V$ **do**
3: Encuentre la arista e_k con peso mínimo que conecte T con un nodo que no esté en T
4: Añada dicho nodo y arista a T, $V_T = V_T \cup \{v_k\}$ y $E_T = E_T \cup \{e_k\}$
5: **end while**

Después de $n-1$ pasos, el grafo resultante será un árbol de expansión mínima. En caso de que se consideren dos aristas con igual peso en el mismo paso, elija una de manera arbitraria.

■ **Ejemplo 4.7** Encuentre el árbol de expansión mínima en el grafo de la Figura
Aplique el algoritmo de Prim en los siguientes pasos

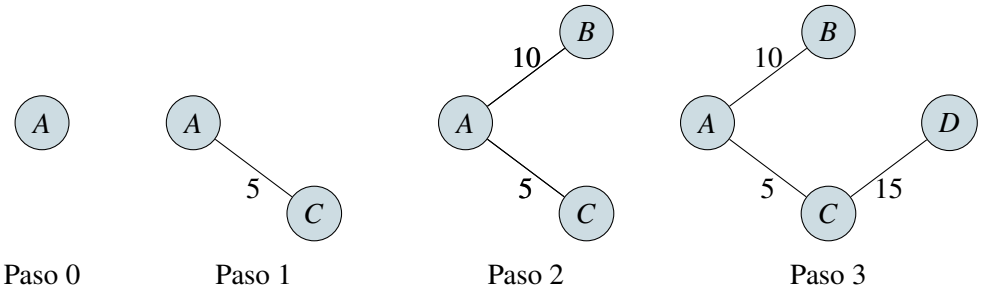


Figura 4.7: Pasos del algoritmo de Prim

Por lo tanto, el árbol de expansión mínima está dado por $T = (V, E_T)$ donde $V = \{A, B, C, D\}$ y $E_T = \{AB, AC, CD\}$. Con un peso total de 30. ■

4.4 Problema del flujo máximo

El Problema de Flujo Máximo es un problema clásico de optimización en la teoría de redes. Busca determinar la cantidad máxima de flujo que puede enviarse desde un nodo origen hasta un nodo destino a través de una red de aristas con restricciones de capacidad. En ingeniería en telecomunicaciones, este concepto es particularmente útil para modelar el tráfico de datos en

redes, optimizar sistemas de transmisión y analizar el transporte de información en infraestructuras complejas.

Consideramos un grafo dirigido $G = (V, E)$, e imponemos la restricción de que el flujo debe enviarse desde el origen hasta el destino sin pérdidas, es decir, ningún nodo intermedio puede retener flujo.

- V: Conjunto de nodos por los que debe atravesar el flujo.
- E: Conjunto de aristas dirigidas que representan las rutas por las que puede viajar el flujo.

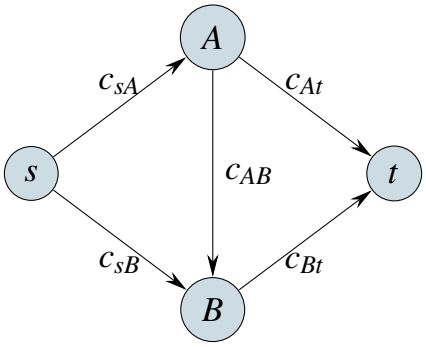


Figura 4.8: Ejemplo de grafo para el problema de flujo máximo

Modelo matemático

Sea $G = (V, E)$ un grafo dirigido con capacidad c_{ij} en cada arista (i, j). Definimos las variables de flujo x_{ij} tales que:

$$x_{ij} = \text{unidades de flujo a través de la arista } (i, j) \quad \text{para} \quad (i, j) \in E$$

El flujo que atraviesa el grafo puede detectarse de dos maneras equivalentes: como el flujo que sale de s o como el flujo que entra en t, por lo tanto, este problema tiene dos funciones objetivo equivalentes

$$\max_{\mathbf{x}} z = \sum_{j \in V} x_{sj} = \sum_{i \in V} x_{it}$$

Por otro lado, tenga en cuenta que ningún nodo intermedio puede retener flujo, por lo que existe una restricción de conservación de flujo. En otras palabras, el flujo que entra y el que sale debe ser el mismo para todos los nodos intermedios. Recordando el protocolo, el flujo que entra se etiqueta como negativo y el que sale como positivo, por lo tanto, las restricciones de conservación de flujo son

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} z & \text{si } i = s, \\ -z & \text{si } i = t, \\ 0 & \text{en otro caso.} \end{cases}$$

Nótese que los vértices especiales s y t no se incluyen como restricciones, porque forman parte de la función objetivo.

Finalmente, el flujo debe satisfacer ciertas restricciones de caja, es decir, el flujo a través de cada arista debe ser positivo y no puede exceder la capacidad de cada arista. Esto es,

$$0 \leq x_{ij} \leq c_{ij}, \quad \text{para} \quad (i, j) \in E$$

El modelo resulta

$$\begin{aligned} \max \quad & z = \sum_{j\in V} x_{sj} = \sum_{i\in V} x_{it} \\ \text{s.t.} \quad & x_{ij} \le c_{ij}, \quad \forall (i,j) \in E \\ & \sum_{j\in V} x_{ij} = \sum_{j\in V} x_{ji}, \quad \forall i \in V \setminus \{s,t\} \\ & x_{ij} \ge 0, \quad \forall (i,j) \in E \end{aligned}$$

■ **Ejemplo 4.8** Considere el grafo en la Figura 4.9, donde las etiquetas de las aristas representan la capacidad máxima de cada arista. Determinemos el flujo máximo que puede enviarse desde s hasta t

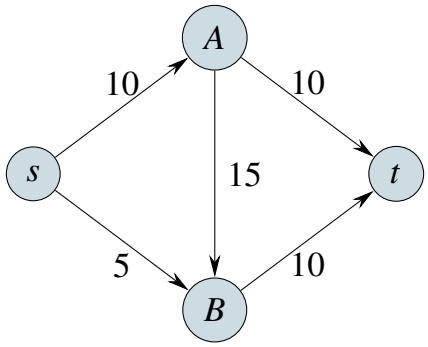


Figura 4.9: Grafo para ejemplo 4.8

$$\begin{aligned} V &= \{s,A,B,t\} \\ E &= \{sA,sB,AB,At,Bt\} \end{aligned}$$

Declare las variables de decisión

$$x_{ij} = \text{unidades de flujo enviadas a través de la arista}$$

para $(i,j) \in E$.

La función objetivo resulta

$$\max_{\mathbf{x}} z = x_{sA} + x_{sB} = x_{At} + x_{Bt}$$

Ahora, las restricciones de conservación de flujo son

$$\begin{aligned} -x_{sA} + x_{AB} + x_{At} &= 0 \quad \text{el vértice } A \text{ conserva el flujo} \\ -x_{sB} - x_{AB} + x_{Bt} &= 0 \quad \text{el vértice } B \text{ conserva el flujo} \end{aligned}$$

Las variables están sujetas a las restricciones de caja

$$\begin{aligned} 0 \le x_{ij} \quad & \forall (i,j) \in E \\ x_{sA} &\le 10 \\ x_{sB} &\le 5 \\ x_{AB} &\le 15 \\ x_{At} &\le 10 \\ x_{Bt} &\le 10 \end{aligned}$$

El modelo completo resulta

$$
\begin{aligned}
\max \quad z \quad & = x_{sA} + x_{sB} = x_{At} + x_{Bt} \\
\text{s.t.} \quad & -x_{sA} + x_{AB} + x_{At} = 0 \\
& -x_{sB} - x_{AB} + x_{Bt} = 0 \\
& x_{sA} \leq 10 \\
& x_{sB} \leq 5 \\
& x_{AB} \leq 15 \\
& x_{At} \leq 10 \\
& x_{Bt} \leq 10 \\
& x_{ij} \geq 0, \quad \forall (i,j) \in E
\end{aligned}
$$

Este modelo se implementa como
Para resolver el problema se emplea el Código 4.2.

```
% Number of variables
numVars = 5;

% ========================
% Objective function
% max z = x_sA + x_sB
% linprog minimizes, so we negate
% ========================
f = -[1; 1; 0; 0; 0];

% ========================
% Equality constraints (flow conservation)
% ========================

% -x_sA + x_AB + x_At = 0
Aeq1 = [-1  0  1  1  0];
beq1 = 0;

% -x_sB - x_AB + x_Bt = 0
Aeq2 = [ 0 -1 -1  0  1];
beq2 = 0;

Aeq = [Aeq1;
       Aeq2];

beq = [beq1;
       beq2];

% ========================
% Bounds
% ========================
lb = zeros(numVars,1);
ub = [10; 5; 15; 10; 10];

% ========================
% Solve the problem
% ========================
[x, fval] = linprog(f, [],[], Aeq, beq, lb, ub);

```

```
% Recover max objective value
z = -fval;

% =========================
% Output
% =========================
disp('*** Solution ***')
fprintf('x_sA = %.2f\n', x(1));
fprintf('x_sB = %.2f\n', x(2));
fprintf('x_AB = %.2f\n', x(3));
fprintf('x_At = %.2f\n', x(4));
fprintf('x_Bt = %.2f\n', x(5));
fprintf('Maximum flow value: %.2f\n', z);
```

Código 4.2: Flujo Máximo

Por lo tanto, el grafo admite un flujo máximo de 15 unidades, utilizando la siguiente distribución:

- Flujo en la arista sA: 10.0
- Flujo en la arista sB: 5.0
- Flujo en la arista AB: 0.0
- Flujo en la arista At: 10.0
- Flujo en la arista Bt: 5.0

■

R En contextos de ingeniería telemática o de telecomunicaciones, resolver el problema de flujo máximo puede ayudar a optimizar la transmisión de datos en redes, garantizar una distribución eficiente del tráfico en infraestructuras críticas y modelar la dinámica del flujo de información en sistemas distribuidos. Comprender las limitaciones del flujo y los principios de conservación es esencial para diseñar redes seguras y eficientes.

4.5 Problema de flujo de costo mínimo

El problema de Flujo de Costo Mínimo es un modelo de optimización fundamental en la teoría de redes. Su objetivo es determinar la forma más eficiente en términos de coste para enviar una cierta cantidad de flujo a través de una red desde nodos de suministro hacia nodos de demanda, sujeto a restricciones de capacidad. Este problema tiene amplias aplicaciones en telecomunicaciones, como la optimización del enrutamiento de tráfico en redes IP, la asignación eficiente de recursos en redes ópticas o la minimización del coste en redes de transporte de datos.

El modelo que se aplicará para la resolución de este tipo de problema incluye todos los modelos anteriores, incluso transporte y asignación.

Consideremos un grafo dirigido $G = (V, E)$, donde:

- V es el conjunto de nodos (que representan routers, estaciones base o puntos de conmutación).
- E es el conjunto de arcos (que representan enlaces de comunicación).
- Cada arco $(i, j) \in A$ tiene:
  - Una capacidad $u_{ij} \geq 0$.
  - Un coste por unidad de flujo c_{ij}.
- Cada nodo $i \in N$ tiene un valor de oferta/demanda b_i, donde:
  - $b_i > 0$: nodo de suministro (origen de tráfico).
  - $b_i < 0$: nodo de demanda (destino de tráfico).
  - $b_i = 0$: nodo de tránsito.

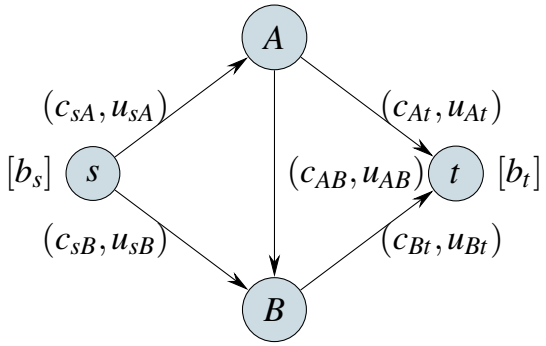


Figura 4.10: Grafo del problema de flujo de coste mínimo

Modelo Matemático

El problema de flujo de coste mínimo puede formularse como:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\ \text{s.a.} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i \quad \forall i \in N, \\ & 0 \le x_{ij} \le u_{ij} \quad \forall (i,j) \in E. \end{aligned}$$

Donde:

- x_{ij} es el flujo en el arco (i, j).
- La primera restricción asegura la conservación del flujo.
- La segunda restricción impone los límites de capacidad.

Teorema 4.5.1 — Existencia de soluciones factibles. Una condición necesaria para que un problema de costo mínimo tenga soluciones factibles es que

$$\sum_{i=1}^{n} b_i = 0$$

Esto es, la oferta y la demanda están equilibradas.

R En caso de $\sum_{i=1}^{n} b_i \neq 0$, se tendrán que ajustar las restricciones de flujo de manera acorde, en la dirección del desbalance, de manera que se pueda encontrar una solución. Generalmente, con ajustar las desigualdades en la dirección en la que hay exceso (oferta o demanda), será suficiente.

■ **Ejemplo 4.9 — Flujo de coste mínimo.** Considere el grafo en la Figura 4.11. Vamos a encontrar el flujo de coste mínimo que se puede enviar desde el nodo s hasta t. Primero, escriba el conjunto de aristas y vértices.

$$\begin{aligned} V &= \{s, A, B, t\} \\ E &= \{sA, sB, AB, At, Bt\} \end{aligned}$$

Declare las variables de decisión

$$x_{ij} : \text{unidades de flujo enviadas desde nodo } i \text{ and nodo } j \; \forall (i,j) \in E.$$

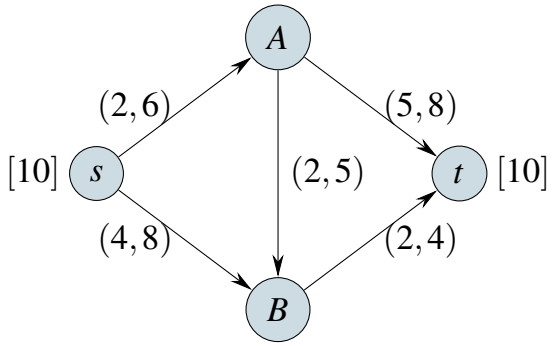


Figura 4.11: Grafo del problema de flujo de coste mínimo

La función objetivo resulta

$$\min_{\mathbf{x}} z = \sum_{(i,j)\in E} c_{ij}x_{ij} = 2x_{sA} + 4x_{sB} + 2x_{AB} + 5x_{At} + 2x_{Bt}$$

Las restricciones de conservación de flujo son

$$\begin{aligned}
x_{sA} + x_{sB} &= 10 \quad \text{el origen envía toda su oferta} \\
-x_{sA} + x_{AB} + x_{At} &= 0 \quad \text{el vértice } A \text{ no retiene flujo} \\
-x_{sB} - x_{AB} + x_{Bt} &= 0 \quad \text{el vértice } B \text{ no retiene flujo} \\
-x_{At} + x_{Bt} &= -1 \quad \text{el destino recibe todo el flujo}
\end{aligned}$$

Notemos que este problema está equilibrado. Las restricciones de capacidad de la red resultan

$$\begin{aligned}
x_{sA} &\leq 6 \\
x_{sB} &\leq 8 \\
x_{AB} &\leq 5 \\
x_{At} &\leq 8 \\
x_{Bt} &\leq 8
\end{aligned}$$

El modelo resulta

$$\begin{aligned}
\min \quad z \quad &= 2x_{sA} + 4x_{sB} + 2x_{AB} + 5x_{At} + 2x_{Bt} \\
\text{s.t.} \quad & x_{sA} + x_{sB} = 10 \\
& -x_{sA} + x_{AB} + x_{At} = 0 \\
& -x_{sB} - x_{AB} + x_{Bt} = 0 \\
& -x_{At} - x_{Bt} = -10 \\
& x_{sA} \leq 6 \\
& x_{sB} \leq 8 \\
& x_{AB} \leq 5 \\
& x_{At} \leq 8 \\
& x_{Bt} \leq 8 \\
& x_{ij} \geq 0, \quad \forall (i,j) \in E
\end{aligned}$$

Para resolver el problema se emplea el Código 4.3.

```
1
2  % Number of variables
3  numVars = 5; % [x_sA, x_sB, x_AB, x_At, x_Bt]
4  % =========================
5  % Objective function
6  % =========================
7  f = [2; 4; 2; 10; 2];
8
9  % =========================
10 % Equality constraints
11 % =========================
12 Aeq1 = [1   1   0   0   0];   % x_sA + x_sB = 10
13 Aeq2 = [-1   0   1   1   0];   % -x_sA + x_AB + x_At = 0
14 Aeq3 = [0  -1  -1   0   1];   % -x_sB - x_AB + x_Bt = 0
15 Aeq4 = [0   0   0  -1   -1];    % -x_At + x_Bt = -10
16
17 Aeq = [Aeq1;Aeq2;Aeq3;Aeq4];
18 beq = [10; 0; 0; -10];
19
20 % =========================
21 % Bounds
22 % =========================
23 lb = zeros(numVars,1);          % x_ij >= 0
24 ub = [6; 8; 5; 8; 8];          % capacities: x_sA<=6, x_sB<=8, x_AB<=5,
       x_At<=8, x_Bt<=3
25
26 % =========================
27 % Solve the problem
28 % =========================
29 options = optimoptions('linprog','Display','none');
30 [x, fval, exitflag, output] = linprog(f, [], [], Aeq, beq, lb, ub,
      options);
31
32 % =========================
33 % Output
34 % =========================
35 disp('*** Solution ***')
36 fprintf('x_sA = %.4f\n', x(1));
37 fprintf('x_sB = %.4f\n', x(2));
38 fprintf('x_AB = %.4f\n', x(3));
39 fprintf('x_At = %.4f\n', x(4));
40 fprintf('x_Bt = %.4f\n', x(5));
41 fprintf('Minimum cost value: %.4f\n', fval);
```

Código 4.3: Flujo de coste mínimo

Con solución $\mathbf{x}^* = [6,4,4,2,8]$ para $z^* = 72$. Por lo tanto, el flujo óptimo resulta, en unidades

de flujo,

$$\begin{aligned} x_{sA} &= 6 \\ x_{sB} &= 4 \\ x_{AB} &= 4 \\ x_{At} &= 2 \\ x_{Bt} &= 8 \end{aligned}$$

para un coste mínimo de 72 unidades. ∎

R En ingeniería de telecomunicaciones, el modelo del flujo de coste mínimo permite optimizar el enrutamiento de tráfico en redes de datos, minimizar costes en redes de transporte óptico y garantizar una asignación eficiente de recursos en redes multiservicio. Al reducir costes y mejorar la calidad del servicio, este enfoque impacta directamente en la eficiencia de la red y en la experiencia del usuario. Recordemos que este modelo engloba todas las aplicaciones vistas anteriormente.

5. Administración de proyectos con PERT/CPM

Para información más detallada, consultar [2, Sec. 9.8].

5.1 Introducción a PERT y CPM

Uno de los retos más importantes a la hora de dirigir proyectos de gran escala es la planificación. Es necesario conocer todas las dependencias, coordinar las partes y tener en cuenta posibles desviaciones. Para realizar este cometido, se utilizan las técnicas **PERT** (Program Evaluation and Review Technique, Técnicas de evaluación y revisión del programa) y **CPM** (Critical Path Method, método del camino crítico), que ayudan al director de proyectos a lidiar con todos los problemas de organización. Ambas técnicas están orientadas a la representación gráfica y análisis temporal de proyectos. Su creación se remonta a la década de los 50.

Definición 5.1.1

- **Actividad**: Tarea indivisible que consume tiempo y recursos.
- **Evento**: Punto en el tiempo que marca el inicio o fin de una o varias actividades.
- **Precedencia**: Una tarea i precede a una tarea j si i no puede ser completada hasta que se completa j.
- **Red de proyecto**: Representación gráfica mediante nodos (actividades) y arcos (relación de precedencia entre actividades) que describe la secuencia lógica del proyecto.
- **Ruta crítica**: Secuencia de actividades que determina la duración mínima del proyecto; cualquier retraso en esta ruta implica retraso en el proyecto completo.



En telecomunicaciones, PERT y CPM son herramientas clave para la planificación de despliegues de redes, instalación de equipos y coordinación de proyectos de gran escala, donde el cumplimiento de plazos es crítico para garantizar la calidad del servicio y la competitividad. Por ejemplo, en el despliegue de fibra óptica, estas técnicas permiten identificar actividades críticas como la instalación de troncales y la configuración de nodos, optimizando recursos y reduciendo riesgos de retraso.

5.2 Ejemplo prototipo: proyecto de Reliable Construction Co

■ **Ejemplo 5.1 — Reliable Construction Co.** La empresa Reliable construction ha ganado un proyecto de 5.4 millones de dólares para construir una nueva fábrica. El contratante necesita que la fábrica esté terminada en el plazo de un año (52 semanas). Por ello, el contrato incluye las siguientes cláusulas:

- Penalización de 300 mil dólares si se acaba en un plazo superior a 47 semanas.
- Gratificación de 150 mil dólares si se acaba en un plazo inferior a 40 semanas.

La Tabla 5.1 muestra las actividades y demás información necesaria para la realización del proyecto:

Tabla 5.1: Tabla de precedencias ejemplo prototipo

Actividad	Descripción	Precedencia	Duración estimada
A	Excavar	-	2 semanas
B	Cimentar	A	4 semanas
C	Levantar muros	B	10 semanas
D	Construir tejado	C	6 semanas
E	Instalar fontanería exterior	C	4 semanas
F	Instalar fontanería interior	E	5 semanas
G	Instalar revestimiento exterior	D	7 semanas
H	Pintar exterior	E,G	9 semanas
I	Instalación eléctrica	C	7 semanas
J	Panelar	F,I	8 semanas
K	Instalar el piso	J	4 semanas
L	Pintar interior	J	5 semanas
M	Instalar accesorios exteriores	H	2 semanas
N	Instalar accesorios interiores	K,L	6 semanas

5.3 Uso de una red para visualizar un proyecto

Una forma muy eficiente de poder visualizar el proyecto, sus distintas tareas e información importantes es mediante el uso de grafos. Estos incluirán las actividades, relaciones de precedencia e información temporal. Se aplicará el enfoque AON (activity on node):

- Cada actividad se representa en un nodo
- Los arcos representan relaciones de precedencia.
- Los número en cada nodo representan la duración de la actividad.

En la Figura 5.1 se representa el resultado de aplicar el enfoque AON.

■

Para realizar esta organización necesitamos dos tipos de información:

- El orden de las actividades que deben realizarse.
- La duración estimada de cada actividad.

5.4 Programación de un proyecto con PERT/CPM

¿Cuánto tiempo va a tardar en realizarse el proyecto? Si sumamos la duración de todas las actividades, obtenemos un resultado de 79 semanas. Para calcular la duración del proyecto es

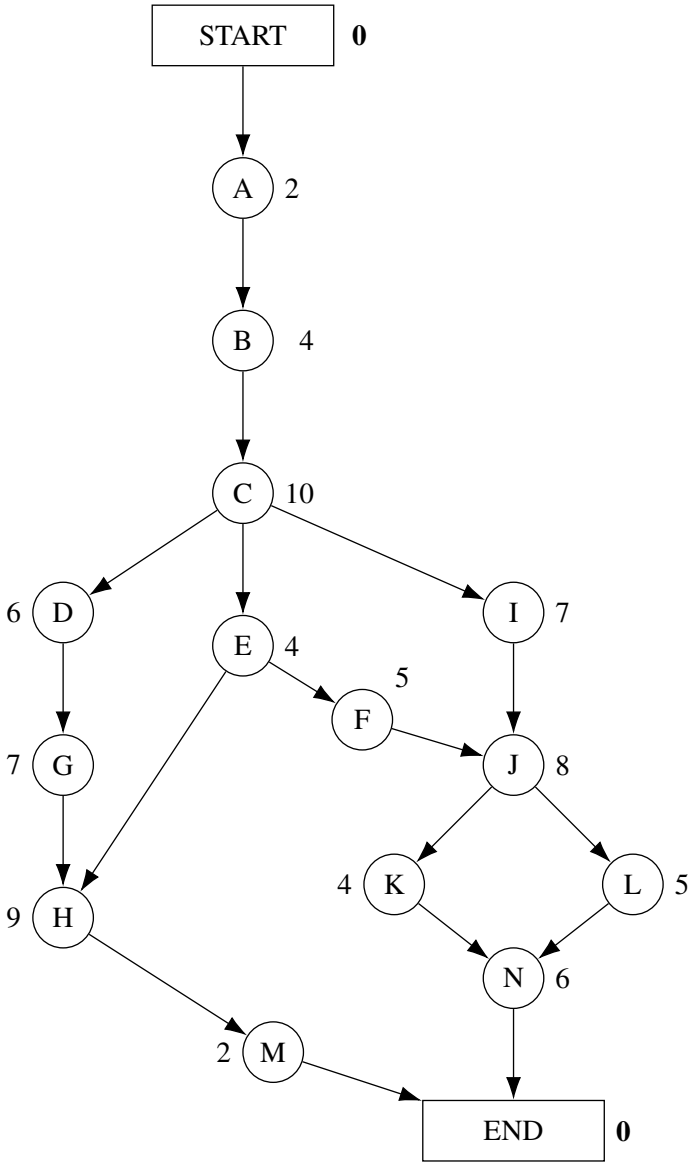


Figura 5.1: Representación AON correspondiente a Ejemplo 5.1

necesario calcular la duración de todos los caminos que van desde origen hasta el fin. La duración estimada del proyecto será aquel camino que tenga una duración mayor. Este camino recibe el nombre de **camino crítico**.

En la Tabla 5.2, se calcula la longitud de cada una de las rutas del nodo START al nodo END. Luego, se concluye que en este proyecto

- Ruta crítica: START $\rightarrow A \rightarrow B \rightarrow C \rightarrow E \rightarrow F \rightarrow J \rightarrow L \rightarrow N \rightarrow$ FIN
- Duración estimada del proyecto 44 semanas.

5.4.1 Cálculo de tiempos mínimos y máximos

Definición 5.4.1 — Tiempo de inicio-finalización mínimo (ES, EF). Al tiempo de inicio de una actividad , si no ocurre ningún retraso, se le conoce como **tiempo de inicio mínimo (ES - early start)**, de la misma forma, se define el tiempo de **finalización mínimo (EF - early finish)** tal que,

Tabla 5.2: Trayectorias y sus longitudes a través de la red de proyecto del Ejemplo 5.1, ver Figura 5.1

Trayectoria	Longitud (semanas)
START $\to A \to B \to C \to D \to G \to H \to M \to$ FIN	$2+4+10+6+7+9+2=40$
START $\to A \to B \to C \to E \to H \to M \to$ END	$2+4+10+4+9+2=31$
START $\to A \to B \to C \to E \to F \to / \to K \to N \to$ FIN	$2+4+10+4+5+8+4+6=43$
START $\to A \to B \to C \to E \to F \to J \to L \to N \to$ FIN	$2+4+10+4+5+8+5+6=44$
START $\to A \to B \to C \to l \to / \to K \to N \to$ FIN	$2+4+10+7+8+4+6=41$
START $\to A \to B \to C \to I \to J \to L \to N \to$ FIN	$2+4+10+7+8+5+6=42$

$$\text{EF} = \text{ES} + \text{duración}$$

Para mejorar la escalabilidad del problema, el ES de la primer actividad se fija en 0. Para ir rellenando el grafo, se utiliza la siguiente regla, el tiempo de inicio mínimo de una actividad es el tiempo de finalización mínimo mayor de entre sus predecesores.

En la Figura 5.2 se calculan los tiempos mínimos de inicio-finalización, de cada actividad del Ejemplo 5.1.

Del mismo modo se pueden definir los tiempos de inicio-finalización máximos.

Definición 5.4.2 — Tiempo de inicio-finalización máximos (LS, LF). Los **tiempos máximos de finalización e inicio** (LF,LS), (last finish, last start) se definen como aquellos que, aun siendo mayores que los mínimos, no modifican el camino mínimo. Estos tiempo se puede definir como

$$LS = LF - \text{duración}$$

Como regla general, el tiempo máximo de finalización de una actividad es el menor tiempo de inicio máximo las actividades sucesoras.

En la Figura 5.3 se calculan los tiempos máximos de inicio-finalización, de cada actividad del Ejemplo 5.1.

R Por lo general para calcular los pares (ES, EF) se va de la tarea START a la tarea FIN, por otro lado, para calcular los pares (LS, LF), se va de la tarea FIN a la tarea START, una vez se conoce la duración mínima del proyecto.

Definición 5.4.3 — Holgura. Las holguras nos permiten saber qué actividades pueden retrasarse sin que eso revierta en un retraso del proyecto. Para calcularlas se utiliza la siguiente fórmula:

$$h = LF - EF$$

El camino crítico tiene holguras 0.

En la Figura 5.4, se ilustra el cálculo de holguras para cada actividad del Ejemplo 5.1.

Luego se concluye que el camino crítico es

$$\text{START} \to A \to B \to C \to E \to F \to J \to L \to N \to \text{FINISH}$$

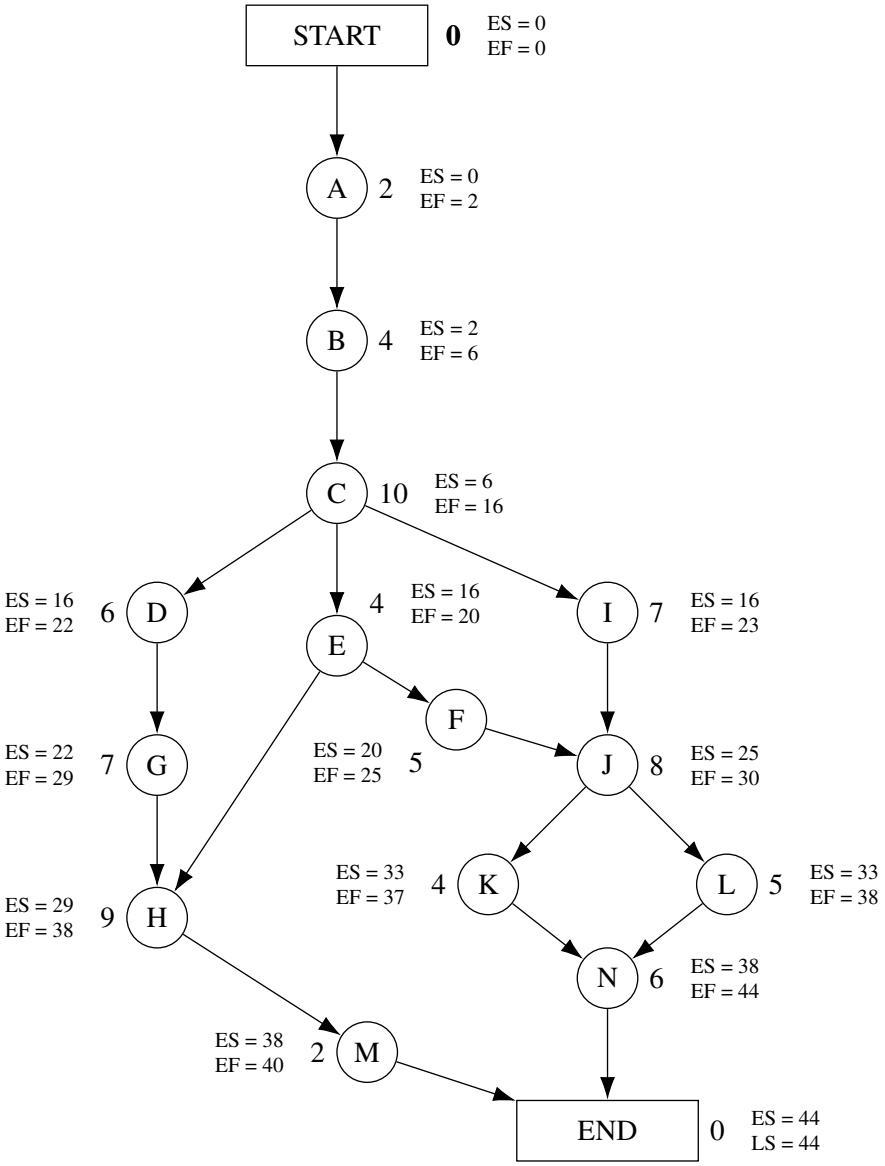


Figura 5.2: Tiempos mínimos de inicio-finalización de cada actividad del Ejemplo 5.1

5.5 Manejo de la incertidumbre en las duraciones de las actividades

PERT introduce tres estimaciones para cada actividad, **optimista (o)**, **probable (m)** y **pesimista (p)**. Denotemos como

- μ_i, duración media de la actividad i. Reservamos μ_p para la duración del proyecto
- σ_i, la desviación típica de la duración de la actividad i. Reservamos σ_p, para la desviación típica de la duración del proyecto.

Definición 5.5.1 — Camino crítico medio (CCM). Definiremos el camino crítico medio (CCM) como el valor del camino crítico si cada actividad tiene una duración igual a su media.

Para calcular los tiempos de duración de un proyecto se siguen los siguientes supuestos

- **Supuesto 1**: El camino más largo es el CCM.
- **Supuesto 2**: Las actividades del camino crítico son estadísticamente independientes, por lo

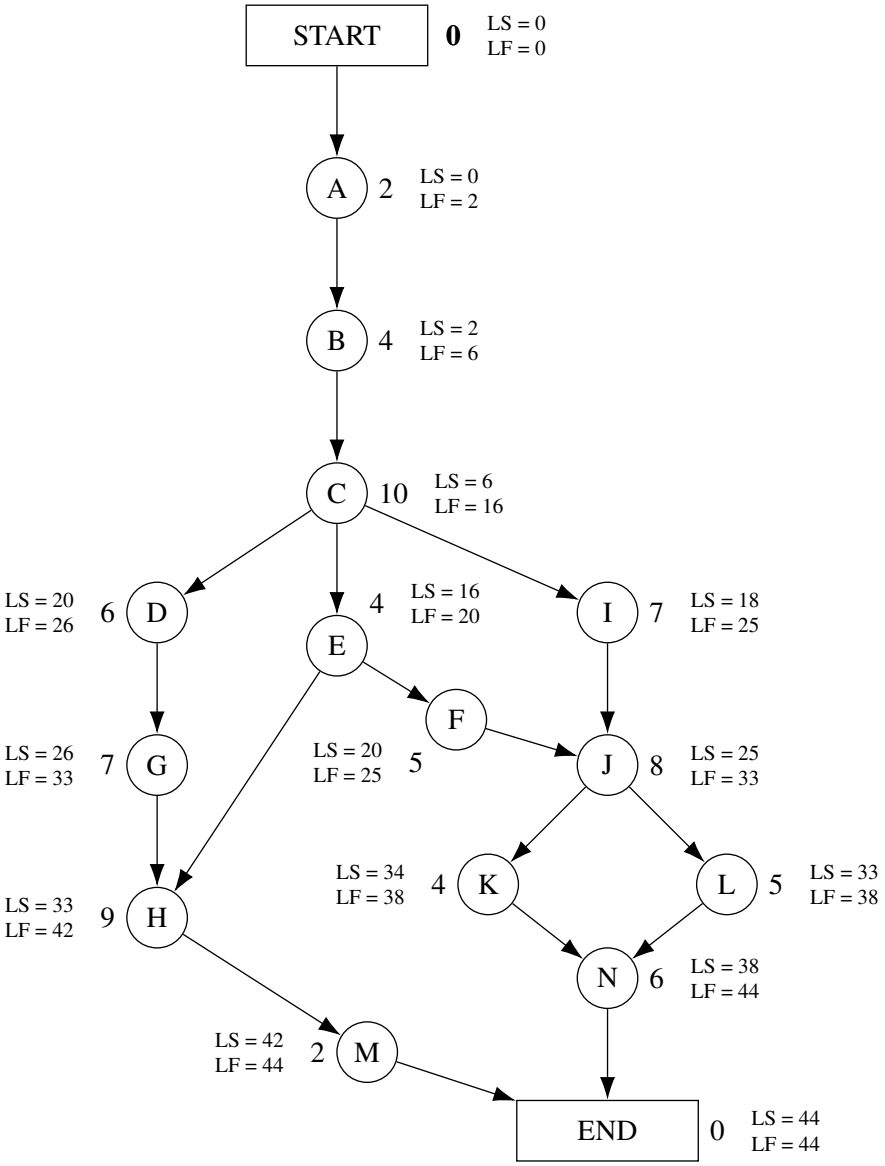


Figura 5.3: Tiempos máximos de inicio-finalización de cada actividad del Ejemplo 5.1

que

$$\mu_p = \sum_i^N \mu_i; \quad \sigma_p^2 = \sum_i^N \sigma_i^2; \quad \sigma_i^2 = \left(\frac{p_i - o_i}{6}\right)^2; \quad \mu_i = \frac{o_i + 4m_i + p_i}{6}$$

Luego se asume que

- μ_p: suma de las medias de las duraciones de las actividades del CCM.
- σ_p^2: suma de las varianzas de las duraciones de las actividades en el CCM.

■ **Ejemplo 5.2** Consultando a expertos se han podido estimar la media y varianza de cada una de las actividades del proyecto del Ejemplo 5.1. Ver Tabla 5.4.

■

- **Supuesto 3**: Se asume que la distribución es normal.
  De esta forma el cálculo de probabilidades vendría por la función de probabilidad acumulada

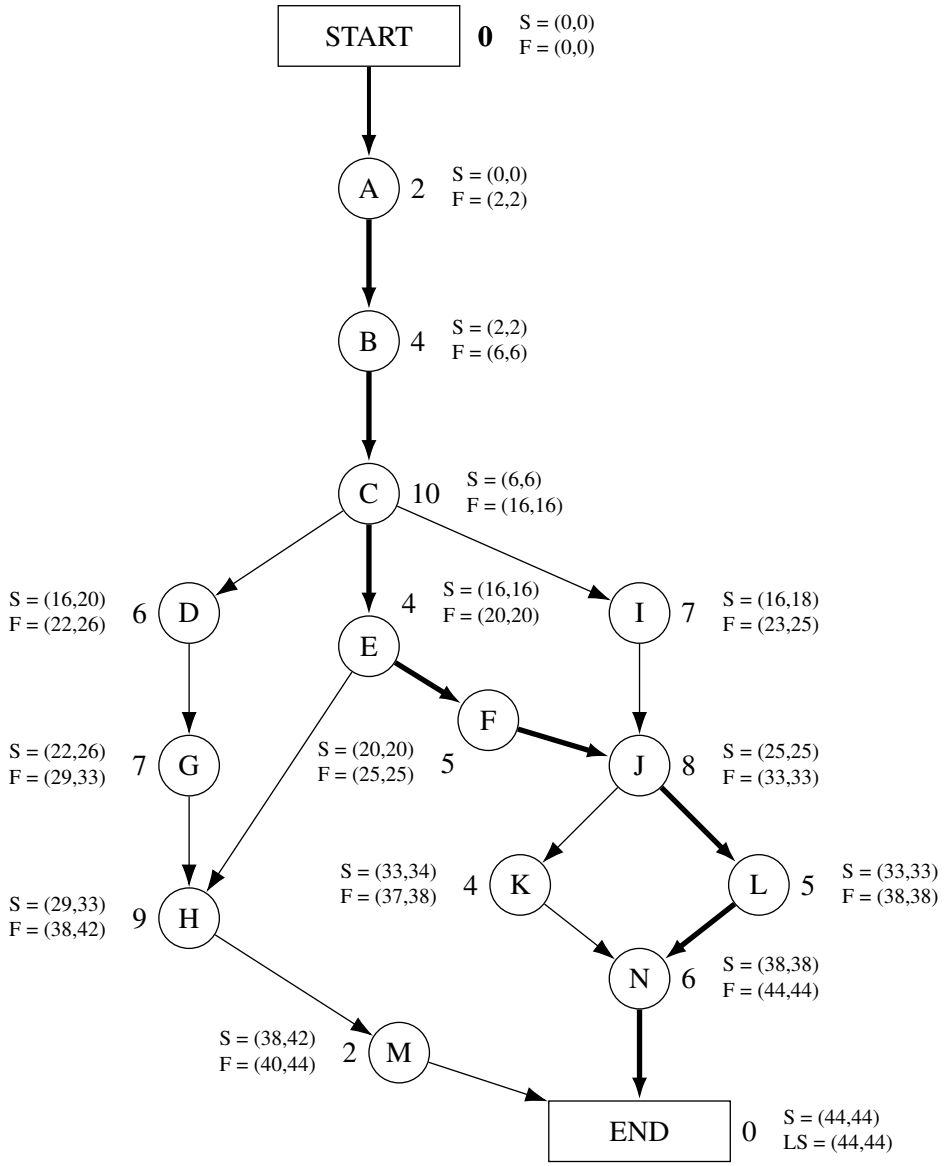


Figura 5.4: Ruta crítica Ejemplo 5.1

de la distribución normal.

$$P(T \leq d) = P(N \leq K_d); \quad K_d = \frac{d - \mu_p}{\sigma_p}$$

En el caso del Ejemplo

■ **Ejemplo 5.3** ¿Cuál es la probabilidad de que el proyecto se complete en 47 semanas o menos?

Aplicamos la función de matlab `normcdf(x,`μ`,`σ`)`, que nos devuelve los valores de la función de distribución acumulada, evaluados en los valores en x para la distribución normal con media μ y desviación estándar σ, de donde se obtiene

$$P(X \leq 47) = 0.8413$$

■

Tabla 5.3: Relación entre holguras y camino crítico

Actividad	Holgura (LF - EF)	Camino crítico
A	0	Sí
B	0	Sí
C	0	Sí
D	4	No
E	0	Sí
F	0	Sí
G	4	No
H	4	No
I	2	No
J	0	Sí
K	1	No
L	0	Sí
M	4	No
N	0	Sí

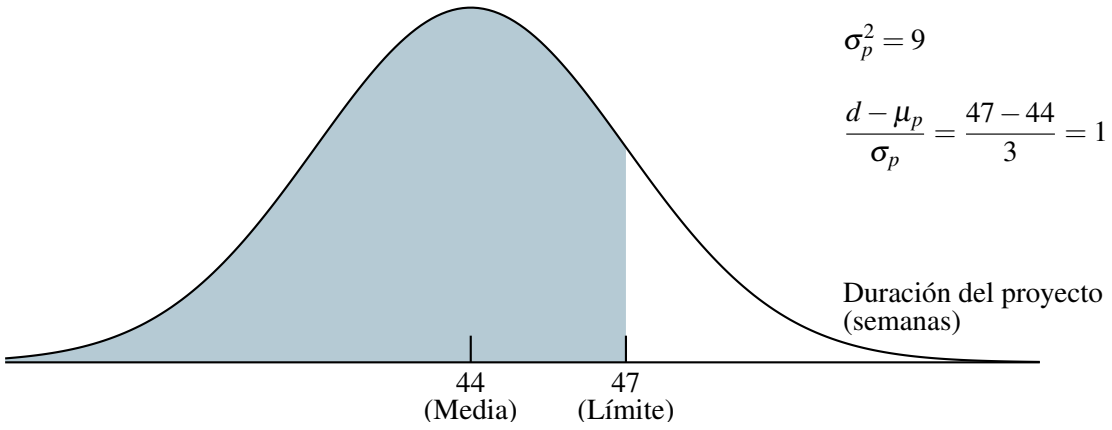


Figura 5.5: Estimación de la duración del proyecto a partir de los datos de la Tabla 5.4, empleando una aproximación normal.

Tabla 5.4: Aplicación de aproximación 2 a Ejemplo 5.1

Actividades en CCM	Media	Varianza
A	2	$\frac{1}{9}$
B	4	1
C	10	4
E	4	$\frac{4}{9}$
F	5	1
J	8	1
L	5	1
N	6	$\frac{4}{9}$
Duración del proyecto	$\mu_p = 44$	$\sigma_p^2 = 9$

5.6 Consideración del trueque entre tiempo y costo. Crashing.

CPM nos permite tener consideraciones en cuanto la reducción del tiempo de proyecto mediante la inyección de dinero a las actividades, en lo que se conoce como **crashing**. Este aporte extra de capital nos permite aplicar lo aprendido en el tema de programación lineal para encontrar soluciones óptimas, es decir, conseguir una reducción de la duración de la obra con la inversión mínima.

En primer lugar necesitaremos tener tabulados las posibles inversiones y su contrapartida en la reducción de la duración de las actividades. En caso de ser valores fijos, deberemos aplicar una interpolación lineal. Una vez realizado esto, debemos formular el modelo de programación matemática que quedará de la siguiente manera,

- Función objetivo: minimización de costes.
- Restricciones:
  - Reducción máxima.
  - No negatividad.
  - Tiempos de inicio, precedencia.
  - Duración del proyecto.

■ **Ejemplo 5.4** Consideremos el proyecto presentado en el Ejemplo 5.1. La empresa matriz nos comunica que desea terminar el proyecto en 40 semanas, para lo que destinará recursos extra. En la Tabla 5.5, se recogen los costes estimados y semanas que se pueden ahorrar aplicando crashing en cada tarea.

Con estos datos, ¿cuál es el coste mínimo que nos permite terminar el proyecto en 40 semanas?

Primeramente consideremos las variables x_i, para $i = A, \ldots, N$, que representan la reducción en tiempo obtenida al realizar crashing en la actividad i. Notemos que $x_i \in \mathbb{Z}_+$.

Luego la función objetivo será

$$\min z = 1000000x_A + 50000x_B + \ldots + 60000x_N$$

En las restricciones hay que incluir la precedencia. Para ello se introducen las variables auxiliares y_i, que representan el inicio de la actividad i.

En cuanto a las restricciones se tendrá:

- Restricciones de crashing, no se puede superar el límite impuesto.

$$x_A \leq 1,\ x_B \leq 2, \ldots, x_N \leq 3$$

- Restricciones de no negatividad: no se pueden tener tiempo o inicios negativos,

$$y_i \geq 0$$

Tabla 5.5: Costes de aplicar crashing al proyecto del Ejemplo 5.1

Actividad	Duración (semanas)		Coste ($)		Máxima reducción en semanas	Costo de crashing por semana ahorrada
	Normal	Con Crash	Normal	Con Crash		
A	2	1	$180,000	$ 280,000	1	$100,000
B	4	2	$320,000	$ 420,000	2	$ 50,000
C	10	7	$620,000	$ 860,000	3	$ 80,000
D	6	4	$260,000	$ 340,000	2	$ 40,000
E	4	3	$410,000	$ 570,000	1	$160,000
F	5	3	$180,000	$ 260,000	2	$ 40,000
G	7	4	$900,000	$1,020,000	3	$ 40,000
H	9	6	$200,000	$ 380,000	3	$ 60,000
I	7	5	$210,000	$ 270,000	2	$ 30,000
J	8	6	$430,000	$ 490,000	2	$ 30,000
K	4	3	$160,000	$ 200,000	1	$ 40,000
L	5	3	$250,000	$ 350,000	2	$ 50,000
M	2	1	$100,000	$ 200,000	1	$100,000
N	6	3	$330,000	$ 510,000	3	$ 60,000

- Restricciones de precedencia, se debe mantener el orden entre actividades

$$y_B \geq 0 + 2 - x_A$$
$$y_H \geq y_G + 7 - x_G$$
$$y_H \geq y_E + 4 - x_E$$

- Restricción de duración total del proyecto

$$y_{FIN} \leq 40$$

■

El modelo matemático resulta

$$\begin{aligned}\min\quad & 100x_A+50x_B+80x_C+40x_D+160x_E+40x_F+40x_G+60x_H\\ & +30x_I+30x_J+40x_K+50x_L+100x_M+60x_N\end{aligned}$$

$$\begin{aligned}\text{s.t.}\quad & y_B \geq 2-x_A\\ & y_C \geq y_B+4-x_B\\ & y_D \geq y_C+10-x_C\\ & y_E \geq y_C+10-x_C\\ & y_F \geq y_E+4-x_E\\ & y_G \geq y_D+6-x_D\\ & y_H \geq y_E+4-x_E\\ & y_H \geq y_G+7-x_G\\ & y_I \geq y_C+10-x_C\\ & y_J \geq y_F+5-x_F\\ & y_J \geq y_I+7-x_I\\ & y_K \geq y_J+8-x_J\\ & y_L \geq y_J+8-x_J\\ & y_M \geq y_H+9-x_H\\ & y_N \geq y_K+4-x_K\\ & y_N \geq y_L+5-x_L\\ & y_{\text{FIN}} \geq y_M+2-x_M\\ & y_{\text{FIN}} \geq y_N+6-x_N\end{aligned}$$

$$\begin{aligned}& 0\leq x_A\leq 1,\ 0\leq x_B\leq 2,\ 0\leq x_C\leq 3,\ 0\leq x_D\leq 2\\ & 0\leq x_E\leq 1,\ 0\leq x_F\leq 2,\ 0\leq x_G\leq 3,\ 0\leq x_H\leq 3\\ & 0\leq x_I\leq 2,\ 0\leq x_J\leq 2,\ 0\leq x_K\leq 1,\ 0\leq x_L\leq 2\\ & 0\leq x_M\leq 1,\ 0\leq x_N\leq 3\end{aligned}$$

$$\begin{aligned}& y_B,y_C,y_D,y_E,y_F,y_G,y_H,y_I,y_J,y_K,y_L,y_M,y_N\geq 0\\ & y_{\text{FIN}}\leq 40\end{aligned}$$

$$x_A,\ldots,x_N\in\mathbb{Z}_+,\quad y_B,\ldots,y_N,y_{\text{FIN}}\in\mathbb{Z}_+$$

Para resolverlo empleamos el Código 5.1.

```
%% T8. Crashing

clear all; close all; clc;

     %xA xB xC xD  xE xF xG xH xI xJ xK xL xM  xN yb yc yd ye yf yg yh
        yi yj yk yl ym yn yFIN
f = [100 50 80 40 160 40 40 60 30 30 40 50 100 60 0  0  0  0  0  0  0
    0  0  0  0  0  0  0];
intcon = 1:length(f);

 %xa xb xc xd xe xf xg xh xi xj xk xl xm xn yb yc yd ye yf yg yh yi yj
    yk yl ym yn yFIN
 A=[-1  0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0
    0  0  0  0  0  0;%B-A
```

```
11      % yB >=0+2-xA==>-2 >=-xA-yB  Each of these equations represents
            the upper row
12      0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0  0
            0  0  0  0  0  0;%C-B
13      % yC >= yB+4-xB==> -4 >= -xB+yB-yC
14      0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0
            0  0  0  0  0  0;%D-C
15      % yD >= yC+10-xC==> -10 >= -xC+yC-yD
16      0  0 -1  0  0  0  0  0  0  0  0  0  0  1  0 -1  0  0  0  0
            0  0  0  0  0  0;%E-C
17      % yE >= yC +10 -xC==> -10 >= -xC+yC-yE
18      0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0
            0  0  0  0  0  0;%F-E
19      % yF >= yE +4 -xE==> -4 >= -xE+yE-yF
20      0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0 -1  0  0
            0  0  0  0  0  0;%G-D
21      % yG >= yD +6 -xD ==> -6 >= -xD+yD-yG
22      0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0 -1  0
            0  0  0  0  0  0;%H-E
23      % yH >= yE + 4 -xE ==> -4 >= -xE+yE-yH
24      0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0
            0  0  0  0  0  0;%H-G
25      % yH >= yG + 7 -xG ==> -7 >= -xG+yG-yH
26      0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0 -1
            0  0  0  0  0  0;%I-C
27      % yI >= yC + 10 -xC ==> -10 >= -xC+yC-yI
28      0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
            -1  0  0  0  0  0;%J-F
29      % yJ >= yF + 5 -xF ==> -5 >= -xF+yF-yJ
30      0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  1
            -1  0  0  0  0  0;%J-I
31      % yJ >= yI + 7 -xI ==> -7 >= -xI+yI-yJ
32      0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0
            1 -1  0  0  0  0;%K-J
33      % yK >= yJ + 8 -xJ ==> -8 >= -xJ+yJ-yK
34      0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0
            1  0 -1  0  0  0;%L-J
35      % yL >= yJ + 8 -xJ ==> -8 >= -xJ+yJ-yL
36      0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0
            0  0  0 -1  0  0;%M-H
37      % yM >= yH + 9 -xH ==> -9 >= -xH+yH-yM
38      0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0
            0  1  0  0 -1  0;%N-K
39      % yN >= yK + 4 -xK ==> -4 >= -xK+yK-yN
40      0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0
            0  0  1  0 -1  0;%N-L
41      % yN >= yL + 5 -xL ==> -5 >= -xL+yL-yN
42      0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0
            0  0  0  1  0 -1;%Fin-M
43      % yFIN >= yM + 2 -xM ==> -2 >= -xM+yM-yFIN
44      0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0
            0  0  0  0  1 -1];%Fin-N
45      % yFIN >= yN + 6 -xN ==> -6 >= -xN+yN-yFIN
46
47    % B  C  D   E  F  G  HE HF  I  JF JI  K  L  M  NK NL YfM YfN
48  b=-[2; 4; 10; 10; 4; 6; 4; 7; 10; 5; 7; 8; 8; 9; 4; 5; 2; 6];
```

```
%This vector contains the durations of the non-crashed tasks of matrix
    A

Aeq = [];
beq = [];

lb = zeros(1,length(f));

%    xa xb xc xd xe xf xg xh xi xj xk xl xm xn  yb yc yd ye yf yg yh yi
    yj yk yl ym yn yFIN
ub =[1  2  3  2  1  2  3  3  2  2  1  2  1  3   inf.*ones(1,13) 40];

%Maximum crashed and regular end times for each task
% 6 16 22 20 25 29 38 23 33 37 38 40 44
[x val] = intlinprog(f,intcon,A,b,Aeq,beq,lb, ub)

cost = val + (180+320+620+260+410+180+900+200+210+430+160+250+100+330)
```

Código 5.1: PERT-CPM con crashing

Solución:
Se concluye que:

- Es posible acabar el proyecto en 40 semanas.
- El coste total ascendería a 4690000 dólares.
- Para ello se debe realizar un crashing sobre la
  - Tarea F de 2 semanas, y
  - Tarea J de 2 semanas.

5.7 Programación y control de los costos del proyecto. Evaluación de PERT/CPM

La gestión rigurosa de los costos en proyectos de ingeniería de telecomunicación resulta esencial para asegurar la viabilidad económica de las infraestructuras y servicios desplegados. La programación y control de costos permite anticipar desviaciones, optimizar el uso de recursos y garantizar que las decisiones técnicas se alineen con las restricciones presupuestarias. En entornos donde la planificación de redes, el despliegue de infraestructura o la operación de sistemas requieren inversiones significativas, disponer de modelos cuantitativos proporciona un soporte sólido a la toma de decisiones.



La programación y control de costos es fundamental en ingeniería de telecomunicación para proyectos como el despliegue de fibra óptica, la instalación de estaciones base, la planificación de enlaces de microondas o la gestión de ampliaciones de capacidad. Un modelo de optimización permite identificar cuellos de botella, priorizar inversiones y evaluar el impacto económico de adelantar hitos clave del proyecto. La formalización matemática proporciona una base sólida para justificar decisiones técnicas y presupuestarias, alineando la ejecución del proyecto con los objetivos estratégicos de la organización.

En proyectos grandes con muchas actividades, se utilizan los llamados paquetes de trabajo, que permiten unir actividades relacionadas en una sola, lo que ayuda a reducir la complejidad del problema. Después, todas estas actividades son controladas en intervalos regulares de tiempo, para observar las posibles desviaciones que pudieran ir ocurriendo y así poder corregirlas. Dependiendo de la complejidad del proyecto, este paso se debe llevar a cabo mediante programas específicos o mediante hojas de cálculo.

IV Programación no Lineal

6 Programación no lineal 93
6.1 Optimización convexa. Condiciones de primer y segundo orden.
6.2 Aplicaciones de muestra. Ilustración gráfica de problemas de programación no lineal
6.3 Optimización no restringida. Resolución con Matlab
6.4 Optimización no lineal con restricciones. Solución con Matlab
6.5 Condiciones de Karush-Kuhn-Tucker

6. Programación no lineal

En este capítulo introducimos problemas más generales que involucran funciones no lineales. Está basado en el [2, Cap. 12]

6.1 Optimización convexa. Condiciones de primer y segundo orden.

La optimización convexa desempeña un papel fundamental en la ingeniería debido a su tratabilidad matemática y su amplia aplicabilidad. Problemas como la reconstrucción de señales en imagen médica, el control óptimo de sistemas de administración de fármacos y los modelos de machine learning para herramientas de diagnóstico suelen basarse en formulaciones convexas para garantizar soluciones fiables y eficientes.

Los problemas convexos resultan especialmente atractivos porque cualquier mínimo local es también un mínimo global, y existen algoritmos potentes capaces de resolverlos de manera eficiente incluso en altas dimensiones.

Definición 6.1.1 — Conjunto convexo. Un conjunto $C \subseteq \mathbb{R}^n$ es *convexo* si, para cualesquiera dos puntos $x_1, x_2 \in C$, el segmento que los une queda completamente contenido en C:

$$\lambda x_1 + (1-\lambda)x_2 \in C \quad \text{para todo } \lambda \in [0,1]$$

Definición 6.1.2 — Función convexa. Una función $f : \mathbb{R}^n \to \mathbb{R}$ es *convexa* si su dominio es convexo y, para todo $x_1, x_2 \in \mathrm{dom}(f)$ y $\lambda \in [0,1]$, se cumple:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

■ **Ejemplo 6.1 — Función convexa.** La función $f(x) = x^2$ es una función convexa. Notemos que $\mathrm{dom}(f) = \mathbb{R}$, es trivialmente un conjunto convexo. Por otro lado, aplicamos la Definición 6.1.2, debemos probar que para todo $x_1, x_2 \in \mathbb{R}$, y $\lambda \in [0,1]$, se cumple

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &= (\lambda x_1 + (1-\lambda)x_2)^2 \\ &\leq \lambda x_1^2 + (1-\lambda)x_2^2 = \lambda f(x_1) + (1-\lambda) f(x_2) \end{aligned}$$

■

Para ello, notemos que

$$\begin{aligned}
\lambda x_1^2 + (1-\lambda)x_2^2 &\geq (\lambda x_1 + (1-\lambda)x_2)^2 \\
(1-\lambda)x_2^2 + \lambda x_1^2 &\geq \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1x_2 + (1-\lambda)^2 x_2^2 \\
\lambda x_1^2 &\geq \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1x_2 + \left[(1-\lambda)^2 - (1-\lambda)\right] x_2^2 \\
\lambda x_1^2 &\geq \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1x_2 + (1-\lambda)[(1-\lambda)-1]x_2 \\
\lambda x_1^2 &\geq \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1x_2 + (1-\lambda)(-\lambda)x_2^2 \\
0 &\geq \left(\lambda^2 - \lambda\right) x_1^2 + 2\lambda(1-\lambda)x_1x_2 + (1-\lambda)(-\lambda)x_2^2 \\
0 &\geq \lambda(\lambda-1)x_1^2 - 2\lambda(\lambda-1)x_1x_2 + \lambda(\lambda-1)x_2^2 \\
0 &\geq \lambda(\lambda-1)\left(x_1 - x_2\right)^2
\end{aligned}$$

que se cumple para $\lambda \in [0,1]$.

Observe en la Figura 6.1 que geométricamente, el gráfico de una función convexa entre does puntos se encuentra por debajo del segmento que une ambos puntos.

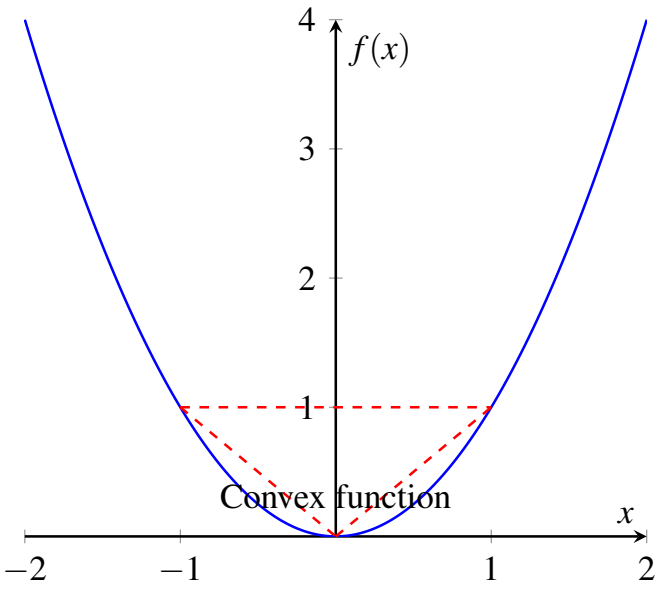


Figura 6.1: Gráfico de $f(x) = x^2$

Teorema 6.1.1 — Funciones Convexas y Dos Veces Diferenciables. Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función para la cual se cumple que $\nabla^2 f(\mathbf{x})$, la matriz Hessiana, es definida positiva para todo $\mathbf{x} \in \mathbb{R}^n$, entonces f es una función convexa.

R Este teorema ofrece una condición fuerte para determinar si una función es convexa o no. Esto es equivalente, en una variable, a pedir que la segunda derivada sea positiva. Se recomienda encarecidamente revisar la definición de matriz Hessiana y de matriz definida positiva en el Apéndice A.

■ **Ejemplo 6.2** La función $f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$ es convexa. Calculando la matriz hessiana de f,

$$\nabla^2 f(\mathbf{x}) = 2\mathrm{Id}_n$$

donde Id_n es la matriz identidad de tamaño $n \times n$. Como la matriz hessiana es definida positiva, aplicamos el Teorema 6.1.1 para concluir que f es una función convexa. ■

Definición 6.1.3 — Función afín. Se define $f : \mathbb{R}^n \to \mathbb{R}^p$ es una *función afín* si se puede escribir $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, donde $A \in \mathbb{R}^{n \times p}$ y $b \in \mathbb{R}^p$

R Una función afín es la suma de una función lineal $A\mathbf{x}$ y una constante $\mathbf{b}$.

Definición 6.1.4 — Máximo Local. Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función, y sea $\mathbf{x}_0 \in \mathbb{R}^n$ un punto en el dominio de f. Decimos que f tiene un *máximo local* en $\mathbf{x}_0$ si existe un entorno U de $\mathbf{x}_0$ tal que:

$$f(\mathbf{x}) \leq f(\mathbf{x}_0) \quad \text{for all} \quad \mathbf{x} \in U.$$

En otras palabras, $f(\mathbf{x}_0)$ es mayor o igual a $f(\mathbf{x})$ para todo $\mathbf{x}$ cerca $\mathbf{x}_0$.

Definición 6.1.5 — Mínimo Local. Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función, y sea $\mathbf{x}_0 \in \mathbb{R}^n$ un punto en el dominio de f. Decimos que f tiene un *mínimo local* en $\mathbf{x}_0$ si existe un entorno U de $\mathbf{x}_0$ tal que:

$$f(\mathbf{x}) \geq f(\mathbf{x}_0) \quad \text{for all} \quad \mathbf{x} \in U.$$

En otras palabras, $f(\mathbf{x}_0)$ es menor o igual que $f(\mathbf{x})$ para todos los $\mathbf{x}$ cercanos a $\mathbf{x}_0$.

R Un punto $\mathbf{x}_0$ donde f tiene un máximo o un mínimo local se denomina *punto crítico*. Estos puntos suelen aparecer donde el gradiente $\nabla f(\mathbf{x}_0)$ es cero, pero no todos los puntos críticos son extremos locales.

■ **Ejemplo 6.3** Considera la gráfica de la siguiente función, $f(x) = x^4 - 17x^2 + x$, en la Figura 6.2. Los mínimos locales se pueden encontrar en

x^*	z^*
-1.11	-2.06
2.9	-69.34

y un máximo local en $x_3^* = 0.03$ para $z_3^* = 0.01$. Obsérvese que $x_1^* = -1.11$ es un mínimo global, porque es el punto más bajo que alcanza la función; sin embargo, x_3^* no es un máximo global, ya que esta función crece hasta infinito cuando el valor de x aumenta. Nótese que esta función no es convexa, la gráfica entre los mínimos locales es *cóncava*.

■

Definición 6.1.6 — Problema de optimización convexo. Consideremos el problema de

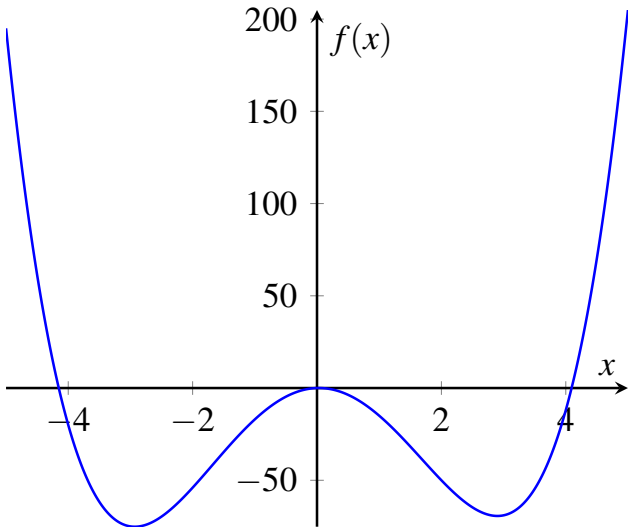


Figura 6.2: Función no convexa

optimización en forma estándar

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1,\dots,m \\ & h_j(\mathbf{x}) = 0, \quad j = 1,\dots,p \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{6.1}$$

decimos que (6.1) es un *problema de optimización convexa* si f y g_i son funciones convexas para $i = 1,\dots,m$ y h_j son funciones afines para $j = 1,\dots,p$. Esto hace que la región factible sea una región convexa.

■ **Ejemplo 6.4 — Convexo y no-convexo.** El problema

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1^4 + e^{x_2} \leq 0 \\ & x_1 + x_2 = 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

es un problema convexo. Obsérvese que f y $x_1^4 + e^{x_2}$ son funciones convexas, y $x_1 + x_2$ es una función afín.

Sin embargo el problema

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = x_1^2 + x_2^3 \\ \text{s.t.} \quad & x_1^4 + e^{x_2} \leq 0 \\ & (x_1 + x_2)^2 = 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

no es un problema convexo porque f no es convexa y la restricción de igualdad no es una función afín, aunque puede transformarse en una. ■

Teorema 6.1.2 — Optimalidad global en optimización convexa. Cualquier óptimo local de un problema convexo es también un óptimo global

6.1.1 Condiciones de primer y segundo orden sin restricciones

La programación no lineal sin restricciones se centra en encontrar el mínimo o el máximo de una función no lineal sin imponer restricciones sobre las variables. Esto resulta especialmente útil en aplicaciones donde la dinámica del sistema es inherentemente no lineal y compleja.

Definición 6.1.7 — Problema Diferenciable No Lineal sin Restricciones. Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función continuamente diferenciable. El Problema Diferenciable No Lineal sin Restricciones (DNLUP) es

$$\min_{\mathbf{x}} f(\mathbf{x})$$

donde $\mathbf{x} \in \mathbb{R}^n$.

La solución de este problema puede deducirse fácilmente a partir de las condiciones de primer y segundo orden estudiadas en Cálculo de varias variables.

Teorema 6.1.3 — Solutions of the DNLUP. Sea $\mathbf{x}^* \in \mathbb{R}^n$, y f definidos anteriormente. Una condición necesaria para que $\mathbf{x}^*$ sea un mínimo local de $f(x)$ es que el gradiente se anule:

$$\nabla f(\mathbf{x}^*) = 0$$

y la matriz hessiana$\nabla^2 f(\mathbf{x}^*)$ sea positiva definida. $\mathbf{x}^*$ será el mínimo global, si $z^* = f(\mathbf{x}^*) \leq f(\mathbf{x})$ para todo $\mathbf{x} \in \mathbb{R}^n$.

En otras palabras, para encontrar máximos o mínimos locales, primero localizamos los puntos críticos, donde el gradiente se anula.

Definición 6.1.8 — Punto crítico. Un punto $\mathbf{x}_0 \in \mathbb{R}^n$ se denomina *punto crítico* de una función diferenciable $f : \mathbb{R}^n \to \mathbb{R}$ si $\nabla f(\mathbf{x}_0) = \mathbf{0}$.

R Los puntos críticos son candidatos a extremos locales, pero no garantizan necesariamente la existencia de máximos o mínimos.

Entonces, debemos aplicar la prueba de la segunda derivada, que ayuda a clasificar los puntos críticos como máximos locales, mínimos o puntos de silla.

Teorema 6.1.4 — Test de la segunda derivada. Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función dos veces diferenciable y sea $\mathbf{x}_0$ un punto crítico. Consideremos la matriz hessiana $H_f(\mathbf{x}_0)$:

- Si $H_f(\mathbf{x}_0)$ es positiva definida, entonces $\mathbf{x}_0$ es un mínimo local.
- Si $H_f(\mathbf{x}_0)$ es definida negativa, entonces $\mathbf{x}_0$ es un máximo local.
- Si $H_f(\mathbf{x}_0)$ tiene autovalores positivos y negativos, entonces $\mathbf{x}_0$ es un punto de silla.
- Si $|H_f(\mathbf{x}_0)| = 0$ se está en presencia de un caso degenerado y no se puede deducir nada acerca de la función desde la matriz hessiana.

Teorema 6.1.5 — Second-Derivative Maximum-Minimum Test for Functions of Two Variables. Sea $f(x,y)$ una función de clase C^2 definida en un abierto U de $\mathbb{R}^2$. un punto (x_0, y_0) es un mínimo local de f si se cumplen las siguientes condiciones:

(i) $\frac{\partial f}{\partial x}(x_0, y_0) = \frac{\partial f}{\partial y}(x_0, y_0) = 0$

(ii) $\frac{\partial^2 f}{\partial x^2}(x_0,y_0) > 0$

(iii) $D = \left(\frac{\partial^2 f}{\partial x^2}\right)\left(\frac{\partial^2 f}{\partial y^2}\right) - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 > 0$ en (x_0,y_0)

(D es el determinante de la matriz hessiana.) Si en (ii) se tiene < 0 en vez de > 0 con la condición (iii), entonces se tendrá un máximo local. Si $D < 0$ (e.g., si $\frac{\partial^2 f}{\partial x^2}(x_0,y_0) = 0$ o $\frac{\partial^2 f}{\partial y^2}(x_0,y_0) = 0$, pero $\frac{\partial^2 f}{\partial x \partial y}(x_0,y_0) \neq 0$), entonces (x_0,y_0) es un punto de silla (no es máximo ni mínimo).

■ **Ejemplo 6.5** Encuentre los máximos relativos, mínimos y puntos de silla de la función

$$f(x,y) = \log\left(x^2+y^2+1\right)$$

Debemos primero localizar los puntos críticos de esta función; por lo tanto, de acuerdo con el Teorema 6.1.5, calculamos

$$\nabla f(x,y) = \frac{2x}{x^2+y^2+1}\mathbf{i} + \frac{2y}{x^2+y^2+1}\mathbf{j}$$

Así, $\nabla f(x,y) = \mathbf{0}$ si y solo si $(x,y) = (0,0)$, y por tanto el único punto crítico de f es $(0,0)$. Ahora debemos determinar si este es un máximo, un mínimo o un punto de silla. Las segundas derivadas parciales son

$$\frac{\partial^2 f}{\partial x^2} = \frac{2\left(x^2+y^2+1\right) - (2x)(2x)}{\left(x^2+y^2+1\right)^2}$$
$$\frac{\partial^2 f}{\partial y^2} = \frac{2\left(x^2+y^2+1\right) - (2y)(2y)}{\left(x^2+y^2+1\right)^2}$$

y

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{-2x(2y)}{\left(x^2+y^2+1\right)^2}$$

Por tanto,

$$\frac{\partial^2 f}{\partial x^2}(0,0) = 2 = \frac{\partial^2 f}{\partial y^2}(0,0) \quad \text{and} \quad \frac{\partial^2 f}{\partial x \partial y}(0,0) = 0$$

lo que implica

$$D = 2 \cdot 2 = 4 > 0$$

Como $\left(\partial^2 f/\partial x^2\right)(0,0) > 0$, concluimos que $(0,0)$ es un mínimo local. (¿Puedes demostrarlo únicamente a partir del hecho de que $\log t$ es una función creciente de $t > 0$?) ■

6.2 Aplicaciones de muestra. Ilustración gráfica de problemas de programación no lineal

En programación lineal todas sus funciones (objetivo y de restricción) son lineales. Este supuesto se cumple en el caso de muchos problemas prácticos, pero no siempre es así. Para tratar con situaciones en las que la programación lineal no puede modular, vamos a tratar con funciones no lineales.

Definición 6.2.1 Problema de programación no lineal De manera general, el problema de programación no lineal consiste en encontrar $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ para

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & g_i(\mathbf{x}) \leq b_i, \text{ para } i = 1, 2, \ldots, m, \\ & \mathbf{x} \geq 0, \end{array}$$

donde $f(\mathbf{x})$ y $g_i(\mathbf{x})$ son funciones dadas de n variables de decisión.

Dada las diferentes formas que pueden tomar las funciones $f(x)$ y $g(x)$ existen diferentes tipos de problemas PNL. Cada uno de estos problemas tiene una resolución específica, por lo que tratarlos individualmente es una tarea compleja. Además, existen ciertas consideraciones que se deben tener en cuenta a la hora de abordar este tipo de problemas.

■ **Ejemplo 6.6 — Wyndor Glass formulación no lineal.** Considere el modelo del Ejemplo 1.1. Si se sustituye la segunda restricción por una restricción no lineal, $9x_1^2 + 5x_2^2 \leq 216$, la solución óptima sigue siendo $(x_1^*, x_2^*) = (2, 6)$, vea Figura 6.3, o Geogebra.

$$\begin{array}{ll} \max\limits_{\mathbf{x}} & z = 3x_1 + 5x_2 \\ \text{s.a.} & x_1 \leq 4 \\ & 9x_1^2 + 5x_2^2 \leq 216 \\ & 3x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{array}$$

■

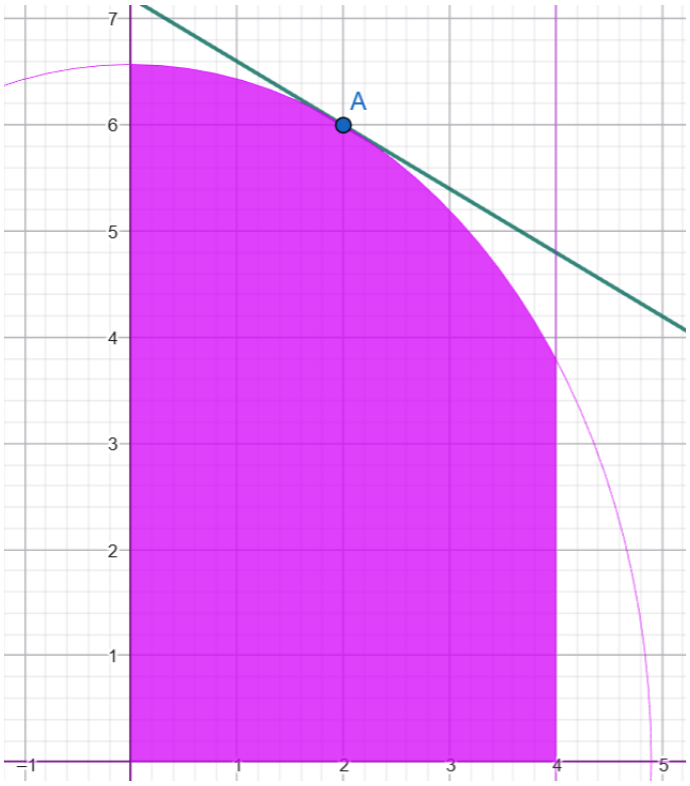


Figura 6.3: Modelo Ejemplo 1.1, con una restricción no lineal

Notemos que en el ejemplo anterior, la solución se encuentra sobre la frontera de la región factible, pero no es una **solución factible de vértice**.

■ **Ejemplo 6.7** Ahora supongamos que las restricciones lineales se conservan, pero la función objetivo cambia, para hacerse no lineal. Si

$$\begin{array}{ll} \max\limits_{\mathbf{x}} z & = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2 \\ \text{s.a.} & x_1 \leq 4 \\ & x_2 \leq 6 \\ & 3x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{array}$$

La solución óptima ahora resulta $(x_1^*, x_2^*) = \left(\frac{8}{3}, 5\right)$, con $z^* = 857$, vea Figura 6.4, o Geogebra. ■

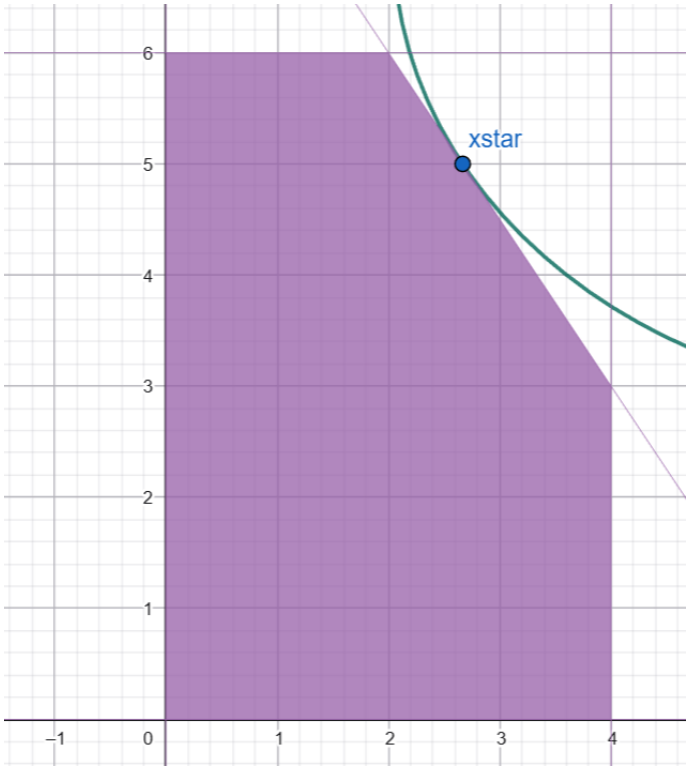


Figura 6.4: Modelo Ejemplo 1.1, con una función objetivo no lineal

Esta solución, nuevamente se encuentra sobre la frontera de la región factible, pero no es una **solución factible de vértice**.

Finalmente consideremos el siguiente caso

■ **Ejemplo 6.8** Conservamos las restricciones lineales, pero la función objetivo cambia a $z = 54x_1 - 9x_1^2 + 78x_2 - 13x_2^2$, para hacerse no lineal. Si

$$\begin{array}{ll} \max\limits_{\mathbf{x}} z & = 54x_1 - 9x_1^2 + 78x_2 - 13x_2^2 \\ \text{s.a.} & x_1 \leq 4 \\ & x_2 \leq 6 \\ & 3x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{array}$$

La solución óptima ahora resulta $(x_1^*, x_2^*) = (3, 3)$, con $z^* = 198$, vea Figura 6.5, o Geogebra. ■

Por lo tanto, es necesario que un algoritmo general para resolver problemas de este tipo tome en cuenta todas las soluciones en la región factible, y no sólo aquellas que están sobre la frontera.

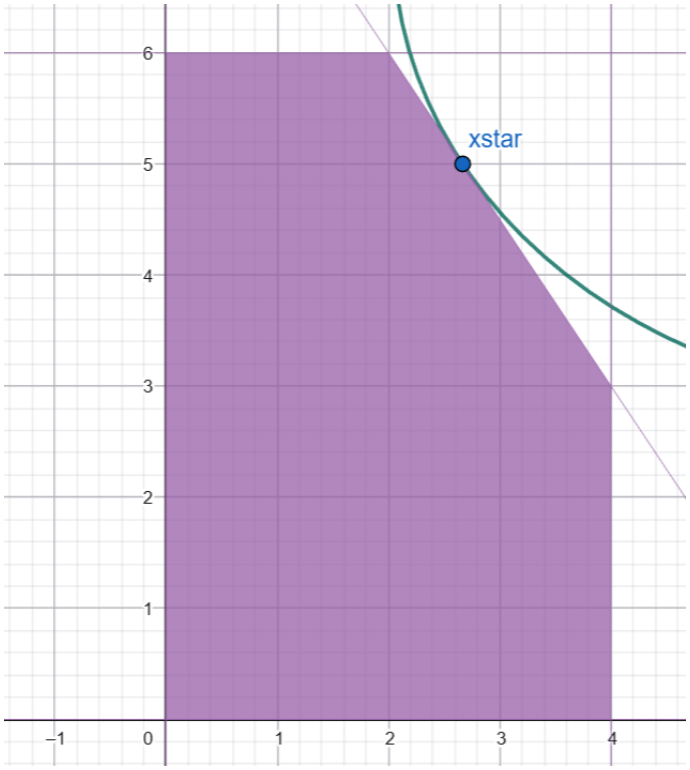


Figura 6.5: Modelo Ejemplo 1.1, con una función objetivo no lineal

Otra complicación que surge en programación no lineal es que un máximo local no necesariamente es un máximo global es decir, no es la solución óptima global, ver Figura 6.6.

Los algoritmos de Programación no Lineal, en general, no pueden distinguir entre máximos y mínimos locales y globales.

Los problemas PNL presentan muchas formas distintas. Al contrario que en el caso lineal, no hay un método general de resolución. Existen pues distintos métodos específicos para cada una de las familias, tales como

- Optimización no restringida: este tipo de problemas no tiene restricciones, por lo que su función objetivo es del tipo

Minimizar min o maximizar max $f(\mathbf{x})$, donde esta función es no lineal.

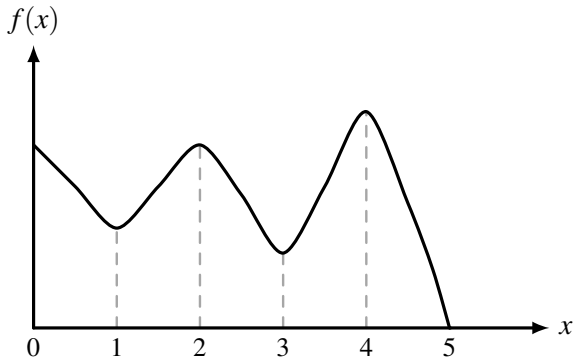


Figura 6.6: Función multimodal: varios óptimos locales

- Optimización restringida linealmente: en este tipo de problemas la única no linealidad aparece en la función objetivo, mientras las restricciones son lineales

  Minimizar min o maximizar max $f(\mathbf{x})$, donde esta función es no lineal, restringida a $g_i(\mathbf{x}) \leq b_i$, funciones lineales.

- **Programación cuadrática**: la función objetivo tiene forma cuadrática, es decir, hay polinomios de segundo grado, ya sea por variables al cuadrado o producto de variables.
- **Programación convexa**: aquí se deben cumplir dos condiciones:
  - $f(\mathbf{x})$ debe ser cóncava o convexa
  - Cada una de las funciones de restricción $g_i(\mathbf{x})$ han de ser convexas

  En este caso se puede asegurar que la solución es mínimo o máximo global.
- Programación no convexa
- **Programación separable**: es un tipo especial de programación convexa, donde además se ha de cumplir
  - $f(\mathbf{x})$ y $g_i(\mathbf{x})$ han de ser separables

  ¿Qué es una función separable? Aquella en la que cada uno de sus términos sólo contiene una variable

$$f(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j)$$

- **Programación no convexa**: este tipo de problemas recoge todos aquellos que no satisfacen los supuestos de la programación convexa. En este tipo de problemas no hay garantía de encontrar máximos globales.
- **Programación geométrica**: es un tipo especial de programación no convexa en el que las restricciones son de tipo polinomios.

$$\begin{aligned} g(\mathbf{x}) &= \sum_{i=1}^{N} c_i P_i(\mathbf{x}) \\ P_i(\mathbf{x}) &= x_1^{a_1 i}, x_2^{a_2 i}, \ldots, x_n^{a_n i} \end{aligned}$$

- **Programación fraccional**: en este tipo de problemas, la función objetivo tiene forma de fracción

$$\max / \min f(\mathbf{x}) = \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \tag{6.2}$$

6.3 Optimización no restringida. Resolución con Matlab

Para optimización no lineal sin restricciones, `Matlab` tiene las funciones

- `fminunc`, basada en el gradiente y
- `fminsearch`, no basada en el gradiente.

Para más información consultar Apéndice G. `fminunc` resuelve problemas del tipo

$$min_x f(x)$$

■ **Ejemplo 6.9** Minimizar la función $2x^2 - 1$ empleando `fminunc`, para encontrar un mínimo alrededor del punto $x_0 = 1$

```
1 fun = @(x)x^2-1; %anonimus function
2 x0 = 1; % starting point
3 fminunc(fun,x0);
```

Código 6.1: Ejemplo de uso de `fminunc`

La función se debe escribir en modo anónimo `@(x)` para declarar la variable y se debe proveer de un punto de inicio para el algoritmo.

La solución es $x^* = 0$ para $z^* = -1$. Ver Código 6.1 ■

■ **Ejemplo 6.10** Tenemos que instalar una nueva antena que permita dar cobertura mediante radio-enlace a cada una de las sedes de nuestra empresa. Estas se encuentran distribuidas tal y como aparece en la Figura 6.7.

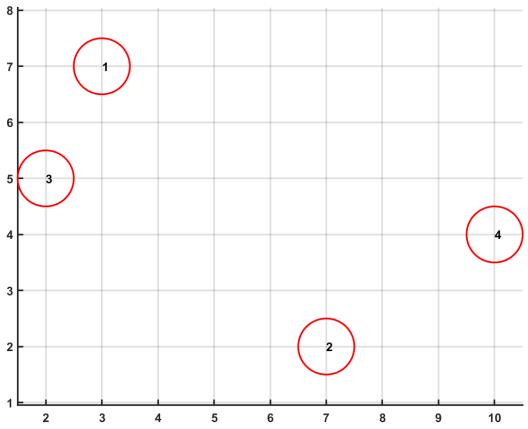


Figura 6.7: Mapa Ejemplo 6.10

Se desea que las perdidas debido a la propagación en espacio libre sea mínima. ¿En qué coordenadas deberemos instalar la antena? (asumir que las distancias están en km).

- Las perdidas debidas a la propagación en espacio libre tienen la siguiente forma

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2$$

- La longitud de onda no va a cambiar para ninguna de las sedes a las que tenemos que darle cobertura, nuestro problema se convierte en la minimización de las distancias entre nuestra antena y nuestras bases.
- Si definimos la posición de nuestra antena como (x_a, y_a) podemos escribir la distancia hasta cada una de nuestras sedes como $d_i = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2}$
- De tal manera que la función a minimizar es

$$z = \sum_{i=1}^{N} d_i; \quad \text{donde N es el número de sedes}$$

```
x0 = [1,1];
[x,fval] = fminunc(@fdist,x0);
function [d] = fdist(x)
pos_sedes = [3,7;
  7,1;
  2,5;
  10,4];
d = 0;
for i = 1:size(pos_sedes,1)
    d = d + sqrt((pos_sedes(i,1)-x(1))^2 +(pos_sedes(i,2)-x(2))^2);
end
end
```

Código 6.2: Ejemplo de uso de `fminunc`

Aplicando la función de *MATLAB*, ver Código 6.2, que hemos visto anteriormente para este problema obtenemos la siguiente solución:

- x = [4.8889,4.6389]
- fval = 14.4654

Es decir, nuestra antena debe colocarse en las coordenadas $(4.8889, 4.6389)$ y la suma de distancias a todas las estaciones es de 14.4654 km, ver Figura 6.8.

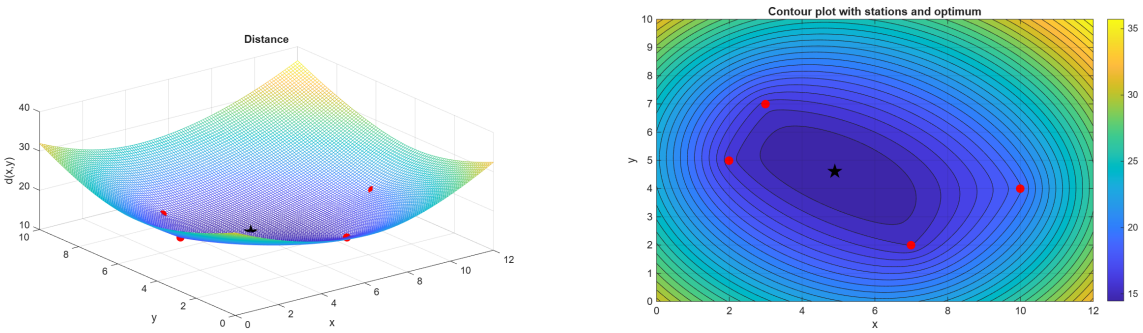


Figura 6.8: Solución Ejemplo 6.10. Izquierda, representación tridimensional del problema y su solución. Derecha: mapa de contornos.

■

6.4 Optimización no lineal con restricciones. Solución con Matlab

Para optimización no lineal con restricciones, emplearemos la función `fmincon` de matlab, basada en el gradiente. Para más información consultar Apéndice G.

■ **Ejemplo 6.11** Maximice la siguiente función

$$\begin{aligned} \max z \quad & = x_1 + x_2 \\ \text{s.a} \quad & x_1^2 + x_2^2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Aplicamos el Código 6.3

```
A=[];b=[];
Aeq=[];beq=[];
lb = [0,0];
ub = [];
x0 = [1,0];

nonlcon = @circlecon;
[x,fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon);

fun = @(x)-(x(1)+x(2)); %function
function [c,ceq] = circlecon(x) %declaring the non-linear constraints
c = (x(1))^2 + (x(2))^2-1;
ceq = [];
end
```

Código 6.3: Solución Ejemplo 6.11

Para obtener $x^* \approx (0.7071, 0.7071)$ for $z^* \approx 1.4142$. ■

■ **Ejemplo 6.12 — Ejemplo prototipo, uso de `fmincon`.** Tenemos que instalar una nueva antena que permita dar cobertura mediante radio-enlace a cada una de las sedes de nuestra empresa, ver Figura 6.7.

Se desea que las perdidas debido a la propagación en espacio libre sea mínima. ¿En qué coordenadas deberemos instalar la antena? (asumir que las distancias están en km).

Además queremos que ninguna de las sedes se encuentre a una distancia superior a 5 km ya que la banda de frecuencias que utilizamos está muy saturada y necesitamos que llegue un nivel de potencia determinado.

Solución: La longitud de onda no va a cambiar para ninguna de las sedes a las que tenemos que darle cobertura, nuestro problema se convierte en la minimización de las distancias entre nuestra antena y nuestras bases.

Si definimos la posición de nuestra antena como (x_a, y_a) podemos escribir la distancia hasta cada una de nuestras sedes como

$$d_i = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2}$$

De tal manera que la función a minimizar es

$$z = \sum_{i=1}^{N} d_i; \quad \text{donde N es el número de sedes}$$

$$st.$$

$$d_i < 5 \quad \text{i =1,2,3,4.}$$

Aplicando la función de Matlab *fmincon* que hemos visto anteriormente para este problema obtenemos la siguiente solución, ver Código 6.4.

```
A=[];b=[];
Aeq=[];beq=[];
lb = [0,0];
ub = [];
x0 = [1,1];
[x,fval] = fmincon(@fdist,x0,A,b,Aeq,beq,lb,ub,@nonlcon);

function [d] = fdist(x) %distance function
pos_sedes = [3,7;
  7,2;
  2,5;
  10,4];
d = 0;
for i = 1:size(pos_sedes,1)
    d = d + sqrt((pos_sedes(i,1)-x(1))^2 +(pos_sedes(i,2)-x(2))^2);
end
end
function [c,ceq] = nonlcon(x) %nonlinear constraint -- distance
pos_sedes = [3,7;
  7,2;
  2,5;
  10,4];
c = zeros(size(pos_sedes,1),1);
for i = 1:size(pos_sedes,1)
    c(i) = sqrt((pos_sedes(i,1)-x(1))^2 + ...
```

```
                (pos_sedes(i,2)-x(2))^2) - 5;
end
ceq = [];
end
```

Código 6.4: Solución Ejemplo 6.12

- x = [5.0321,4.5652]
- fval = 14.4675

Es decir, nuestra antena debe colocarse en las coordenadas $(5.0321, 4.5652)$ y la suma de distancias a todas las estaciones es de 14.4675 km, ver Figura 6.9

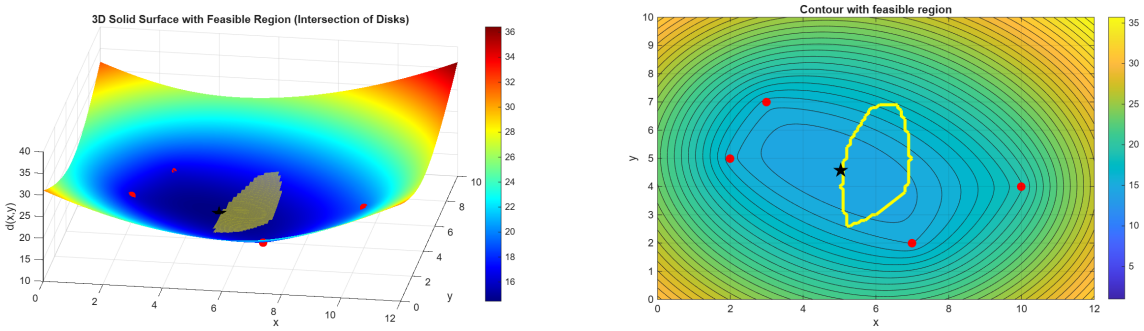


Figura 6.9: Solución Ejemplo 6.12. Izquierda, representación tridimensional del problema y su solución. Derecha: mapa de contornos. En amarillo la intersección de los discos inducida por las nuevas restricciones de distancia.

■

6.5 Condiciones de Karush-Kuhn-Tucker

Muchos problemas de optimización en ingeniería implican relaciones no lineales y están sujetos a restricciones derivadas de consideraciones físicas, biológicas o de seguridad. Por ejemplo, optimizar la dosis de un fármaco debe respetar los límites fisiológicos, y el diseño de una prótesis debe ajustarse a las restricciones anatómicas. Estos problemas se modelan mediante *programación no lineal con restricciones*, donde tanto la función objetivo como las restricciones pueden ser no lineales.

Definición 6.5.1 — Formulación canónica problema no lineal con restricciones. Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función objetivo, y sean $g_i(x) \leq 0$ y $h_j(x) = 0$ las restricciones de desigualdad y de igualdad, respectivamente. La forma general de un problema de programación no lineal con restricciones es:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

Surge una pregunta natural: ¿existe alguna generalización de las anteriores condiciones (las dadas en la Sección 6.1) de punto crítico y de segundo orden para encontrar las soluciones óptimas? La respuesta es sí. La solución óptima debe satisfacer las condiciones de Karush-Kuhn-Tucker (KKT), que generalizan el método de los multiplicadores de Lagrange para manejar restricciones de desigualdad.

Las condiciones de Karush-Kuhn-Tucker (KKT) son un conjunto de condiciones necesarias para que una solución en programación no lineal sea óptima, dadas ciertas condiciones de regulari-

dad. Estas condiciones generalizan el método de los multiplicadores de Lagrange para manejar restricciones de desigualdad y son fundamentales en problemas de optimización con restricciones.

Consideremos el siguiente problema de optimización:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{6.3}$$

Supongamos que f, g_i y h_j son diferenciables en la región factible $\mathscr{F} \subset \mathbb{R}^n$.

Definición 6.5.2 — Función de Lagrange. Definimos la función de Lagrange asociada a la ecuación (6.3) como

$$\mathscr{L}(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g(\mathbf{x}) + \sum_{j=1}^{p} \mu_j h(\mathbf{x})$$

donde $\lambda = (\lambda_1, \ldots, \lambda_m) \in \mathbf{R}^m$ y $\mu = (\mu_1, \ldots, \mu_p) \in \mathbf{R}^p$ son vectores.

Teorema 6.5.1 — KKT-conditions. El puntox^* es un mínimo local **si y solo si**, existen multiplicadores de Lagrange $\lambda_i^* \geq 0$ and $\mu_j^* \in \mathbb{R}$ tales que:

(Condición estacionaria)	$\nabla \mathscr{L} = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) = 0$
(Condición de factibilidad primal)	$\begin{cases} g_i(x^*) \leq 0, & i = 1, \ldots, m \\ h_j(x^*) = 0, & j = 1, \ldots, p \end{cases}$
(Condición de positividad/negatividad)	$\lambda_i^* \geq 0 (\text{mínimos}), \quad i = 1, \ldots, m$ $\lambda_i^* \leq 0 (\text{máximos}), \quad i = 1, \ldots, m$
(Condición de holgura)	$\lambda_i^* g_i(x^*) = 0, \quad i = 1, \ldots, m$

Si el problema es de maximización, como el único cambio es en los multiplicadores, estos serán un indicio para detectar la naturaleza del punto.



- Este mínimo local de x^* coincide con el mínimo $\ell^* = (x_1, \ldots, x_n, \lambda_1, \lambda_m, \mu_1, \ldots, \mu_p)$ de la función Lagrangiana $\mathscr{L}$.
- Las condiciones KKT transforman el proceso de optimización en una búsqueda de raíces de sistemas de ecuaciones.
- Si $p = 0$, es decir, no hay restricciones de igualdad, este resultado se reduce al caso habitual de los multiplicadores de Lagrange.
- La condición de estacionariedad se refiere a la condición de primer orden que hace que el gradiente de una función sea igual a cero.
- La factibilidad primal obliga a que la solución pertenezca a la región factible.
- La factibilidad dual fija el signo de los λ para que sea positivo/negativo según la naturaleza de mínimo/máximo.
- La holgura complementaria fuerza una de las siguientes condiciones: $\lambda_i = 0$ o la restricción g_i está activa en la solución dada.

■ **Ejemplo 6.13 — Resolviendo un problema empleando KKT.** Encuentre el máximo y el mínimo del siguiente problema

$$\begin{aligned} f(x_1,x_2) &= \log(x_1+1)+x_2 \\ \text{s.t.} \quad 2x_1+x_2 &\leq 3 \\ x_1,x_2 &\geq 0 \end{aligned}$$

Solución: Primero, escribamos el problema en la formulación canónica (6.1), que consiste en considerar un problema de minimización y expresar todas las desigualdades en la versión $\leq$, lo que produce

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(x_1,x_2) = \log(x_1+1)+x_2 \\ \text{s.t.} \quad & 2x_1+x_2-3 \leq 0 \\ & -x_1 \leq 0 \\ & -x_2 \leq 0 \end{aligned}$$

Luego se tiene

$$\begin{aligned} f(x_1,x_2) &= \log(x_1+1)+x_2 \\ g_1(x_1,x_2) &: 2x_1+x_2-3 \leq 0 \\ g_2(x_1,x_2) &: -x_1 \leq 0 \\ g_3(x_1,x_2) &: -x_2 \leq 0 \end{aligned}$$

con

$$\begin{aligned} \nabla f(x_1,x_2) &= \left(\frac{1}{x_1+1},1\right) \\ \nabla g_1(x_1,x_2) &= (2,1) \\ \nabla g_2(x_1,x_2) &= (-1,0) \\ \nabla g_3(x_1,x_2) &= (0,-1) \end{aligned}$$

Luego, $m=3$, y $\lambda=(\lambda,\nu_1,\nu_2)$, por simplicidad. Dado que el valor de λ_2 y λ_3 está ligado a los valores de x_1 y x_2, respectivamente, los renombramos como $\lambda_2=\nu_1$ y $\lambda_3=\nu_2$. Por tanto, las condiciones KKT producen

Stationarity 1:	$\frac{1}{x_1+1}+2\lambda-\nu_1$	$=0$
Stationarity 2:	$1+\lambda-\nu_2$	$=0$
Primal Feasibility 1:	$2x_1+x_2-3$	≤ 0
Primal Feasibility 2:	x_1,x_2	≥ 0
Dual Feasibility:	λ,ν_2,ν_2	≥ 0
Complementary slackness 1:	$\lambda(2x_1+x_2-3)$	$=0$
Complementary slackness 2:	$-\nu_1 x_1$	$=0$
Complementary slackness 3:	$-\nu_2 x_2$	$=0$

La solución óptima debe satisfacer estas condiciones. Nótese que, a partir de la Estacionariedad 2, se tiene que $1+\lambda=\nu_2$, lo cual implica por factibilidad dual que $\nu_2>0$, y por tanto, la holgura complementaria 2 implica que $x_2=0$, para todos los posibles valores de λ.

Consideremos $\lambda \neq 0$

Si $\lambda_1 \neq 0$, entonces, a partir de la holgura complementaria 1, tenemos $x_1 = \frac{3}{2}$ y, de la holgura complementaria 1, obtenemos $\nu_1 = 0$. Sustituyendo en la Estacionariedad 1

$$\frac{1}{x_1+1} + 2\lambda - \nu_1 = 0$$
$$2\lambda = \nu_1 - \frac{2}{5}$$

lo que implica que $\nu_1 > \frac{2}{5}$, dado que λ es positivo; por tanto, tenemos una **contradicción**. Por consiguiente, este caso no es válido y $\lambda = 0$.

Hasta ahora, el vector solución es $\ell^* = (x_1, 0, 0, \nu_1, 1)$.

Consideremos ahora, estacionaridad 1

$$\frac{1}{x_1+1} - \nu_1 = 0$$
$$\nu_1 = \frac{1}{x_1+1}$$

lo cual, combinado con la holgura complementaria 2, devuelve

$$0 = \nu_1 x_1 = \frac{1}{x_1+1} \cdot x_1$$

lo que implica $x_1 = 0$.

Por tanto, el mínimo de la Lagrangiana produce $\ell^* = (0,0,0,1,1)$, y, en consecuencia, el mínimo de este problema es $x^* = (0,0)$ para $z^* = 0$.

Para encontrar el máximo, podemos cambiar estacionariedad 1 y 2 a

$$\begin{aligned} \text{Estacionariedad 1':} \quad & -\frac{1}{x_1+1} + 2\lambda - \nu_1 = 0 \\ \text{Estacionariedad 2':} \quad & -1 + \lambda - \nu_2 = 0 \end{aligned}$$

manteniendo el resto de las condiciones igual. Nótese ahora que estacionariedad 2' ($\lambda = \nu_2 + 1$), y la factibilidad dual implican $\lambda \geq 1$, lo cual significa automáticamente (vía holgura complementaria 1) que $2x_1 + x_2 = 3$, lo que hace que la primera desigualdad sea activa.

Consideremos el caso: $\nu_2 \neq 0$, luego $x_2 = 0$, y, al introducir este valor en la estacionariedad 1', se obtiene

$$\lambda = \frac{\nu_1}{2} + \frac{1}{5} \geq 1$$

porque $\lambda \geq 1$. Por tanto, $\nu_1 \geq \frac{8}{5}$ forzando, vía holgura complementaria 3, que $x_1 = 0$, lo cual es una **contradicción**.

Por lo tanto, debemos considerar el **caso** $\nu_2 = 0$. Este caso fuerza $\lambda = 1$, así que, a partir de la estacionariedad 1', tenemos

$$1 + \frac{1}{x_1+1} = \nu_1$$

lo que obliga a que $\nu_1 \geq 1$, porque $x_1 \geq 0$. Si $\nu_1 \neq 0$, entonces, por holgura complementaria 2, $x_1 = 0$, y a partir de la condición

$$2x_1 + x_2 = 3$$

se deduce $x_2 = 3$.

Por tanto, tenemos $\ell^* = (0,3,1,2,0)$, y el máximo del problema es $x^*_{max} = (0,3)$ para $z^*_{max} = 3$.

■

Tabla 6.1: Puntos crítcos Ejemplo 6.14

x	y	λ_1	λ_2
0	0	0	0
1	0	-1/2	0
-1	0	0	-1/2
-3	0	-3/2	0
3	0	0	-3/2
0	$\sqrt{3}$	1/2	1/2
0	$-\sqrt{3}$	1/2	1/2
$-1/2$	$-\sqrt{15}/2$	1	0
$-1/2$	$\sqrt{15}/2$	1	0
1/2	$-\sqrt{15}/2$	0	1
1/2	$\sqrt{15}/2$	0	1

6.5.1 Solución con Matlab

- Para poder asegurar que un punto cumple las condiciones necesarias de optimalidad es obligatorio que este punto cumpla alguna de las hipótesis de cualificación de restricciones.
- Demostrar estas hipótesis se sale fuera del alcance de la asignatura.
- Aplicaremos la siguiente: Un punto cumple las KKT si el conjunto del gradiente de las restricciones activas es linealmente independiente.

■ **Ejemplo 6.14 — Condiciones de KKT en matlab.** Dado el siguiente problema de optimización, encuentre todos los puntos críticos y clasifíquelos de acuerdo a las condiciones KKT de primer orden. Compruebe los puntos factibles obtenidos.

$$\begin{aligned} \min / \max_{(x,y,z)} \quad & x^2 - y^2 \\ \text{s.t.} \quad & x^2 + 2x + 1 + y^2 \leq 4 \\ & x^2 - 2x + 1 + y^2 \leq 4 \end{aligned}$$

Solución:

1. Obtener el lagrangiano

$$\mathscr{L} \quad = \quad f + \lambda_1 g_1 + \lambda_2 g_2$$

2. Aplicar la condición estacionaria. Calcular el gradiente del Lagrangiano e igualarlo a cero ($\nabla \mathscr{L} = 0$). En Matlab puedes utilizar el comando `gradient`.
3. Aplicar la condición de holgura. Igualar a cero las restricciones de desigualdad multiplicadas por λ_i, llamados multiplicadores de Lagrange.
4. Resolver el sistema formado por las cuatro ecuaciones anteriores. En Matlab puedes utilizar el comando `solve`.
5. Los puntos críticos (x, y) obtenidos son, ver Tabla 6.1.
6. Las condiciones KKT de primer grado son las siguientes
   - **Condición estacionaria y de holgura**: se cumplen a la hora de resolver el sistema de ecuaciones
   - **Condición de factibilidad**: Comprobar que se cumplen las restricciones

$$\begin{aligned} h_i(x^*) &= 0 \qquad i = 1, \ldots, m \\ g_j(x^*) &= 0 \qquad j = 1, \ldots, p \end{aligned}$$

Tabla 6.2: Clasificación de puntos críticos Ejemplo 6.14

x	y	λ_1	λ_2	Factible	Máx/Mín	Valor
0	0	0	0	Sí	¿?	0
1	0	-1/2	0	Sí	Máx	1
-1	0	0	-1/2	Sí	Máx	1
-3	0	-3/2	0	No	-	-
3	0	0	-3/2	No	-	-
0	$\sqrt{3}$	1/2	1/2	Sí	Mín	-3
0	$-\sqrt{3}$	1/2	1/2	Sí	Mín	-3
$-1/2$	$-\sqrt{15}/2$	1	0	No	-	-
$-1/2$	$\sqrt{15}/2$	1	0	No	-	-
1/2	$-\sqrt{15}/2$	0	1	No	-	-
1/2	$\sqrt{15}/2$	0	1	No	-	-

- **Condición de positividad/negatividad**: El signo de λ_i indica el tipo de punto: $\lambda_i \leq 0$ indica máximo, $\lambda_i \geq 0$ indica mínimo, ver Tabla 6.2.

7. Comprobar los puntos factibles. Utilizamos `subs` para sustituir los puntos obtenidos en las restricciones y comprobaremos si se cumplen `subs(g1, [x,y], [sol.x, sol.y])`
8. Comprobar los valores obtenidos para los puntos factibles. Utilizaremos la función `fmincon`, ya que vamos a optimizar una función no lineal con restricciones para cada punto, ver Código 6.5.

```
x0 =[sol.x,sol.y];
A=[]; b=[]; Aeq = []; beq = []; lb =[]; ub = [];
fun = @(x) x(1)^2 - x(2)^2;
[x,fval]=fmincon(fun, x0, A,b,Aeq,beq,lb,ub,@nonlcon);

function[c, ceq] = nonlcon(x)
    c = [ x(1)^2 + 2*x(2) - 3 + x (2)^2 , x(1)^2 - 2 *x(2) - 3 + x
        (2) ^2 ];
    ceq = [];
end
```

Código 6.5: Aplicación de `fmincon`

El flujo de trabajo completo de este ejemplo se puede reproducir a partir del Código 6.6.

```
syms  x y m1 m2
Z = x^2-y^2;
g1 = x^2+2*x-3+y^2;
g2 = x^2-2*x-3+y^2;

L = Z + m1*g1 + m2*g2;
nLx = gradient(L,x);
nLy = gradient(L,y);
eq1 = nLx == 0;
eq2 = nLy == 0;
eq3 = m1*g1 == 0;
eq4 = m2*g2 == 0;
sist = [eq1;eq2;eq3;eq4];
sol = solve(sist);

```

```
for i = 1:11
    Fac(i) = subs(g1,[x,y],[double(sol.x(i)),double(sol.y(i))])<=0 &&
        subs(g2,[x,y],[double(sol.x(i)),double(sol.y(i))])<=0;
    mx(i) = sol.m1(i)<=0 && sol.m2(i)<=0;
    mn(i) = sol.m1(i)>=0 && sol.m2(i)>=0;
end

for i = 1:11
    x0 = double([sol.x(i),sol.y(i)]);
    if Fac(i) && mn(i)
        disp('min')
        i, x0
        fun = @(x) x(1)^2-x(2)^2;
        [x,fval] = fmincon(fun,x0,[],[],[],[],[],[],@nonlcon)
    end
    if Fac(i) && mx(i)
        disp('max')
        i, x0
        fun = @(x) -x(1)^2+x(2)^2;
        [x,fval] = fmincon(fun,x0,[],[],[],[],[],[],@nonlcon)
    end
end
%

function [c,ceq] = nonlcon(x)
c = [x(1)^2+2*x(1)-3+x(2)^2, x(1)^2-2*x(1)-3+x(2)^2];
ceq = [];
end
```

Código 6.6: Código para Ejemplo 6.14

Se debe tener en cuenta que

- Las condiciones KKT son expresiones de primer orden que pueden dar lugar a máximos, mínimos o puntos de silla.
- Si aplicamos las condiciones KKT de segundo orden podemos, conocer la verdadera naturaleza de los puntos obtenidos. Además, este tipo de condiciones son suficientes para la optimalidad.
- Estas condiciones vienen marcadas por condiciones sobre el Hessiano:

$$\mathbf{z}^{\mathbf{T}} \mathscr{H}_{\mathbf{x}^{*2}} \mathscr{L}(\mathbf{x}^*, \lambda^*, \mu^*)\mathbf{z} > 0$$
$$\forall z \begin{bmatrix} \nabla \mathbf{h}(\mathbf{x}^*) \\ \nabla \mathbf{g}_{act}(\mathbf{x}^*) \end{bmatrix} \mathbf{z} = 0$$

■

■ **Ejemplo 6.15** Dado el siguiente problema de optimización, encuentre todos los puntos críticos y clasifíquelos de acuerdo a las condiciones KKT de primer orden. Compruebe los puntos factibles obtenidos.

$$\begin{aligned} \min / \max_{(x,y,z)} \quad & x^2 + y^2 \\ \text{s.t.} \quad & x^2 + 2x - 1 + y^2 \leq 4 \\ & x^2 - 2x + 1 + y^2 \leq 3 \end{aligned}$$

Los puntos críticos obtenidos están en la Tabla 6.3. ■

Tabla 6.3: Clasificación puntos críticos Ejemplo 6.15

x	y	λ_1	λ_2	Valor	Mín/Máx
0	0	0	0	0	Mín/Máx
$1-\sqrt{3}$	0	0	$\sqrt{3}/3-1$	0.5359	Máx
$-1-\sqrt{6}$	0	$-\sqrt{6}/6-1$	0	11.899	-
3/4	$-\sqrt{47}/4$	-1/2	-1/2	12.3125	Máx
3/4	$\sqrt{47}/4$	-1/2	-1/2	12.3125	Máx
$1+\sqrt{3}$	0	$-1/2$	$-\sqrt{3}/3-1$	7.4641	
$-1+\sqrt{6}$	0	$\sqrt{6}/6-1$	0	2.1010	Máx

6.5.2 Software: Windows solver

Para resolver un problema no lineal en el solver de Windows solo seleccione la opción `Problem: non-linear` y escriba el problema teniendo en cuenta que las potencias deben escribirse usando `**`. Véase el Apéndice B para más información.

V

Teoría de Colas

7 Teoría de colas 117

7.1 Introducción
7.2 Estructura básica de los modelos de colas. Ejemplo prototipo
7.3 Ejemplos de sistemas de colas reales. Aplicaciones
7.4 Proceso de nacimiento y muerte
7.5 Modelos de colas basados en procesos de nacimiento y muerte. Ejemplos

8 Resolución analítica de modelos de colas 123

8.1 Modelos de colas con disciplina de prioridades
8.2 Papel de las distribuciones exponenciales
8.3 Nomenclatura a utilizar para los sistemas de colas
8.4 Sistemas de un solo canal
8.5 Sistemas de varios canales en paralelo
8.6 Sistemas de población finita
8.7 Redes de colas
8.8 Modelos de colas con distribuciones no exponenciales

7. Teoría de colas

Para este capítulo se emplea como referencia [3]. Para más detalles acerca de la formulación de los modelos esta referencia es sumamente útil.

7.1 Introducción

La teoría de colas estudia el comportamiento de sistemas donde existe un conjunto limitado de recursos para atender las peticiones generadas por los usuarios, de tal manera que cuando un usuario envía una tarea al sistema, ésta podrá tener que esperar para ser atendida por algún recurso, o, incluso, podrá ser rechazada si el sistema no tiene capacidad suficiente para almacenarla en espera de ser atendida. El estudio de estos sistemas implicará el modelado no sólo del sistema en sí, sino también del comportamiento aleatorio del tráfico ofrecido por los usuarios al sistema. Este tráfico ofrecido por los usuarios se modela mediante dos procesos estocásticos: proceso de llegadas de las tareas y proceso de servicio demandado por las mismas, usualmente considerados independientes entre sí.

El conjunto más amplio de resultados analíticos en teoría de colas se encuentran para los procesos de llegada de Poisson y proceso de servicio demandado aleatorio puro modelado por variables aleatorias con distribución exponencial, que caracterizan sistemas de espera en los que tanto las llegadas de tareas de usuarios como la demanda que realizan son completamente aleatorias, en el sentido de que la evolución del sistema sólo depende de su condición presente y no de su pasado. Si bien en la práctica surgen otros modelos de colas con caracterización del tráfico mediante otros tipos de procesos, su análisis matemático es más complejo y no se dispone, en la actualidad, de una solución general para todos los casos posibles de interés.

Para el estudio de sistemas modelados mediante modelos de colas cuya solución analítica se desconoce, o bien es muy difícil de obtener, se suele recurrir a técnicas de simulación que, mediante el empleo de programas de computación, permiten obtener estimaciones de los parámetros de interés del comportamiento del sistema.

El objetivo de este capítulo es el estudio de esta disciplina enmarcada dentro de la investigación de operaciones por lo que en los siguientes apartados nos centraremos en el estudio de un conjunto de modelos de colas con solución analítica conocida.

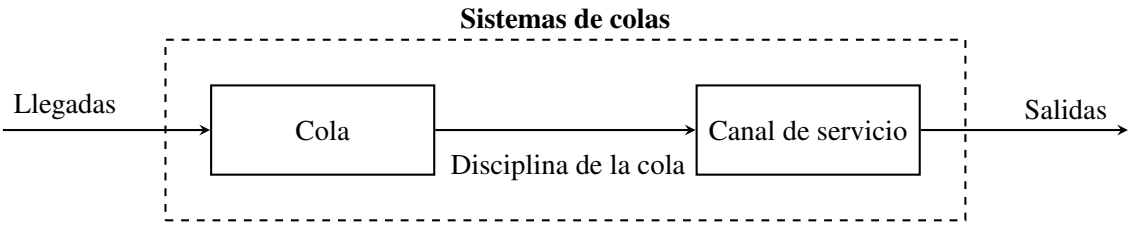


Figura 7.1: Estructura genérica de una cola

La teoría de colas estudia aquellos sistemas en los que existe contienda por los recursos, lo que hace que en un momento dado pueda haber más demanda de recursos que recursos disponibles, por lo que algunas de estas solicitudes de servicio deberán esperar a ser atendidas, formándose así una cola de acceso de las tareas a los recursos. e incluso podrán ser rechazadas por falta de capacidad de almacenamiento.

La mayoría de resultados de la teoría de colas se obtienen suponiendo que los sistemas estudiados son conservativos. Un sistema de espera es conservativo si los recursos no están desocupados cuando hay tareas esperando a ser servidas, y si la demanda de servicio de dichas tareas no depende del orden en que son servidas.

En los modelos de colas existentes, las tareas generadas por los usuarios de una población llegan al sistema en un instante de tiempo aleatorio. Estas tareas demandan una cantidad de servicio también aleatoria, y se sirven una a una o por grupos. El orden en el que se sirven las tareas que ocupan la cola del sistema lo determina una disciplina de gestión de cola.

En los siguientes apartados se describen detalladamente los elementos necesarios para la caracterización de un modelo de colas cualquiera, tanto el modelado del tráfico ofrecido por los usuarios, como el modelado del sistema encargado de aceptar y atender sus peticiones de servicio.

Importante Todos los resultados obtenidos en este curso parten de la suposición de que nuestros sistemas son **conservativos**, es decir, no se puede dar el caso de servidores inactivos cuando hay tareas por realizar.

7.2 Estructura básica de los modelos de colas. Ejemplo prototipo

La **Teoría de Colas** estudia el comportamiento de los sistemas donde existe un conjunto limitado de recursos para atender las peticiones generadas por los usuarios.

Un modelo genérico de colas puede representarse como la Figura 7.1. El principio básico de funcionamiento es el siguiente: un cliente **llega** a través de *Llegadas*, **ingresa** o no al sistema, representado por líneas discontinuas. una vez dentro del sistema, respeta una disciplina de cola, para llegar al canal de servicio. Una vez atendido, **egresa** o sale del sistema.

Terminología 7.1 — Modelos de colas. Los modelos de colas cuentan con varias partes básicas que hay que tener en cuenta, siendo estas:

- **Población**: La población representa el conjunto de clientes potenciales que pueden requerir el servicio. Puede ser finita, cuando el número de posibles clientes está limitado (por ejemplo, máquinas en una fábrica), o infinita, cuando no existe un límite práctico al número de clientes que pueden llegar al sistema, como ocurre en muchos servicios abiertos al público.
- **Llegadas**: Las llegadas describen la forma en que los clientes acceden al sistema. Se caracterizan por la frecuencia y regularidad con que ocurren y suelen modelarse mediante procesos aleatorios, siendo el proceso de Poisson uno de los más utilizados en teoría de colas.

- **Impaciencia**: La impaciencia se refiere al comportamiento de los clientes cuando el tiempo

Tabla 7.1: Ejemplos de sistemas de colas y sus elementos

Sistemas	Ejemplo	Clientes	Canales
Transporte	Puertos	Barcos	Muelles
	Aeropuertos	Aviones	Pistas
	Autopistas	Coches	Casillas de peaje
	Terminales	Buses	Plataforma
	Carga y descarga	Camiones	Islas
Computación	Procesamiento	Trabajos	CPU
	Impresión	Trabajos	Impresoras
Telefonía	Centrales	Llamados	Líneas
Producción	Elaboración	Productos	Máquinas
	Reparación	Solicitudes	Operarios
	Control de calidad	Productos	Inspectores
Atención al público	Bancos	Personas	Cajeros
	Supermercados	Personas	Cajeros
	Estaciones de servicio	Coches	Surtidores
	Negocios	Personas	Empleados
	Atención médicos	Pacientes	Ambulancias
	Alquiler de coches	Personas	Empleados
Administración	Dpto. compras	Solicitudes	Compradores
	Juzgados	Causas	Jueces

de espera es excesivo. Algunos clientes pueden abandonar la cola antes de ser atendidos (abandono) o decidir no entrar al sistema al observar una cola demasiado larga (rechazo).

- **Capacidad**: La capacidad del sistema indica el número máximo de clientes que pueden estar simultáneamente en la cola y en servicio. Cuando esta capacidad es limitada, los clientes que llegan al sistema lleno no pueden ingresar y se pierden.
- **Modo de llegadas**: El modo de llegadas especifica si los clientes llegan individualmente o en lotes, así como si las llegadas dependen o no del estado del sistema. Este aspecto influye directamente en la congestión y el desempeño del sistema.
- **Disciplina de cola**: La disciplina de cola define el orden en que los clientes son atendidos. La más común es primero en llegar, primero en ser atendido, aunque también existen otras como prioridad, atención aleatoria o último en llegar, primero en ser atendido.
- **Duración del servicio**: La duración del servicio corresponde al tiempo necesario para atender a un cliente. Este tiempo puede ser constante o aleatorio y se modela mediante distintas distribuciones de probabilidad, dependiendo de la naturaleza del sistema.
- **Modo de atención**: El modo de atención describe la forma en que se presta el servicio, incluyendo el número de servidores disponibles y si estos trabajan de manera independiente o coordinada. Puede tratarse de atención individual, en paralelo o en etapas sucesivas.
- **Ingresos y salidas de clientes**: Este aspecto considera el flujo de clientes dentro del sistema, desde su ingreso hasta su salida tras recibir el servicio o abandonar la cola. El análisis de estos flujos permite evaluar el desempeño del sistema en términos de tiempos de espera, longitud de cola y nivel de servicio.

7.3 Ejemplos de sistemas de colas reales. Aplicaciones

Algunos ejemplos de sistemas de colas pueden consultarse en la Tabla 7.1.

7.4 Proceso de nacimiento y muerte

El **proceso de nacimiento y muerte** es un tipo particular de proceso estocástico en tiempo continuo que modela la evolución del número de elementos en un sistema mediante transiciones entre estados consecutivos. Este tipo de proceso constituye la base matemática de muchos modelos clásicos de teoría de colas.

Sea $\{N(t), t \geq 0\}$ el número de clientes en el sistema en el instante t. El proceso es de nacimiento y muerte si sólo son posibles las siguientes transiciones:

$$n \longrightarrow n+1 \quad \text{(nacimiento)}$$

$$n \longrightarrow n-1 \quad \text{(muerte)}$$

donde:

- La tasa de nacimiento en el estado n es λ_n.
- La tasa de muerte en el estado n es μ_n.

Estas tasas representan, respectivamente, la llegada de nuevos clientes al sistema y la finalización del servicio.

Este proceso es una **cadena de Markov en tiempo continuo**, ya que el futuro del sistema depende únicamente del estado actual y no de la historia previa.

7.4.1 Ecuaciones de equilibrio

En régimen estacionario, i.e., bajo estabilización de las variables del sistema; las probabilidades de que hayan n clientes en el sistema, $p(n) = \lim_{t\to\infty} \mathbb{P}(N(t) = n)$ satisfacen las ecuaciones de equilibrio:

$$\lambda_{n-1} p(n-1) + \mu_{n+1} p(n) = (\lambda_n + \mu_n) p(n)$$

para $n \geq 1$, junto con:

$$\mu_1 p(1) = \lambda_0 p(0).$$

A partir de estas relaciones se obtiene de forma recursiva:

$$p(n) = \left(\prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}} \right) p(0).$$

El valor de $p(0)$ se determina mediante la condición de normalización:

$$\sum_{n=0}^{\infty} p(n) = 1.$$

Para que exista régimen estacionario es necesario que la serie anterior converja. Para este fin, o bien se limita el largo de la cola, o en muchos modelos prácticos esto se traduce en la condición $\rho < 1$ donde ρ representa la utilización del sistema (menos clientes entrantes que la capacidad de procesamiento).

Muchos de los modelos fundamentales de teoría de colas pueden formularse como procesos de nacimiento y muerte, debido a que el número de clientes en el sistema cambia únicamente por llegadas y salidas.

En estos modelos, el estado del sistema se describe completamente por el número de clientes presentes, lo que permite aplicar directamente las ecuaciones del proceso de nacimiento y muerte.

En sistemas de telecomunicaciones:

- Los nacimientos representan la llegada de paquetes, llamadas o sesiones.
- Las muertes representan la finalización del procesamiento o transmisión.

- Los estados representan el número de paquetes en cola o en servicio.

El marco de nacimiento y muerte permite modelar de forma rigurosa fenómenos como la congestión, el bloqueo y el retardo, proporcionando expresiones analíticas para evaluar el rendimiento del sistema.

El enfoque basado en procesos de nacimiento y muerte es fundamental porque:

- Proporciona una base matemática unificada para numerosos modelos de colas.
- Permite obtener distribuciones estacionarias de forma sistemática.
- Facilita el análisis de sistemas con múltiples servidores o capacidad limitada.
- Sirve como punto de partida para modelos más complejos.

7.5 Modelos de colas basados en procesos de nacimiento y muerte. Ejemplos

En el un modelo con llegadas y salidas que se comporten como un proceso de Poisson, con una sola estación de servicio (modelo $M/M/1$), se tendrá que

- Las llegadas siguen un proceso de Poisson con tasa constante λ, por lo que $\lambda_n = \lambda$.
- El tiempo de servicio es exponencial con tasa μ, por lo que $\mu_n = \mu$ para $n \geq 1$.

Las probabilidades estacionarias resultan:

$$P_n = (1-\rho)\rho^n, \quad \text{donde } \rho = \frac{\lambda}{\mu}.$$

Por otro lado, cuando existen m servidores ($M/M/m$), la tasa de servicio depende del número de clientes en el sistema:

$$\mu_n = \begin{cases} n\mu, & n < m, \\ m\mu, & n \geq m. \end{cases}$$

Este modelo describe sistemas con múltiples canales de servicio, típicos en centrales telefónicas, servidores de red o estaciones base.

Si bajo las condiciones anteriores, la capacidad del sistema está limitada a K clientes, ($M/M/m/K$), se tiene:

$$\lambda_K = 0,$$

lo que significa que nuevas llegadas son rechazadas cuando el sistema está lleno. Este modelo es especialmente relevante para el dimensionamiento de buffers en routers y dispositivos de red.

8. Resolución analítica de modelos de colas

Una gran parte del espectro de los sistemas de colas no pueden resolverse de manera exacta, en dichos casos se recurre a la simulación. Sin embargo un subgrupo, bajo ciertas hipótesis, tiene solución exacta. En este capítulo se estudiarán la mayoría de modelos de colas cuya solución analítica es conocida. Para ellos se definirán los principales parámetros y variables de un sistema de colas. Este capítulo está basado en [3, Cap. 5].

8.1 Modelos de colas con disciplina de prioridades

La disciplina de cola se encarga de establecer el acceso de las tareas a los diferentes recursos. Los tipos de disciplinas más utilizados son:

- FIFO (*First-In-First-Out*): Los clientes se atienden de acuerdo a su orden de llegada a la cola.
- LIFO (*Last-In-First-Out*): Los clientes se atienden en orden inverso al que llegan a la cola.
- SIRO (Service In Random Order): Los clientes son atendidos de forma aleatoria.
- SIFO (*Shortest-In, First-Out*): Los clientes con servicios más cortos se atienden primero.
- RR (*Round Robin*): Se atiende a varios clientes a la vez segmentando sus tareas.

Notación de un sistema de espera

La forma más usual de representar los sistemas de colas es mediante la notación de Kendall

$$A/B/m/K/N/Z$$

- A: Distribución de la variable aleatoria tiempo entre llegadas (población finita).
- B: Distribución de la variable aleatoria tiempo de servicio.
- m: Número de servidores en el sistema.
- K: Capacidad del sistema, ∞ si no se especifica.
- N: Tamaño de la población, ∞ si no se especifica.
- Z: Disciplina de gestión de cola, FIFO si no se especifica.

8.2 Papel de las distribuciones exponenciales

Las distribuciones exponenciales constituyen una herramienta fundamental en el análisis y modelado de fenómenos aleatorios relacionados con tiempos entre eventos. En ingeniería de telecomunicaciones, su utilidad es particularmente notable en el estudio del tráfico de paquetes, la modelización de llegadas en colas, la fiabilidad de sistemas y la caracterización del tiempo entre fallos en dispositivos y enlaces. Su simplicidad analítica y sus propiedades de memoria nula permiten formular modelos tratables que capturan adecuadamente el comportamiento estocástico de múltiples procesos relevantes.

Las principales distribuciones que se emplean en el modelado de sistemas de colas son:

- M: Distribución exponencial, ver Figura 8.1.

$$f(x) = \lambda e^{-\lambda x}$$

- D: Distribución determinista (constante).

$$f(x) = \delta(x-a)$$

- U: Distribución uniforme, ver Figura 8.2.

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{resto} \end{cases}$$

- E_k: Distribución Erlang de parámetro k, ver Figura 8.3.

$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda k}}{(k-1)!}$$

- H_k: Distribución hiperexponencial de parámetro k, ver Figura 8.4.

$$f(x) = \sum_{i=1}^{k} \alpha_i \lambda_i e^{-\lambda_i x}, \quad \sum_{i}^{k} \alpha_i = 1$$

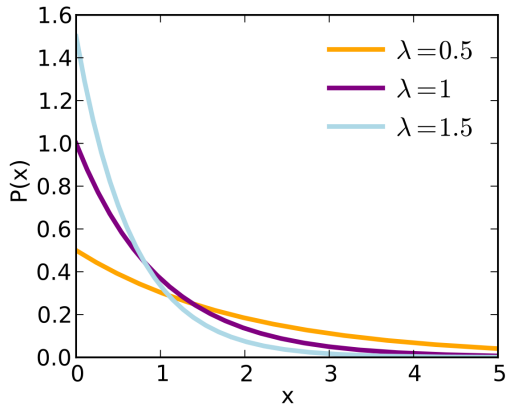


Figura 8.1: Distribución Exponencial

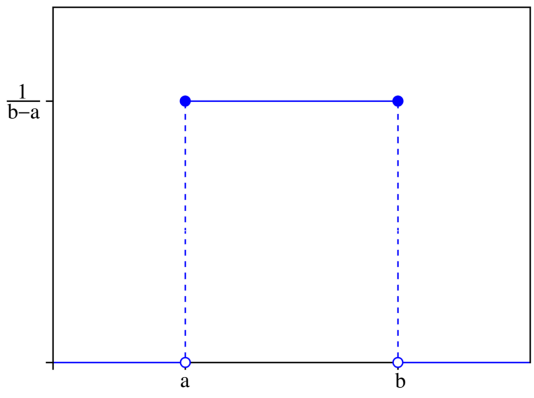


Figura 8.2: Distribución Uniforme

Importante — Régimen de trabajo. El régimen de trabajo se refiere a la fase en la cual se encuentra el sistema. Existen dos regímenes de trabajo:

- **Régimen transitorio:** Cuando el valor de las variables de un sistema depende de las condiciones iniciales del mismo, decimos que nos encontramos en **régimen transitorio**, ver Figura 8.5.

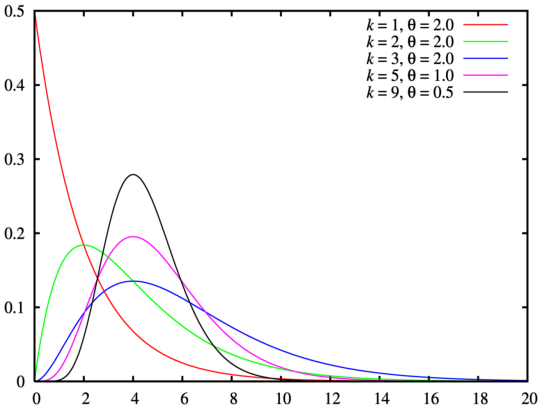


Figura 8.3: Distribución Erlang

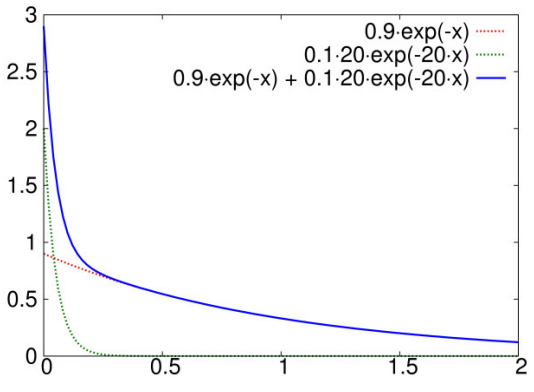


Figura 8.4: Distribución Hiperexponencial

- **Régimen permanente**: En este curso nos **centraremos** en el **régimen permanente**, es decir, cuando las variables han alcanzado un estado estable, ver Figura 8.5.

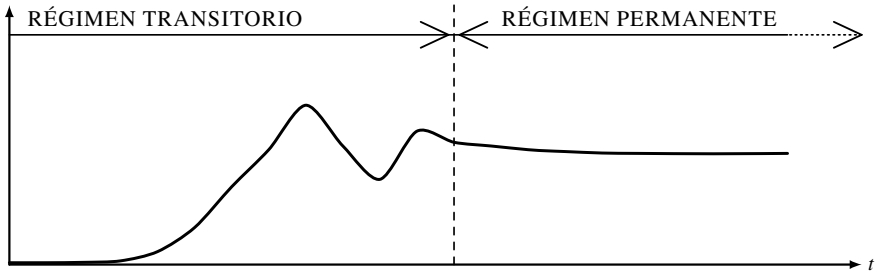


Figura 8.5: Diferencia de regímenes

R Las distribuciones exponenciales desempeñan un papel central en la ingeniería de telecomunicaciones. Su aplicación incluye el dimensionamiento de enlaces en redes de paquetes, modelización del tráfico en sistemas de colas $M/M/1$, análisis de fiabilidad de equipos y enlaces, planificación de capacidad en redes móviles y estudios de interrupciones en servicios. Su estructura matemática sencilla facilita la integración en modelos más complejos como cadenas de Markov, procesos de Poisson o simulaciones dinámicas de redes.

8.3 Nomenclatura a utilizar para los sistemas de colas

En general vamos a emplear la siguiente nomenclatura

- λ : Número promedio de clientes que arriban al sistema por unidad de tiempo (tasa promedio de arribos)
- T_a : Intervalo promedio entre arribos de clientes (llamado también tiempo promedio de arribos)

$$T_a = \frac{1}{\lambda}$$

- μ : Número promedio de clientes que puede atender un canal por unidad de tiempo (velocidad promedio de atención).
- T_s : Duración promedio del servicio (o Tiempo promedio de atención)

$$T_s = \frac{1}{\mu}$$

- ρ : Factor de tránsito (o de tráfico)
$$\rho = \frac{\lambda}{\mu}$$
- n: Estado del sistema (es la cantidad de clientes que hay dentro del sistema en un momento determinado).
- $p(n)$: Probabilidad de que el sistema se encuentre en el estado n.
- K: Capacidad del sistema (cantidad máxima de clientes que puede haber en el sistema, tanto en los canales como en la cola).
- m: Cantidad de canales dispuestos en paralelo en la unidad operativa de atención.
- H: Número promedio de canales activos
- $\mathrm{L_C}$: Longitud promedio de la cola (número esperado de clientes esperando ser atendidos).
$$\mathrm{L_C} = \sum_{n=\mathrm{m}+1}^{\mathrm{K}} (\mathrm{n}-\mathrm{m}) \cdot \mathrm{p(n)}$$
- L: Longitud promedio del sistema (número esperado de clientes dentro del sistema).
$$\mathrm{L} = \sum_{n=0}^{K} \mathrm{n} \cdot \mathrm{p(n)}$$
Además: $\mathrm{L} = \mathrm{L_C} + \mathrm{H}$
- λ_n : Tasa promedio de ingreso de clientes al sistema cuando el estado del mismo es n.
En términos generales:
$$\lambda_\mathrm{n} = \lambda \cdot \mathrm{p(i/n)}$$
- μ_n : Tasa promedio de egreso de clientes del sistema cuando el estado del mismo es n.
- $\bar{\lambda}$: Tasa promedio de ingreso de clientes al sistema. En forma general, tendremos que
$$\bar{\lambda} = \sum_{n=0}^{K} \lambda_n p(n)$$
- $\bar{\mu}$: Tasa promedio de egreso de clientes del sistema. En forma general:
$$\bar{\mu} = \sum_{n=0}^{K} \mu_n p(n)$$
También: $\bar{\mu} = \mu \cdot H$
- PA: Porcentaje de actividad del canal
$$\mathrm{PA} = \frac{\mathrm{H}}{\mathrm{M}}$$
- $\mathrm{W_C}$: Tiempo promedio de espera en cola de un cliente.
$$\mathrm{W_C} = \frac{\mathrm{L_C}}{\bar{\lambda}}$$
- W: Tiempo promedio de permanencia en el sistema de un cliente.
$$\mathrm{W} = \mathrm{W_C} + \mathrm{T_s}$$
o también:
$$\mathrm{W} = \frac{\mathrm{L}}{\bar{\lambda}}$$
La expresión anterior es la fórmula de Little, y establece que el tiempo promedio de residencia en el sistema está dado por la relación entre el número promedio de clientes que permanece en el mismo y el flujo de circulación de clientes que pasa por él.

- $\omega_C(t)$: Probabilidad de que un cliente tenga que esperar un tiempo superior a t en la cola

$$\omega_{\mathrm{C}}(\mathrm{t}) = \mathrm{p}\left(\mathrm{w}_{\mathrm{C}} > \mathrm{t}\right)$$

- $\omega(\mathrm{t})$: Probabilidad de que un cliente tenga que permanecer un tiempo superior a " t " en el sistema.

$$\omega(\mathrm{t}) = \mathrm{p}(\ \mathrm{W} > \mathrm{t})$$

- $\overline{\mathrm{R}}$: Tasa promedio de clientes rechazados (por capacidad finita o por impaciencia).

$$\overline{\mathrm{R}} = \lambda - \bar{\lambda}$$

- PR: Porcentaje de rechazos del sistema

$$\mathrm{PR} = \frac{\overline{\mathrm{R}}}{\lambda}$$

- $\overline{\mathrm{A}}$: Tasa promedio de clientes que abandonan el sistema sin recibir el servicio.

$$\bar{\lambda} = \bar{\mu} + \overline{\mathrm{A}}$$

- PAB: Porcentaje de abandonos del sistema

$$\mathrm{PAB} = \frac{\overline{\mathrm{A}}}{\bar{\lambda}}$$

- λ_{r} : Número promedio de veces por unidad de tiempo que un mismo cliente requiere un servicio en el sistema (aplicable a sistemas de población finita).
- $\mathrm{U}\left(\mathrm{oT_r}\right)$: Tiempo promedio que transcurre para que un cliente solicite un servicio desde que sale del sistema (aplicable a sistemas de población finita).

$$\mathrm{U} = \frac{1}{\lambda_{\mathrm{r}}}$$

- N: Tamaño de la población
- J: Número promedio de clientes fuera del sistema (aplicable a población finita)
- D: Probabilidad de que un cliente que arriba al sistema tenga que esperar para recibir el servicio.
- x: Factor de servicio de un sistema de población finita

$$\mathrm{x} = \frac{\mathrm{T_S}}{\mathrm{T_S} + \mathrm{U}}$$

- F: Factor de eficiencia de un sistema de población finita

$$\mathrm{F} = \frac{\mathrm{T_S} + \mathrm{U}}{\mathrm{T_S} + \mathrm{U} + \mathrm{W_C}}$$

- α : Coeficiente de impaciencia promedio de la población de clientes del sistema.

Parámetros económicos

Para el análisis económico denotaremos

- c_e como el coste derivado de esperar en el sistema.
- c_s costo de atención de un canal a velocidad unitaria.
- c_L como el coste de cada lugar en la cola.
- u como el precio de cada servicio
- c_e como el coste derivado de mantener una unidad en el sistema, y
- c_c como el coste por unidad de tiempo cada canal.

8.4 Sistemas de un solo canal

En esta sección se estudian modelos con un solo servidor/canal $m = 1$.

8.4.1 Modelo M/M/1

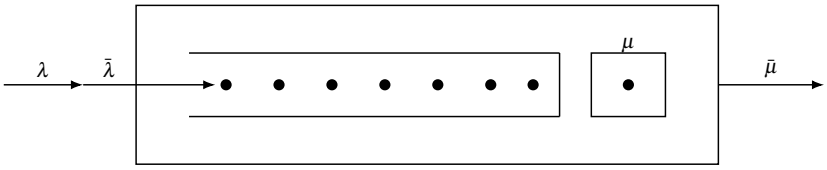


Figura 8.6: Modelo M/M/1

El primer modelo a estudiar es el M/M/1, bajo las siguientes **hipótesis:**

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de un solo canal de servicio.
- La capacidad del sistema es infinita.
- Los clientes no presentan impaciencia.
- La población es infinita.

Este modelo se puede representar como la Figura 8.6

Parámetros

- $p(n)$ probabilidad de que en el sistema haya n clientes (estado n del sistema).
- λ: tasa promedio de llegadas al sistema, el cliente llega, a espera de entrar o no.
- $T_a = \dfrac{1}{\lambda}$ intervalo medio entre llegadas
- μ: tasa promedio de salidas del sistema.
- $T_s = \dfrac{1}{\mu}$ duración promedio de servicios
- $m = 1$: un solo canal de atención (servidor).
- $\rho = \dfrac{\lambda}{\mu}$: factor de tránsito

Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes (estado n del sistema).
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- $\overline{\lambda}$: tasa promedio de ingresos de clientes al sistema, clientes entran al sistema.
- μ: tasa promedio de egresos de clientes del sistema, clientes salen del sistema.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de permanencia en el sistema.
- $\omega_c(t)$: probabilidad de espera mayor a t en la cola.
- $\omega(t)$: probabilidad de espera mayor a t en el sistema.

Determinación de variables

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \left(\frac{\lambda}{\mu}\right)^n p(0) = \rho^n p(0) = \rho^n(1-\rho)$$
$$p(n > k) = \rho^{k+1}$$
$$p(n \geq k) = \rho^k$$
$$\rho < 1 \leftarrow \lambda < \mu$$

- Determinación de L: la longitud del sistema es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = E[p(n)] = \sum_{n=0}^{\infty} n\rho^n(1-\rho) = \frac{\lambda}{\mu - \lambda}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-1$ clientes en cola.

$$L_c = L - \rho = \frac{\overline{\lambda}^2}{(\overline{\mu} - \overline{\lambda})\overline{\mu}}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$H = \rho$$
$$L = L_c + H$$
$$PA = \frac{H}{m} = \rho \text{ (Porcentaje de actividad)}$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$ o $\overline{\mu}$ (Fórmula de Little)

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{\overline{\lambda}}{\overline{\mu}(\overline{\mu} - \overline{\lambda})}$$

- Determinación de W: el tiempo de permanencia en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención. Además, verifica la Fórmula de Little.

$$W = W_c + T_s$$
$$W = \frac{L}{\overline{\lambda}}$$

- Determinación de $\omega(t)$: la probabilidad de que un cliente tenga que permanecer en el sistema un tiempo superior a t, es decir, $\omega(t) = p(W > t)$

$$\omega(t) = p(W > t) = e^{-(\overline{\mu} - \overline{\lambda})t}$$

- Determinación de $\omega_c(t)$: la probabilidad de que un cliente tenga que permanecer en la cola un tiempo superior a t, es decir, $\omega_c(t) = p(W_c > t)$

$$\omega_c(t) = p(W_c > t) = \rho e^{-(\overline{\mu} - \overline{\lambda})t}$$

Análisis Económico

- La velocidad de media de atención, $\overline{\mu}$, es una variable de decisión.
- Podemos modificarla para obtener un balance entre el coste de mejorar el servicio y las pérdidas por mal funcionamiento.
- Definimos c_e como el coste derivado de esperar en el sistema, y
- c_s costo de atención de un canal a velocidad unitaria.
- Podemos definir una función de costes tal que:

$$Z = c_e L + c_s \overline{\mu}$$

- Minimizando la función anterior, obtenemos la velocidad de servicio óptima como:

$$\mu_0 = \overline{\lambda} + \sqrt{\overline{\lambda}\frac{c_e}{c_s}}$$

Ejemplo de aplicación

■ **Ejemplo 8.1 — Modelo M/M/1.** A la ventanilla de un banco llegan usuarios según un proceso de Poisson de tasa 5 usuarios por hora. Sabiendo que estos consumen un tiempo aleatorio cuya distribución puede suponerse exponencial de media $\frac{1}{6}$ de hora y que la disciplina de atención es FIFO. Se pide:

a) Calcular los valores medios de clientes en el sistema y en la cola.
Notemos que $\lambda = 5$ usuarios/hora, y $T_s = \frac{1}{6} = \frac{1}{\mu}$, por tanto $\mu = 6$ usuarios/hora. Luego, se tendrá

$$L = \frac{\lambda}{\mu - \lambda} = 5$$

Por otro lado,

$$L_c = L - \rho = \frac{25}{6} \approx 4.1$$

Se concluye que el sistema tendrá un número medio de usuarios de 5 y una longitud de cola media de 4.1 usuarios.

b) ¿Qué porcentaje de los usuarios llega al sistema y lo encuentra desocupado?

$$p(0) = 1 - \rho = 1 - \frac{5}{6} = \frac{1}{6}$$

Aproximadamente **un sexto** de los usuarios encuentran desocupado el sistema.

c) Calcular el tiempo medio total de estancia en el centro de un usuario

$$W = \frac{L}{\lambda} = 1$$

Luego, el tiempo medio de un usuario en el sistema es de **una hora**.

d) Si en la sala de espera hay 4 sillas, ¿cuál es la probabilidad de que un usuario llegue al sistema y tenga que estar de pie?
Esto es equivalente a preguntar la probabilidad de que haya estrictamente más de 4 usuarios en el sistema, esto es

$$p(n > 4) = \rho^{k+1} = \rho^5 \approx 0.402$$

La probabilidad de que un usuario llegue al sistema y tenga que estar de pie es de aproximadamente un 40%.

e) Calcular los valores medios del tiempo de espera en cola y del tiempo de respuesta.

$$W_c = W_c = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{5}{6}$$

Por lo que se concluye que el tiempo de espera en cola medio es de 50 minutos, y el tiempo de respuesta medio es de 1 hora ($W = 1$).

f) ¿Cuál es la probabilidad de que un usuario espere más de una hora en ser atendido?

$$\omega(t)|_{t=1} = p(W_c > t) = \rho e^{-(\mu-\lambda)t} \approx 0.301$$

La probabilidad de que un usuario tenga que esperar más de una hora es de aproximadamente un 30%.

g) ¿Cuántas sillas se necesitarían para que un usuario al llegar al sistema tenga una probabilidad menor del 10% de esperar de pie?
En esta pregunta se nos indica calcular la cantidad de sillas k, necesarias para que la probabilidad de tener más usuarios que $k+2$ (k sillas en la cola, 1 usuario que llega y 1 usuario siendo atendido) sea menor que el 10%. Para ello empleamos

$$\begin{aligned} p(n > k+1) = \rho^{k+2} &\leq 0.1 \\ k &\leq -\frac{1}{\log \rho} - 2 \\ k &\geq 10.6 \end{aligned}$$

Por lo tanto se necesitarían 11 sillas para que la probabilidad de un cliente de pie sea menor que el 10%

■

8.4.2 Modelo M/M/1/K

El siguiente modelo a estudiar es el M/M/1/K, ver figura 8.7, bajo las siguientes hipótesis.

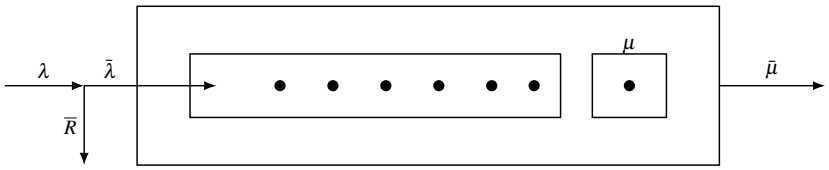


Figura 8.7: representación gráfica de un modelo M/M/1/K

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes que llegan forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de un solo canal de servicio.
- **La capacidad del sistema está limitada a un valor** K (cola finita).
- Los clientes no presentan impaciencia.
- La población de clientes potenciales es infinita.

Parámetros

- λ: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = \frac{1}{\lambda}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- $m = 1$: número de servidores, en este caso 1.
- K: **capacidad del sistema**. Esto significa que no pueden haber más clientes que K, si hay más clientes, estos serán rechazados.

Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema (número esperado de clientes esperando dentro del sistema).
- L_c: longitud promedio de la cola (número esperado de clientes esperando ser atendidos).
- $\overline{\lambda}$: tasa promedio de ingreso de clientes al sistema.
- $\overline{\mu}$: tasa promedio de egreso de clientes del sistema.
- W_c: tiempo promedio de espera en cola.
- W: tiempo promedio de espera en el sistema.
- H: promedio de canales activos.
- $\overline{R}$: tasa promedio de clientes rechazados.
- PR: porcentaje de rechazo.

Determinación de variables.

Importante Para varias de las variables se cumplirá que la fórmula presenta una indeterminación del tipo $0/0$ en el caso $\rho = 1$. Para resolverla se aplicará la regla de L'Hòpital, con respecto a ρ.

- $p(n)$: el **estado de sistema** es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} \rho^n \frac{1}{K+1} & \text{Si } \rho = 1 \\ \rho^n \frac{1-\rho}{1-\rho^{K+1}} & \text{si no} \end{cases}$$

- L: la **longitud del sistema** representa el número esperado de clientes en el sistema, se deduce como el valor esperado de la v.a. número de clientes en el sistema

$$L = \begin{cases} \rho \frac{1-(K+1)\rho^K + K\rho^{K+1}}{(1-\rho)(1-\rho^{K+1})} & \rho \neq 1 \\ \frac{K}{2} & \rho = 1 \end{cases}$$

- **Longitud esperada de la cola** L_c: representa el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-1$ clientes en cola, clientes esperando ser atendidos.

$$L_c = L - (1 - p(0)) = \rho^2 \frac{1 - K\rho^{K-1} + K\rho^K}{(1-\rho)(1-\rho^{K+1})}$$

- H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= 1 - p(0) = L - L_c \\ L &= L_c + H \end{aligned}$$

- Porcentaje de actividad: $PA = \frac{H}{m} = 1 - p(0)$
- Tasa promedio de egresos $\overline{\mu}$,

$$\overline{\mu} = \mu[1 - p(0)] = \mu \cdot H$$

  Los egresos se producen solo en los estados $n \neq 0$.
- Tasa promedio de ingresos $\overline{\lambda}$

$$\overline{\lambda} = \lambda \cdot [1 - p(K)]$$

  Se producirán ingresos al sistema mientras no haya K clientes, esto es $n \leq K$.
  Bajo condiciones de **equilibrio**, se tendrá que $\overline{\lambda} = \overline{\mu}$.

- W_c: El tiempo medio de espera en cola viene dado por la relación entre L_c y $\overline{\lambda}$ (o $\overline{\mu}$) (Fórmula de Little).

$$W_c = \frac{L_c}{\overline{\mu}} = \frac{L_c}{\mu\left(1-\frac{1-\rho}{1-\rho^{K+1}}\right)} W_c \qquad = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{\lambda \cdot [1-p(K)]} = \frac{L_c}{\lambda \cdot [1-\rho^k p(0)]}$$

- W: El tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

- $\overline{R}$: el número promedio de clientes rechazados por unidad de tiempo es la tasa promedio de llegadas multiplicada por la probabilidad de no ingresar.

$$\overline{R} = \lambda p(K) = \lambda \rho^K \frac{(1-\rho)}{1-\rho^{K+1}}$$
$$\lambda = \overline{\lambda} + \overline{R}$$

- PR: el porcentaje de rechazos, serán aquellas llegadas que serán rechazadas por encontrarse el sistema ocupado se puede escribir como:

$$PR = \frac{\overline{R}}{\overline{\lambda}}$$

Análisis Económico

- En este tipo de sistemas se intentan balancear las perdidas por ingresos no percibidos y el coste asociado a la infraestructura de espera.
- Definimos c_L como el coste de cada lugar en la cola, y u como el precio de cada servicio.
- Variable de desición K, lugares habilitados para cola.
- Podemos definir una función de costes tal que:

$$Z = c_L(K-1) + u\overline{R} = c_L(K-1) + u\overline{\lambda}\rho^K \frac{1-\rho}{1-\rho^{K+1}}$$

- Problema de minimización en enteros.

Ejemplo de aplicación

■ **Ejemplo 8.2 — Ejemplo 2.4 de [3] – Modelo M/M/1/10.** A un sistema M/M/1/10 llega un ciudadano cada 5 minutos, en promedio. Si no hay sillas vacías, el ciudadano se va. La duración media del servicio es de 6 minutos. Calcule:

a) Probabilidad de que un ciudadano no tenga que esperar para recibir el servicio.
Primeramente calculemos los parámetros del sistema

$$\lambda = \frac{1}{5}\text{cl/min} = 12\text{cl/h}$$
$$\mu = \frac{1}{6}\text{cl/min} = 10\text{cl/h}$$
$$\rho = \frac{\lambda}{\mu} = 1.2$$

Por lo que se tiene que la probabilidad de llegar y no tener que esperar es de

$$p(0) = \frac{1-\rho}{1-\rho^n} \approx 0.0310$$

b) Porcentaje de ocupación.
El porcentaje de ocupación del canal se calcula como

$$\text{PA} = \frac{H}{m} = H = 1 - p(0) \approx 0.9689 = 96.89\%$$

El porcentaje de ocupación será aproximadamente del 96.89%

c) Probabilidad de encontrar 3 o más sillas ocupadas.

$$\begin{aligned} p(n > 3) &= 1 - p(0) - p(1) - p(2) \approx 0.8872 \\ p(1) &= \rho p(0) \approx 0.0372 \\ p(2) &= \rho^2 p(0) \approx 0.0446 \end{aligned}$$

Por tanto, la probabilidad de encontrar 3 o más sillas ocupadas es aproximadamente 4.46%

d) Probabilidad de que haya menos de 2 sillas ocupadas.

$$p(n \leq 1) = p(0) + p(1) \approx 0.0682$$

Por tanto, la probabilidad de que hayan menos de 2 clientes en el sistema es aproximadamente 6.82%

e) Número promedio de ciudadanos en la oficina.

$$L = \rho \frac{1 - (K+1)\rho^K + K\rho^{K+1}}{(1-\rho)(1-\rho^{K+1})} \approx 6.701$$

Por tanto, el número promedio de clientes en la oficina será de aproximadamente 6.701.

f) Número promedio de sillas ocupadas.

$$L_c = L - [1 - p(0)] \approx 5.7418$$

Por tanto, el número promedio de sillas ocupadas será de aproximadamente 5.7418.

g) Ingreso esperado, si cada servicio se cobra a 30 €.

$$\text{Ingreso} = \overline{\mu} \cdot u = u \cdot \mu \cdot [1 - p(0)] \approx 290.67€$$

h) Tiempo promedio de espera en cola.

$$W_c = \frac{L_c}{\overline{\lambda}} \approx 0.5926 \approx 35.55\text{min}$$

Por tanto, el tiempo promedio de espera en cola es de 35.55min.

i) Tiempo promedio de permanencia en el sistema.

$$W = W_c + T_s = 0.5926 + 0.1 = 0.6926 \approx 41.55\text{min}$$

Por tanto, el tiempo promedio de permanencia en el sistema es de 41.55min.

■

8.5 Sistemas de varios canales en paralelo

En esta sección se estudian modelos con más de un servidor/canal $m \geq 1$.

8.5.1 Modelo M/M/m

El siguiente modelo a estudiar es el M/M/m, ver figura 8.8, bajo las siguientes hipótesis.

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- **Se dispone de m canales de servicio**, en paralelo, con la misma velocidad de atención.
- La capacidad del sistema es infinita.
- Los clientes no presentan impaciencia.
- La población es infinita.

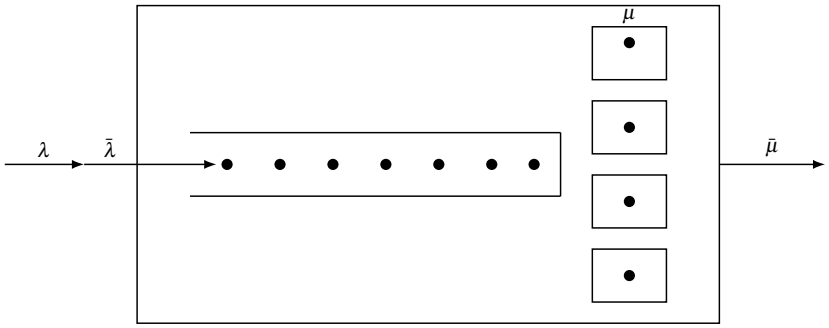


Figura 8.8: Modelo M/M/m

Parámetros

- λ: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = \frac{1}{\lambda}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- m: número de servidores.

Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de espera en el sistema.
- $\omega_c(t)$: probabilidad de espera mayor a t en la cola.
- $\omega(t)$: probabilidad de espera mayor a t en el sistema.

Determinación de variables

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} \frac{\rho^n}{n!} p(0), & \text{si } n < m \\ \left(\frac{\rho}{m}\right)^n \frac{m^m}{m!} p(0), & \text{si } n \geq m \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} \frac{\rho^n}{n!} + \frac{\rho^m}{(m-1)!(m-\rho)} \right]^{-1}$$

$$p(n \geq k) = \frac{\rho^k}{(k-1)!(k-\rho)} p(0)$$

  Luego se debe cumplir que $\frac{\rho}{m} < 1$, para que el sistema esté en equilibrio y funcione sin colapsar.
- Determinación de L: la longitud del sistemas es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = E[p(n)] = \frac{\rho^m \mu \lambda}{(m-1)!(m\mu - \lambda)^2} p(0) + \rho$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n - m$ clientes en cola.

$$L_c = L - \rho = \frac{\rho^m \mu \lambda}{(m-1)!(m\mu - \lambda)^2} p(0)$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= \rho \\ L &= L_c + H = L_c + \rho \\ PA &= \frac{H}{m} = \frac{\rho}{m} \text{ (Porcentaje de actividad de cada canal)} \end{aligned}$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y λ

$$W_c = \frac{L_c}{\lambda}$$

- Determinación de W: el tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención. Además, verifica la Fórmula de Little.

$$\begin{aligned} W &= W_c + T_s \\ W &= \frac{L}{\lambda} \end{aligned}$$

- Determinación de $\overline{\mu}$, $\overline{\lambda}$: tasas promedio de ingresos y egresos del sistema.

$$\begin{aligned} \overline{\mu} &= \mu \cdot H, \\ \overline{\lambda} &= \lambda \end{aligned}$$

- Determinación de $\omega(t)$: la probabilidad de que un cliente tenga que permanecer en el sistema un tiempo superior a t, es decir, $\omega(t) = p(W > t)$

$$\omega(t) = p(W > t) = e^{-\mu t}\left[\frac{1+(m\rho)^m p(0)[1-e^{\mu(m-1-m\rho)t}]}{m!(1-\rho)(m-1-m\rho)}\right]$$

- Determinación de $\omega_c(t)$: la probabilidad de que un cliente tenga que permanecer en la cola un tiempo superior a t, es decir, $\omega_c(t) = p(W_c > t)$

$$\omega_c(t) = p(W_c > t) = \frac{e^{-\mu(m-\rho)t}\rho^m p(0)}{(m-1)!(m-\rho)}$$

Análisis Económico

- En este tipo de sistemas, el número de servidores m es la variable de decisión.
- Podemos modificarla para minimizar el coste operativo del sistema.
- c_e como el coste derivado de mantener una unidad en el sistema, y
- c_c como el coste por unidad de tiempo cada canal.
- Podemos definir una función de costes tal que:

$$z = c_e L + c_c m$$

- Problema de minimización en enteros.

Ejemplo de aplicación

■ **Ejemplo 8.3 — Ejemplo 3.1 de [3] – Modelo M/M/4.** Un taller de reparaciones cuenta con cuatro técnicos (sistema M/M/4). En promedio, llegan pedidos cada 5 minutos y la duración media de las reparaciones es de 10 minutos. Determinar:

a) El número mínimo de operarios para que el sistema funcione.
Cálculo de parámetros

$$\lambda = \frac{1}{5}\frac{\text{cl}}{\text{min}} \cdot 60 \cdot \frac{\text{min}}{h} = 12\frac{\text{cl}}{h}$$
$$\mu = \frac{1}{10}\frac{\text{cl}}{\text{min}} \cdot 60 \cdot \frac{\text{min}}{h} = 6\frac{\text{cl}}{h}$$
$$\rho = \frac{\lambda}{\mu} = 2$$

Nótese que para que el sistema no colapse y entre en estado de equilibrio se tiene que dar que

$$\frac{\rho}{m} < 1$$

por lo tanto a partir de 3 operarios serán suficientes para que el sistema funcione.

b) La probabilidad de que una pieza sea reparada conforme llega.
La probabilidad de que una pieza sea reparada al llegar es la probabilidad de tener 3 o menos clientes en el sistema, esto es

$$p(n \leq 3) = p(0) + p(1) + p(2) + p(3) = 1 - p(n \geq 4) \approx 0.8261$$

Por tanto, la probabilidad de que una pieza sea reparada al llegar es aproximadamente de 82.61%.

c) Porcentaje de ocupación de los técnicos.
El porcentaje de ocupación de los técnico es el porcentaje de actividad del canal, esto es

$$PA = \frac{H}{m} = \frac{\rho}{m} = \frac{1}{2}$$

Por tanto, el porcentaje de ocupación de los técnicos es del 50%.

d) Probabilidad de que haya menos de dos piezas en el sistema.

$$p(n<2) = p(0) + p(1) = (1+\rho)p(0) \approx 0.3912$$

Por tanto, la probabilidad de que haya menos de dos piezas en el sistema es aproximadamente 39.12%.

e) Número medio de piezas en el taller.

$$L \approx 2.1739$$

Por tanto la media de piezas en el sistema es de aproximadamente 2.17.

f) Número promedio de piezas a la espera de ser reparadas.

$$L_c = L - \rho = 0.1739$$

Por tanto el promedio de piezas a la espera de ser reparadas es de 0.1739.

g) Ganancias esperadas, si cada reparación se cobra a 50 €.
Los ingresos esperados por hora serán el promedio de ingresos (egresos) por hora multiplicados por la ganancia, esto es

$$\text{Ingreso} = u \cdot \overline{\lambda} = 50 \cdot 12 = 600€/h$$

Por tanto los ingresos esperados por hora son de 600 €.

h) Tiempo promedio de permanencia de una pieza en el taller.
El tiempo promedio de permanencia de la pieza en el taller se calcula como

$$W = W_c + T_s = \frac{L_c}{\lambda} + \frac{1}{\mu} \approx 0.1812h$$

Por tanto, el tiempo promedio de permanencia de una pieza en el taller es de aproximadamente 0.1812 horas.

■

8.5.2 Modelo M/M/m/K

El siguiente modelo a estudiar es el M/M/m/K, ver figura 8.9, bajo las siguientes hipótesis.

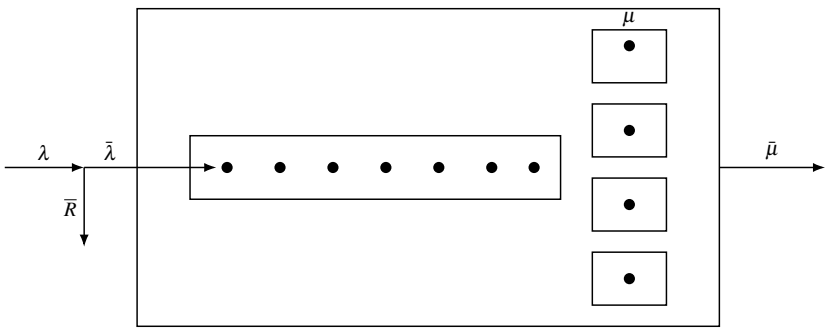


Figura 8.9: Modelo de colas M/M/m/K

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- **Se dispone de m canales de servicio**, en paralelo, con la misma velocidad de atención.
- **La capacidad del sistema está limitada a un valor K.**
- Los clientes no presentan impaciencia.
- La población es infinita.

Parámetros

- λ: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = \frac{1}{\lambda}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- m: número de servidores.
- K: capacidad del sistema.

Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de espera en el sistema.
- $\overline{R}$: tasa promedio de clientes rechazados.

Determinación de variables

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} \frac{\rho^n}{n!} p(0) & \text{Si } n < m \\ \left(\frac{\rho}{m}\right)^n \frac{m^m}{m!} p(0) & \text{Si } m \leq n \leq K \end{cases}$$

$$p(0) = \left[\frac{\rho^m}{(m-1)!} \left(1 - \left[\frac{\rho}{m}\right]^{K-m+1} \right) \frac{1}{(m-\rho)} + \sum_{n=0}^{m-1} \frac{\rho^n}{n!} \right]^{-1}$$

- Determinación de L: la longitud del sistema es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$\begin{aligned} L &= \sum_{n=0}^{K} n p(n) \\ &= \frac{\frac{\rho}{m} \cdot \rho^m \cdot p(0)}{m! \cdot \left(1 - \frac{\rho}{m}\right)^2} \cdot \left[1 - \left(\frac{\rho}{m}\right)^{K-m} - (K-m) \cdot \left(\frac{\rho}{m}\right)^{K-m} \cdot \left(1 - \frac{\rho}{m}\right) \right] + \rho \cdot \left[1 - \left(\frac{\rho}{m}\right)^K \cdot \frac{K^K}{m!} \cdot p(0) \right] \end{aligned}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado

de que en el sistema haya $n-m$ clientes en cola.

$$\begin{aligned} L_c &= \sum_{n=m}^{K} (n-m)p(n) \\ &= \frac{\frac{\rho}{m}\cdot\rho^m}{m!\cdot\left(1-\frac{\rho}{m}\right)^2}\cdot\left[1-\left(\frac{\rho}{m}\right)^{K-m}-(K-m)\cdot\left(\frac{\rho}{m}\right)^{K-m}\cdot\left(1-\frac{\rho}{m}\right)\right]\cdot p(0) \end{aligned}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= \rho\,[1-p(K)] \\ PA &= \frac{H}{m} \text{ (Porcentaje de actividad de cada canal)} \end{aligned}$$

- Determinación de $\overline{\mu}$, $\overline{\lambda}$: tasas promedio de ingresos/egresos del sistema.

$$\begin{aligned} \overline{\mu} &= \mu H, \\ \overline{\lambda} &= \lambda\,[1-p(K)] \end{aligned}$$

- Determinación de W_c: El tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$ o $\overline{\mu}$.

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{\lambda(1-p(K))}$$

- Determinación de W: El tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

- Determinación de $\overline{R}$: el número promedio de clientes rechazados por unidad de tiempo es la tasa promedio de llegadas multiplicada por la probabilidad de no ingresar.

$$\overline{R} = \lambda\,p(K) = \lambda - \overline{\lambda}$$

Análisis Económico

- Si elegimos el número de canales disponibles m como variable a optimizar, definimos
  - c_e como el coste de esperar una unidad de tiempo en el sistema.
  - c_c el coste por unidad de tiempo de cada canal.

$$\min(Z) = c_e L + c_c m$$

- Si elegimos el tamaño de la cola $K-m$ como variable a optimizar, definimos
  - c_L como el coste de cada plaza de la cola.
  - u precio de una unidad de servicio.

$$\min(Z) = c_L(K-m) + u\overline{R}$$

- Si elegimos la velocidad de servicio μ como variable a optimizar
  - c_s costo de atención de un canal a velocidad unitaria

$$\min(Z) = c_s M\mu + u\overline{R}$$

Ejemplo de aplicación

■ **Ejemplo 8.4 — Ejemplo 3.4 de [3] – Sistema M/M/2/4.** Un taller mecánico cuenta con 2 técnicos y una capacidad de 4 coches, cada 5 minutos llega un coche. Si no hay huecos vacíos, el coche se va. La duración media de las consultas es de 10 minutos. Calcule:

a) Probabilidad de que un coche no tenga que esperar para ser atendido.

$$p(n \leq 1) = p(0) + p(1) = \frac{1}{3}$$
$$p(0) = \frac{1}{9}$$
$$p(1) = \rho p(0) = \frac{2}{9}$$

Por tanto, la probabilidad de no tener que esperar para ser atendido es de aproximadamente un 33%.

Nota: al calcular $p(0)$, algunas calculadoras pueden dar 0 (lo que es imposible en este caso). La razón es que $m = \rho$ produce una indeterminación. Para calcular $p(0)$ hay que resolverla analíticamente.

b) Porcentaje de ocupación.

$$PA = \frac{H}{m} = \frac{\rho[1 - p(K)]}{m} = \frac{7}{9}$$

Por lo que se concluye que el porcentaje de ocupación es de aproximadamente 77%.

c) Probabilidad de que un coche no pueda ingresar.
Esto es, la probabilidad de que el sistema esté lleno, $n = K$

$$p(4) = \frac{2}{9}$$

Por tanto, la probabilidad de que un coche no pueda entrar es de aproximadamente el 22%.

d) Tasa promedio de coches que no son atendidos.
Esto es la tasa de rechazos $\overline{R}$

$$\overline{R} = \lambda p(K) = \frac{8}{3} \text{ cl/h}$$

Por tanto, la tasa de rechazos será de aproximadamente $2,666$ coches por hora.

e) Número promedio de clientes esperando ser atendidos.

$$L_c = \sum_{n=m}^{K} (n-m)p(n) = p(3) + 2p(4) = \frac{2}{3} \text{ coches}$$

Por tanto, el número promedio de coches esperando ser atendidos es de aproximadamente 0.666.

f) Número promedio de coches en el taller.

$$L = L_c + H = \frac{2}{3} + 2[1 - p(4)] = \frac{20}{9} \text{ coches}$$

Por tanto, el número de clientes siendo atendidos es de aproximadamente $2,22$.

g) Facturación esperada, si cada servicio se cobra a 50 €.

$$\text{Ingreso} = u \cdot \overline{\lambda} = u\lambda[1 - p(K)] \approx 466,68 \text{ €}/h$$

Por tanto la facturación esperada es de aproximadamente $466,68$ €$/h$.

h) Tiempo promedio de espera en cola.

$$W_c = \frac{L_c}{\overline{\lambda}} \approx 0.0714\ h$$

i) Si el cliente es aceptado por el taller a las 9h, ¿a qué hora esperaría irse con su coche reparado?

$$W = W_c + T_s \approx 0.2381\ h = 14.286 \text{ min}$$

Por tanto, el cliente espera poder irse aproximadamente a las 9 : 14 horas.

j) Pérdida de facturación, si cada reparación se factura a 50 €.

$$\text{Loss} = u \cdot \overline{R} \approx 133.33\ €/h$$

■

8.6 Sistemas de población finita

Si la población fuente de clientes de un sistema de colas es finita pero lo suficientemente grande como para que la tasa media de arribos no se vea afectada de manera apreciable por la entrada o salida de un cliente del sistema, el modelo puede analizarse, en la práctica, como si la población fuese infinita.

En cambio, cuando la fuente de usuarios es reducida y la llegada de un cliente o la finalización de un servicio modifica sensiblemente la probabilidad de arribos futuros, se trata de un sistema con población finita, que requiere un tratamiento específico.

En estos sistemas, los clientes atendidos no abandonan definitivamente el conjunto de potenciales usuarios, sino que regresan a la población fuente del sistema.

Para el análisis de sistemas con población finita, la observación no se centra en el flujo global de llegadas, sino en el comportamiento individual de cada cliente. Con este fin se introduce el parámetro T_r (frecuentemente denotado por la letra "U"), que representa el tiempo medio entre dos arribos consecutivos de un mismo cliente al sistema; es decir, el tiempo medio que un cliente permanece fuera del sistema.

Se define entonces λ_r como el inverso de T_r. Este parámetro expresa la frecuencia con la cual un usuario solicita el servicio, es decir, el número medio de veces por unidad de tiempo que un cliente, estando fuera del sistema, ingresa al mismo para recibir atención.

Si se denota por N el tamaño total de la población (número de clientes tanto dentro como fuera del sistema), todo arribo necesariamente proviene de los $N - n$ clientes que no se encuentran en el sistema en ese instante.

A continuación se formulará el modelo correspondiente al caso en que no existen restricciones de capacidad. No obstante, el mismo esquema de modelización puede adaptarse a sistemas con capacidad finita, limitando el número de estados posibles, de forma análoga a lo estudiado previamente para esos sistemas.

Los sistemas de población finita se definen sobre los siguientes supuestos.

- Si la población de un sistema es lo suficientemente grande, la tasa de llegadas no se ve afectada cuando aparece una llegada o una salida, como en los casos anteriores.
- En caso contrario, hay que tener en cuenta las siguientes consideraciones:
  - Las observaciones se hace sobre cada cliente, en vez de sobre la población.
  - Definiremos el parámetro T_r o U, como el tiempo entre llegadas para un mismo cliente.
  - Llamaremos N al tamaño de la población, es decir, al número de clientes dentro y fuera del sistema.

8.6.1 Modelo M/M/1/-/N

Un sistema M/M/1/-/N, ver figura 8.10 se basa en las siguientes hipótesis.

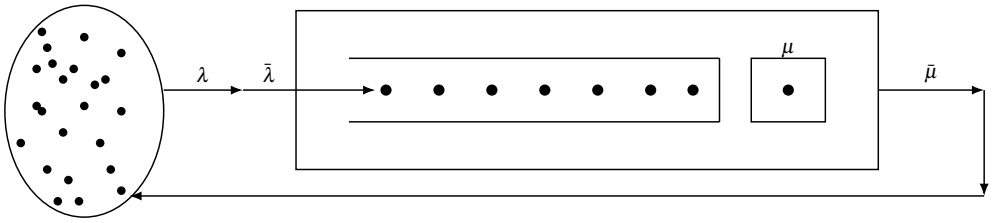


Figura 8.10: Modelo M/M/1/-/N

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de **un solo canal de servicio**.
- La **capacidad del sistema es infinita**.
- Los clientes no presentan impaciencia.
- La **población es finita y relativamente pequeña**.

Parámetros

- λ_r: frecuencia de requerimientos de servicios por el cliente. También podemos referirnos como intervalo medio entre requerimientos: $T_r = U = \frac{1}{\lambda_r}$.
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas: $T_s = \frac{1}{\mu}$.
- m: número de servidores, en este caso 1.
- N: número total de clientes, tamaño de la población.

Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- P_{in}: porcentaje de clientes que se encuentran en el sistema.
- L_c: longitud promedio de la cola.
- P_q: porcentaje de clientes que se encuentran en la cola.
- H: promedio de canales activos.
- P_{active}: porcentaje de clientes que están siendo atendidos.
- P_{out}: porcentaje de clientes que están fuera del sistema.
- W: tiempo promedio de espera en el sistema.
- W_c: tiempo promedio de espera en cola.
- J: número promedio de clientes fuera del sistema.

Determinación de variables

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \frac{N!}{(N-n)!}\rho_r^n p(0)$$
$$p(0) = \frac{1}{\sum_{n=0}^{N} \frac{N!}{(N-n)!}\rho_r^n}$$

- Determinación de L: la longitud del sistemas es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = E[p(n)] = \sum_{0}^{\infty} np(n) = N - \frac{1-p(0)}{\rho_r}$$

- El porcentaje de clientes que se encuentran en el sistema viene dado por

$$P_{in} = \frac{L}{N}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-1$ clientes en cola.

$$L_c = N - \frac{(1+\rho_r)[1-p(0)]}{\rho_r}$$

- El porcentaje de clientes que se encuentran esperando en la cola viene dado por

$$P_{queue} = \frac{L_c}{N}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$H = 1 - p(0)$$
$$P_{active} = \frac{H}{N} \qquad \text{Porcentaje de clientes siendo atendidos}$$
$$PA = \frac{H}{m} = 1 - p(0) \qquad \text{Porcentaje de actividad}$$

- Determinación de J: el número promedio de usuarios que se encuentran fuera del sistema, viene dado por:

$$J = N - L$$
$$J = \frac{H}{\rho_r}$$

- El porcentaje de clientes que se encuentran fuera del sistema viene dado por la expresión

$$P_{out} = \frac{J}{N}$$

- Determinación de $\overline{\mu}/\overline{\lambda}$: tasas promedio de ingresos/egresos del sistema.

$$\overline{\mu} = \mu[1-p(0)] = \overline{\lambda} = \lambda_r J$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$(o $\overline{\mu}$).

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{J\lambda_r} = \frac{L_c}{\mu[1-p(0)]} = \frac{L_c}{\overline{\mu}}$$

- Determinación de W: el tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

Análisis Económico

- Como hay un solo canal, la velocidad de media de atención, $\overline{\mu}$, se escoge usualmente como variable de decisión.
- Podemos modificarla para obtener un balance entre el coste de mejorar el servicio y las perdidas por mal funcionamiento.
- Definimos c_e como el coste derivado de esperar en el sistema, y c_s como el coste por atender a una velocidad unitaria.
- Podemos definir una función de costes tal que:

$$\min(Z) = c_e L + c_s \overline{\mu}$$

Ejemplo de aplicación

■ **Ejemplo 8.5 — Ejemplo 4.1 de [3] – modelo M/M/1/-/5.** En un sistema M/M/1/-/5, cada cliente permanece en promedio 0.5 horas fuera del sistema. La duración media del sistema es de 20 minutos. Determinar:

a) Probabilidad de que el sistema se encuentre ocupado.
Cálculo de parámetros

$$\begin{aligned} U &= 0.5\ \text{h/pedido} \Rightarrow \lambda_r = 2\ \text{pedidos}/h \\ \mu &= 3\ \text{pedidos}/h \\ \rho_r &= \frac{2}{3} \end{aligned}$$

Las probabilidades de cada estado resultan:

$$\begin{aligned} p(0) &= 0.0142 \\ p(1) &= 0.0473 \\ p(2) &= 0.1261 \\ p(3) &= 0.2524 \\ p(4) &= 0.3362 \\ p(5) &= 0.2241 \end{aligned}$$

Por tanto la probabilidad de que el sistema esté ocupado es de

$$p(1)+p(2)+p(3)+p(4)+p(5) = 1-p(0) \approx 0.9858$$

aproximadamente un 98%.

b) Porcentaje de clientes que se encuentran fuera del sistema.

$$P_{out} = \frac{J}{N} = \frac{N-L}{N} = 1 - \frac{1}{N}\left(N - \frac{1-p(0)}{\rho_r}\right) \approx 0.2958$$

o usando $J = 1.4789$, se concluye igualmente que el porcentaje de clientes fuera del sistema es de aproximadamente 30%.

c) Cantidad de clientes que se atienden en el sistema por hora, en promedio.

$$\overline{mu} = \mu \cdot H = \mu(1 - p(0)) \approx 2.957$$

Por tanto, se atienden de media aproximadamente 3 clientes por hora.

d) Longitud promedio de la cola.

$$L_c = N - \frac{(1+\rho_r)(1-p(0))}{\rho_r} \approx 2.5353$$

Por tanto, la longitud media de la cola es de aproximadamente 2.5353 clientes.

e) Tiempo de espera promedio de un cliente en cola.

$$W_c = \frac{L_c}{J \cdot \lambda_r} \approx 0.8573\ h$$

Por tanto, el tiempo de espera promedio de un cliente en cola es de 0.8573 h.

f) Tiempo promedio de permanencia de un cliente dentro del sistema.

$$W = W_c + T_s \approx 1.19\ h$$

Por tanto, el tiempo promedio de permanencia de un cliente en cola es de 1.19 h.

■

■ **Ejemplo 8.6 — Ejemplo 4.2 de [3] – modelo M/M/1/-/5.** Un CPD tiene disponibles 4 racks de servidores idénticos, de los cuales están en funcionamiento 3 y el cuarto está de backup, es decir, entra en funcionamiento cuando uno de los anteriores sufre algún problema. Sabiendo que los servidores fallan, en promedio, cada 12 horas de funcionamiento y que existe un equipo de reparación que tardas 3 horas en arreglar los desperfectos. Tenga en cuenta que si hay más de dos servidores fuera de servicio el CPD debe desconectarse.

a) Deducir y calcular las probabilidades asociadas a cada uno de los estados posibles.
b) Determinar la longitud promedio de cola de servidores averiados en esperas de ser reparados .
c) Sabiendo que el beneficio que produce una máquina en funcionamiento es de 1500€, calcular las pérdidas esperadas.

■

8.6.2 Modelo M/M/m/-/N

Un sistema M/M/m/-/N, ver figura 8.11 se basa en las siguientes hipótesis.

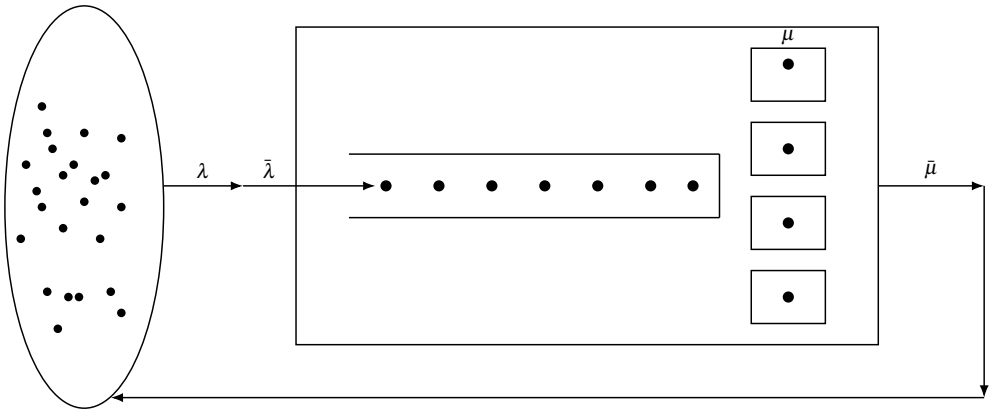


Figura 8.11: Sistema M/M/m/-/N

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de m **canales de servicio** en paralelo, con la misma velocidad de atención.
- La capacidad del sistema es infinita (cola infinita).
- Los clientes no presentan impaciencia.
- La **población es finita** y relativamente pequeña.

Parámetros

- λ_r: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = U = \frac{1}{\lambda_r}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- m: número de servidores.
- N: número total de clientes.

Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de espera en el sistema.
- J: número promedio de clientes fuera del sistema
- D: probabilidad de esperar para recibir servicio.
- X: factor de servicio.
- F: factor de eficiencia.

Determinación de variables

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} C_N^n \rho_r^n p(0) = \frac{N!}{(N-n)!n!} \rho_r^n p(0) \text{ Si } n < m \\ \frac{n! m^m}{m! m^n} C_N^n \rho_r^n p(0) \text{ Si } m \leq n \leq N \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} C_N^n \rho_r^n + \sum_{n=m}^{N} C_N^n \frac{n! m^m}{m! m^n} \rho_r^n \right]^{-1}$$

  Recordemos que C_N^n denota el número combinatorio

$$C_N^n = \frac{N!}{n!(N-n)!}$$

- Determinación de L: la longitud del sistemas es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = \sum_{n=0}^{m-1} n C_N^n \rho_r^n p(0) + \sum_{n=m}^{N} n C_N^n \frac{n! m^m}{m! m^n} \rho_r^n p(0)$$

- El porcentaje de clientes que se encuentran en el sistema viene dado por

$$P_{in} = \frac{L}{N}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-m$ clientes en cola.

$$L_c = \sum_{n=m}^{N} (n-m)p(n) = \sum_{n=m}^{N} (n-m) \cdot C_N^n \cdot \frac{n!}{m!} \cdot \frac{m^m}{m^n} \cdot \rho_r^n \cdot p(0)$$

- El porcentaje de clientes que se encuentran esperando en la cola viene dado por

$$P_{queue} = \frac{L_c}{N}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media es el número de clientes atendiéndose en el sistema.

$$H = \sum_{n=1}^{m-1} np(n) + \sum_{n=m}^{N} mp(n)$$

$$P_{active} = \frac{H}{N} \qquad \text{Porcentaje de clientes siendo atendidos}$$

$$PA = \frac{H}{m} \qquad \text{Porcentaje de actividad}$$

- Determinación de J: el número promedio de usuarios que se encuentran fuera del sistema, viene dado por:

$$J = N - L$$

$$J = \frac{H}{\rho_r}$$

- El porcentaje de clientes que se encuentran fuera del sistema viene dado por la expresión

$$P_{out} = \frac{J}{N}$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$ (o $\overline{\mu}$).

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{J\lambda_r}$$

- Determinación de $\overline{\mu}/\overline{\lambda}$: tasas promedio de ingresos y egresos del sistema.

$$\overline{\mu} = \mu \cdot H = \overline{\lambda} = \lambda_r J$$

- Determinación de W: el tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

$$W = \frac{L}{\overline{\lambda}} = \frac{L}{J\overline{\lambda_r}}$$

- Si el tamaño de N es demasiado grande ($N > 10$), es conveniente usar las aproximaciones enunciadas por Peck y Hazelwood. Para ello se definen el factor de servicio X y el factor de eficiencia F, que cumplen

$$X = \frac{T_s}{T_s + U}$$

$$F = \frac{T_s + U}{T_s + U + W_c}$$

$$H = FNX$$

$$J = FN(1-X)$$

$$L_c = (1-F)N$$

Análisis económico

- En este tipo de sistemas, el número de servidores es la variable de decisión.
- Podemos modificarla para minimizar el coste operativo del sistema.
- Definimos c_e como el coste derivado de esperar en el sistema, y c_c como el coste de atención de un servidor.
- Podemos definir una función de costes tal que:

$$Z = c_e L + c_c m$$

- Problema en enteros positivos.

Ejemplo de aplicación

■ **Ejemplo 8.7 — Ejemplo 4.4 de [3] – M/M/m/-/N.** Una empresa cuenta con 5 máquinas idénticas que necesitan supervisión, en promedio, cada 2 horas. Las operaciones de supervisión tienen una duración promedio de 1 hora. La empresa cuenta con dos operarios para realizar esta actividad. Cada máquina proporciona una ganancia esperada de 500€ por hora.

a) Deducir y calcular las probabilidades asociadas a cada uno de los estados posibles.
Calculamos los parámetros

$$\rho_{\mathrm{r}} = \frac{\lambda_{\mathrm{r}}}{\mu} = \frac{0.5}{1} = 0.5$$

Para calcular las probabilidades empleamos Matlab, para obtener

$$\begin{aligned} p(1) &= 0.2763 \\ p(2) &= 0.2763 \\ p(3) &= 0.2073 \\ p(4) &= 0.1036 \\ p(5) &= 0.0260 \end{aligned}$$

b) Determinar, L, L_c, H y J.

$$\begin{aligned} L &= 0 \cdot p(0) + 1 \cdot p(1) + 2 \cdot p(2) + 3 \cdot p(3) + 4 \cdot p(4) + 5 \cdot p(5) = 1.9948 \\ L_c &= 1 \cdot p(3) + 2 \cdot p(4) + 3 \cdot p(5) = 0.4922 \\ H &= L - Lc = 1,5026 \\ J &= N' - L = 3,0052 \end{aligned}$$

c) Determinar el factor de servicio, el factor de eficiencia y el porcentaje de actividad de cada trabajador.

$$\begin{aligned} X &= \frac{T_S}{T_S + U} = 0.3333 \\ \bar{\mu} &= \mu \cdot H = 1.5026 \\ W_C &= \frac{L_C}{\bar{\mu}} = 0.3276 \\ F &= \frac{T_S + U}{T_S + U + W_C} = 0.9016 \\ PA &= \frac{H}{M} = 0.7513 \end{aligned}$$

d) Calcular las pérdidas.

$$\mathrm{LC} = 500 \cdot \mathrm{L} = 997,41 \frac{€}{\mathrm{h}}$$

■

8.7 Redes de colas

En los sistemas de colas analizados hasta el momento se ha supuesto que cada sistema ofrecía un único tipo de servicio, prestado en un único centro de atención compuesto por uno o varios canales dispuestos en paralelo, con determinadas características operativas tales como capacidad, modalidad de atención y disciplina de servicio.

No obstante, en numerosos sistemas reales coexisten varios centros de atención dentro de un mismo sistema, en los cuales se brindan distintos servicios que, en muchos casos, se encuentran vinculados de manera secuencial. En estas configuraciones, un cliente que ingresa al sistema recibe un primer servicio en un centro de atención (o subsistema) y, una vez finalizado, se dirige a un segundo centro para recibir un nuevo servicio, continuando este proceso hasta completar la secuencia y abandonar el sistema.

Esta organización de centros de atención conectados secuencialmente se denomina disposición **en serie** o **en cascada**.

8.7.1 Redes en serie y en paralelo

Dentro de las redes en serie podemos diferenciar dos tipos: sistemas secuenciales, ver Figura 8.12 y sistemas paralelos, ver Figura, 8.13.



Figura 8.12: Sistema en serie

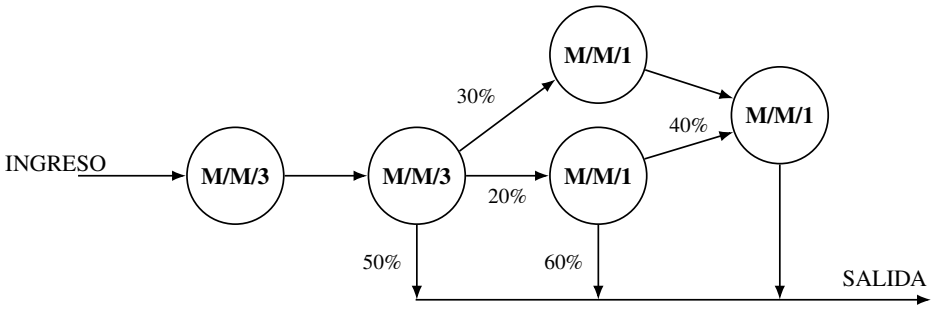


Figura 8.13: Sistema en paralelo

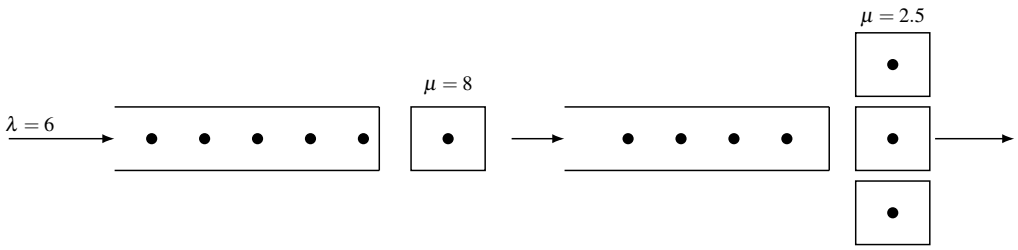


Figura 8.14: Recepción de pacientes

■ **Ejemplo 8.8 — Ejemplo aplicación redes secuenciales.** Supongamos una sala de urgencias, en donde los pacientes deben formar una cola única a la espera de recibir asignación en uno de

3consultorios, ver Figura 8.14. En promedio la llegada de pacientes se realiza a razón de 6 por hora. El recepcionista tarda una 0.125 horas en llamar a un paciente, y el tiempo medio de tratamiento en cada uno de los consultorios es de 0.4 horas. Obtener los parámetros básicos del sistema.

- Subsistema I

$$\rho_1 = 0.75 \qquad L_1 = 3$$
$$H_1 = 0.75 \qquad W_{c1} = 0.375$$
$$p(0)_1 = 0.25$$
$$L_{c1} = 2.25 \qquad W_1 = 0.5$$

Para el segundo subsistema, la tasa de llegadas es igual a la tasa efectiva de salidas del primer subsistema, es decir, $\lambda_2 = \overline{\mu_1} = \mu_1 H = 6$

- Subsistema II

$$\rho_2 = 2.4 \qquad L_2 = 4.99$$
$$H_2 = 2.4 \qquad W_{c2} = 0.4315$$
$$p(0)_2 = 0.0562 \qquad W_2 = 0.8315$$
$$L_{c2} = 2.59 \qquad \overline{\mu_2} = \mu_2 H_2 = 6$$

- Sistema completo

$$H = H_1 + H_2 = 3.15 \qquad L = L_1 + L_2 = 7.99$$
$$p(0) = p(0)_1 p(0)_2 = 0.01405 \qquad W_c = W_{c1} + W_{c2} = 0.8065$$
$$L_c = L_{c1} + L_{c2} = 4.84 \qquad W = W_1 + W_2 = 1.3315$$

■

El trabajo con sistemas en paralelo se basa en los siguientes principios.

1. **Teorema de Burke**: la división de un flujo de clientes de un proceso de Poisson, probabilísticamente lleva a procesos que también son de tipo Poisson, ver Figura 8.15.

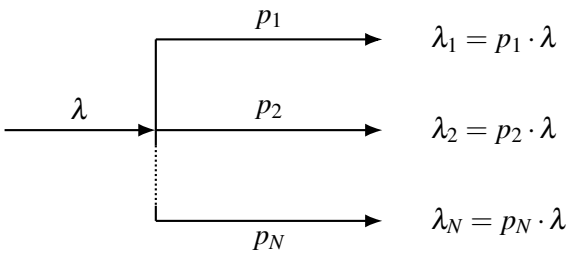


Figura 8.15: División de flujos de Poisson.

2. La combinación de flujos independientes de clientes de procesos de Poisson lleva a un proceso de naturaleza poissoniana, ver Figura 8.16.
3. **Teorema de Jackson**: para un sistema con m centros de atención, la probabilidad de que el sistema se encuentre en el estado $n_1, n_2, \ldots, n_m$ viene dada por:

$$p(n_1, n_2, \ldots, n_m) = \prod_{i=1}^{m} p_i(n_i)$$

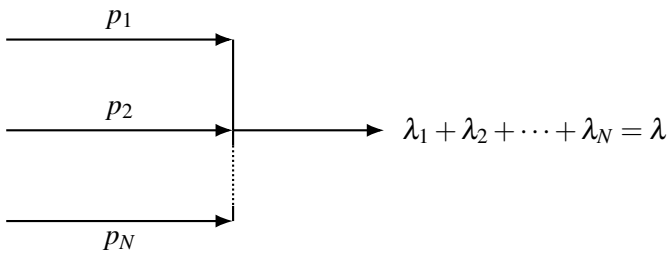


Figura 8.16: Combinación de flujos de Poisson.

4. En condiciones de equilibrio, las tasa efectivas de ingresos y egresos son iguales ($\overline{\mu} = \overline{\lambda}$). Por tanto, se cumple:

$$\overline{\lambda} = \mu \cdot H = \frac{1}{T_s} \cdot H$$
$$T_s = \frac{H}{\overline{\lambda}}$$

5. Los sistemas de colas cumplen la fórmula de Little

$$W = \frac{L}{\overline{\lambda}}$$
$$W_c = \frac{L_c}{\overline{\lambda}}$$

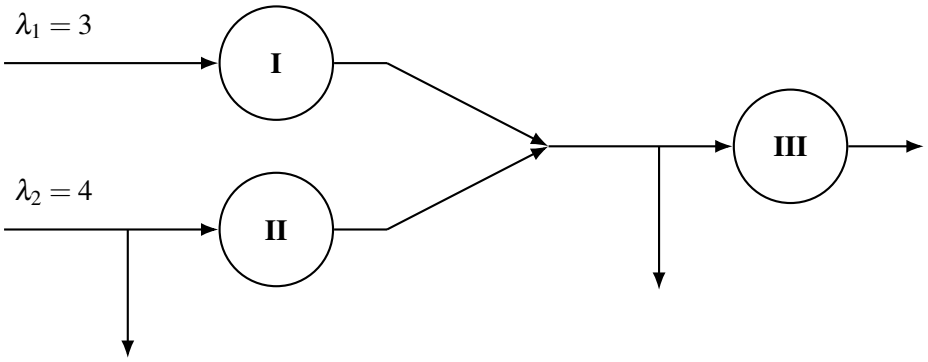


Figura 8.17: Problema de tres estaciones

■ **Ejemplo 8.9 — Problema con tres estaciones.** Considere el sistema de la figura 8.17, en donde la estación 1 es M/M/1 con $\mu_1 = 6$, la estación 2 es M/M/2/2 con $\mu_2 = 2$ y la estación 3 es M/M/1/2 con $\mu_3 = 5$, estando los valores de λ y μ expresados en usuarios por día, sabiendo que cada servicio ofrecido en la estación III se cobran 100€ y que cada canal tiene un coste de 50€ día, calcular la ganancia esperada diaria.

Para resolver este problema tenemos que encontrar la tasa efectiva de clientes que dejan el sistema $\overline{\mu_3}$, lo cual indicará la facturación, luego como el sistema tiene en total $m = m_1 + m_2 + m_3 = 4$ canales, tendrá un coste operativo de 200 €/día.

Notar que como el nodo 3 es M/M/1/2, se deduce que $\overline{\mu_3} = \overline{\lambda_3} = \lambda_3 \cdot [1 - p_3(2)]$. Luego, la estructura del problema indica que se tendrá que encontrar $\lambda_3 = \overline{\mu_1} + \overline{\mu_2}$ y $p_3(2)$.

- Estación 1 $--$ M/M/1, $\lambda_1 = 3$, $\mu_1 = 6$.
  - Dado que no hay restricciones de capacidad $\overline{\mu_1} = \overline{\lambda_1} = \lambda_1 = 3$

- Estación 2 −− M/M/2/2, $\mu_2 = 2$.
  - $p_2(0) = 0.2$, $p_2(1) = 0.4$ y $p_2(2) = 0.4$.
  - $\overline{\mu_2} = \mu_2 \cdot [1 - p_2(2)] = 2.4$
- Estación 3 −− M/M/1/2, $\mu_3 = 5$.
  - $\lambda_3 = \overline{\mu_1} + \overline{\mu_2} = 5.4$.
  - $p_3(0) = 0.308$, $p_3(1) = 0.3326$ y $p_3(2) = 0.3593$.
  - $\overline{\mu_3} = 3.4598$
- Luego se concluye que la tasa de clientes atendidos en el sistema 3, y que por tanto, dejan el sistema es, $\overline{\mu_3} = 3.4598$.
  - Beneficio $B = 100€/\text{día} \cdot \overline{\mu_3} - 50€/\text{día} \cdot m \cdot = 146$ €/día.

■

8.7.2 Redes con realimentación

Las redes con realimentación (también denominado como redes con reciclaje) responden a la siguiente situación, ver Figura 8.18:

- En algunos sistemas reales ocurre que parte de los clientes atendidos en una determinada unidad del sistema tienen que retornar a una unidad anterior, para ser atendidos nuevamente.
- En estos casos, es necesario tener en cuenta el flujo de clientes que van por la rama realimentada.
- Es importante notar que, siendo los procesos de llegadas de tipo Poisson, el flujo resultante se comporta como si fuera de Poisson, aunque no lo es estrictamente hablando.

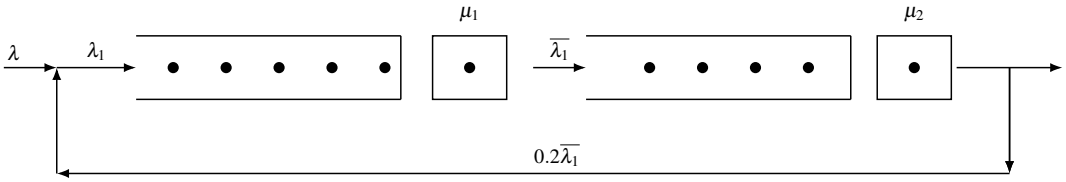


Figura 8.18: Ejemplo de red con retroalimentación

■ **Ejemplo 8.10 — Red simple con realimentación.** Por ejemplo, supongamos un sistema de elaboración de un producto en una máquina, al que llegan piezas con una tasa $\overline{\lambda}$ para ser procesadas a una velocidad $\overline{\mu_1}$. Una vez fabricadas, las piezas son sometidas a un control de calidad, que se realiza a una velocidad promedio de $\overline{\mu_2}$. Un 20% de las piezas inspeccionadas requiere ser reprocesadas, por lo que vuelven a la cola de la máquina 1, ver Figura 8.18.

La filosofía para la resolución de este tipo de sistemas es considerar que la tasa de entrada a los sistemas que reciben la realimentación es una combinación de la tasa media con el flujo que se recibe de otros nodos.

En este ejemplo específico se trata con dos sistemas $M/M/1$, por lo que se tendrá que

$$\overline{\lambda_1} = \overline{\mu_1} = \lambda_1.$$

Luego para el segundo subsistema, se tendrá de manera análoga:

$$\overline{\lambda_2} = \overline{\mu_2} = \overline{\lambda_1} = \lambda_1.$$

- Tasa efectiva de entrada al primer subsistema

$$\lambda_1 = \lambda + 0.2\lambda$$
$$\lambda_1 = \frac{\lambda}{0.8}$$

- Centro 1

$$\rho_1 = \frac{\lambda_1}{\mu_1} \qquad L_1 = \frac{\overline{\lambda}}{\mu_1 - \overline{\lambda}} \qquad H_1 = \rho_1$$

$$W_{c1} = \frac{L_{c1}}{\lambda_1}$$

$$p(n)_1 = \rho_1^n(1-\rho_1) \qquad L_{c1} = \frac{\lambda^2}{\overline{\mu_1}(\overline{\mu_1} - \lambda_1)} \qquad W_1 = W_{c1} + T_{s1}$$

- Centro 2

$$\rho_2 = \frac{\lambda_1}{\overline{\mu_2}} \qquad L_2 = \frac{\lambda_1}{\mu_2 - \lambda_1} \qquad H_2 = \rho_2$$

$$W_{c2} = \frac{L_{c2}}{\lambda_1}$$

$$p(n)_2 = \rho_2^n(1-\rho_2) \qquad L_{c2} = \frac{\lambda_1^2}{\mu_2 \cdot (\mu_2 - \lambda_1)} \qquad W_2 = W_{c1} + T_{s2}$$

■

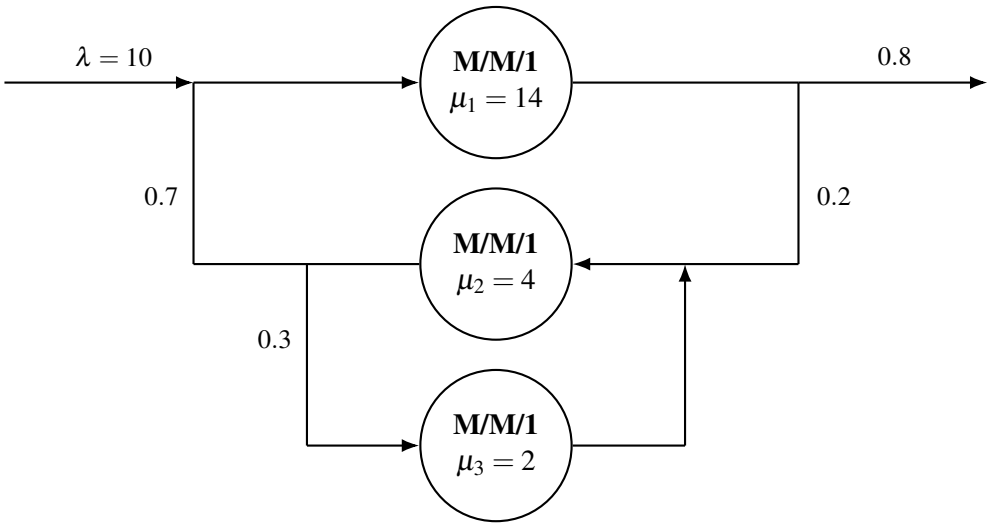


Figura 8.19: Red de Ejemplo 8.11

■ **Ejemplo 8.11 — Sistema con realimentación en paralelo.** Considere el sistema de la Figura 8.19. Se tendrán tres subsistemas

- Subsistema 1: M/M/1 – $\mu_1 = 14$
- Subsistema 2: MM1 – $\mu_1 = 4$
- Subsistema 3: M/M/1 – $\mu_3 = 2$

Los datos están expresados en clientes/hora. Determinar cuántos clientes salen del sistema, calcular la longitud promedio de clientes en cada centro, y encontrar la probabilidad de que el sistema esté vacío.

Solución: Al sistema ingresan en media $\lambda = 10$ clientes por hora. De acuerdo a los flujos de clientes, se tendrá

Centro 1: **Centro 2**: **Centro 3**:

$$\lambda_1 = \lambda + 0.7 \qquad \lambda_2 = \overline{\mu_3} + 0.2\overline{\mu_1} \qquad \lambda_3 = 0.3\overline{\mu_2}$$

$$\overline{\lambda_1} = \overline{\mu_1} = \lambda_1 \qquad \overline{\lambda_2} = \overline{\mu_2} = \lambda_2 \qquad \overline{\lambda_3} = \overline{\mu_3} = \lambda_3$$

Por tanto las ecuaciones de **balance de flujo**, resultan

$$\begin{cases} \lambda_1 = \lambda + 0.7\lambda_2 \\ \lambda_2 = 0.2\lambda_1 + \lambda_3 \\ \lambda_3 = 0.3\lambda_2 \end{cases}$$

Resolviendo este sistema se obtiene:

$$\lambda_1 = 12.5, \qquad \lambda_2 = 3.5714, \qquad \lambda_3 = 1.0714$$

Luego se tiene que, del sistema salen

$$\overline{\mu} = 0.8\overline{\mu_1} = 10$$

clientes por hora.
La **longitud promedio de cada subsistema** será

$$L_1 = 8.3333, \qquad L_2 = 8.3327, \qquad L_3 = 1.1538$$

clientes.
La **probablidad de que el sistema esté vacío** se obtiene aplicando el Teorema de Jackson,

$$p(0,0,0) = p(0)_1 \cdot p(0)_2 \cdot p(0)_3 = (1-\rho_1)(1-\rho_2)(1-\rho_3) = 0.0053$$

donde se ha empleado que

$$\rho_1 = 0.8929, \qquad \rho_2 = 0.8929, \qquad \rho_3 = 0.5357$$

En resumen:

- $\overline{\mu} = 10$ clientes/hora,
- $L_1 = 8.333$, $L_2 = 8.3327$ y $L_3 = 1.1538$ clientes.
- $p(vacio) = 0.0053$.

■

8.8 Modelos de colas con distribuciones no exponenciales

En el estudio clásico de la teoría de colas, gran parte de los resultados analíticos se apoyan en la suposición de **llegadas y tiempos de servicio exponenciales**. Esta hipótesis conduce a modelos matemáticamente manejables (como $M/M/1$, $M/M/m$, etc.) gracias a la propiedad de **falta de memoria** de la distribución exponencial, que permite modelar el sistema como una **cadena de Markov**.

Sin embargo, en muchos sistemas reales de telecomunicaciones, esta suposición no es realista.

El tráfico de datos, la duración de las conexiones, el tamaño de los paquetes, el comportamiento de los usuarios y los tiempos de procesamiento en dispositivos de red presentan con frecuencia **variabilidad, asimetría y dependencia temporal** que no pueden describirse adecuadamente mediante distribuciones exponenciales.

Por ello, es fundamental extender el análisis hacia **modelos con distribuciones generales**, que permiten capturar de manera más fiel el comportamiento de los sistemas reales.

Limitaciones del supuesto exponencial

En redes de telecomunicaciones se observan fenómenos como:

- Tráfico **autocorrelacionado** (por ejemplo, tráfico de video o descargas masivas).
- Tiempos de servicio con **baja variabilidad** (procesamiento casi determinista en routers).
- Tiempos de servicio con **alta variabilidad** (procesamiento variable en servidores, retransmisiones, colisiones).
- Distribuciones **pesadas en la cola** (heavy–tailed), típicas en tamaños de archivos o sesiones de usuario.

Estos comportamientos producen desviaciones significativas respecto a las predicciones de los modelos $M/M/1$ o $M/M/c$. En particular, la variabilidad de los tiempos de servicio impacta directamente en:

- El tiempo medio de espera.
- La longitud media de la cola.
- La probabilidad de congestión.

Notación general: modelos $G/G/1$

Para describir sistemas con distribuciones arbitrarias se utiliza la notación de Kendall extendida:

$$A/B/c$$

donde:

- A describe la distribución de los tiempos entre llegadas.
- B describe la distribución de los tiempos de servicio.
- c es el número de servidores.

La letra G (General) indica que no se asume ninguna forma específica para la distribución.

Así, el modelo más general de un solo servidor es:

$$G/G/1$$

que representa un sistema con llegadas y servicios de distribución arbitraria.

Importancia del coeficiente de variación

En modelos no exponenciales, la forma exacta de la distribución es menos relevante que su **media** y su **variabilidad**. Esta última se mide mediante el **coeficiente de variación**:

$$C_X = \frac{\sigma_X}{\mathbb{E}[X]}$$

donde X puede representar tiempos entre llegadas o tiempos de servicio.

- Para la exponencial: $C_X = 1$.
- Para la determinista: $C_X = 0$.
- Para distribuciones muy variables: $C_X > 1$.

Este parámetro resulta clave porque el rendimiento del sistema depende fuertemente de la variabilidad, incluso cuando la media se mantiene constante.

Aproximación de Kingman para $G/G/1$

Dado que el modelo $G/G/1$ no admite una solución exacta sencilla, se utilizan aproximaciones. La más importante es la **fórmula de Kingman** para el tiempo medio de espera en cola:

$$W_q \approx \frac{\rho}{1-\rho} \cdot \frac{C_A^2 + C_S^2}{2} \cdot \mathbb{E}[S]$$

donde:

- $\rho = \lambda \mathbb{E}[S]$ es la utilización del sistema.
- C_A es el coeficiente de variación de los tiempos entre llegadas.
- C_S es el coeficiente de variación de los tiempos de servicio.
- $\mathbb{E}[S]$ es el tiempo medio de servicio.

Esta expresión muestra de forma explícita que **la variabilidad influye tanto como la carga del sistema**.

8.8.1 Casos particulares relevantes

Algunos modelos intermedios muy utilizados en telecomunicaciones son:

- $M/G/1$: llegadas Poisson y servicios generales.
- $G/M/1$: llegadas generales y servicios exponenciales.
- $D/G/1$: llegadas deterministas y servicios generales.
- $G/D/1$: servicios deterministas.

Por ejemplo, en un router que procesa paquetes a velocidad casi constante, el modelo $M/D/1$ describe mejor el comportamiento que el $M/M/1$.

8.8.2 Implicaciones prácticas en el diseño de redes

El uso de modelos no exponenciales permite:

- Dimensionar buffers con mayor precisión.
- Estimar retardos reales en enlaces y routers.
- Analizar efectos de tráfico multimedia y tráfico bursty.
- Evaluar impacto de protocolos que introducen variabilidad (retransmisiones, backoff, etc.).

En muchos casos, ignorar la variabilidad conduce a **subestimar los retardos y la congestión**, lo que resulta crítico en aplicaciones sensibles al tiempo como VoIP, videoconferencia y servicios en tiempo real.

Los modelos con distribuciones no exponenciales representan una herramienta esencial para analizar sistemas de telecomunicaciones modernos. Aunque su tratamiento matemático es más complejo, el uso de métricas como el coeficiente de variación y aproximaciones como la fórmula de Kingman permite obtener resultados prácticos y aplicables en el diseño y evaluación de redes reales.

VI Apéndice. Bibliografía. Índice de contenidos

9 **Apéndice** 161

A: Hessian Matrix
B: Pyomo solver para Windows. Ayuda.
C: Guía de Geogebra
D: Formulario Teoría de Colas
E: Excel Guide
F: Pyomo Guide
G: Funciones de Matlab

Referencias 178

Index 181

9. Apéndice

A: Hessian Matrix

Definición .0.1 — Hessian Matrix. Suppose $f : U \subset \mathbb{R}^n \to \mathbb{R}$ has second-order continuous derivatives $(\partial^2 f/\partial x_i \partial x_j)(\mathbf{x}_0)$ for $i, j = 1, \ldots, n$ at a point $\mathbf{x}_0 \in U$. The **Hessian matrix** of f at $\mathbf{x}_0$ is defined as

$$\nabla^2 f(\mathbf{x}_0) = H_f(\mathbf{x_0}) \left[\begin{array}{ccc} \dfrac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \frac{\partial^2 f}{\partial x_i \partial x_j} & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 f}{\partial x_n \partial x_n} \end{array} \right]_{\mathbf{x}=\mathbf{x_0}}$$

■ **Ejemplo .1 — Hessian matrix.** Determine the Hessian matrix of the function $f(x,y,z) = x^2 y + yz^3 + \sin(xz)$

Compute the first-order partial derivatives

$$\frac{\partial f}{\partial x} = 2xy + z\cos(xz) \qquad \frac{\partial f}{\partial y} = x^2 + z^3 \qquad \frac{\partial f}{\partial z} = 3yz^2 + x\cos(xz)$$

Compute the second-order partial derivatives

$$\frac{\partial^2 f}{\partial x^2} = 2y - z^2\sin(xz) \qquad \frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = 2x \qquad \frac{\partial^2 f}{\partial x \partial z} = \frac{\partial^2 f}{\partial z \partial x} = \cos(xz) - xz\sin(xz)$$

$$\frac{\partial^2 f}{\partial y^2} = 0 \qquad \frac{\partial^2 f}{\partial y \partial z} = \frac{\partial^2 f}{\partial z \partial y} = 3z^2 \qquad \frac{\partial^2 f}{\partial z^2} = 6yz - x^2\sin(xz)$$

The Hessian matrix is

$$H_f = \begin{bmatrix} 2y - z^2\sin(xz) & 2x & \cos(xz) - xz\sin(xz) \\ 2x & 0 & 3z^2 \\ \cos(xz) - xz\sin(xz) & 3z^2 & 6yz - x^2\sin(xz) \end{bmatrix}$$

■

R Note that if the function $f \in C^2(D)$, continuous second partial derivatives, then Clairaut's theorem holds, and the hessian matrix must be **symmetric**.

Positive Definite and Negative Definite Matrices via Principal Minors

Definición .0.2 — Leading Minor. Let $A \in \mathbb{R}^{n\times n}$ be a square matrix. The *k-leading minor* of A, denoted Δ_k, is the determinant of the $k \times k$ leading principal submatrix of A. Specifically, it is the submatrix formed by the first k rows and the first k columns of A.

More formally, if $A = [a_{ij}]$ is the matrix, then the k-leading principal submatrix A_k is:

$$A_k = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix}.$$

The k-leading minor Δ_k is then given by:

$$\Delta_k = \det(A_k).$$

Definición .0.3 — Positive Definite Matrix. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *positive definite* if and only if all its principal minors are positive. That is, for each k (where $1 \leq k \leq n$), the leading principal minors Δ_k of A satisfy:

$$\Delta_k > 0.$$

Here, Δ_k is the determinant of the $k \times k$ leading principal submatrix of A.

Definición .0.4 — Negative Definite Matrix. A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *negative definite* if and only if the signs of the principal minors alternate, starting with a negative sign for the first principal minor. Specifically, for each k (where $1 \leq k \leq n$), the leading principal minors Δ_k satisfy:

$$\Delta_k \text{ is } (-1)^{k+1} \text{ times a positive number.}$$

In other words:

- $\Delta_1 < 0$
- $\Delta_2 > 0$
- $\Delta_3 < 0$
- $\ldots$
- $\Delta_n = (-1)^{n+1}$ times a positive number

R The signs of the principal minors provide a convenient criterion for definiteness of symmetric matrices. Positive definiteness is indicated by all positive principal minors, while negative definiteness is indicated by alternating signs of principal minors, starting with a negative sign.

■ **Ejemplo .2** Consider the function $f(x,y,z) = x^2 + y^2 + z^2$. The gradient is:

$$\nabla f(x,y) = \begin{bmatrix} 2x \\ 2y \\ 2z \end{bmatrix}.$$

The only critical point is $(0,0,0)$. The Hessian matrix is:

$$H_f(0,0,0) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

The determinant of the leading minors are

$$\Delta_1 = |2| = 2 > 0 \qquad \Delta_2 = \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4 > 0 \qquad \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{vmatrix} = 8 > 0$$

Therefore, the Hessian matrix is definite positive and $(0,0,0)$ is a local minimum. ∎

B: Pyomo solver para Windows. Ayuda.

Esta sección proporciona una guía para utilizar la interfaz gráfica de usuario (GUI) basada en Windows, diseñada para construir y resolver modelos de investigación operativa con Pyomo. La GUI simplifica la creación de modelos al permitir que los usuarios introduzcan los componentes del problema mediante campos estructurados.

Importante — Pyomo solver no estará disponible para su uso en exámenes. El solver de Pyomo no estará disponible durante los exámenes. Durante los exámenes se empleará como optimizador única y exclusivamente los módulos de Matlab descritos en clase.

Introduciendo el modelo

1. **Number of Variables:** Enter an integer representing the number of decision variables. *Example:* For variables x_1, x_2, enter `2`.
2. **Problem Category:** Choose between `Linear` or `Non-Linear` depending on the nature of your objective function and constraints.
3. **Domain:** Select the type of variables from the following options:

Domain	Description
`NonNegativeReals`	All real numbers ≥ 0
`NonPositiveReals`	All real numbers ≤ 0
`Reals`	All real numbers
`PositiveReals`	All real numbers > 0
`NegativeReals`	All real numbers < 0
`Integers`	All integers
`NonNegativeIntegers`	All integers ≥ 0
`PositiveIntegers`	All integers > 0
`Binary`	Only 0 or 1

Tabla 1: Summary of variable domains in Pyomo

4. **Direction (sense):** Specify whether the problem is a `Minimize` or `Maximize` problem.
5. **Objective Function:** Enter the objective using variable names `x1, x2, ...`
   Supported mathematical functions include:
   - `exp()`, `log()`, `sqrt()`, `sin()`, `cos()`, `tan()`, `abs()`
   - Multiplication: `*`

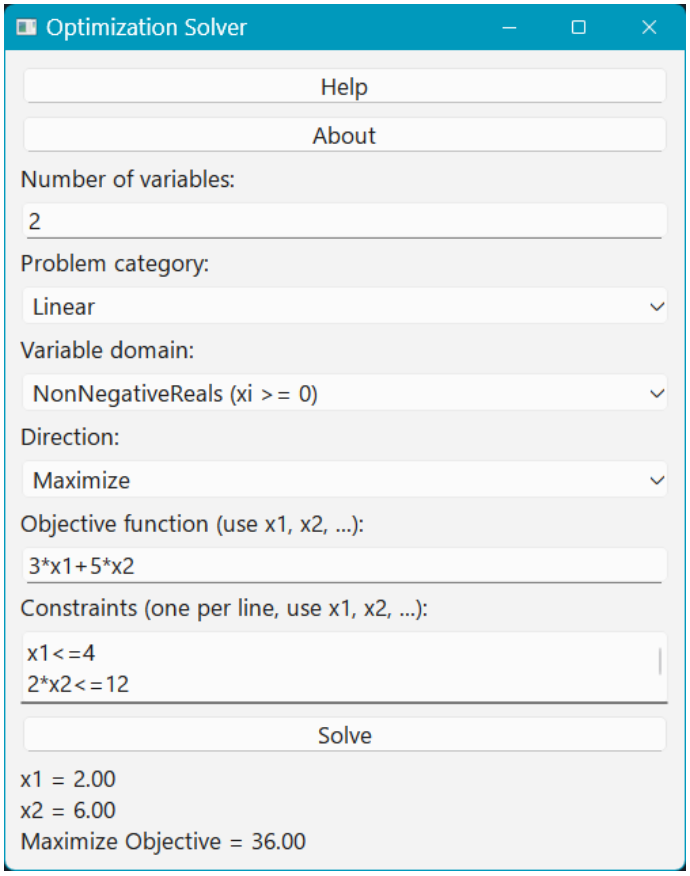


Figura 1: Resolución de Ejemplo 1.1 con solver de Pyomo

- Power: `**`

Examples: `3*x1 + 2*x2` `exp(x1) + x2**2`

6. **Constraints:** Enter one constraint per line using the same variable names.
   Examples:

```
x1 + x2 <= 10
x1 - x2 >= 2
```

7. **Valid Constraint Operators:** Use one of the following:
   - <= for less than or equal to
   - >= for greater than or equal to
   - == for equality constraints

This GUI is intended to streamline the modeling process for users who prefer a visual interface over scripting. Once the model is entered, the GUI will use supported solvers to find a solution.

See Figure 1 as reference.

C: Guía de Geogebra

Esta es una breve guía para el uso del graficador de Geogebra.

> **Importante** Geogebra estará disponible durante los exámenes.

Graficando la región factible y la función objetivo

1. **Acceder al GeoGebra Graphing Calculator**
   Disponible en: `https://www.geogebra.org/graphing`
2. **Introducir las restricciones**
   Escriba cada restricción como una desigualdad en la barra de entrada. Por ejemplo:
   - `x + y <= 10`
   - `x >= 0`
   - `y >= 0`

   GeoGebra etiquetará las restricciones, por ejemplo, como `a`, `b` y `c`. Para encontrar la intersección de las restricciones, utilice
   `a && b && c`
   Esto crea la intersección de las restricciones como un nuevo objeto. A continuación, haz clic en los círculos junto a las restricciones para desactivar su visualización. **Nota:** `&&` es equivalente al operador lógico `AND`, lo que significa intersectar las regiones factibles.
3. **Representar la Función Objetivo como una Curva de Nivel**
   Supongamos que la función objetivo es: `f(x, y) = 3x + 2y`.
   Para visualizarla, se puede representar una curva de nivel, por ejemplo:
   - `3x + 2y = k`

   Esto genera un control deslizante. Ajusta los valores de `k` para observar cómo se desplaza la recta. Este procedimiento permite identificar el punto óptimo en la frontera de la región factible.
   Este ejemplo está disponible en este enlace.

D: Formulario Teoría de Colas

> **Importante** Este formulario se entregará para su uso durante el examen. Es la **única versión oficial**.

Modelo M/M/1

$$p(n) = \rho^n(1-\rho)$$

$$p(n > k) = \rho^{k+1}$$

$$p(n \geq k) = \rho^k$$

$$\rho < 1 = \lambda < \mu$$

$$L = \frac{\lambda}{\mu - \lambda}$$

$$L_c = L - \rho = \frac{\lambda^2}{(\mu-\lambda)\mu}$$

$$H = \rho = L - L_c$$

$$PA = \frac{H}{m} = \rho$$

$$\overline{\lambda} = \lambda$$

$$\overline{\mu} = \mu[1 - p(0)]$$

$$W_c = \frac{\lambda}{\mu(\mu-\lambda)}$$

$$W = W_c + T_s = \frac{L}{\overline{\lambda}} = \frac{L}{\lambda}$$

$$\omega(t) = p(W > t) = e^{-(\mu-\lambda)t}$$

$$\omega_c(t) = p(W_c > t) = \rho e^{-(\mu-\lambda)t}$$

$$Z = c_e L + c_s \overline{\mu}$$

$$\mu_0 = \lambda + \sqrt{\lambda \frac{c_e}{c_s}}$$

Modelo M/M/1/K

$$p(n) = \begin{cases} \rho^n \frac{1}{K+1} & \text{Si } \rho = 1 \\ \rho^n \frac{1-\rho}{1-\rho^{K+1}} & \text{si no} \end{cases}$$

$$\overline{\mu} = \mu[1 - p(0)] = \mu \cdot H$$

$$\overline{\lambda} = \lambda[1 - p(K)]$$

$$L = \rho \frac{1 - (K+1)\rho^K + K\rho^{K+1}}{(1-\rho)(1-\rho^{K+1})}$$

$$L_c = L - (1 - p(0))$$

$$H = 1 - p(0) = L - L_c$$

$$L = L_c + H$$

$$PA = \frac{H}{m} = 1 - p(0)$$

$$W = W_c + T_s$$

$$W = \frac{L_c}{\overline{\lambda}} = \frac{L}{\overline{\lambda}} = \frac{L}{\lambda[1 - p(K)]}$$

$$W_c = \frac{L_c}{\overline{\mu}} = \frac{L_c}{\mu[1 - p(0)]} = \frac{L_c}{\mu\left(1 - \frac{1-\rho}{1-\rho^{K+1}}\right)}$$

$$\overline{R} = \lambda p(K) = \lambda \cdot \rho^K \frac{1-\rho}{1-\rho^{K+1}}$$

$$PR = \frac{\overline{R}}{\lambda}$$

$$\min(Z) = c_L(K-1) + u\overline{R}$$

Modelo M/M/m

$$p(n) = \begin{cases} \frac{\rho^n}{n!} p(0), & \text{si } n < m \\ \left(\frac{\rho}{m}\right)^n \frac{m^m}{m!} p(0), & \text{si } n \geq m \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} \frac{\rho^n}{n!} + \frac{\rho^m}{(m-1)!(m-\rho)}\right]^{-1}$$

$$p(n \geq k) = \frac{\rho^k}{(k-1)!(k-\rho)} p(0)$$

$$L = E[p(n)] = \frac{\rho^m \mu \lambda}{(m-1)!(m\mu - \lambda)^2} p(0) + \rho$$

$$L_c = L - \rho$$

$$H = \rho$$

$$L = L_c + H$$

$$PA = \frac{H}{m}$$

$$W_c = \frac{L_c}{\lambda}$$

$$W = W_c + T_s$$

$$W = \frac{L}{\lambda}$$

$$\omega(t) = e^{-\overline{\mu}t}\left[\frac{1 + (m\rho)^m p(0)[1 - e^{-\overline{\mu}(m-1-m\rho)t}]}{m!(1-\rho)(m-1-m\rho)}\right]$$

$$\omega_c(t) = \frac{e^{-\overline{\mu}(m-\rho)t}\rho^m p(0)}{(m-1)!(m-\rho)}$$

$$Z = c_e L + c_c m$$

$$\overline{\mu} = \mu H$$

$$\overline{\lambda} = \lambda$$

$$\frac{\rho}{m} < 1$$

Modelo M/M/m/K

$$p(n) = \begin{cases} \frac{\rho^n}{n!} p(0) & \text{Si } n < m \\ \left(\frac{\rho}{m}\right)^n \frac{m^m}{m!} p(0) & \text{Si } m \leq n \leq K \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} \frac{\rho^n}{n!} + \frac{\rho^m}{(m-1)!}\left(1 - \left[\frac{\rho}{m}\right]^{K-m+1}\right)\frac{1}{(m-\rho)}\right]^{-1}$$

$$L = \sum_{n=0}^{K} n p(n)$$

$$L_c = \sum_{n=m}^{K} (n-m) p(n)$$

$$H = \rho\,[1 - p(K)]$$

$$PA = \frac{H}{m}$$

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{\lambda(1 - p(K))}$$

$$W = W_c + T_s$$

$$\overline{R} = \lambda p(K) = \lambda - \overline{\lambda}$$

$$\overline{\mu} = \mu H$$

$$\overline{\lambda} = \lambda[1 - p(K)]$$

$$Z = c_e L + c_c m$$

$$Z = c_L(K - m) + u\overline{R}$$

$$Z = c_s M\mu + u\overline{R}$$

Modelo M/M/1/-/N

$$p(n) = \frac{N!}{(N-n)!}\rho_r^n p(0)$$

$$p(0) = \frac{1}{\sum_{n=0}^{N} \frac{N!}{(N-n)!}\rho_r^n}$$

$$L = N - \frac{1 - p(0)}{\rho_r}$$

$$P_{in} = \frac{L}{N}$$

$$L_c = N - \frac{(1+\rho_r)[1 - p(0)]}{\rho_r}$$

$$P_{queue} = \frac{L_c}{N}$$

$$H = 1 - p(0)$$

$$P_{active} = \frac{H}{N}$$

$$PA = \frac{H}{m}$$

$$J = N - L$$

$$J = \frac{H}{\rho_r}$$

$$P_{out} = \frac{J}{N}$$

$$W_c = \frac{L_c}{J\lambda_r}$$

$$W = W_c + T_s$$

$$Z = c_e L + c_s \overline{\mu}$$

$$\overline{\mu} = \mu[1 - p(0)]$$

$$\overline{\lambda} = \lambda_r J$$

Modelo M/M/m/-/N

$$p(n) = \begin{cases} C_N^n \rho_r^n p(0) = \frac{N!}{(N-n)!n!}\rho_r^n p(0) \text{ Si } n < m \\ \frac{n!m^m}{m!m^n} C_N^n \rho_r^n p(0) \text{ Si } m \le n \le N \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} C_N^n \rho_r^n + \sum_{n=m}^{N} C_N^n \frac{n!m^m}{m!m^n}\rho_r^n\right]^{-1}$$

$$L = \sum_{n=0}^{N} np(n) = \sum_{n=0}^{m-1} nC_N^n \rho_r^n p(0) + \sum_{n=m}^{N} nC_N^n \frac{n!m^m}{m!m^n}\rho_r^n p(0)$$

$$P_{in} = \frac{L}{N}$$

$$L_c = \sum_{n=m}^{N} (n-m)p(n) = \sum_{n=m}^{N} (n-m)\cdot C_N^n \cdot \frac{n!}{m!}\cdot\frac{m^m}{m^n}\cdot\rho_r^n \cdot p(0)$$

$$P_{queue} = \frac{L_c}{N}$$

$$H = \sum_{n=1}^{m-1} np(n) + \sum_{n=m}^{N} mp(n) = L - L_c$$

$$P_{active} = \frac{H}{N}$$

$$PA = \frac{H}{m}$$

$$J = N - L$$

$$J = \frac{H}{\rho_r}$$

$$P_{out} = \frac{J}{N}$$

$$W_c = \frac{L_c}{J\overline{\lambda_r}}$$

$$W = W_c + T_s$$

$$W = \frac{L}{J\overline{\lambda_r}}$$

$$X = \frac{T_s}{T_s + U}$$

$$F = \frac{T_s + U}{T_s + U + W_c}$$

$$H = FNX$$

$$J = FN(1 - X)$$

$$L_c = (1 - F)N$$

$$Z = c_e L + c_c m$$

$$\overline{\mu} = \mu H$$

$$\overline{\lambda} = \lambda_r J$$

Redes de colas

$$p(n_1, n_2, \ldots, n_m) = \prod_{i=1}^{m} p_i(n_i)$$

$$\overline{\lambda} = \overline{\mu} = \overline{\mu} H$$

E: Excel Guide

This is a guide on how to use the Excel provided template in Aula Virtual and the Free Excel software available on laboratories.

Importante — Excel template is not provided on the exam. The laboratories include a distribution of FreeOffice. On the exam the Excel template is **not** provided.

The first time, before executing the excel, we must enable to solver add-on in Excel.

1. **Open Excel**
2. **Go to Options**: Click on `File` → `Options`.
3. **Access Add-ins**: In the Excel Options window, click on `Add-ins` from the left sidebar.
4. **Manage Add-ins** (Complementos): At the bottom, in the "Manage" dropdown, select `Excel Add-ins` and click `Go`.
5. **Enable Solver**: In the Add-ins window, check the box next to `Solver Add-in`. Click `OK`.
6. **Verify Installation**: Go to the `Data` tab in the Excel ribbon. You should now see a `Solver` button in the "Analysis" group.

■ **Ejemplo .3 — Applying the Excel template to Example 1.1.** In Aula Virtual there is a template on how to use the excel solver. After installing the solver complement in the excel distribution and introducing the correct formulas in the cells. Find the `Solver` button in the `Data` tab, see Figure 2.

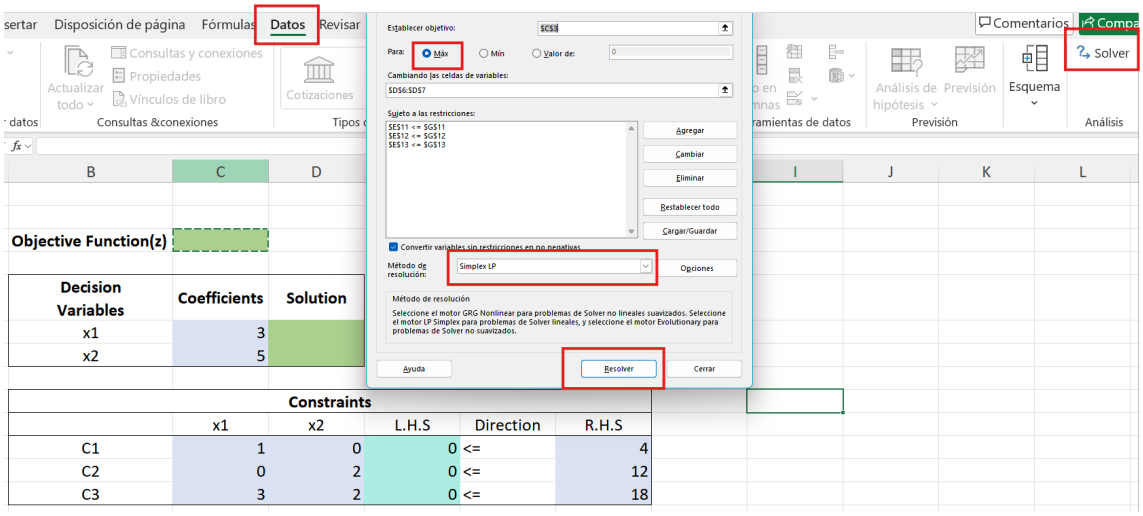


Figura 2: The excel template applied to Example 1.1

Special cells

- C2: contains the **objective function** formula as = `C6*D6+C7*D7`
- E11: contains LHS of the first constraint, as a formula, `=C11*D6+D11*D7`
- E12: contains LHS of the second constraint, as a formula, `=C12*D6+D12*D7`
- E13: contains LHS of the third constraint, as a formula, `=C13*D6+D13*D7`

Introduce as goal the cell with the objective function formula. Select `Max` as the direction of the problem. Introduce the cells to store the value of the variables. Introduce the formula cells and the directions of the constraints representing the relations between LHS and RHS. Select `Simplex LP` as the algorithm and press Solve. If everything is correct the output would look like Figure 3.

From this we conclude that the optimal solution is $\mathbf{x}^* = (2,6)$ for $z^* = 36$. ■

To use Free Excel, follow the instructions provided in Figure 4 and 5. Remember the formulations of the objective function and the LHS of the constraints.

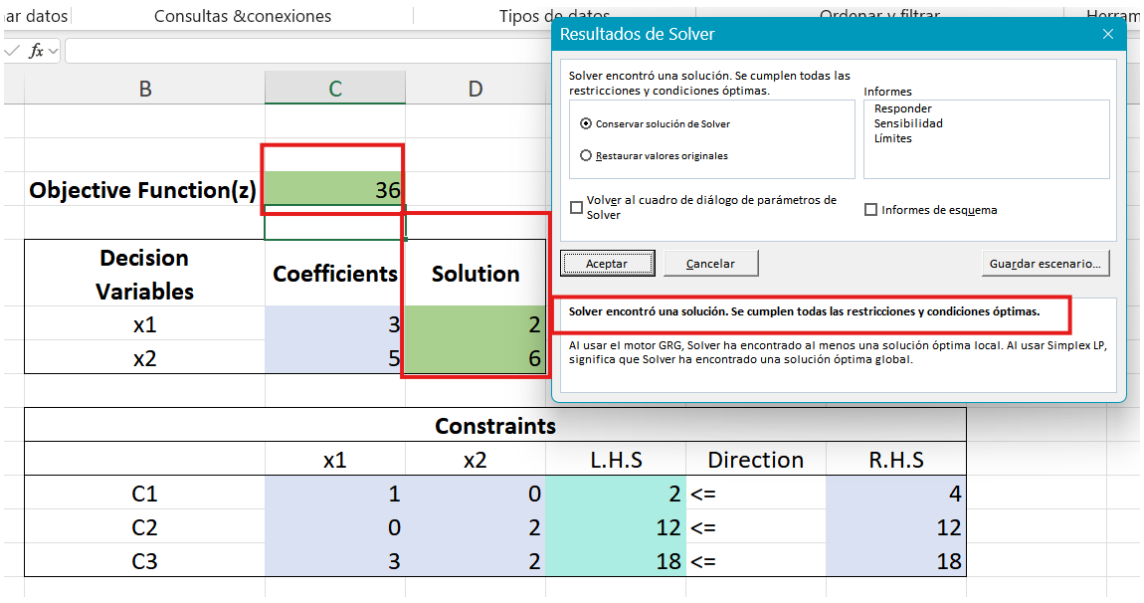


Figura 3: The excel solution of Example 1.1

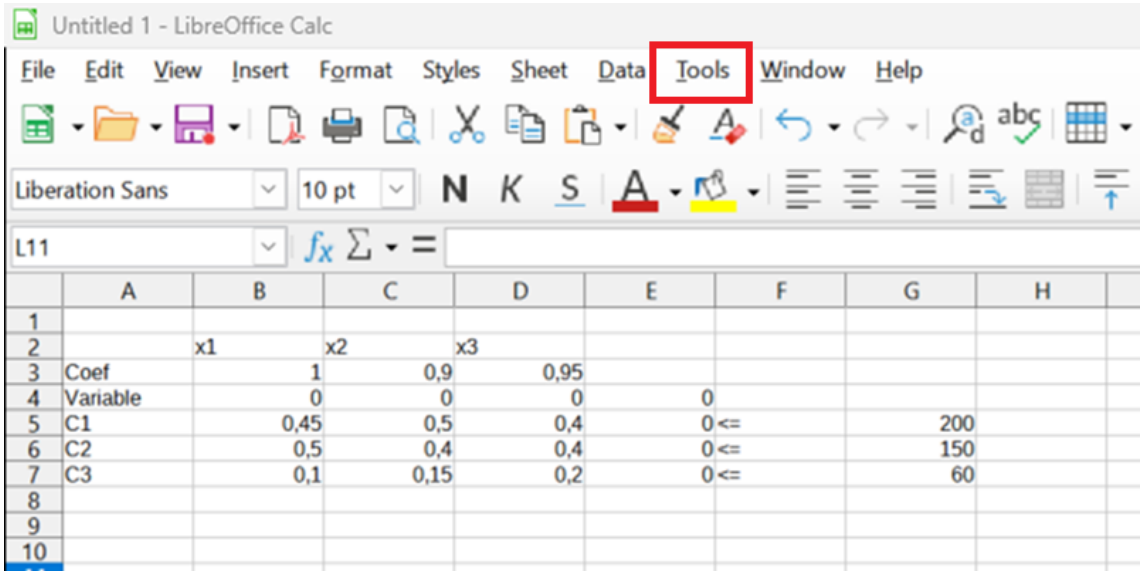


Figura 4: Free Excel, setting the problem

F: Pyomo Guide

Pyomo (*Python Optimization Modeling Objects*) is an open-source Python-based package designed for the formulation and solution of optimization problems. It allows users to express mathematical models using a syntax that is close to standard algebraic notation, while taking advantage of the flexibility and ecosystem of the Python programming language.

Pyomo was originally developed at Sandia National Laboratories, a U.S. Department of Energy research center. The project started in the mid-2000s as part of the **COIN-OR** (Computational Infrastructure for Operations Research) initiative, which promotes open-source software for optimization. The first public release of Pyomo was in 2008, with the goal of providing a Python-based alternative to AMPL, a well-known algebraic modeling language. Unlike AMPL, Pyomo is fully embedded in Python, allowing users to combine optimization modeling with data processing,

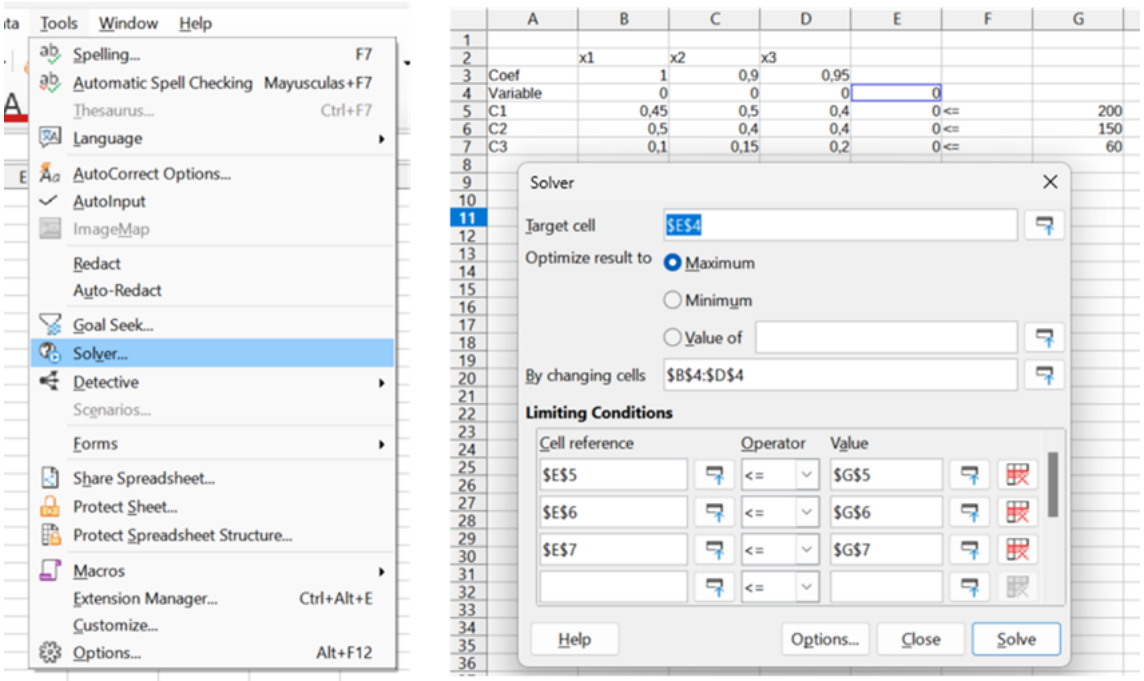


Figura 5: Free Excel, using the solver

visualization, and other computational tasks.

Pyomo is widely used in both academia and industry because it combines ease of use, extensibility, and the ability to handle a broad spectrum of optimization problems. Among the most important features, Pyomo can model:

- **Linear Programming (LP)** problems
- **Integer and Mixed-Integer Linear Programming (MILP/MIP)** problems
- **Nonlinear Programming (NLP)** problems
- **Mixed-Integer Nonlinear Programming (MINLP)** problems
- **Stochastic, dynamic, and network optimization** problems

One of the greatest advantages of Pyomo is that it is not tied to a specific solver. Instead, it acts as a modeling language, allowing users to connect seamlessly to a wide variety of optimization solvers such as **GLPK, CBC, IPOPT, Gurobi, and CPLEX**. This provides flexibility for both research and practical applications.

Since Pyomo is written entirely in Python, it integrates naturally with other tools for:

- **Data analysis** (NumPy, pandas)
- **Visualization** (Matplotlib, Seaborn)
- **Simulation and machine learning** (SciPy, scikit-learn, TensorFlow)

This makes Pyomo not only a modeling environment but also a complete optimization workflow tool, where data can be pre-processed, optimization problems can be solved, and results can be analyzed in a single framework. Overall, Pyomo provides an accessible, powerful, and flexible way to apply optimization methods to real-world engineering problems, making it an ideal tool for both learning and practice.

This guide details the steps to use the Pyomo package through a pre-installed Miniconda environment on the teaching lab computers at ETSIT, which will be used during the Optimization course.

Anaconda and Installation

During the course, we will use the Python programming language. It is assumed that students already have knowledge of Python. Python version 3.6 will be used for the exercises within the Miniconda development environment, a lightweight version of Anaconda.

Anaconda is a free distribution framework for the R and Python languages. One of its main advantages is its module management, making it easy to handle modules using the `conda` tool. It is widely used in data science and machine learning. See this site for more information.

This guide allows students to access a pre-configured Miniconda environment to complete the course exercises.

Environments are Anaconda setups with a specific set of modules and a Python version. This way, different configurations can coexist on the same computer. For instance, suppose the Physiological Signal Processing course requires Python 3.7 with `scipy` and `numpy` modules, while the Optimization course needs Python 3.6 with `pyomo` and `glpk`. Environments allow clear and efficient management of different setups without conflicts. Anaconda enables easy switching between environments.

The following steps explain how to access the specific environment for the Optimization course.

Step 1: Connect to the Lab Computers

First, we need to access the teaching lab computers. The most intuitive way is through VNC access at this link: https://labs.etsit.urjc.es/vnc/.

Once inside, we need to enter our specific username and password for these Linux machines.

Step 2: Add Miniconda to the PATH

The Linux lab computers have a Miniconda distribution located at `/opt/miniconda3/`. The first step is to add the path to access Miniconda's executables. In a terminal, execute the following command:

```
~$ export PATH="$PATH:/opt/miniconda3/bin"
```

After this, to initialize the shell for Miniconda3 usage in the user profile, execute:

```
~$ conda init
```

With this, it won't be necessary to re-execute the `export` command the next time you log in.

Step 3: Access the Environment for the Optimization Course

In this step, we will access the specific environment for the Optimization course, which contains the necessary packages to solve all course exercises.

To run `conda` commands, we must be within the Miniconda path.

When multiple environments are available, it's necessary to specify which one we want to activate (only one environment can be active at a time).

We can list all available environments with:

```
~$ conda env list
```

The output should look something like this:

```
# conda environments:
#
base *                  /opt/miniconda3/
opt_env                 /opt/miniconda3/envs/opt_env
```

The asterisk indicates the active environment.
To activate the `opt_env` environment, execute the following command:

```
~$ source activate opt_env
```

The active environment is displayed in the terminal path.

Step 4: Open Spyder

For easier creation and execution of code, we can use the Spyder editor. To do this, simply run the following command in the terminal:

```
~$ spyder
```

With this, we have a basic setup to complete the course exercises.

Template

A pyomo template is given as supplementary material in Aula Virtual, to use in laboratories and personal computers.

Importante — Pyomo template is not provided on exams. The python distribution in the laboratories include **Pyomo** and it will be available on the exam. However, the **pyomo template** provided in Aula Virtual will **NOT** be available. It is expected that the student is capable of write a full script tailored to the problem is trying to solve.

Below is a line-by-line explanation of the Pyomo script for a linear programming maximization problem of Example 1.1.

■ **Ejemplo .4 — Using Pyomo template.**

- **Imports:**

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

Código 1: Import Pyomo modules

These lines import all necessary Pyomo classes and functions. ‘pyomo.environ‘ includes ‘ConcreteModel‘, ‘Var‘, ‘Objective‘, ‘Constraint‘, etc. ‘pyomo.opt‘ provides ‘SolverFactory‘ to solve optimization models.

- **Create model:**

```
model = ConcreteModel()
```

Código 2: Create a concrete model

Creates a concrete Pyomo model instance, meaning all variables, constraints, and objectives are fully defined at creation.

- **Define decision variables:**

```
model.x_1 = Var(within=NonNegativeReals)
model.x_2 = Var(within=NonNegativeReals)
```

Código 3: Define decision variables

Defines the decision variables x_1 and x_2, constrained to be non-negative real numbers (≥ 0). The options for variables domains are given in Table 2.

- **Define objective function:**

Domain	Description
`NonNegativeReals`	All real numbers ≥ 0
`NonPositiveReals`	All real numbers ≤ 0
`Reals`	All real numbers
`PositiveReals`	All real numbers > 0
`NegativeReals`	All real numbers < 0
`Integers`	All integers
`NonNegativeIntegers`	All integers ≥ 0
`PositiveIntegers`	All integers > 0
`Binary`	Only 0 or 1

Tabla 2: Summary of variable domains in Pyomo

```
model.obj = Objective(expr= 3*model.x_1 + 5*model.x_2, sense=
    maximize)
```

Código 4: Objective function

Sets the objective function to maximize: $3x_1 + 5x_2$. Supported mathematical functions include:

- `exp()`, `log()`, `sqrt()`, `sin()`, `cos()`, `tan()`, `abs()`
- Multiplication: `*`
- Power: `**`

The option sense can be set as `maximize` or `minimize` according to the problem. In this case `sense=maximize` indicates that this is a maximization problem.

- **Define constraints:**

```
model.con1 = Constraint(expr= model.x_1 <= 4)
model.con2 = Constraint(expr= 2*model.x_2 <= 12)
model.con3 = Constraint(expr=3*model.x_1 + 2*model.x_2 <= 18)
```

Código 5: Constraints

Defines the feasible region of the problem:

- con1: $x_1 \leq 4$
- con2: $2x_2 \leq 12$
- con3: $3x_1 + 2x_2 \leq 18$

- **Specify solver:** In this course, we deal with three options

Solver	Description
`glpk`	GNU Linear Programming Kit. Use it for **linear problems**
`ipopt`	Interior Point Optimizer. General solver, preferred for **non-linear problems**
`mindtpy`	Mixed-Integer Optimization problems. Is a combination of `glpk` with `ipopt`

Tabla 3: Solvers in Pyomo available for this course

```
solver = SolverFactory('glpk')
```

Código 6: Select solver

Creates a solver object using the open-source solver GLPK.

- **Solve the model:**

```
results = solver.solve(model)
```

Código 7: Solve the model

Solves the optimization problem and stores the results.

- **Display the solution:**

```
print()
print('*** Solution *** :')
print('x_1:', value(model.x_1))
print('x_2:', value(model.x_2))
print('objective:', value(model.obj))
```

Código 8: Print solution

Prints the solution:

  - x_1 and x_2 values at the optimum
  - Optimal value of the objective function

This script defines the maximization problem with two decision variables, three constraints. After executing the script we get the solution

```
*** Solution *** :
x_1: 2.0
x_2: 6.0
objective: 36.0
```

Código 9: Solution of the Pyomo model

■

G: Funciones de Matlab

Esta sección es una guía muy resumida para las principales funciones de matlab a emplear durante el curso. PAra información más detallada consultar directamente Mathworks.

Función `linprog` en MATLAB

La función `linprog` resuelve problemas de **programación lineal** de la forma:

$$\min\ f^T x$$

sujeto a:

$$Ax \leq b, \qquad A_{eq}x = b_{eq}, \qquad \ell \leq x \leq u$$

Sintaxis básica

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,options)
[x,fval,exitflag,output] = linprog(...)
```

Parámetros

Parámetro	Descripción
`f`	Vector de coeficientes de la función objetivo
`A, b`	Restricciones de desigualdad $Ax \leq b$
`Aeq, beq`	Restricciones de igualdad $A_{eq}x = b_{eq}$
`lb, ub`	Cotas inferior y superior de las variables
`options`	Opciones del solver (creadas con `optimoptions`)

Resultados

Salida	Significado
`x`	Solución óptima
`fval`	Valor mínimo de la función objetivo
`exitflag`	Estado de terminación del algoritmo
`output`	Información del proceso de optimización

Ejemplo

```
f = [-1 -2];
A = [1 1; -1 2; 2 1];
b = [2; 2; 3];
lb = [0 0];

x = linprog(f,A,b,[],[],lb,[])
```

Este ejemplo resuelve el problema de minimizar $-x_1 - 2x_2$ bajo las restricciones dadas.

Función `intlinprog` en MATLAB

La función `intlinprog` resuelve problemas de **programación lineal entera mixta** (MILP) de la forma:

$$\min \ f^T x$$

sujeto a:

$$Ax \leq b, \qquad A_{eq}x = b_{eq}, \qquad \ell \leq x \leq u$$

donde algunas variables están restringidas a tomar **valores enteros**.

Sintaxis básica

```
x = intlinprog(f,intcon,A,b)
x = intlinprog(f,intcon,A,b,Aeq,beq)
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,options)
[x,fval,exitflag,output] = intlinprog(...)
```

Parámetros

Parámetro	Descripción
`f`	Vector de coeficientes de la función objetivo
`intcon`	Índices de las variables que deben ser enteras
`A, b`	Restricciones de desigualdad $Ax \leq b$
`Aeq, beq`	Restricciones de igualdad $A_{eq}x = b_{eq}$
`lb, ub`	Cotas inferior y superior de las variables
`options`	Opciones del solver (creadas con `optimoptions`)

Resultados

Salida	Significado
`x`	Solución óptima encontrada
`fval`	Valor mínimo de la función objetivo
`exitflag`	Estado de terminación del algoritmo
`output`	Información del proceso de optimización

Ejemplo

```
f = [-1 -2];
intcon = [1 2];
A = [1 1; -1 2; 2 1];
b = [2; 2; 3];
lb = [0 0];

x = intlinprog(f,intcon,A,b,[],[],lb,[])
```

Este ejemplo minimiza $-x_1 - 2x_2$ imponiendo que ambas variables sean enteras y satisfagan las restricciones dadas.

Función `fminunc` en MATLAB

La función `fminunc` resuelve problemas de **optimización no lineal sin restricciones** de la forma:

$$\min\ f(x)$$

donde $f(x)$ es una función no lineal y derivable.

Sintaxis básica

```
x = fminunc(fun,x0)
x = fminunc(fun,x0,options)
[x,fval,exitflag,output] = fminunc(...)
```

Parámetros

Parámetro	Descripción
`fun`	Función objetivo a minimizar (handle de función)
`x0`	Punto inicial
`options`	Opciones del solver (creadas con `optimoptions`)

Resultados

Salida	Significado
`x`	Punto donde se alcanza el mínimo local
`fval`	Valor de la función objetivo en x
`exitflag`	Estado de terminación del algoritmo
`output`	Información del proceso de optimización

Ejemplo

```
fun = @(x) x(1)^2 + 2*x(2)^2 + x(1)*x(2);
x0 = [1 1];

x = fminunc(fun,x0)
```

Este ejemplo minimiza una función cuadrática no lineal a partir de un punto inicial dado.

Función `fmincon` en MATLAB

La función `fmincon` resuelve problemas de **optimización no lineal con restricciones** de la forma:

$$\min\ f(x)$$

sujeto a:

$$Ax \leq b, \qquad A_{eq}x = b_{eq}, \qquad \ell \leq x \leq u,$$

$$c(x) \leq 0, \qquad c_{eq}(x) = 0$$

donde $f(x)$, $c(x)$ y $c_{eq}(x)$ pueden ser funciones no lineales.

Sintaxis básica

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
[x,fval,exitflag,output] = fmincon(...)
```

Parámetros

Parámetro	Descripción
`fun`	Función objetivo a minimizar (handle de función)
`x0`	Punto inicial
`A, b`	Restricciones lineales de desigualdad $Ax \leq b$
`Aeq, beq`	Restricciones lineales de igualdad $A_{eq}x = b_{eq}$
`lb, ub`	Cotas inferior y superior de las variables
`nonlcon`	Función con restricciones no lineales
`options`	Opciones del solver (creadas con `optimoptions`)

Resultados

Salida	Significado
`x`	Punto donde se alcanza el mínimo local factible
`fval`	Valor de la función objetivo en x
`exitflag`	Estado de terminación del algoritmo
`output`	Información del proceso de optimización

Ejemplo

```
fun = @(x) x(1)^2 + x(2)^2;
x0 = [1 1];

A = [1 1];
b = 1.5;

nonlcon = @(x) deal([], x(1)^2 + x(2) - 1);

x = fmincon(fun,x0,A,b,[],[],[0 0],[],nonlcon)
```

Este ejemplo minimiza una función no lineal sujeta a restricciones lineales y no lineales.

Función `fminsearch` en MATLAB

La función `fminsearch` resuelve problemas de **optimización no lineal sin restricciones** usando el método de **Nelder-Mead (simplex)**:

$$\min f(x)$$

donde $f(x)$ es una función escalar continua de una o varias variables.

Sintaxis básica

```
x = fminsearch(fun,x0)
x = fminsearch(fun,x0,options)
[x,fval,exitflag,output] = fminsearch(...)
```

Parámetros

Parámetro	Descripción
`fun`	Función objetivo a minimizar (handle de función)
`x0`	Punto inicial
`options`	Opciones del solver (creadas con `optimset`)

Resultados

Salida	Significado
`x`	Punto donde se alcanza el mínimo local
`fval`	Valor de la función objetivo en x
`exitflag`	Estado de terminación del algoritmo
`output`	Información del proceso de optimización

Ejemplo

```
fun = @(x) (x(1)-2)^2 + (x(2)+3)^2;
x0 = [0 0];

x = fminsearch(fun,x0)
```

Este ejemplo minimiza una función cuadrática de dos variables a partir del punto inicial dado, encontrando el mínimo local.

Referencias

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. `https://web.stanford.edu/~boyd/cvxbook/`.

[2] Frederick S. Hillier and Gerald J. Lieberman. *Introducción a la investigación de operaciones*. McGraw-Hill, 2015.

[3] Miguel Miranda. *Teoría de colas*. Primera Edición, Educa, 2005.

[4] Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, 2009.

[5] Hamdy A. Taha. *Investigación de operaciones*. Pearson Educación, México, 9 edition, 2012.

Index

colas
- canal único, 124
- canales en paralelo, 130
- disciplina, 119
- distribuciones, 120
- ejemplos, 115
- glosario de conceptos, 114
- impaciencia, 115
- llegadas, 114
- M/M/1, 124
- M/M/1/-/N, 139
- M/M/1/K, 127
- M/M/m, 131
- M/M/m/-/N, 142
- M/M/m/K, 134
- modelo genérico, 114
- notación de Kendall, 119
- notación y nomenclatura, 121
- población, 114
- población finita, 138
- realimentación, 149
- redes, 146
- redes en serie y en paralelo, 146
- régimen permanente, 120

investigación operativa
- formulación matemática estándar, 19

optimización
- dirección, 19
- función objetivo, 19
- región factible, 19
- región no acotada, 19
- restricción activa, 19
- solución óptima, 19
- taxonomía, 22
- variables de decisión, 19

planificación de proyectos
- AON, 76
- CPM, 76
- crashing, 83
- PERT, 76

precios sombra, 36
Programación Lineal, 27
programación Lineal
- algoritmo símplex, 32

programación lineal
- método analítico, 31
- método gráfico, 33

programación lineal entera
- enteros no negativos $\mathbb{Z}_+$, 39
- problema, 39
- problema relajado, 41
- solvers, 41
- variable binaria, 42

programación no lineal
- función convexa, 91
- función de Lagrange, 104
- KKT, 103

matlab solvers, 107
problema, 104
pyomo solver, 110

redes
Algoritmo de Prim, 65
camino, 60
camino más corto, 61
ciclo, 60
flujo de coste mínimo, 69
flujo máximo, 65
grafo conexo, 60
grafos, 59
árbol, 61
árbol soporte, 61

Investigación Operativa

Grado En Ingeniería en Sistemas de Telecomunicación
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Telemática

Javier Quintero & Almudena Buelta

Diapositivas – Material docente en abierto de la Universidad Rey Juan Carlos
Repositorio Institucional de la Universidad Rey Juan Carlos (BURJC Digital)
`https://burjcdigital.urjc.es/home`

2025 – 2026 v1.02.00

Copyright © 2026. Algunos derechos reservados. Este trabajo se distribuye bajo la licencia "Creative Commons **Attribution – Share alike**"

Temario

0 Introducción a IO

1 Programación lineal con variable continua

2 Programación Entera

3 Problemas de transporte y asignación

4 Modelos de optimización de redes

5 Administración de proyectos con PERT/CPM

6 Programación no lineal

7 Teoría de colas

Para esta parte se emplean como referencia principal los textos [Hillier and Lieberman, 2015], [Taha, 2012] y [Boyd and Vandenberghe, 2004].

La **Investigación Operativa**, conocida también como *Operational Research* o *Management Science*, es una disciplina científica orientada al apoyo de la toma de decisiones mediante la aplicación de métodos analíticos avanzados. Su objetivo fundamental es proporcionar soluciones racionales y óptimas a problemas complejos relacionados con la asignación eficiente de recursos escasos en sistemas organizativos.

Origen

El origen histórico de la Investigación Operativa se sitúa en el contexto de la Segunda Guerra Mundial, cuando las administraciones militares británica y estadounidense recurrieron a grupos interdisciplinarios de científicos –matemáticos, físicos, ingenieros y estadísticos– con el fin de mejorar la planificación y ejecución de operaciones militares. Estos equipos fueron encargados de analizar problemas estratégicos y tácticos tales como la asignación de radares, la optimización de convoyes, la defensa aérea o la gestión de inventarios de armamento, empleando por primera vez modelos matemáticos y métodos cuantitativos de manera sistemática.

Evolución

- Este surgimiento no fue un hecho aislado, sino que estuvo estrechamente ligado al aumento progresivo de la complejidad organizativa derivada de la Revolución Industrial. El crecimiento sostenido de empresas, instituciones y redes de transporte incrementó de forma notable la dificultad asociada a la gestión de recursos humanos, materiales y financieros, poniendo de manifiesto la necesidad de enfoques científicos para la toma de decisiones.
- El notable éxito alcanzado por los equipos de Investigación Operativa durante la guerra impulsó su rápida expansión en el ámbito civil tras el conflicto. Durante la década de 1950 se produjo la consolidación formal de la disciplina con la creación de sociedades científicas especializadas, entre las que destaca la *Operational Research Society*, fundada en el Reino Unido y considerada la sociedad científica más antigua dedicada específicamente a este campo.
- El desarrollo y la difusión de la Investigación Operativa se vieron además favorecidos por varios factores tecnológicos y metodológicos clave. En primer lugar, la disponibilidad de técnicas matemáticas adecuadas al nivel de desarrollo científico de la época permitió formular modelos cada vez más realistas. En segundo lugar, la revolución informática facilitó la realización de cálculos complejos que, hasta entonces, resultaban inabordables. Finalmente, la aparición de los ordenadores personales contribuyó decisivamente a la generalización de estas técnicas, haciéndolas accesibles a empresas y organizaciones de todos los tamaños.

Evolución

- En este contexto resulta especialmente relevante la figura de George B. Dantzig (1913–2005), creador del método *Simplex*, uno de los algoritmos más influyentes para la resolución de problemas de programación lineal. Este método constituyó un hito fundamental en la consolidación de la Investigación Operativa como disciplina aplicada, al proporcionar una herramienta eficiente para resolver problemas reales de gran escala.
- La estrecha relación entre la evolución de la Investigación Operativa y el desarrollo de la computación queda ilustrada por ejemplos históricos concretos, como la llegada en 1959 del IBM 560 Magnetic Drum Calculator a España, adquirido por RENFE para la gestión de líneas ferroviarias. Este hecho pone de manifiesto cómo los primeros sistemas informáticos fueron empleados de forma temprana para resolver problemas logísticos y de planificación, ámbitos naturales de aplicación de la Investigación Operativa.
- En síntesis, la Investigación Operativa surge de la confluencia entre necesidades prácticas de decisión, avances científicos y progreso tecnológico. Desde sus orígenes militares hasta su consolidación como herramienta esencial en sectores como el transporte, la logística, la construcción o las telecomunicaciones, la disciplina ha mantenido como rasgo distintivo una visión global y organizativa, orientada a optimizar el funcionamiento del sistema en su conjunto y no únicamente de sus componentes individuales.

Objetivo de la IO

- La Investigación Operativa es una disciplina orientada al análisis, gestión y coordinación de actividades dentro de una organización, con el objetivo de mejorar su funcionamiento global. Sus ámbitos de aplicación son muy amplios e incluyen, entre otros, el transporte y la logística, la construcción, las telecomunicaciones, la producción industrial, la energía y los servicios. En todos estos contextos, la Investigación Operativa proporciona un marco metodológico para abordar problemas complejos de decisión en los que intervienen múltiples variables, restricciones y objetivos potencialmente contrapuestos.
- Desde un punto de vista metodológico, la Investigación Operativa se caracteriza por un enfoque sistemático y estructurado, que permite transformar situaciones reales, a menudo poco definidas y de gran complejidad, en problemas susceptibles de ser analizados mediante modelos matemáticos. Este enfoque no persigue únicamente la obtención de soluciones factibles, sino la identificación de soluciones óptimas o cuasi óptimas que mejoren de forma cuantificable el desempeño del sistema considerado.

El desarrollo de un estudio de Investigación Operativa suele organizarse en una serie de fases bien definidas, que, aunque se presentan de forma secuencial, deben entenderse como parte de un proceso iterativo:

1. **Observación y formulación del problema.** Esta fase inicial consiste en el análisis detallado de la situación real, la identificación de los objetivos del sistema y la recopilación de los datos relevantes. Una correcta formulación del problema es esencial, ya que determina en gran medida la utilidad y validez de los resultados posteriores.
2. **Construcción del modelo matemático.** A partir de la formulación del problema, se desarrolla un modelo matemático que represente de forma simplificada pero suficientemente realista el sistema bajo estudio. Este modelo debe capturar los elementos esenciales del problema, como las variables de decisión, las restricciones y la función objetivo, de modo que las conclusiones obtenidas sean aplicables a la situación real.
3. **Fase experimental y validación.** El modelo propuesto se somete a pruebas mediante datos reales o simulados con el fin de verificar su coherencia y su capacidad para reproducir el comportamiento del sistema. En esta etapa pueden detectarse deficiencias que requieran modificar las hipótesis iniciales o ajustar el modelo, lo que pone de manifiesto el carácter iterativo del proceso.
4. **Obtención de la solución óptima.** Una vez validado el modelo, se aplican las técnicas analíticas o computacionales adecuadas para identificar la mejor solución entre las múltiples alternativas posibles. Esta solución debe interpretarse en el contexto del problema original y servir de apoyo a la toma de decisiones.

- Un aspecto fundamental de la Investigación Operativa es la adopción de una *perspectiva organizativa*. A diferencia de enfoques parciales que optimizan únicamente componentes individuales de un sistema, la Investigación Operativa busca resolver los conflictos de interés entre las distintas partes de una organización, de modo que la decisión final sea la más conveniente para el sistema en su conjunto.
- Esta visión global es especialmente relevante en entornos complejos, donde las decisiones locales pueden tener efectos adversos a nivel global.
- En consecuencia, la Investigación Operativa no debe entenderse únicamente como un conjunto de técnicas matemáticas, sino como una forma de abordar los problemas de decisión desde una perspectiva integradora, rigurosa y orientada a la mejora continua del desempeño organizativo.

Panorama del enfoque de modelado en investigación de operaciones

El enfoque de modelado constituye el núcleo metodológico de la Investigación Operativa y define su forma característica de abordar problemas complejos de toma de decisiones. Este enfoque se basa en la construcción y el análisis de modelos que representan, de manera simplificada pero rigurosa, sistemas reales en los que intervienen múltiples variables, restricciones y objetivos. El propósito fundamental del modelado es proporcionar una base racional que permita evaluar alternativas y apoyar decisiones de forma objetiva y coherente.

Un estudio de Investigación Operativa no debe entenderse como un ejercicio puramente matemático, sino como un proceso integral que combina análisis cuantitativo, conocimiento del sistema real y juicio profesional. En la práctica, el desarrollo de modelos matemáticos es solo una parte del trabajo total, y su utilidad depende en gran medida de la correcta formulación del problema, de la validez de los supuestos adoptados y de la adecuada interpretación de los resultados.

El proceso de modelado en Investigación Operativa suele estructurarse en una serie de etapas interrelacionadas, que no deben considerarse estrictamente secuenciales, sino como fases de un proceso iterativo y dinámico:

Enfoque de un problema de IO

1. **Definición del problema:** El punto de partida es la identificación clara del problema que se desea resolver, así como de los objetivos que se persiguen. Esta fase requiere comprender el contexto organizativo en el que se toma la decisión, delimitar el alcance del estudio y determinar los criterios de desempeño relevantes.
2. **Recolección y análisis de datos:** Una vez definido el problema, se recopilan los datos necesarios para describir el sistema. Estos datos pueden proceder de registros históricos, mediciones directas, estimaciones o hipótesis razonables. Su calidad y fiabilidad son determinantes para la validez del modelo.
3. **Formulación del modelo:** En esta etapa se construye un modelo matemático que represente los aspectos esenciales del problema real. El modelo incluye variables de decisión, parámetros, restricciones y una función objetivo que refleja los objetivos del sistema. La formulación implica un compromiso entre realismo y simplicidad, ya que un modelo excesivamente complejo puede resultar intratable, mientras que uno demasiado simple puede carecer de utilidad práctica.
4. **Obtención de soluciones:** Una vez formulado el modelo, se emplean métodos analíticos o algoritmos computacionales para obtener soluciones. En muchos casos existen múltiples soluciones factibles, por lo que el análisis se orienta a identificar aquellas que optimizan el criterio establecido en la función objetivo.
5. **Validación y análisis del modelo:** Las soluciones obtenidas deben evaluarse para comprobar que el modelo representa adecuadamente el sistema real. Esto puede implicar comparar los resultados con datos observados, realizar análisis de sensibilidad o modificar los supuestos iniciales. Si el modelo no resulta satisfactorio, es necesario reformularlo y repetir el proceso.

Enfoque de un problema de IO

6. **Implementación y seguimiento:** La etapa final consiste en aplicar los resultados del estudio al problema real. La implementación puede requerir ajustes organizativos, tecnológicos o operativos, así como un seguimiento posterior para verificar que los beneficios esperados se materializan en la práctica.

Un rasgo esencial del enfoque de modelado en Investigación Operativa es su orientación global. Las decisiones se analizan desde la perspectiva del sistema completo, y no únicamente desde el punto de vista de sus componentes individuales. Este enfoque permite identificar y resolver conflictos entre objetivos parciales, garantizando que la solución adoptada sea la más adecuada para el conjunto de la organización.

Conclusión

En conclusión, el enfoque de modelado proporciona un marco sistemático para analizar problemas complejos de decisión, integrando modelos matemáticos, datos y criterios organizativos. Su correcta aplicación permite transformar situaciones reales difíciles de gestionar en problemas estructurados, cuyas soluciones ofrecen un apoyo sólido y cuantificable a la toma de decisiones en ingeniería y gestión.

Observación

Para una discusión más ampliada del tema de esta sección remitimos al lector al Capítulo 2 de [Hillier and Lieberman, 2015].

Algoritmos y paquetes de Investigación Operativa

La aplicación práctica de la Investigación Operativa se apoya de manera fundamental en el desarrollo de algoritmos eficientes y en el uso de paquetes de software especializados. Los algoritmos constituyen procedimientos sistemáticos diseñados para resolver modelos matemáticos asociados a problemas de decisión, garantizando, cuando es posible, la obtención de soluciones óptimas en un tiempo computacional razonable.

A lo largo de la evolución de la disciplina se han desarrollado numerosos algoritmos específicos, como métodos para programación lineal, entera y no lineal, algoritmos para problemas de redes, modelos de colas y métodos de simulación. La elección del algoritmo adecuado depende tanto de la naturaleza del problema como de su tamaño y complejidad, siendo habitual recurrir a aproximaciones heurísticas o metaheurísticas cuando los métodos exactos resultan computacionalmente inviables.

El uso de paquetes de Investigación Operativa permite implementar estos algoritmos sin necesidad de desarrollarlos desde cero, facilitando su aplicación en entornos reales. Estos paquetes integran rutinas de optimización, herramientas de análisis y capacidades de gestión de datos, lo que los convierte en instrumentos esenciales tanto en el ámbito académico como en el profesional. No obstante, el uso efectivo de estas herramientas requiere una comprensión sólida de los modelos y algoritmos subyacentes, ya que el software debe entenderse como un apoyo al análisis y no como un sustituto del razonamiento científico propio de la disciplina.

Observación

Los programas que permiten el acceso e implementación de diferentes algoritmos para la resolución de modelos de optimización se conocen con **solvers**.

Paquetes de software **no comerciales** y de **acceso libre**

- **OR-Tools** – Conjunto de herramientas de optimización de código abierto desarrollado por Google, que incluye solvers para programación lineal y entera, además de herramientas para problemas de rutas y redes.
- **GLPK (GNU Linear Programming Kit)** – Biblioteca libre para programación lineal y programación entera mixta, ampliamente utilizada en entornos académicos y compatible con muchos lenguajes y entornos de modelado.
- **CBC (COIN-OR Branch and Cut)** – Solver de programación lineal entera mixta parte del proyecto COIN-OR, adecuado para problemas de tamaño pequeño y mediano.
- **SCIP** – Sistema de optimización de código abierto capaz de resolver problemas combinatorios y de programación entera, con interfaces para múltiples lenguajes.
- **Pyomo** – Lenguaje de modelado basado en Python para definir y resolver aplicaciones de optimización estructurada, que soporta una amplia variedad de tipos de problemas, desde lineales hasta no lineales y estocásticos.
- **JuMP** – Lenguaje de modelado algebraico embebido en Julia que facilita la declaración de variables, restricciones y funciones objetivo para múltiples clases de problemas de optimización.
- **SimPy** – Marco de simulación de eventos discretos basado en Python que permite modelar sistemas dinámicos, colas y procesos estocásticos en tiempo discreto.
- **OpenModelica** – Entorno libre para modelado, simulación y análisis de sistemas dinámicos complejos, útil para problemas de simulación y control en Ingeniería.
- **PyORlib** – Biblioteca Python de acceso abierto diseñada para explorar conceptos de investigación operativa y probar distintos métodos de optimización de forma sencilla.
- **COIN-OR Optimization Suite** – Conjunto de herramientas interoperables que incluye múltiples solvers y utilidades para programación lineal, entera, no lineal y otros métodos de optimización.

Paquetes libres y de acceso abierto

Para este curso se emplearán de manera específica dos solvers:

- Optimization Toolbox: Módulo de optimización de Matlab, licenciado en los ordenadores de campus.
- Pyomo solver for Windows: Herramienta de código abierto desarrollada para Windows, disponible en Aula Virtual, bajo la pestaña de Recursos. Este solver está basado en Pyomo y fue desarrollado específicamente para los fines de este curso. Más información y ayuda en Apéndice B.

Además en los Apéndices F y E se ofrecen instrucciones para la implementación de modelos de optimización en Python y Excel, como alternativas a las herramientas oficiales del curso.

Introducción

La Investigación Operativa desempeña un papel esencial en la Ingeniería de Telecomunicaciones, donde el diseño de redes eficientes, la gestión de recursos y la planificación de servicios dependen, en gran medida, de la resolución de problemas matemáticos de decisión. En términos generales, la Investigación Operativa se define como el proceso de determinar la mejor solución dentro de un conjunto de alternativas factibles, de acuerdo con un objetivo específico, como minimizar la latencia en la transmisión de datos, maximizar el ancho de banda disponible o reducir los costes de despliegue de infraestructura.

La Investigación Operativa se enmarca dentro de la disciplina matemática de Optimización. Los problemas abordados, desde un punto de vista puramente matemático, pueden clasificarse en diversas categorías:

- **Programación Lineal:** Problemas en los que tanto la función objetivo como las restricciones son lineales. Ejemplos incluyen la asignación de recursos en redes de telecomunicaciones o la planificación de capacidad en sistemas de transmisión.
- **Programación No Lineal:** Problemas que involucran funciones objetivo o restricciones no lineales, frecuentes en el diseño de antenas, optimización de potencia en sistemas inalámbricos o control de interferencias en redes.
- **Programación Entera y Mixta:** Problemas en los que algunas o todas las variables deben tomar valores enteros. Son comunes en la planificación de rutas en redes IP, asignación de canales en sistemas móviles o ubicación óptima de estaciones base.
- **Optimización con Restricciones:** Problemas que incluyen limitaciones físicas, regulatorias o de calidad de servicio, por ejemplo, garantizar que la latencia no supere ciertos umbrales en redes críticas.
- **Optimización sin Restricciones:** Problemas sin restricciones explícitas, que suelen aparecer en estimación de parámetros o ajuste de modelos en algoritmos de aprendizaje aplicados a telecomunicaciones.
- **Optimización Dinámica:** Problemas donde las decisiones evolucionan en el tiempo, relevantes para la gestión de tráfico en redes, control de congestión o planificación de despliegue de infraestructura.
- **Optimización Estocástica:** Problemas que incorporan incertidumbre, como variabilidad en la demanda de tráfico, fluctuaciones en la calidad del canal o comportamiento impredecible de usuarios.

Protocolo de resolución de problemas A lo largo de este curso estudiaremos los diferentes tipos de problemas abordados por la Investigación Operativa, aprenderemos las herramientas matemáticas necesarias para su formulación y resolución, y exploraremos aplicaciones que impactan directamente en la práctica profesional de la Ingeniería de Telecomunicaciones.
Cuando existe una decisión que requiere obtener el mejor resultado posible, decimos que estamos frente a un problema de Investigación Operativa. Surgen entonces varias preguntas: ¿Cómo definir qué significa "el mejor resultado"? ¿Dónde reside el poder de decisión? ¿Son válidas todas las soluciones en el contexto dado? Resolver un problema de Investigación Operativa implica seguir cuatro pasos fundamentales, que denominaremos a partir de ahora el protocolo:

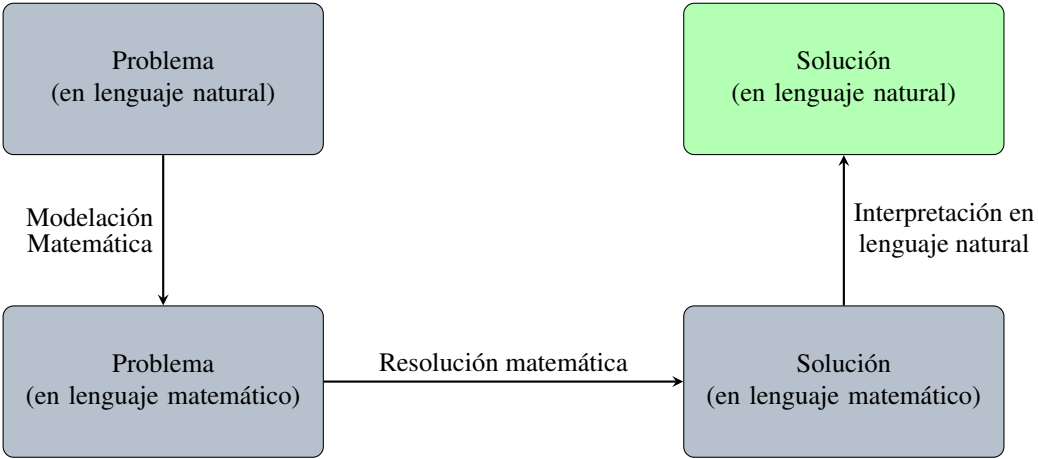


Figura 1: Protocolo de solución de problemas de Investigación Operativa

Como ingenieros, podemos encontrarnos con una pregunta (un problema expresado en lenguaje natural) y buscamos una respuesta (una solución igualmente en lenguaje natural). En este curso estudiaremos el proceso que permite obtener dicha respuesta mediante técnicas matemáticas avanzadas, concluyendo con una solución clara y directa al problema original. El objetivo principal de esta asignatura es proporcionar al estudiante las herramientas necesarias para afrontar este proceso en un entorno profesional, sirviendo como apoyo al responsable de la toma de decisiones para que pueda partir de un problema y llegar a una respuesta sin necesidad de involucrarse en los detalles matemáticos internos.

Por simplicidad, en este curso nos centraremos en problemas deterministas, es decir, aquellos en los que todos los parámetros del modelo son conocidos y no dependen de factores incontrolables. La formulación matemática de un problema determinista en Investigación Operativa se expresa de la siguiente manera:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m, \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p, \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{1}$$

Al conjunto de ecuaciones 1 le llamaremos **modelos de optimización**. Sus elementos son

Elementos del modelo de optimización

- **Dirección:** `min` para problemas de minimización y `max` para problemas de maximización. Por defecto, asumimos un problema de minimización en la forma estándar.
- **Variables de decisión** $\mathbf{x} \in \mathbb{R}^n$: las cantidades desconocidas que deben determinarse. Cada componente x_k representa una variable controlable, como la asignación de ancho de banda, la potencia de transmisión o la inclusión de nodos en una red. La dimensión n corresponde al número de variables de decisión.

Elementos del modelo de optimización

- **Función objetivo** $f(\mathbf{x})$: la función que se desea minimizar (o maximizar). Cuantifica el criterio de rendimiento, por ejemplo, minimizar la latencia, minimizar el consumo energético o maximizar la capacidad de la red.
- **Región factible** $\mathscr{F}$: el conjunto de todos los $\mathbf{x}$ que satisfacen las restricciones:

$$\mathscr{F} = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \leq 0,\ h_j(\mathbf{x}) = 0\}.$$

  Solo las soluciones dentro de $\mathscr{F}$ son válidas; a estas soluciones válidas las denominamos **soluciones factibles**, y el conjunto de todas ellas se denomina **región factible**.
- **Solución factible:** Cualquier punto $\mathbf{x} \in \mathscr{F}$.
- **Solución no factible:** Cualquier punto $\mathbf{x} \notin \mathscr{F}$.
- **Solución óptima:** Un punto $\mathbf{x}^* \in \mathscr{F}$ que satisface $f(\mathbf{x}^*) \leq f(\mathbf{x})$ para todo $x \in \mathbb{X}$. En caso de un problema de maximización, esta caracterización cambia a $f(\mathbf{x}^*) \geq f(\mathbf{x})$.
- **Restricción activa:** Una restricción activa es aquella que se cumple en igualdad. Por ejemplo, considere la desigualdad $f_1 : 2x_1 + 3x_2 \leq 5$; la solución $x^* = (1,1)$ satisface la desigualdad en igualdad, porque $2(1) + 3(1) = 5$. Decimos que la restricción f_1 está activa en x^*.

Observación

- Por simplicidad, la forma estándar de un problema de Investigación Operativa se escribe como un problema de minimización. Un problema de maximización puede transformarse en uno de minimización considerando que

$$\max_{\mathbf{x}} f(x) = -\min_{\mathbf{x}} -f(x).$$

- Para las restricciones de desigualdad asumimos como forma estándar $g_i(x) \leq 0$, dado que la condición $g(x) \geq 0$ es equivalente a $-g(x) \leq 0$.
- La región factible debe ser cerrada, es decir, las restricciones deben incluir las fronteras. Si una frontera de la región no está incluida, el problema puede no tener solución.
- Si la región factible es no acotada, es decir, no puede contenerse en ningún círculo de radio finito, entonces decimos que la región factible es **no acotada**. Véase el Ejemplo 1.

Ejemplo (Región No Acotada)

Considere la siguiente región factible determinada por las desigualdades,

$$0 \leq x_1 \leq 4,$$
$$0 \leq x_2$$

que corresponde a la Figura 2. Esta región factible es no acotada.

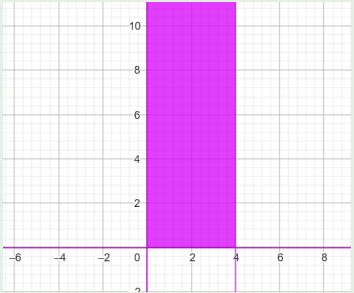


Figura 2: Región factible no acotada de un problema de optimización bidimensional

Caracterización de la Solución Óptima de un Problema

La "solución óptima" de un problema de optimización debe almacenar la siguiente información:

- Valor Óptimo $z^* = f(\mathbf{x}^*)$, el valor mínimo de la función objetivo, que se alcanza en el punto de la
- Solución Óptima $\mathbf{x}^* \in \mathscr{F}$, el cual es el vector que hace que el valor de la función objetivo sea mínimo y, al mismo tiempo, pertenece a la región factible.

Ejemplo

Considere el siguiente **problema de optimización**

$$\begin{array}{lll} \max & 2x_1 + x_2 & \\ \text{s.t.} & x_1 + x_2 & \leq 2 \\ & x_1, x_2 \geq 0 & \end{array}$$

Este es un problema de programación lineal porque tanto la función objetivo como todas las restricciones son funciones lineales. Los elementos son:

- **Variables de decisión**: $x_1, x_2 \in \mathbb{R}^n$
- **Función Objetivo**: $z = f(x_1, x_2) = 2x_1 + x_2$

Ejemplo

- **Restricciones de Desigualdad**: $x_1 + x_2 \leq 2$
- **Restricciones de dominio o de no-negatividad**: $x_1, x_2 \geq 0$
- **Región Factible**: La combinación en $\mathbb{R}^2$ de las restricciones anteriores:

$$\mathscr{F} = \{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 \leq 2, x_1 \geq 0, x_2 \geq 0\}$$

que corresponde con la Figura 3
Observa que esta región factible está acotada. En casos con más de dos variables, una representación gráfica no siempre es útil ni posible. La **solución** de este problema es el par $\mathbf{x}^* = (x_1^*, x_2^*) = (2, 1)$ con valor óptimo $z^* = f(x_1^*, x_2^*) = 5$. Más detalles sobre cómo resolver este tipo de problemas se presentan en el Capítulo 2.

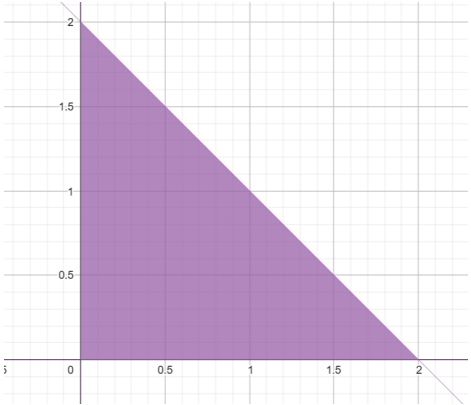


Figura 3: Región factible de un problema de optimización en dos dimensiones

Considera el siguiente problema de optimización:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1,\ldots,m, \\ & h_j(\mathbf{x}) = 0, \quad j = 1,\ldots,p. \end{aligned}$$

Proposición

La solución óptima $\mathbf{x}^*$ se caracteriza por las siguientes propiedades:

1. **Factibilidad:** La solución debe pertenecer al conjunto factible

$$\mathscr{F} = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \leq 0,\ h_j(\mathbf{x}) = 0\}.$$

   Si no existe tal punto, se dice que el problema es *infactible*.

2. **Optimalidad:** Una solución óptima $\mathbf{x}^*$ satisface

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathscr{F}.$$

   Si la desigualdad es estricta, la solución es *única*.

Proposición

3. **Tipos de optimalidad:**
   - *Óptimo global:* la mejor solución en todo el conjunto factible.
   - *Óptimo local:* la mejor solución solo en una vecindad de $\mathbf{x}^*$.

   En problemas convexos, todo óptimo local también es global; en problemas no convexos pueden existir múltiples óptimos locales.

1. **Condiciones de optimalidad:** Para problemas suaves, las soluciones óptimas se caracterizan mediante las *condiciones de Karush–Kuhn–Tucker (KKT)*. Más detalles en la Subsección 6.
2. **Calidad de la solución:** En la práctica, los algoritmos numéricos pueden devolver soluciones aproximadas.
   - ε-optimalidad: la solución está dentro de una tolerancia respecto al óptimo verdadero.
   - Tolerancia en restricciones: la factibilidad se satisface dentro de un pequeño error numérico.

En este curso vamos a estudiar problemas dentro de la taxonomía mostrada en la Figura 4. Los problemas se clasifican según la naturaleza matemática de los elementos del modelo involucrados.

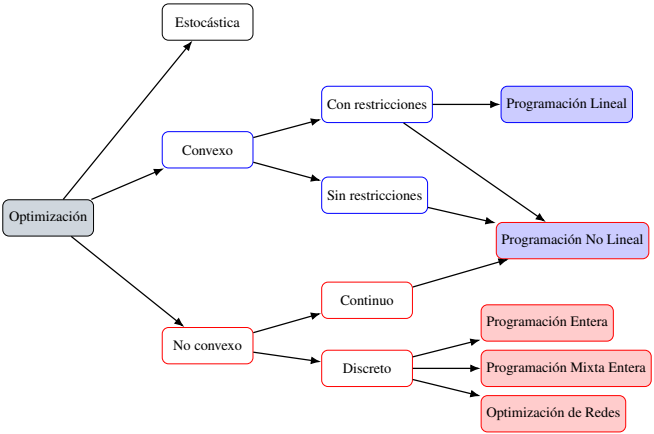


Figura 4: Taxonomía simplificada de los problemas de investigación operativa; para una taxonomía más detallada, consulte este sitio.

Los problemas de investigación operativa pueden clasificarse según las propiedades matemáticas de la función objetivo $f(\mathbf{x})$ y la región factible $\mathscr{F}$. Estas características determinan tanto la dificultad del problema como los tipos de algoritmos que pueden aplicarse para resolverlo.

1. Problemas lineales vs. no lineales

- **Programación Lineal (LP):** tanto la función objetivo como las restricciones son lineales. Estos problemas están bien estudiados y pueden resolverse de manera eficiente mediante métodos como el algoritmo Simplex o los métodos de Punto Interior. Véase el Capítulo 2.
- **Programación No Lineal (NLP):** al menos una de las funciones, ya sea la función objetivo o las restricciones, es no lineal. Estos problemas pueden ser significativamente más complejos, especialmente si son no convexos. Véase el Capítulo 7.

2. Problemas convexos vs. no convexos

- **Optimización convexa:** la función objetivo es convexa y el conjunto factible es un conjunto convexo. Cualquier óptimo local es también óptimo global, lo que hace que estos problemas sean tratables con sólidas garantías teóricas. De hecho, la solución de un problema convexo está **garantizada** de existir y ser única.
- **Optimización no convexa:** la función objetivo o la región factible es no convexa. Estos problemas pueden tener múltiples mínimos locales, y encontrar el óptimo global puede ser computacionalmente difícil o incluso intratable.

Ejemplo

La función objetivo en el siguiente problema es no convexa, véase la Figura 5

$$\begin{aligned} \min & f(x) = x^4 - 3x^2 + 2 \\ & x \in \mathbb{R} \end{aligned}$$

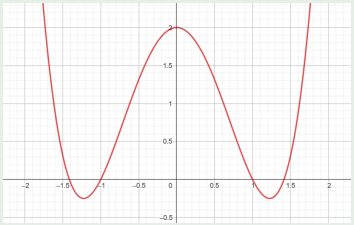


Figura 5: Función objetivo no convexa en dos dimensiones.

- La función objetivo $f(x)$ es un polinomio cuártico con múltiples mínimos locales y un mínimo global.
- Esta no convexidad implica que las técnicas estándar de optimización convexa no pueden garantizar encontrar el óptimo global.

Observación

Una función no convexa es aquella cuya segunda derivada no mantiene signo constante. Esto conduce a la existencia de puntos de inflexión y, por tanto, a la presencia de múltiples extremos locales. En la literatura, esto también se conoce como funciones multimodales y representa un problema complejo para abordar con técnicas clásicas de optimización.

3. Problemas discretos vs. continuos

- **Optimización continua:** las variables de decisión $\mathbf{x}$ pueden tomar cualquier valor real dentro del conjunto factible.
- **Optimización discreta:** las variables de decisión están restringidas a valores discretos (por ejemplo, enteros o binarios). Estos problemas surgen con frecuencia en aplicaciones de planificación, asignación y diseño.
- **Optimización mixta-entera (MIP/MINLP):** algunas variables son continuas mientras que otras están restringidas a valores discretos, combinando la complejidad de ambos mundos. Estos son los problemas más comunes en aplicaciones reales.

4. Problemas con restricciones vs. sin restricciones

- **Optimización sin restricciones:** no existen restricciones explícitas; la región factible es todo el espacio $\mathbb{R}^n$. Estos problemas suelen ser más simples de analizar y resolver. Véase el Ejemplo 4.
- **Optimización con restricciones:** la región factible está limitada por igualdades y/o desigualdades. Estas restricciones representan limitaciones físicas, operativas o tecnológicas en aplicaciones reales. Véase la Figura 3.

De los cursos de cálculo multivariable tenemos las herramientas para tratar problemas convexos y continuos. Aplicando simplemente las pruebas de primera y segunda derivada, podemos encontrar fácilmente los valores óptimos de cualquier problema que sea dos veces derivable. Por lo tanto, vamos a comenzar con el caso más sencillo de optimización convexa, según la naturaleza de los elementos del modelo: el Problema de Programación Lineal (LP).

§I. Programación lineal con variable continua

- Modelos de programación lineal. Elementos.
- Ejemplo prototipo.
- Solución de problemas LP.
  - Método analítico. Algoritmo Símplex
  - Método gráfico.
  - Solvers de Matlab.
- Análisis post-óptimo. Precios sombra.

Introducción

La programación lineal (LP) constituye una de las clases más fundamentales y ampliamente estudiadas dentro de los problemas de optimización. Su relevancia radica tanto en su aplicabilidad práctica como en la sólida teoría matemática que sirve de base para métodos de optimización más generales. Los modelos de programación lineal se emplean de manera extensa en diversas áreas de la ingeniería, incluyendo las telecomunicaciones, así como en economía y ciencias de la gestión. Entre las aplicaciones típicas se incluyen:

- **Asignación de Recursos:** determinar la distribución óptima de recursos limitados entre actividades que compiten entre sí, como ancho de banda, potencia de transmisión o canales de comunicación.
- **Planificación de Redes:** optimizar costes, capacidad y rendimiento en el diseño y operación de redes de telecomunicaciones, sujetos a restricciones de tráfico y calidad de servicio.
- **Transporte y Logística de Datos:** minimizar costes de enrutamiento y transmisión cumpliendo requisitos de demanda y disponibilidad de enlaces.
- **Gestión de Infraestructura:** optimizar la asignación de equipos, frecuencias y recursos en sistemas de comunicación para garantizar eficiencia y fiabilidad.

Objetivo

En este capítulo se estudia el problema de LP, los elementos que conforman un modelo lineal, los métodos de solución, las interpretaciones geométricas y las particularidades que hacen que este modelo sea tan relevante en el ámbito de la optimización aplicada a las telecomunicaciones.

Definición (Modelo de Programación Lineal)

Un **problema de Programación Lineal (LP)** es un problema de optimización en el que tanto la función objetivo como las restricciones son lineales en las variables de decisión. Formalmente, la forma estándar (matricial o canónica) de un problema de programación lineal se expresa como:

$$\begin{array}{lll} \min\limits_{\mathbf{x}} z = & c_1x_1 + c_2x_2 + \cdots + c_nx_n & \\ \text{s.a.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 & \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 & \\ & \vdots & \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m & \\ & x_1, x_2, \ldots, x_n \geq 0 & \end{array} \quad \Leftrightarrow \quad \boxed{\begin{array}{ll} \min\limits_{\mathbf{x}} z = & \mathbf{c}^t\mathbf{x} \\ \text{s.a.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array}} \tag{2}$$

donde:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$ es el vector de variables de decisión.
- $\mathbf{c} = (c_1, c_2, \ldots, c_n)^t \in \mathbb{R}^n$ es el vector de coeficientes de la función objetivo (vector de costes).
- $(a_{ij}) = A \in \mathbb{R}^{m \times n}$ es la matriz de coeficientes de las m restricciones:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \cdots & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

- $\mathbf{b} = (b_1, b_2, \ldots, b_m)^t \in \mathbb{R}^m$ es el vector de términos independientes en las restricciones (vector de recursos).
- El conjunto de restricciones $\mathbf{x} \geq 0 \Rightarrow x_1, x_2, \ldots, x_n \geq 0$ se conoce como restricciones de no negatividad.

Observación

- La función objetivo $\mathbf{c}^t\mathbf{x}$ es lineal en las variables de decisión.
- La región factible, definida por las desigualdades lineales y las restricciones de no negatividad, es un poliedro convexo.
- Cualquier solución óptima, si existe, se encuentra en un *vértice* de la región factible.

Ejemplo (Asignación de Producción)

Una empresa tecnológica ha decidido ampliar su catálogo de productos avanzados en el ámbito de las telecomunicaciones, incorporando dos nuevos dispositivos:

- **Producto 1**: un módulo de transmisión que requiere un bastidor metálico y ensamblaje final.
- **Producto 2**: un kit de diagnóstico de red que requiere una carcasa de polímero y ensamblaje final.

La empresa dispone de tres instalaciones especializadas organizadas de la siguiente manera:

- Instalación 1 fabrica **componentes y bastidores metálicos**,
- Instalación 2 produce **partes de polímero y compuestos**,
- Instalación 3 se encarga del **ensamblaje y pruebas finales de los dispositivos de telecomunicaciones**.

Ejemplo (Asignación de Producción)

Los requisitos de producción son los siguientes:

- El Producto 1 consume capacidad de las Instalaciones 1 y 3,
- El Producto 2 consume capacidad de las Instalaciones 2 y 3.

Cada producto se fabricará en lotes de 20 unidades. La tasa de producción será el número de lotes producidos por semana que satisfacen las restricciones. Los lotes pueden producirse de manera fraccionada.

Ejemplo (Asignación de Producción)

Para formular cuidadosamente el modelo matemático, se han estimado, Tabla 1, los siguientes datos:

- el número de horas de producción disponibles por semana en cada instalación para estos dos nuevos productos,
- el número de horas de producción requeridas por lote de cada producto en cada instalación,
- el beneficio por lote de cada producto.

Tabla 1: Datos de Ejemplo 6

Instalación	**Tiempo de producción por lote (horas)**		**Tiempo disponible por semana (horas)**
	Producto		
	1	**2**	
1	1	0	4
2	0	2	12
3	3	2	18
Beneficio por lote	$3,000	$5,000	

El problema consiste en determinar los niveles óptimos de producción semanal de los dos productos, es

Ejemplo (Asignación de Producción)

- Para resolver este problema, activamos el protocolo de la Figura 1. Tenemos un problema en lenguaje natural, por lo que el primer paso es extraer los datos para elaborar un modelo como (2).
- Primero, observamos que la decisión se centra en los niveles de producción, es decir, el número de dispositivos de cada tipo que la empresa fabrica. Por tanto, declaramos las siguientes variables x_i: número de lotes del producto i que la empresa produce cada semana, para $i = 1, 2$. El problema indica que pueden producirse lotes fraccionados, pero no pueden ser negativos, por lo tanto, $x_1, x_2 \geq 0$.

Además, el beneficio depende directamente del número de lotes producidos, es decir:

$$\text{beneficio} = 3000x_1 + 5000x_2.$$

Dado que el objetivo es maximizar el beneficio, la función objetivo se expresa como:

$$z = 3x_1 + 5x_2$$

en miles de dólares.

Ejemplo (Asignación de Producción)

Existen varias restricciones de producción asociadas a los recursos disponibles, en este caso el tiempo de producción en cada instalación. Por ejemplo, en la instalación 1 hay 4 horas disponibles por semana, y dado que cada lote del producto 1 consume una hora, la restricción correspondiente es $x_1 \leq 4$. Las otras instalaciones generan restricciones adicionales: $2x_2 \leq 12$ y $3x_1 + 2x_2 \leq 18$, respectivamente.
Una vez extraída toda la información, estamos listos para el segundo paso del protocolo 1. Escribimos esta información en lenguaje matemático utilizando la forma estándar proporcionada en (2):

$$
\begin{array}{lrrrrr}
\max\limits_{\mathbf{x}} & z & = 3x_1 & +5x_2 & & \\
\text{s.a.} & & x_1 & & \leq & 4 \\
& & & 2x_2 & \leq & 12 \\
& & 3x_1 & +2x_2 & \leq & 18 \\
& & & x_1, x_2 & \geq & 0
\end{array}
$$

Hipótesis de un problema LP

Existen varias suposiciones que hacen viable un modelo de programación lineal (LP):

1. **Proporcionalidad.** Esta suposición implica que:
   a. la contribución de cada actividad al valor de la función objetivo Z es proporcional al nivel de dicha actividad, y
   b. la contribución de cada actividad a cada restricción funcional es proporcional al nivel de dicha actividad.
   c. **Ejemplo:** El coste total de transmisión es proporcional al número de enlaces activados en una red. Si se duplican los enlaces, se duplica el coste.
2. **Aditividad.** Esta suposición implica que toda función en un modelo de programación lineal se expresa como la suma de las contribuciones individuales de las respectivas actividades. **Ejemplo:** El consumo total de ancho de banda en una red es la suma del ancho de banda utilizado por cada conexión individual.
3. **Divisibilidad.** Esta suposición implica que las variables de decisión en un modelo de programación lineal pueden tomar cualquier valor, incluyendo valores no enteros. **Ejemplo:**La asignación de potencia en un sistema de transmisión puede realizarse en valores fraccionarios, por ejemplo, asignar 2.5 W a un canal.
4. **Certeza.** Se asume que los valores asignados a cada parámetro en un modelo de programación lineal son conocidos y permanecen constantes. **Ejemplo:** Se asume que la demanda de tráfico y la capacidad de los enlaces son conocidas y constantes durante el periodo de planificación.

Observación

Estas suposiciones son fundamentales para la tratabilidad matemática de la programación lineal. Garantizan que las relaciones en el modelo permanezcan lineales, lo que permite aplicar métodos de solución eficientes como el algoritmo símplex. Sin embargo, en aplicaciones reales, una o más de estas suposiciones pueden no cumplirse estrictamente. Por ejemplo, las variables de decisión pueden necesitar tomar valores enteros, los parámetros pueden involucrar incertidumbre o las relaciones pueden ser no lineales. En tales casos, se requieren extensiones de la programación lineal, como la programación entera, la programación no lineal o la programación estocástica, para modelar el problema con mayor precisión.

Un problema lineal es **por definición** un problema convexo. Un problema convexo garantiza la existencia de una solución óptima única. En el caso de problemas lineales, el principio de optimalidad en los vértices nos permite encontrar dicha solución de manera directa.

Definición (Solución en Vértice)

Se denomina solución en vértice a una solución del problema (2) que se obtiene como la intersección de dos restricciones. Es decir, $\mathbf{x}^+$ es una **solución en vértice** si y solo si

$$\mathbf{x}^+ \in \{g_i\} \cap \{g_j\} \cap \mathscr{F}$$

donde g_i, g_j son dos restricciones del problema (1).

Teorema (Optimalidad en Vértices)

Un problema de programación lineal (LP) con variables continuas y región factible acotada satisface:

- *la solución óptima se encuentra en el vértice con mejor valor de la función objetivo, o*
- *si dos vértices tienen el mismo valor de la función objetivo, entonces el problema tiene infinitas soluciones situadas en el segmento que une ambos vértices.*

Ejemplo (Soluciones en Vértices)

Retomemos el Ejemplo 6. Utilizando Geogebra podemos graficar la región factible como en la Figura 6:

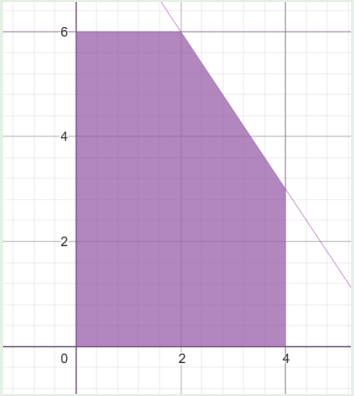


Figura 6: Región factible del Ejemplo 6

Ejemplo (Soluciones en Vértices)

Dado que la región es acotada, según el Teorema 12, la solución óptima se encuentra en los vértices de la región. Calculamos los vértices mediante la intersección de las restricciones y evaluamos cada punto en la función objetivo:

x	z
$(0,0)$	0
$(4,0)$	12
$(4,3)$	27
$\boxed{(2,6)}$	$\boxed{36}$
$(0,6)$	30

Ejemplo

Este problema es de maximización, por lo tanto, dado que $(2, 6)$ devuelve el valor máximo de la función objetivo, concluimos que esta es la solución óptima.
Finalmente, cerramos el último paso del protocolo 1, expresando la solución en los mismos términos del problema, es decir, traducimos la solución matemática a lenguaje natural:
"El nivel óptimo de producción semanal es fabricar 2 *lotes del producto* 1 *y* 6 *lotes del producto* 2*, para maximizar el beneficio a $36 mil dólares."*
Obsérvese que en esta respuesta se resume el objetivo principal del problema: nivel de producción semanal para beneficio máximo, utilizando las unidades correspondientes.

Este ejemplo ilustra cómo aplicar el método analítico a un problema bidimensional.

Algorithm 1 Método Analítico

1: Aplicar únicamente si la región factible es acotada.
2: Encontrar el conjunto V de vértices factibles mediante la intersección $\{g_i\} \cap \{g_j\}$ para todo $i = 1, \ldots, m+n$, donde g_i para $i = m+1, \ldots, n+m$ corresponde a las últimas n restricciones de no negatividad.
3: Para cada $v_i \in V$, calcular $f(v_i)$.
4: La solución óptima corresponde al menor (mayor) valor $z^* = f(v_i^*)$ en un problema de minimización (maximización).

Este principio está detrás del algoritmo Símplex. El **algoritmo Símplex** es uno de los métodos más famosos en optimización. Fue desarrollado en 1947 por **George Dantzig**, quien trabajaba en problemas de planificación para la Fuerza Aérea de los EE.UU. Desde entonces, se ha convertido en un pilar de la **investigación operativa** y se ha aplicado en innumerables campos, desde *ingeniería y economía* hasta *telecomunicaciones*.

La idea principal detrás del algoritmo es simple pero poderosa: en un problema de programación lineal, las soluciones factibles forman una figura geométrica llamada **politopo** (un polígono multidimensional). Dantzig demostró que la **solución óptima siempre se encuentra en uno de los vértices** de este politopo. En lugar de verificar todos los puntos posibles, el algoritmo Símplex se desplaza inteligentemente de vértice en vértice, mejorando la solución en cada paso, hasta alcanzar la mejor.

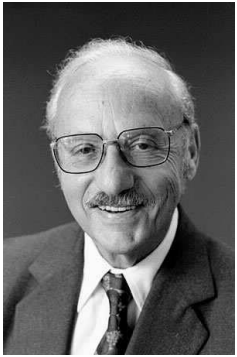


Figura 7: George B. Dantzig (1913–2005) Creador del método SÍMPLEX

Algunas características importantes del método Símplex

- Proporciona no solo la **solución óptima**, sino también información sobre la **sensibilidad** del modelo (cómo afectan los cambios en recursos o coeficientes al resultado).
- En la práctica, es extremadamente **eficiente**, aunque en teoría puede requerir tiempo exponencial en el peor caso.
- Sentó las bases de la optimización moderna e inspiró muchas extensiones, como el **Símplex Revisado** y el **Símplex Dual**.

Hoy en día, incluso con algoritmos más recientes (como los **Métodos de Punto Interior**), el método Símplex sigue siendo ampliamente utilizado debido a su **eficiencia práctica, robustez e interpretabilidad**.

Ejemplo (Soluciones infinitas)

Consideremos el siguiente problema de optimización:

$$\begin{array}{lll} \max & x_1 + x_2 & \\ \text{s.a.} & x_1 + x_2 & \leq 2 \\ & x_1, x_2 \geq 0 & \end{array}$$

La región factible tiene tres vértices:

x	z
$(0,0)$	0
$(2,0)$	2
$(0,2)$	2

Obsérvese cómo el valor de la función objetivo coincide en dos vértices. Esto significa, según el Teorema 12, que las soluciones óptimas se encuentran en el segmento que une ambos vértices, es decir, el segmento:

$$\mathbf{x}^*(\lambda) = \lambda(2,0) + (1-\lambda)(0,2)$$

para $\lambda \in [0,1]$, y $z^* = 2$. Dado que para cada valor de λ existe una solución diferente, concluimos que este problema tiene infinitas soluciones.

Método gráfico. Curvas de nivel

¿Qué significa geométricamente el principio de optimalidad (Teorema 12)? Añadamos una curva de nivel que represente la función objetivo a la región factible del Ejemplo 6. Para ello, utilizamos la calculadora gráfica Geogebra (ver más en el Apéndice C). Para esta sección, los gráficos pueden consultarse aquí.

Algorithm 2 Método Gráfico

1: Aplicar únicamente a problemas bidimensionales.
2: Graficar la región factible mediante la intersección de todas las restricciones.
3: **if** la región factible no está vacía **then**
4: Crear una curva de nivel correspondiente a la función objetivo.
5: Disminuir (problema de minimización) o aumentar (problema de maximización) el nivel de la curva hasta que toque el último punto factible posible.
6: La solución óptima corresponde al menor (mayor) valor de la curva de nivel en un problema de minimización (maximización). La solución $\mathbf{x}^*$ es el último punto factible que toca la curva de nivel.
7: **else**El problema es infactible
8: **end if**

Ejemplo (Encontrar la asignación óptima de producción mediante el método gráfico)

El gráfico izquierdo en la Figura 8 corresponde al paso inicial del método gráfico descrito anteriormente. La línea verde es la curva de nivel correspondiente a la función objetivo. En el gráfico derecho se observa cómo, al aumentar (es un problema de maximización) el valor de la curva de nivel, encontramos la solución óptima en $x^* = (2, 6)$ (el último punto factible) para $z^* = 36$ (el valor de la curva de nivel).

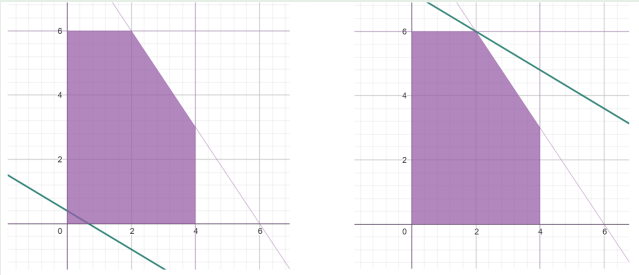


Figura 8: Método gráfico aplicado al Ejemplo 6

Ejemplo (Soluciones infinitas)

Existen casos en los que la función objetivo es paralela a alguna de las fronteras del politopo de la región factible, ver Figura 9. En tales casos, tendremos soluciones óptimas infinitas, como se muestra en el Ejemplo 16. Para visualizar la región y la implementación del método gráfico, consulte este gráfico en Geogebra.

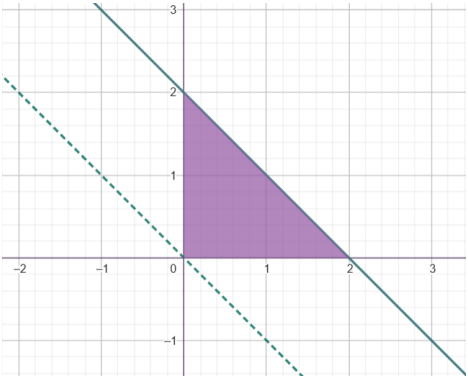


Figura 9: Método gráfico aplicado al Ejemplo 16

Hasta ahora, el método presentado en esta sección permite resolver problemas pequeños y bidimensionales. Sabemos que, en la práctica, la mayoría de los problemas son mucho más grandes que eso. Por lo tanto, introducimos dos solucionadores computacionales para abordar esta limitación. Comenzaremos con el solver de Matlab para problemas lineales.

Importante (La plantilla de Matlab no estará disponible durante los exámenes)

Durante los exámenes se empleará como optimizador única y exclusivamente los módulos de Matlab descritos en clase. La plantilla de Aula Virtual no estará disponible, por lo que un conocimiento de la sintaxis y los comandos básicos para la resolución del problema mediante un script de Matlab será requerido.

Para resolver un problema de programación lineal se empleará la función de Matlab conocida como `linprog`, se puede consultar su configuración en Mathworks, o escribiendo `help linprog` en la consola de Matlab. Para más información consultar Apéndice G.
La función `linprog` resuelve los problemas LP con variable continua, esto es, el problema tiene que ser **lineal**, y las variables tienen permitido asumir valores no-enteros. La lista de argumentos se resume en la Tabla 2.

Tabla 2: Entradas/salidas de la función `linprog` en MATLAB

Entrada / Salida	Descripción	Tipo / Valores
Entradas		
`f`	Vector de coeficientes de la función objetivo	Vector columna
`A`	Matriz de coeficientes de desigualdades lineales	Matriz $m \times n$ (opcional)
`b`	Vector de términos independientes de desigualdades	Vector columna (opcional)
`Aeq`	Matriz de coeficientes de ecuaciones lineales	Matriz $p \times n$ (opcional)
`beq`	Vector de términos independientes de ecuaciones lineales	Vector columna (opcional)
`lb`	Vector de límites inferiores de las variables	Vector columna (opcional)
`ub`	Vector de límites superiores de las variables	Vector columna (opcional)
`x0`	Vector inicial de la solución	Vector columna (opcional)
Salidas		
`x`	Vector solución óptima que satisface las restricciones	Vector columna
`fval`	Valor de la función objetivo en la solución óptima	Escalar
`exitflag`	Indica el resultado de la optimización	Escalar entero
`output`	Información detallada del solver (iteraciones, tiempo, etc.)	Estructura
`lambda`	Multiplicadores de Lagrange asociados a las restricciones lineales	Estructura (si aplica)

En el Código 1 se ejemplifica el uso de la función `linprog` para la resolución del Ejemplo 6. Nótese que `linprog` solamente acepta problemas de mínimo, con restricciones de menor o igual; problemas en forma canónica, por lo que se debe cambiar el sentido de la desigualdad y considerar el opuesto de la función objetivo. Ver código 1.

```
% vector de costes
f = -[3 5];
%vector de recursos
b =  [4 12 18];
%cotas
ub = []; %superior
lb = zeros(1,2); % inferior
%matriz de restricciones
     %x1,x2
A = [ 1 0;
      0 2;
      3 2];
%igualdades
Aeq = []; beq= [];
%optimizar
[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub)
%imprimir resultados
x
-fval
```

Este software se presenta como material complementario del curso para usuarios de Windows, y consiste simplemente en una codificación de `Pyomo` para ejecutarse en ordenadores personales con Windows. Está disponible en Aula Virtual.
Esta solución es una interfaz gráfica sencilla (GUI) que contiene varios paquetes y solucionadores de `Pyomo`. Su uso es intuitivo y sigue las mismas instrucciones que el script de `Pyomo`. Más información en el Apéndice B.

Importante (Pyomo solver no estará disponible para su uso en exámenes)

El solver de Pyomo no estará disponible durante los exámenes. Durante los exámenes se empleará como optimizador única y exclusivamente los módulos de Matlab descritos en el curso.

Ejemplo (Resolviendo el problema de asignación de producción)

Recordemos el Ejemplo 6. La GUI de este solucionador solicita los elementos básicos del problema. Introduciendo cada uno en el área correspondiente, como se muestra en la Figura 10, obtenemos el resultado deseado.

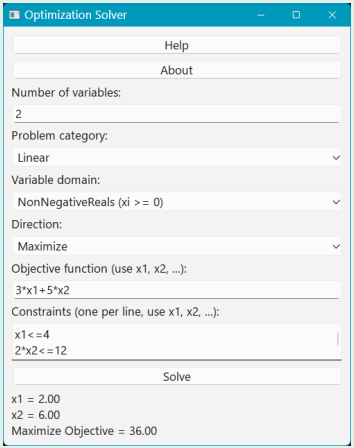


Figura 10: Resolución del Ejemplo 6 con el solucionador para Windows

Análisis post-óptimo

El análisis post-óptimo es un análisis *what-if* que realizamos después de encontrar la solución óptima. Este análisis responde a preguntas sobre cuán sólida y/o robusta es la solución actual, y qué acciones debemos llevar a cabo para mejorarla. También es una herramienta para estimar el impacto de incluir nuevas variables o mejorar/limitar recursos en el modelo.

Para realizar el análisis post-óptimo, introducimos cambios en los parámetros del modelo uno por uno, manteniendo el resto constante. Este análisis nos permite estimar con alta precisión cómo afectan los cambios a la solución actual sin necesidad de resolver un nuevo modelo.

En este curso nos centraremos en la obtención y significado práctico de los precios sombra.

Consideremos un problema como (2), con solución óptima $\mathbf{x}^*$, valor óptimo z^* y coeficiente del lado derecho (RHS) $\mathbf{b}$, al que denominamos LP. Consideremos un problema asociado $\hat{\text{LP}}_i$, con solución óptima $\hat{\mathbf{x}}^*$, valor óptimo $\hat{z^*}$ y vector RHS $\hat{\mathbf{b}}_i = (b_1, \ldots, b_i + 1, \ldots, b_n)$.

Definición (Precio Sombra)

El **precio sombra** s_i correspondiente a la i-ésima restricción cuantifica la mejora marginal en la función objetivo z^* resultante de un incremento unitario en b_i. Sea $\hat{z^*}$ la solución del problema donde $\hat{b_i} = b_i + 1$, entonces:

$$\hat{z^*} = z^* + s_i$$

Si $s_i^* = 0$, entonces los cambios en b_i no afectan el valor óptimo de la función objetivo.

Observación

- Los cambios en b_i se denotan como Δb_i, es decir, $\hat{b}_i - b_i = \Delta b_i$.
- El precio sombra s_i asociado al coeficiente b_i, de la restricción $i = 1, \ldots, m$ contiene la información acerca de cuánto cambia la función objetivo si el coeficiente b_i se incrementa en una unidad.
- Si $s_i = 0$, esto significa que aumentar el recurso i no mejora la solución, y por lo tanto el recurso no es crítico.
- Si $s_i \neq 0$, se tiene que el recurso i es crítico y que aumentarlo en una unidad cambia el valor óptimo del problema.

Ejemplo (Precios sombra de Ejemplo 6)

Para obtener los precios sombra se extrae la información dual que brinda `linprog` almacenda en la estructura `lambda`. Ver Código 2.

```
f = -[3 5]; % vector de costes
b =  [4 12 18]; %vector RHS de recursos
lb = zeros(1,2); % cota inferior
%matriz de restricciones
     %x1,x2
A = [ 1 0;
      0 2;
      3 2];
%optimizar
[x,fval,exitflag,output,lambda] = linprog(f,A,b,[],[],lb)
%imprimir resultados
x
-fval
[lambda.ineqlin;lambda.lower]
```

Código 2: Extracción de precios sombra

La información de `lambda` se extrae de la estructura. Como en este caso contamos con desigualdades de recursos (`b`), se emplea `lambda.ineqlin` para acceder a los precios sombra.
De acuerdo a la información obtenida, se tiene que

$$\mathbf{s} = (0, 1.5, 1)$$

Esto significa que:

- $s_1 = 0$, se refiere a la desigualdad $x_1 \leq 4$. Si se aumenta la disponibilidad de horas semanales en la planta 1, las ganancias no mejoran. Luego concluimos que el tiempo en disponible en la planta 1 **no** es un recurso crítico.
- $s_2 = 1.5$, se refiere a la desigualdad $x_2 \leq 12$. Si se aumenta la disponibilidad de horas semanales en la planta 2, de 12 a 13, las ganancias aumentarán en 1.5 unidades, de 36 a 37.5. Por lo tanto, las horas en la planta 2 son un recurso crítico.
- $s_3 = 1$, se refiere a la desigualdad $3x_1 + 2x_2 \leq 18$. Si se aumenta la disponibilidad de horas semanales en la planta 3, de 18 a 19, las ganancias aumentarán en 1 unidad, de 36 a 37. Por lo tanto, las horas en la planta 3 son un recurso crítico.

§II. Programación Entera

- Modelos de programación entera puros y mixtos.
- Solución numérica.
- Modelación con variables binarias.
  - Condicionales y pre-requisitos
  - Implicaciones lógicas.
  - Inclusión de restricciones.
  - Acotación.
  - Costes fijos.

La **Programación Lineal Entera (ILP)** es una extensión fundamental de la Programación Lineal (LP) en la que algunas o todas las variables de decisión deben tomar valores enteros. Este tipo de modelado es esencial cuando se abordan problemas que implican decisiones discretas, como planificación de horarios, ubicación de instalaciones o diseño de redes. Particularmente en aplicaciones de telecomunicaciones, esto incluye determinar el número de enlaces, asignación de canales o planificación de nodos en una red.

Definición (Problema de Programación Lineal Entera)

La forma general de un problema de ILP puede escribirse como:

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.a.} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \in \mathbb{Z}^n \text{ (si todas las variables son enteras).} \end{aligned} \tag{3}$$

Donde:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ es el vector de variables de decisión.
- $\mathbf{c}^T \mathbf{x}$ es la función objetivo lineal.
- $A\mathbf{x} \leq \mathbf{b}$ representa el sistema de restricciones lineales.
- La condición de integralidad exige que algunas o todas las variables pertenezcan al conjunto de enteros.

Observación

Nótese que $\mathbb{Z}_+ = \mathbb{Z} \cap \{x > 0\}$, es decir, x es un entero estrictamente positivo, excluyendo el cero. Sin embargo, para los fines de este curso, dado que las regiones factibles deben ser cerradas, cuando nos referimos a $\mathbb{Z}_+$, incluiremos el cero, es decir, redefinimos $\mathbb{Z}_+ = \{x \geq 0\} \cap \mathbb{Z}$.

En un modelo **entero puro**, *todas las variables de decisión* deben tomar valores enteros, mientras que en un modelo **mixto** (**MILP**), solo un subconjunto de las variables son enteras y las demás permanecen continuas.
Los modelos ILP y MILP son más complejos de resolver que los modelos LP, ya que las restricciones de integralidad hacen que la región factible deje de ser convexa. Para resolver estos problemas de manera eficiente se emplean algoritmos especializados como *branch-and-bound*, *branch-and-cut* y *cutting planes*.
En la práctica, ¿cómo resolver un problema ILP? Primero, veamos qué ocurre si tratamos este problema como un problema LP, con el siguiente ejemplo:

Ejemplo

Consideremos el siguiente problema de optimización:

$$\begin{aligned} \max \quad & z = 5x_1 + 3x_2 \\ \text{s.a.} \quad & -1.4x_1 + x_2 \leq 2, \\ & 3x_1 + x_2 \leq 12, \\ & x_1, x_2 \geq 0, \end{aligned}$$

Ejemplo

Para resolver este problema, relajamos la condición que obliga a las variables a ser enteras. Trabajamos con el problema **relajado**:

$$\begin{aligned} \max \quad & z = 5x_1 + 3x_2 \\ \text{s.a.} \quad & -1.4x_1 + x_2 \leq 2, \\ & 3x_1 + x_2 \leq 12, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Aplicando el método analítico y encontrando los vértices de la región factible, obtenemos la siguiente tabla:

x	z
$(0,0)$	0
$(0,2)$	6
$(4,0)$	20
$(2.25,5.18)$	26.91

Concluimos que la solución óptima del problema relajado es $\mathbf{x}^* = (2.25, 5.18)$ para $\mathbf{z}^* = 26.91$. También podemos aplicar el método gráfico y obtener el mismo resultado, ver Figura 11 izquierda.

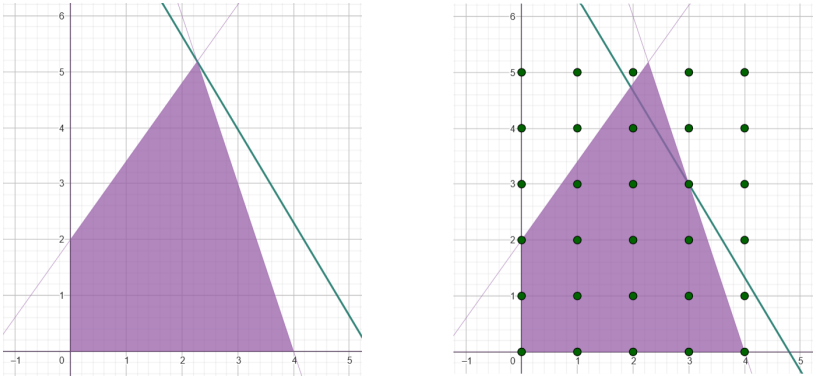


Figura 11: Problema relajado vs. problema entero

Ejemplo

Sin embargo, las soluciones deseadas deben ser enteras, por lo que aplicar los métodos anteriores ya no es válido. Volviendo al problema original, debemos considerar únicamente los puntos dentro de la región factible con coordenadas enteras, ver Figura 11 derecha. Ahora, aumentamos el valor de las curvas de nivel de la función objetivo hasta alcanzar el último punto factible con coordenadas enteras. De esta manera, concluimos que la solución óptima del problema entero es $\mathbf{x}^* = (3,3)$ con $z^* = 24$.

Definición (Problema Relajado)

Consideremos el problema entero (ILP) (3). Llamamos **Problema Relajado (RLP)** al problema asociado:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.a.} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x} \geq 0. \end{aligned} \tag{4}$$

En otras palabras, relajamos la condición que restringe las variables a ser enteras.

Proposición (ILP versus RLP)

Relación entre soluciones:

- En general, la solución del ILP y la del RLP no son iguales.
- No es correcto calcular la solución del ILP como una aproximación al punto entero más cercano de la solución del RLP.
- El único caso en el que ambas soluciones coinciden es cuando la matriz de restricciones es unimodular.

Relación entre regiones factibles:

- La región factible del ILP siempre es más pequeña que la del RLP, es decir, contiene menos soluciones.
- La región factible del ILP es un subconjunto de la región factible del RLP; todas las soluciones del ILP son también soluciones del RLP.
- La solución del ILP no está obligada a encontrarse en la frontera de la región factible (no existe una frontera continua).

Relación entre soluciones óptimas:

- Dado que la región factible del ILP es un subconjunto de la del RLP, la solución del RLP, z^*_{LP}, siempre es mejor que la solución del ILP, z^*_{IP}, proporcionando así una cota:
- Minimización: $z^*_{IP} \geq z^*_{LP}$,
- Maximización: $z^*_{IP} \leq z^*_{LP}$.

Para algunos problemas puede ser útil considerar solo un subconjunto de variables como enteras y permitir que el resto sean continuas. A estos problemas los denominamos **Problemas de Programación Entera Mixta**.

Definición (Problema de Programación Entera Mixta)

Decimos que un problema LP es un problema de programación entera mixta si un subconjunto de las variables está restringido a ser entero, es decir:

$$\begin{aligned} \min \quad & \mathbf{c}_1^T \mathbf{x}_1 + \mathbf{c}_2^T \mathbf{x}_2 \\ \text{s.a.} \quad & A\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{x}_2 \in \mathbb{Z}. \end{aligned} \tag{5}$$

En otras palabras, relajamos la condición que restringe todas las variables a ser enteras.

¿Cómo resolver un problema entero? Para resolver un problema entero en este curso, utilizamos software:

- **Matlab `intlinprog`:** de manera análoga a la solución de problemas lineales de variable continua, Matlab implementa la función `intlinprog` para la resolución de ILP y MILP. Más información acerca de estas funciones puede ser consultada en Mathworks o escribiendo `help intlinprog` en la consola de Matlab. Para más información consultar Apéndice G.
- **Pyomo Solver:** podemos especificar el dominio de las variables eligiendo la opción `NonNegativeIntegers/NonPositiveIntegers/Integers` en el menú contextual "Domain", más detalles en el Apéndice B. Al establecer esta opción, todas las variables se consideran enteras, por lo que no es posible resolver MILP.

Tabla 3: Entradas y salidas de la función `intlinprog` en MATLAB

Entrada / Salida	Descripción	Tipo / Valores
Entradas		
`f`	Vector de coeficientes de la función objetivo	Vector columna
`intcon`	Índices de las variables que deben ser enteras	Vector de enteros
`A`	Matriz de coeficientes de desigualdades lineales	Matriz m×n (opcional)
`b`	Vector de términos independientes de desigualdades	Vector columna (opcional)
`Aeq`	Matriz de coeficientes de ecuaciones lineales	Matriz p×n (opcional)
`beq`	Vector de términos independientes de ecuaciones lineales	Vector columna (opcional)
`lb`	Vector de límites inferiores de las variables	Vector columna (opcional)
`ub`	Vector de límites superiores de las variables	Vector columna (opcional)
`x0`	Vector inicial de la solución	Vector columna (opcional)
Salidas		
`x`	Vector solución óptima que satisface las restricciones	Vector columna
`fval`	Valor de la función objetivo en la solución óptima	Escalar
`exitflag`	Indica el resultado de la optimización	Escalar entero
`output`	Información detallada del solver (iteraciones, tiempo, etc.)	Estructura
`lambda`	Multiplicadores de Lagrange asociados a las restricciones lineales	Estructura (si aplica)

Ejemplo

Para resolver el ejemplo anterior empleamos el código 3.

```
%intlinprog example
f = [-5; -3];                % Cambiado de signo (max -> min)
A = [-1.4  1;
      3    1];
b = [2;
     12];
intcon = [1 2];             % Ambas variables son enteras
lb = [0; 0];                % x1, x2 >= 0
ub = [];                    % Sin cotas superiores explicitas
[x_opt, fval] = intlinprog(f, intcon, A, b, [], [], lb, ub);
z_opt = -fval              % Volvemos a cambiar el signo
x_opt
```

Código 3: Código para Ejemplo 23

En ocasiones necesitamos modelar una decisión de tipo sí/no; en tales casos utilizamos variables binarias.

Definición (Variable Binaria)

Se denomina variable binaria o variable 0-1 a una variable entera cuyos únicos valores posibles son 0 o 1, es decir:

$$x_j = \begin{cases} 1, & \text{si la condición } j \text{ se cumple,} \\ 0, & \text{en caso contrario.} \end{cases}$$

Ejemplo (Uso de variables binarias)

Una empresa tecnológica está considerando la activación de la producción de cuatro tipos de dispositivos de telecomunicaciones: A, B, C y D, para el próximo año. La producción de los dispositivos A, B, C y D implica:

- un coste fijo asociado a la puesta en marcha de la línea de producción (en EUR), y
- un requerimiento de horas de trabajo por unidad producida.

La tabla resume el coste fijo, la mano de obra requerida por unidad y el beneficio esperado por unidad:

Dispositivo	Coste Fijo (EUR)	Mano de obra por unidad (horas)	Beneficio por unidad (EUR)
A	14 500	10	50
B	10 000	15	60
C	8 000	5	55
D	9 000	14	80

La empresa dispone de 20 000 horas de mano de obra en esta temporada y se requiere que:

I. si se activa la producción de B, no se puede activar la producción de D,

II. si se activa la producción de A, entonces al menos C o D también deben activarse,

III. al menos una línea de producción debe ser activada,

IV. se debe producir C o D,

Para cada tipo de producción, si se activa, la empresa se compromete a producir un mínimo de 1000, 1500, 3000 y 2000 unidades de los dispositivos A, B, C y D, respectivamente. Formule el modelo de programación matemática que permita a la empresa planificar su producción anual, con el objetivo de **maximizar** su beneficio total.

Ejemplo

En este problema, debemos tomar dos tipos de decisiones: decidir si se activa una línea de producción y el nivel de producción de cada tipo de producto. La primera decisión es binaria (sí/no), y la segunda representa simplemente cuántas unidades de cada producto se producen. Por lo tanto, podemos utilizar como variables de decisión:

- x_j: unidades del producto j, donde $j \in I = A, B, C, D$, y $x_j \in \mathbb{Z}_+$ para todo $j \in I$; y
- $y_i = \begin{cases} 1, & \text{si la línea de producción } i \text{ está activada,} \\ 0, & \text{en caso contrario} \end{cases}$ para $i \in I$.

Obsérvese que usamos I como nuestro conjunto índice y declaramos la activación de las líneas de producción como una variable binaria.

Proposición

Condicionales y Prerrequisitos

1. j es condicional respecto a i. Para completar j, i debe completarse:

$$x_j - x_i \leq 0 \quad (x_j = 1 \Rightarrow x_i = 1)$$

2. i es un pre-requisito para j:

$$x_i - x_j = 0 \quad (x_i \Leftrightarrow x_j)$$

3. i y j son mutuamente excluyentes (no pueden ser ambos iguales a 1 simultáneamente):

$$x_i + x_j \leq 1$$

Ejemplo

1. Para llevar a cabo el proyecto 5, el proyecto 2 debe completarse primero: $x_5 \leq x_2$.
2. Los materiales se compran (actividad i) solo si la tarea correspondiente es seleccionada: $x_j - x_i = 0$.
3. Los ítems 4 y 2 no pueden seleccionarse al mismo tiempo: $x_4 + x_2 \leq 1$.

Esto significa que, volviendo al Ejemplo 30, la condición "si se activa la producción de B, no se puede activar la producción de D" se escribe como

$$y_B + y_D \leq 1$$

porque ambos no pueden ser iguales a 1 simultáneamente. Esto conduce a un conjunto más general de selección de k actividades de entre n.

Proposición (Número de Actividades a Completar)

1. Completar como máximo k actividades de entre n: $\sum_{i \in I} x_i \leq k$
2. Completar al menos k actividades de entre n: $\sum_{i \in I} x_i \geq k$
3. Completar exactamente k actividades de entre n: $\sum_{i \in I} x_i = k$

Ejemplo

1. De 5 miembros, seleccionar como máximo 2: $x_1 + x_2 + x_3 + x_4 + x_5 \leq 2$.
2. De 5 ítems, tomar al menos 3: $x_1 + x_2 + x_3 + x_4 + x_5 \geq 3$.
3. De 5 proyectos, completar exactamente 2: $x_1 + x_2 + x_3 + x_4 + x_5 = 2$.

Esto significa que, volviendo al Ejemplo 30, la condición "al menos una línea de producción debe ser activada" se escribe como

$$y_A + y_B + y_C + y_D \geq 1$$

porque al menos una de ellas debe ser igual a 1.

Proposición (Inclusión de Restricciones OR)

El modelo debe incluir una de dos restricciones, $g_1(x) \leq b_1$ o $g_2(x) \leq b_2$. Se introduce una nueva variable binaria δ para representar la no activación de una restricción, y las restricciones se incluyen como

$$g_1(x) \leq b_1 + M_1\delta$$
$$g_2(x) \leq b_2 + M_2(1-\delta)$$

Las constantes M_1, M_2 deben elegirse lo suficientemente grandes para trivializar la restricción no activada (típicamente una constante positiva muy grande).

Ejemplo

Suponga que para viajar de A a B, se debe tardar menos de 10 minutos o pagar menos de 2 € por transporte. Sea $g_1(x)$ el tiempo de viaje y $g_2(x)$ el coste de viaje. Entonces el modelo incluiría:

$$g_1(x) \leq 10 + M_1\delta$$
$$g_2(x) \leq 2 + M_2(1-\delta)$$

con la elección adecuada de M. Nótese que si $\delta = 0$, la restricción de tiempo de viaje está activa; si $\delta = 1$, la restricción de coste está activa.

Esto es particularmente útil cuando se enfrenta al tipo de restricción $P \Rightarrow Q$, donde P y Q son proposiciones. Volviendo al Ejemplo 30, la condición «si se activa la producción de A, entonces al menos C o D también deben activarse» significa P: producción de A activada, $y_A = 1$ implica Q: al menos C o D activadas, $y_C = 1$ o $y_D = 1$.

Para modelar este tipo de implicaciones, las reducimos a implicaciones OR usando la siguiente equivalencia lógica

$$(P \Rightarrow Q) \Leftrightarrow (\neg P \vee Q) \tag{6}$$

donde $\neg$ significa la negación de P y $\vee$ significa OR (al menos una de las condiciones se cumple). Una vez obtenida la proposición OR, procedemos a usar el mecanismo OR como antes.

¿Cómo aplicar esto a variables binarias?

Ejemplo (Implicación lógica)

La condición "si se activa la producción de A, entonces al menos C o D también deben activarse" significa P: producción de A activada, $y_A = 1$ implica Q: al menos C o D activadas, $y_C = 1$ o $y_D = 1$. Esto es
$y_A = 1 \Rightarrow y_C = 1$ o $y_D = 1$.
Como trabajamos con variables binarias, la condición $y_A = 1$ es equivalente a $y_A \geq 1$.

Ejemplo (Implicación lógica)

Por otro lado, la condición $y_C = 1$ o $y_D = 1$ es equivalente a la restricción $y_C + y_D \geq 1$. Así, el esquema $P \Rightarrow Q$ se convierte en $y_A \geq 1 \Rightarrow y_C + y_D \geq 1$.
La negación de P sería $\neg P = (y_A \leq 0)$, por lo que según la ecuación (6) obtenemos la relación OR

$$y_A \leq 0 \text{ o } y_C + y_D \geq 1$$

Use la Proposición 80 para concluir que la relación OR anterior puede incluirse en nuestro modelo como dos restricciones relacionadas mediante una nueva variable binaria

$$\delta = \begin{cases} 1, & \text{si la primera restricción está activada} \\ 0, & \text{si la segunda restricción está activada} \end{cases}$$

como

$$y_A \leq M_1 \delta$$
$$y_C + y_D \geq 1 + M_2(1 - \delta)$$

eligiendo el valor adecuado de M_1 y M_2. Nótese que si $\delta = 0$, activamos la primera restricción, lo que significa que la segunda debe ser trivial. Trivializar restricciones que involucran variables binarias es simple: basta elegir una condición que siempre se cumpla. Por ejemplo, $y_C + y_D \geq 0$, o cualquier número negativo, así que sea $1 + M_2 \leq 0$, lo que implica que $M_2 \leq -1$, por lo tanto cualquier valor menor que -1 es adecuado para M_2.
Repita el mismo razonamiento para M_1. Sea $\delta = 1$, entonces la segunda restricción está activada y la primera debe ser trivial. Para ello, elija $M_1 \geq 1$. Así, las restricciones resultan

$$y_A \leq \delta$$
$$y_C + y_D \geq \delta$$

Observación

Del último ejemplo surge una pregunta natural: ¿Se pueden combinar las dos restricciones anteriores en una sola? En este caso la respuesta es sí. Sin embargo, en general no se recomienda. Considere el contraejemplo "si $y_A = 1$ y $y_B = 1$ implica $y_C = 1$". Usando el mecanismo anterior, obtenemos las restricciones

$$y_A + y_B \leq 1 + \delta$$
$$y_C \geq \delta$$

Si unimos estas dos restricciones obtenemos $y_A + y_B \leq 1 + y_C$, lo que permite el caso $y_A = 1$, $y_B = 0$ y $y_C = 1$, que no es factible.

Proposición (Inclusión de K de N restricciones)

El modelo debe incluir K de N restricciones del tipo $g_j(x) \leq b_j$, para $j = 1, \ldots, N$. Se introduce una nueva variable binaria δ_j para representar la no activación de cada restricción, y las restricciones se incluyen como

$$
\begin{aligned}
g_1(x) &\leq b_1 + M_1 \delta_1 \\
g_2(x) &\leq b_2 + M_2 \delta_2 \\
&\vdots \\
g_N(x) &\leq b_N + M_N \delta_N
\end{aligned}
$$

con

$$\sum_{j=1}^{N} \delta_j = N - K$$

Las constantes M_j deben elegirse adecuadamente de forma que trivialicen las restricciones no activadas.

Ejemplo

Para los nuevos laboratorios biomédicos que la empresa abrirá en Madrid, al menos dos de tres protocolos medioambientales deben respetarse: emisiones $g_1(x) \leq 10$, consumo energético $g_2(x) \leq 20$, o residuos $g_3(x) \leq 30$. En el modelo, estos se incluirían como

$$g_1(x) \leq 10 + M_1\delta_1$$
$$g_2(x) \leq 20 + M_2\delta_2$$
$$g_3(x) \leq 30 + M_3\delta_3$$

con

$$\sum_{j=1}^{3} \delta_j = 3 - 2 = 1$$

Proposición (Acotación)

En el caso de un problema que involucra variables binarias y enteras, en el cual el valor de la variable entera depende de la activación de algunas variables binarias, podemos utilizar la técnica de **acotación**. Por ejemplo, el nivel de producción de un bien x depende de realizar un pedido de material y, siendo la variable binaria y la que indica si el envío se solicita o no. Esta restricción se introduce en el modelo como $\mathbf{x} \leq \mathbf{My}$, donde M es una constante que excede los límites de los valores de x.

Ejemplo

Retomando el ejemplo 30, la condición «Para cada tipo de producción, si está activada, la empresa se compromete a producir un mínimo de 1000, 1500, 3000 y 2000 unidades de los dispositivos A, B, C y D, respectivamente», se escribe como un conjunto de cuatro restricciones:

$$M_1 y_A \leq x_A \leq M y_1$$
$$M_2 y_B \leq x_B \leq M y_2$$
$$M_3 y_C \leq x_C \leq M y_3$$
$$M_4 y_D \leq x_D \leq M y_4$$

donde los valores de las constantes son 1000, 1500, 3000 y 2000, respectivamente.

El valor de M y su inclusión obligan a que las variables x_i sean cero si y_i es cero, lo cual constituye el caso útil. La constante M se elige lo suficientemente grande para evitar una restricción artificial de acotación superior que no está en el modelo.

Proposición (Funciones con N valores posibles)

Sea f una función que puede tomar los valores $d_1, d_2, \ldots, d_N$. Entonces f puede expresarse como

$$f(x) = \sum_{i=1}^{N} d_i y_i$$

$$\sum_{i=1}^{N} y_i = 1$$

donde y_i son variables binarias que indican qué valor asume f.

Ejemplo

Las horas de operación de una planta farmacéutica están dadas por $g(x)$. La actividad de la planta no debe exceder las 18 horas, pero solo puede operar en bloques de 6 horas. Por lo tanto, $g(x)$ puede tomar los valores 6, 12 o 18. Esto puede representarse como

$$g(x) = 6y_1 + 12y_2 + 18y_3$$

$$y_1 + y_2 + y_3 = 1$$

Esto significa, retomando el Ejemplo 30, que la función objetivo se escribe como

$$\max_{\mathbf{x}} z = 50x_A + 60x_B + 55x_C + 80x_D - (14500y_A + 10000y_B + 8000y_C + 9000y_D)$$

Por lo tanto, el modelo del Ejemplo 30 queda

$$\begin{aligned}
\max_{\mathbf{x}} \quad & z = 50x_A + 60x_B + 55x_C + 80x_D - (14500y_A + 10000y_B + 8000y_C + 9000y_D) \\
s.t \quad & 10x_A + 15x_B + 5x_C + 14x_D \leq 20000 \\
& y_B + y_D \leq 1 \\
& y_A + y_B + y_C + y_D \geq 1 \\
& y_A \leq \delta \\
& y_C + y_D \geq \delta \\
& y_C + y_D \geq 1 \\
& 1000y_A \leq x_A \leq My_A \\
& 1500y_B \leq x_B \leq My_2 \\
& 3000y_C \leq x_C \leq My_C \\
& 2000y_D \leq x_D \leq My_D
\end{aligned}$$

para $x_j, \geq 0, j \in I$, enteras y $\delta, y_i \in \{0,1\},$ para $i \in I$. M suficientemente grande, por ejemplo $M = 10^6$.

La solución es

$$\mathbf{x}^* = (0,0,4000,0,0,0,0,1,0) \qquad z^* = 212\,000$$

Así, para maximizar el beneficio la empresa debe activar únicamente la línea C y producir 4000 unidades del producto C para obtener un beneficio de 212 000 euros.
En el Código 4 se muestra el código empleado para la resolución, utilizando `intlinprog`.

```
% vector de costes
% xA    xB  xC  xD  yA  yB  yC  yD  delta
f = -[50 60 55 80 -14500 -10000 -8000 -9000 0];
M = 90000; %M grande
b =  [20000 1 -1 0 0 -1 zeros(1,8)]; %vector de recursos
%cotas
ub = [M M M M 1 1 1 1 1];
lb = zeros(1,9);
intcon = 1:9; % indices de variables enteras
%matriz de restricciones
     % xA    xB  xC  xD  yA  yB  yC  yD  delta
A = [ 10 15 5 14 0 0 0 0 0;
      0 0 0 0 0 1 0 1 0;
      0 0 0 0 -1 -1 -1 -1 0;
      0 0 0 0 1 0 0 0 -1;
      0 0 0 0 0 0 -1 -1 1;
      0 0 0 0 0 0 -1 -1 0;
      1 0 0 0 1000 0 0 0 0;
      0 -1 0 0 0 1500 0 0 0;
      0 0 -1 0 0 0 3000 0 0;
      0 0 0 -1 0 0 0 2000 0;
      1 0 0 0 -M 0 0 0 0;
      0 1 0 0 0 -M 0 0 0;
      0 0 1 0 0 0 -M 0 0;
      0 0 0 1 0 0 0 -M 0];
Aeq = []; beq= []; %igualdades
[x,fval] = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub); %optimizar
```

Para incluir variables binarias, en la definición de los parámetros `lb`, `ub`, se incluyen como $0, 1$, respectivamente.
Notemos que, para Matlab, el modelo debe ser escrito en forma canónica, esto es, dirección de mínimo y desigualdades en forma de menor o igual. Luego, las restricciones de tipo

$$1500y_B \leq x_B \leq My_2$$

se codifican en el modelo como dos restricciones

$$-x_B + 1500y_B \leq 0$$
$$x_B - My_2 \leq 0$$

Nótese, además, que se debe adicionar una nueva variable δ, binaria, como una entrada extra a la función objetivo y con una columna en la matriz de restricciones.
Se define la constante $M = 9000$, para emplearla como cota superior, tanto del problema como en las restricciones de acotación de producción.

§III. Problemas de transporte y asignación

- Problema de transporte.
- Problema de asignación.

Para ampliar el contenido de este capítulo, consultar [Hillier and Lieberman, 2015, Cap. 8].

El *problema de transporte* es un modelo clásico en investigación operativa que aborda la distribución de un producto desde varias fuentes (o nodos de oferta) hacia varios destinos (o nodos de demanda) de manera que se minimice el coste total de transporte. Fue introducido por primera vez en la década de 1940 y sigue siendo una de las aplicaciones más estudiadas de la programación lineal debido a su elegancia teórica y relevancia práctica.

Consideramos la siguiente situación:

- Existen m puntos de oferta (fábricas, almacenes, plantas de producción), donde cada punto de oferta i tiene una capacidad total disponible $a_i \geq 0$.
- Existen n puntos de demanda (mercados, hospitales, zonas de clientes), donde cada punto de demanda j requiere una demanda total $b_j \geq 0$.
- El coste unitario de transporte desde el punto de oferta i al punto de demanda j es c_{ij}, donde $I = \{1, \ldots, m\}$ y $J = \{1, \ldots, n\}$.

Esta situación se representa como un grafo en la Figura 12.

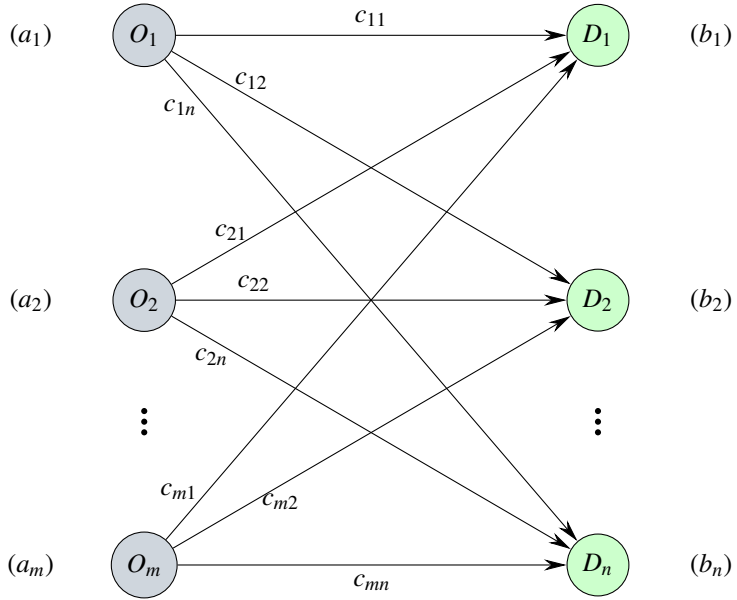


Figura 12: Representación como grafo del problema de transporte

Definición (Problema de transporte)

En este problema, la decisión consiste en cuántas unidades enviar desde el origen i al destino j, por lo que definimos las variables de decisión:

$$x_{ij} = \text{número de unidades enviadas desde el punto de oferta } i \text{ al punto de demanda } j.$$

donde $I = \{1,\dots,m\}$ y $J = \{1,\dots,n\}$.

El objetivo es minimizar el coste total de transporte $\boxed{\min_{\mathbf{x}} z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}}$

sujeto a las restricciones (una para cada punto de oferta y destino):

$$\sum_{j=1}^{n} x_{ij} \le a_i, \qquad i = 1,\dots,m,$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j, \qquad j = 1,\dots,n,$$

$$x_{ij} \ge 0, \qquad \forall i,j \in I \times J.$$

Estas restricciones garantizan que los envíos desde cada nodo de oferta no excedan su disponibilidad y que la demanda en cada destino sea satisfecha. En el caso especial en que $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, el problema se denomina *problema de transporte balanceado*, y la solución está garantizada. En caso contrario, se pueden introducir nodos ficticios de oferta o demanda para equilibrar el sistema.

Ejemplo

Una empresa necesita enviar materia prima desde sus 2 almacenes a 3 fábricas. El inventario en unidades de materia prima de cada almacén, las demandas en unidades de materia prima de cada fábrica y los costes unitarios de transporte (euros por unidad de materia prima enviada entre cada origen y destino) se muestran en la Tabla 4. Formule y resuelva el modelo de optimización que minimiza los costes de transporte.

	Fábrica 1	Fábrica 2	Fábrica 3	Oferta
Almacén 1	4	6	8	20
Almacén 2	5	4	3	30
Demanda	10	25	15	

Tabla 4: Matriz de costes de transporte con ofertas y demandas.

El problema de transporte también puede visualizarse como un *grafo bipartito*, donde los nodos de la izquierda representan los almacenes (ofertas), los nodos de la derecha representan las fábricas (demandas) y las aristas representan los posibles envíos con sus costes asociados. La situación del problema se representa en el grafo de la Figura 13

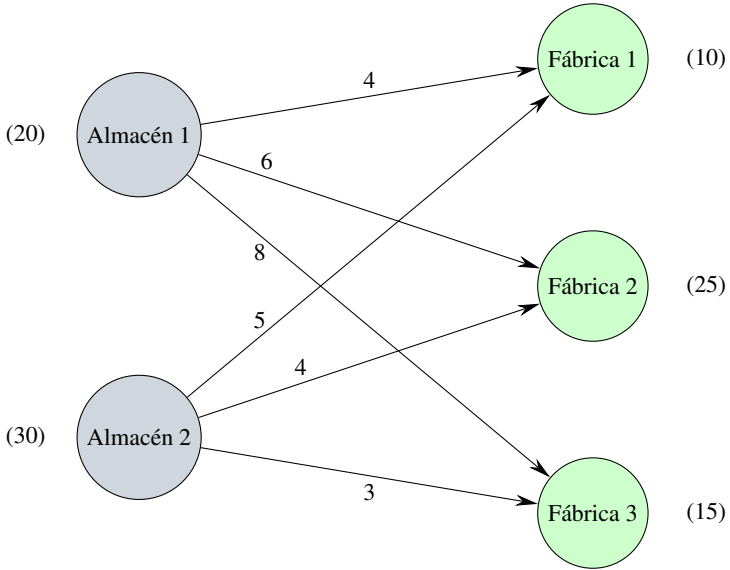


Figura 13: Grafo del Ejemplo 41

Ejemplo

Dado que tenemos 2 almacenes y 3 fábricas, se tiene que $m = 2$ y $n = 3$. Así, $I = \{1,2\}$ y $J = \{1,2,3\}$, por lo tanto las variables de decisión son:

$$x_{ij} : \text{unidades enviadas desde el almacén } i \text{ a la fábrica } j \text{ para } (i,j) \in I \times J, x_{ij} \geq 0$$

Minimizar el coste de transporte implica minimizar la función de coste $\sum_{(i,j)\in I\times J} c_{ij}x_{ij}$, es decir,

$$\min_{\mathbf{x}} z = 4x_{11} + 6x_{12} + 8x_{13} + 5x_{21} + 4x_{22} + 3x_{23}$$

Este problema implica dos tipos de restricciones: un conjunto de demandas y otro conjunto de ofertas. Las restricciones de demanda significan que todas las fábricas deben recibir al menos las unidades demandadas, esto es

$$\sum_{i=1}^{2} x_{ij} \geq b_j \qquad j = 1,2,3$$

De igual forma, las restricciones de oferta significan que cada almacén puede enviar como máximo su inventario, lo cual es

$$\sum_{j=1}^{3} x_{ij} \leq a_i \qquad i = 1,2$$

Ejemplo

Por lo tanto, el modelo queda

$$\min_{\mathbf{x}} z = 4x_{11} + 6x_{12} + 8x_{13} + 5x_{21} + 4x_{22} + 3x_{23}$$

$$\begin{aligned}
\text{s.a.} \quad & x_{11} + x_{12} + x_{13} \leq 20 \quad \text{(El almacén 1 puede suministrar como máximo 20 unidades)} \\
& x_{21} + x_{22} + x_{23} \leq 30 \quad \text{(El almacén 2 puede suministrar como máximo 30 unidades)} \\
& x_{11} + x_{21} \geq 10 \quad \text{(La fábrica 1 demanda al menos 10 unidades)} \\
& x_{12} + x_{22} \geq 25 \quad \text{(La fábrica 2 demanda al menos 25 unidades)} \\
& x_{13} + x_{23} \geq 15 \quad \text{(La fábrica 3 demanda al menos 15 unidades)} \\
& x_{ij} \geq 0, \quad \forall (i,j) \in I \times J.
\end{aligned}$$

Para obtener la solución podemos usar `intlinprog` (Matlab) o el solver de Windows.

$$x^* = (10, 10, 0, 0, 15, 15) \qquad z^* = 205$$

Por lo tanto, para minimizar el coste de transporte, debemos enviar 10 unidades desde el almacén 1 a la fábrica 1, 10 unidades desde el almacén 1 a la fábrica 2, 15 unidades desde el almacén 2 a la fábrica 2 y 15 unidades desde el almacén 2 a la fábrica 3, para un coste mínimo de 205 euros.

El *problema de asignación* es un modelo fundamental en investigación operativa que aborda la asignación óptima de recursos a tareas. Su objetivo es asignar un conjunto de agentes (por ejemplo, trabajadores, máquinas o proveedores de servicios) a un conjunto de tareas (por ejemplo, trabajos, ubicaciones o responsabilidades) de manera que se minimice el coste total, el tiempo o la distancia de las asignaciones. Este problema es un caso particular de programación lineal y optimización combinatoria, y tiene aplicaciones ampliamente extendidas en planificación, logística y sistemas de toma de decisiones.

Consideramos el siguiente escenario:

- Existen m agentes, cada uno capaz de realizar exactamente una tarea.
- Existen n tareas, cada una requiere exactamente un agente.
- El coste de asignar el agente i a la tarea j se denota por c_{ij}, donde $I = \{1, \ldots, m\}$ y $J = \{1, \ldots, n\}$.

Si $m = n$, entonces el problema está balanceado y cada tarea tendrá asignado su respectivo agente. Si $m > n$, existe un exceso de agentes y habrá agentes sin asignar; por otro lado, si $n > m$, existe un exceso de tareas y habrá tareas sin asignar. Esta situación se representa típicamente como un *grafo bipartito completo*, véase la Figura 14, donde un conjunto de nodos corresponde a los agentes y el otro a las tareas, y cada arista representa una posible asignación con un coste asociado.

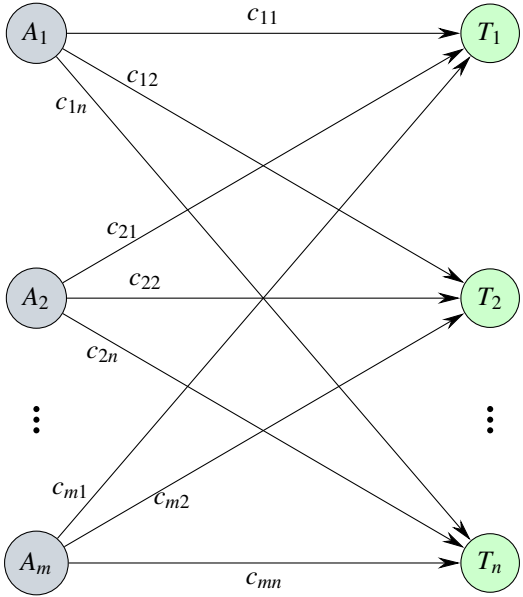


Figura 14: Representación empleando un grafo del problema de asignación

Definición (Problema de Asignación)

Definimos las variables de decisión $x_{ij} = \begin{cases} 1, & \text{si el agente } i \text{ se asigna a la tarea } j, \\ 0, & \text{en caso contrario.} \end{cases}$

donde $I = \{1, \ldots, m\}$ y $J = \{1, \ldots, n\}$.

El objetivo es minimizar el coste total de asignación $\boxed{\min_{\mathbf{x}} z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}}$

sujeto a las restricciones:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \text{para todo } i = 1, \ldots, m \quad \text{(cada agente se asigna a una tarea)}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \text{para todo } j = 1, \ldots, n \quad \text{(cada tarea se asigna a un agente)}$$

$$x_{ij} \in \{0, 1\}, \quad \text{para todo } i, j \in I \times J.$$

Observación

El problema de asignación es un caso particular del problema de transporte donde todas las ofertas y demandas son iguales a uno. Su naturaleza binaria y estructura balanceada permiten algoritmos eficientes y lo convierten en un pilar de la optimización combinatoria.

Ejemplo

Una empresa necesita asignar 4 empleados a 4 tareas diferentes. La matriz de costes (en horas requeridas para completar cada tarea) se muestra en la Tabla 5. Formule y resuelva el problema de asignación para minimizar el tiempo total empleado.

	Tarea 1	Tarea 2	Tarea 3	Tarea 4
Empleado 1	9	2	7	8
Empleado 2	6	4	3	7
Empleado 3	5	8	1	8
Empleado 4	7	6	9	4

Tabla 5: Matriz de costes para el problema de asignación (en horas).

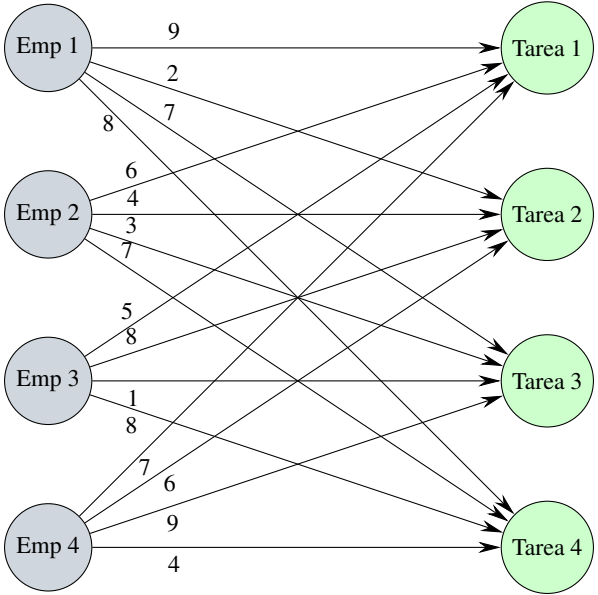


Figura 15: Representación empleando un grafo bipartito del problema de asignación

Ejemplo

Sea x_{ij} una variable binaria que indica si el empleado i se asigna a la tarea j. Definimos las variables de decisión:

$$x_{ij} = \begin{cases} 1, & \text{si el agente } i \text{ se asigna a la tarea } j, \\ 0, & \text{en caso contrario.} \end{cases}$$

donde $I = \{1, \ldots, m\}$ y $J = \{1, \ldots, n\}$. La función objetivo es:

$$\begin{aligned} \min_{\mathbf{x}} z = & 9x_{11} + 2x_{12} + 7x_{13} + 8x_{14} + 6x_{21} + 4x_{22} + 3x_{23} + 7x_{24} \\ & + 5x_{31} + 8x_{32} + 1x_{33} + 8x_{34} + 7x_{41} + 6x_{42} + 9x_{43} + 4x_{44} \end{aligned}$$

Sujeto a:

$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &= 1 \quad \text{(Empleado 1 asignado a una tarea)} \\ x_{21} + x_{22} + x_{23} + x_{24} &= 1 \quad \text{(Empleado 2 asignado a una tarea)} \\ x_{31} + x_{32} + x_{33} + x_{34} &= 1 \quad \text{(Empleado 3 asignado a una tarea)} \\ x_{41} + x_{42} + x_{43} + x_{44} &= 1 \quad \text{(Empleado 4 asignado a una tarea)} \\ x_{11} + x_{21} + x_{31} + x_{41} &= 1 \quad \text{(Tarea 1 asignada a un empleado)} \\ x_{12} + x_{22} + x_{32} + x_{42} &= 1 \quad \text{(Tarea 2 asignada a un empleado)} \\ x_{13} + x_{23} + x_{33} + x_{43} &= 1 \quad \text{(Tarea 3 asignada a un empleado)} \\ x_{14} + x_{24} + x_{34} + x_{44} &= 1 \quad \text{(Tarea 4 asignada a un empleado)} \\ x_{ij} &\in \{0,1\} \quad \forall i,j \in I \times J. \end{aligned}$$

```
n = 4; % Number of employees and tasks
% Cost matrix
C = [ 9  2  7  8; 6  4  3  7; 5  8  1  8; 7  6  9  4 ];
f = C(:); % Vectorize cost matrix (column-wise)
numVars = n^2; % Number of variables
intcon = 1:numVars; % Integer variables (all binary)
lb = zeros(numVars,1); ub = ones(numVars,1); % Bounds
%Equality constraints % Each employee is assigned exactly one task
Aeq1 = zeros(n, numVars);
for i = 1:n
    Aeq1(i, (i-1)*n+1:i*n) = 1;
end
% Each task is assigned exactly one employee
Aeq2 = zeros(n, numVars);
for j = 1:n
    Aeq2(j, j:n:end) = 1;
end
Aeq = [Aeq1; Aeq2];
beq = ones(2*n,1);
%Solve the problem
[x, fval] = intlinprog(f, intcon, [], [], Aeq, beq, lb, ub);
X = reshape(x, n, n); % Reshape solution
disp('*** Solution ***') % Output
for i = 1:n
    for j = 1:n
        fprintf('x_%d%d: %d\n', i, j, X(i,j));
    end
end
fprintf('Objective: %f\n', fval);
```

Código 5: Código para Ejemplo 45

Ejemplo

$$x^* = (0,1,0,0,1,0,0,0,0,0,1,0,0,0,0,1), \quad z^* = 13$$

Así, la asignación óptima es:

- Empleado 1 $\rightarrow$ Tarea 2
- Empleado 2 $\rightarrow$ Tarea 1
- Empleado 3 $\rightarrow$ Tarea 3
- Empleado 4 $\rightarrow$ Tarea 4

Para un mínimo de 13 horas de trabajo.

§IV. Modelos de optimización de redes

- Teoría de grafos.
- Problema del camino más corto.
- Árbol de expansión mínima.
- Problema del flujo máximo.
- Problema del flujo de costo mínimo.

La referencia principal para este capítulo es [Hillier and Lieberman, 2015, Cap. 9].

La Teoría de Grafos es una rama de la matemática discreta que estudia las relaciones entre objetos mediante estructuras denominadas *grafos*. Un grafo proporciona una forma natural de representar y analizar la conectividad, el flujo y las estructuras de red, lo que lo convierte en una herramienta fundamental en matemáticas, informática, ingeniería y numerosas ciencias aplicadas.

Definición (Grafo)

Un *grafo* $G = (V, E)$ consiste en un conjunto no vacío de vértices V y un conjunto de aristas E, donde cada arista es un par no ordenado de vértices distintos de V.

Ejemplo

Considere el grafo representado en la Figura 16. Los vértices son los puntos etiquetados con letras y las aristas son las líneas que los conectan. Así, tenemos el conjunto de vértices $V = \{A, B, C, D\}$ y el conjunto de aristas $E = \{AB, AC, BC, BD, CD\}$. Para representar los vértices se suelen utilizar números y/o letras, y para las aristas, el par de vértices conectados.

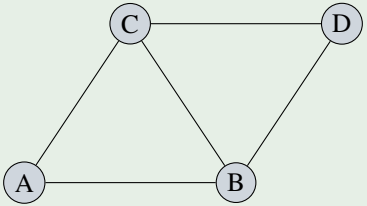


Figura 16: Un grafo con cuatro vértices y cinco aristas

Definición (Digrafo)

Un *grafo dirigido*, o *digrafo*, es un grafo en el que cada arista tiene una dirección y se representa como un par ordenado de vértices. Al escribir el conjunto de aristas de un digrafo, el **orden** es importante.

Ejemplo

Los digrafo son adecuados para representar relaciones asimétricas, como calles de sentido único, flujo de información o relaciones de precedencia. Considere la Figura 17. Aquí $V = \{A, B, C\}$ y $E = \{AB, BC, CA\}$; observe que ahora el orden de los vértices que etiquetan una arista indica la dirección de dicha arista.

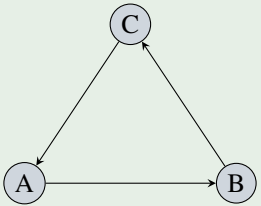


Figura 17: Digraph

Definición (Cadena / Camino)

Un *camino* es una secuencia de vértices tal que cada par consecutivo está conectado por una arista. Los caminos pueden ser abiertos (primer y último vértice diferentes) o cerrados (cuando el primer y el último vértice coinciden).

Observación

En un grafo conectado, los caminos deben respetar la dirección de las aristas.

Definición (Loop (Ciclo))

Un *ciclo* es un camino en el que el primer y el último vértice son iguales. Si un grafo no contiene ciclos se denomina grafo *acíclico*.

Ejemplo

Considere el grafo en la Figura 16

- Existen varios caminos que conectan A con D, por ejemplo, $P_1 = \{AC, CD\}$, $P_2 = \{AC, CB, BD\}$.
- El camino $\{AC, CB, AB\}$ es un ciclo, porque comienza y termina en el mismo vértice A.

Los caminos son fundamentales para analizar la alcanzabilidad y la conectividad dentro de un grafo.

Definición (Grafo conexo)

Un grafo se dice *conexo* si siempre existe un camino entre cada par de vértices.

Ejemplo

Los grafos en las Figuras 16, 17 y 19 son conexos; sin embargo, el grafo en la Figura 18 no es conexo. Observe que no existe ningún camino que comience en A y termine en E.

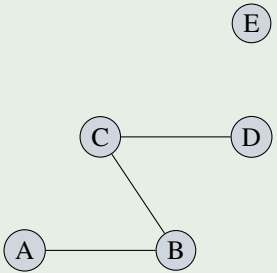


Figura 18: Grafo no conexo

Definición (Árbol)

Un *árbol* es un grafo conexo y acíclico.

Definición (Subgrafo)

Sean dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$. Decimos que G_1 es un subgrafo de G_2, y lo denotamos como $G_1 \subset G_2$, si $V_1 \subset V_2$ y $E_1 \subset E_2$.

Definición (Árbol Soporte)

Dado un grafo conexo G, un *árbol soporte* es un subgrafo de G que incluye todos los vértices de G y que es, a su vez, un árbol.

Ejemplo

- El grafo en la Figura 19 es un subgrafo de la Figura 16.
- El grafo en la Figura 19 es un grafo conexo y acíclico, por lo tanto, un árbol.
- Además, la Figura 19 es el árbol soporte de la Figura 16.

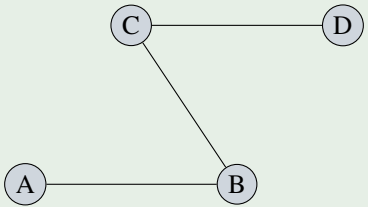


Figura 19: Spanning Tree (Árbol Soporte)

Los árboles son estructuras mínimas que conectan todos sus vértices sin ciclos y desempeñan un papel esencial en modelos jerárquicos y estructuras de datos. Los árboles soporte son útiles en problemas de optimización, como la minimización del coste de conexión de una red.

El problema del camino más corto

El *problema del camino más corto* es un concepto fundamental en teoría de grafos y optimización, ampliamente utilizado en logística, diseño de redes y aplicaciones biomédicas como el análisis de imágenes médicas, el mapeo de conectividad neuronal y los sistemas de enrutamiento hospitalario. El objetivo es determinar el camino entre dos nodos en un grafo de manera que el coste total (o distancia) sea mínimo.
Consideramos un grafo dirigido $G = (V, E)$ donde:

- V es el conjunto de vértices (nodos), que representan ubicaciones, estados o entidades.
- E es el conjunto de aristas (arcos), cada una con un coste no negativo asociado c_{ij} y una dirección para la arista $(i,j) \in E$.

Sea $s \in V$ el nodo origen y $t \in V$ el nodo destino. El objetivo es encontrar un camino desde s hasta t tal que la suma de los costes a lo largo del camino sea mínima.

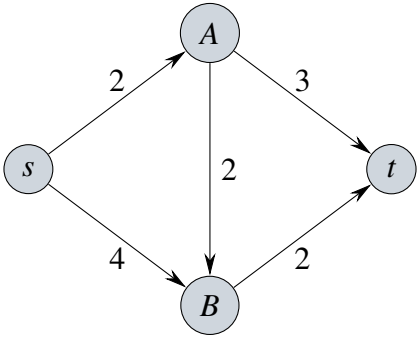


Figura 20: Ejemplo de grafo del problema del camino más corto

Definición (El problema del camino más corto)

El objetivo de este problema es encontrar el camino más corto entre dos nodos predefinidos en un grafo. Dado que un camino es una secuencia de aristas, debemos decidir qué arista se incluye en el camino. Para ello, definimos las variables de decisión binarias:

$$x_{ij} = \begin{cases} 1 & \text{si la arista } (i,j) \text{ se incluye en el camino,} \\ 0 & \text{si no.} \end{cases}$$

para $(i,j) \in E$. Así, dado que el coste de usar la arista (i,j) es $c_{ij}x_{ij}$, la función objetivo se escribe como

$$\boxed{\min_{\mathbf{x}} z = \sum_{(i,j)\in E} c_{ij}x_{ij}}$$

Finalmente, si el viajero toca un vértice, eso significa que utiliza una arista que entra en el vértice (una arista que termina en el vértice), y dado que debe continuar viajando, debe salir del vértice utilizando una arista que sale del vértice (una arista que comienza en el vértice). Esta restricción se denomina, por razones que estudiaremos más adelante, restricción de conservación de flujo.

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} 1 & \text{si } i = s, \\ -1 & \text{si } i = t, \\ 0 & \text{en otro caso.} \end{cases}$$

El protocolo establece que, si una arista entra en un vértice, se utiliza signo negativo, y si sale, se utiliza signo positivo. Existen dos excepciones a este comportamiento: s (origen) y t (destino). Desde el origen, el viajero solo puede salir, y de esta forma debemos usar únicamente una arista; lo mismo ocurre con el vértice destino, el viajero solo puede entrar, utilizando únicamente una arista, por lo que consideramos únicamente las aristas que entran.

Ejemplo

Considere el grafo en la Figura 20. Vamos a encontrar el camino más corto desde s hasta t. Primero, escriba el conjunto de aristas y vértices: $V = \{s, A, B, t\}$ y $E = \{sA, sB, AB, At, Bt\}$. Declare las variables binarias

$$x_{ij} = \begin{cases} 1 & \text{si la arista } (i,j) \text{ está incluida en el camino,} \\ 0 & \text{en caso contrario.} \end{cases}$$

para $(i,j) \in E$. La función objetivo resulta $\boxed{\min_{\mathbf{x}} z = \sum_{(i,j)\in E} c_{ij}x_{ij} = 2x_{sA} + 4x_{sB} + 2x_{AB} + 3x_{At} + 2x_{Bt}}$

Las restricciones de conservación de flujo son

$$\begin{aligned} x_{sA} + x_{sB} &= 1 && \text{el origen envía al viajero} \\ -x_{sA} + x_{AB} + x_{At} &= 0 && \text{vertex } A \text{ está en equilibrio} \\ -x_{sB} - x_{AB} + x_{Bt} &= 0 && \text{vertex } B \text{ está en equilibrio} \\ -x_{At} + x_{Bt} &= -1 && \text{el destino recibe al viajero} \end{aligned}$$

El modelo resulta

$$
\begin{aligned}
\min \quad & z \quad = 2x_{sA} + 4x_{sB} + 2x_{AB} + 3x_{At} + 2x_{Bt} \\
\text{s.t.} \quad & x_{sA} + x_{sB} = 1 \\
& -x_{sA} + x_{AB} + x_{At} = 0 \\
& -x_{sB} - x_{AB} + x_{Bt} = 0 \\
& -x_{At} + x_{Bt} = -1 \\
& x_{ij} \in \{0,1\}, \quad \forall (i,j) \in E
\end{aligned}
$$

Para resolver el problema se emplea el Código 6.

```
numVars = 5; % Number of variables
% Cost vector (objective function)
% [x_sA, x_sB, x_AB, x_At, x_Bt]
f = [2; 4; 2; 3; 2];
intcon = 1:numVars; % Integer (binary) variables
lb = zeros(numVars,1); % Bounds
ub = ones(numVars,1);
% Constraints ; % RHS resources
Aeq1 = [1  1  0  0  0]; beq1 = 1;
Aeq2 = [-1  0  1  1  0]; beq2 = 0;
Aeq3 = [0 -1 -1  0  1]; beq3 = 0;
Aeq4 = [0  0  0 -1  1]; beq4 = -1;
% Assemble equality constraints
Aeq = [Aeq1;Aeq2;Aeq3;Aeq4];
beq = [beq1;beq2;beq3;beq4];
[x, fval] = intlinprog(f, intcon, [], [], Aeq, beq, lb, ub); %solve
disp('*** Solution ***')
fprintf('x_sA = %d\n', x(1));
fprintf('x_sB = %d\n', x(2));
fprintf('x_AB = %d\n', x(3));
fprintf('x_At = %d\n', x(4));
fprintf('x_Bt = %d\n', x(5));
fprintf('Objective value: %f\n', fval);
```

Código 6: Camino más corto

El Problema del Árbol de Expansión Mínima es un concepto fundamental en teoría de grafos y optimización combinatoria. Consiste en encontrar un subconjunto de aristas en un grafo conexo y ponderado que conecte todos los vértices con el peso total mínimo posible y sin formar ciclos. En ingeniería en telecomunicaciones, los árboles de expansión mínima pueden utilizarse para diseñar redes de transmisión eficientes, optimizar la topología de redes de sensores en sistemas IoT o planificar infraestructuras de comunicación con coste mínimo.

Considere el grafo G en la figura 21

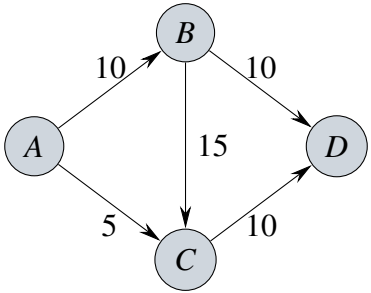


Figura 21: A graph with loops

Árbol de Expansión Mínima

El problema aquí consiste en extraer un subgrafo conexo de G que no contenga ciclos (un árbol) con los menores pesos en las aristas. Por lo tanto, se requiere que G sea un grafo conexo, no dirigido, con pesos positivos. El objetivo es obtener un conjunto de aristas que conecte **todos** los nodos cuyo peso total sea mínimo. Nótese que si una red contiene n nodos, entonces se requieren exactamente $n-1$ aristas para conectarlos a todos. Dichos arcos deben seleccionarse de manera que el peso total se minimice. Este problema puede aplicarse en el despliegue de redes, transporte y distribución.
Para resolver este problema aplicamos el algoritmo de Prim.

Algorithm 3 Algoritmo de Prim

1: Sea $T = (V_T, E_T)$, donde $V_T = E_T = \emptyset$. Elija un nodo arbitrario $v_0 \in V = \{v_0, \ldots, v_{n-1}\}$ y añádalo al subgrafo T, así, $V_T = \{v_0\}$
2: **while** existan nodos que aún no estén conectados a T, es decir, $V_T \neq V$ **do**
3: Encuentre la arista e_k con peso mínimo que conecte T con un nodo que no esté en T
4: Añada dicho nodo y arista a T, $V_T = V_T \cup \{v_k\}$ y $E_T = E_T \cup \{e_k\}$
5: **end while**

Después de $n-1$ pasos, el grafo resultante será un árbol de expansión mínima. En caso de que se consideren dos aristas con igual peso en el mismo paso, elija una de manera arbitraria.

Ejemplo

Encuentre el árbol de expansión mínima en el grafo de la Figura
Aplique el algoritmo de Prim en los siguientes pasos

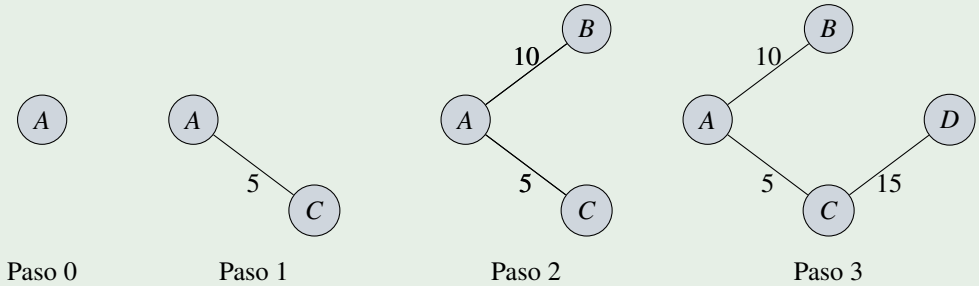


Figura 22: Pasos del algoritmo de Prim

Por lo tanto, el árbol de expansión mínima está dado por $T = (V, E_T)$ donde $V = \{A, B, C, D\}$ y $E_T = \{AB, AC, CD\}$. Con un peso total de 30.

Problema del flujo máximo

El Problema de Flujo Máximo es un problema clásico de optimización en la teoría de redes. Busca determinar la cantidad máxima de flujo que puede enviarse desde un nodo origen hasta un nodo destino a través de una red de aristas con restricciones de capacidad. En ingeniería en telecomunicaciones, este concepto es particularmente útil para modelar el tráfico de datos en redes, optimizar sistemas de transmisión y analizar el transporte de información en infraestructuras complejas. Consideramos un grafo dirigido $G = (V, E)$, e imponemos la restricción de que el flujo debe enviarse desde el origen hasta el destino sin pérdidas, es decir, ningún nodo intermedio puede retener flujo.

- V: Conjunto de nodos por los que debe atravesar el flujo.
- E: Conjunto de aristas dirigidas que representan las rutas por las que puede viajar el flujo.

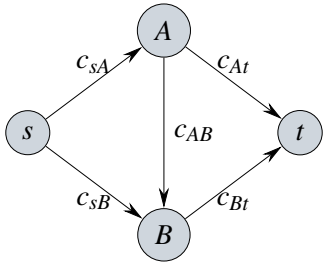


Figura 23: Ejemplo de grafo para el problema de flujo máximo

Definición (Problema del flujo máximo)

Sea $G = (V, E)$ un grafo dirigido con capacidad c_{ij} en cada arista (i,j). Definimos las variables de flujo x_{ij} tales que:

$$x_{ij} = \text{unidades de flujo a través de la arista } (i,j) \quad \text{para} \quad (i,j) \in E$$

El flujo que atraviesa el grafo puede detectarse de dos maneras equivalentes: como el flujo que sale de s o como el flujo que entra en t, por lo tanto, este problema tiene dos funciones objetivo equivalentes

$$\max_{\mathbf{x}} z = \sum_{j \in V} x_{sj} = \sum_{i \in V} x_{it}$$

Por otro lado, tenga en cuenta que ningún nodo intermedio puede retener flujo, por lo que existe una restricción de conservación de flujo. En otras palabras, el flujo que entra y el que sale debe ser el mismo para todos los nodos intermedios. Recordando el protocolo, el flujo que entra se etiqueta como negativo y el que sale como positivo, por lo tanto, las restricciones de conservación de flujo son

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} z & \text{si } i = s, \\ -z & \text{si } i = t, \\ 0 & \text{en otro caso.} \end{cases}$$

Nótese que los vértices especiales s y t no se incluyen como restricciones, porque forman parte de la función objetivo. Finalmente, el flujo debe satisfacer ciertas restricciones de caja, es decir, el flujo a través de cada arista debe ser positivo y no puede exceder la capacidad de cada arista. Esto es, $0 \le x_{ij} \le c_{ij}, \quad \text{para} \quad (i,j) \in E$. El modelo resulta

$$\begin{aligned} \max \quad & z = \sum_{j \in V} x_{sj} = \sum_{i \in V} x_{it} \\ \text{s.t.} \quad & x_{ij} \le c_{ij}, \quad \forall (i,j) \in E \\ & \sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji}, \quad \forall i \in V \setminus \{s,t\} \\ & x_{ij} \ge 0, \quad \forall (i,j) \in E \end{aligned}$$

Ejemplo

Considere el grafo en la Figura 24, donde las etiquetas de las aristas representan la capacidad máxima de cada arista. Determinemos el flujo máximo que puede enviarse desde s hasta t

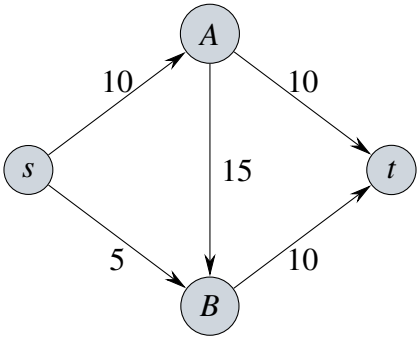


Figura 24: Grafo para ejemplo 65

Ejemplo

Primeramente, denotemos las aristas y vértices del grafo como $V = \{s, A, B, t\}$ $E = \{sA, sB, AB, At, Bt\}$.

Declare las variables de decisión

$$x_{ij} = \text{unidades de flujo enviadas a través de la arista}$$

para $(i,j) \in E$. La función objetivo resulta

$$\max_{\mathbf{x}} z = x_{sA} + x_{sB} = x_{At} + x_{Bt}$$

Ahora, las restricciones de conservación de flujo son

$$-x_{sA} + x_{AB} + x_{At} = 0 \quad \text{el vértice } A \text{ conserva el flujo}$$
$$-x_{sB} - x_{AB} + x_{Bt} = 0 \quad \text{el vértice } B \text{ conserva el flujo}$$

Las variables están sujetas a las restricciones de caja

$$0 \leq x_{ij} \quad \forall (i,j) \in E$$
$$x_{sA} \leq 10$$
$$x_{sB} \leq 5$$
$$x_{AB} \leq 15$$
$$x_{At} \leq 10$$
$$x_{Bt} \leq 10$$

El modelo resulta

$$
\begin{aligned}
\max \quad & z \quad = x_{sA} + x_{sB} = x_{At} + x_{Bt} \\
\text{s.t.} \quad & -x_{sA} + x_{AB} + x_{At} = 0 \\
& -x_{sB} - x_{AB} + x_{Bt} = 0 \\
& x_{sA} \leq 10 \\
& x_{sB} \leq 5 \\
& x_{AB} \leq 15 \\
& x_{At} \leq 10 \\
& x_{Bt} \leq 10 \\
& x_{ij} \geq 0, \quad \forall (i,j) \in E
\end{aligned}
$$

Para resolver el problema se emplea el Código 7.

```
numVars = 5; % Number of variables
f = -[1; 1; 0; 0; 0]; % Objective function
% Equality constraints (flow conservation)
Aeq1 = [-1  0  1  1  0]; beq1 = 0;
Aeq2 = [ 0 -1 -1  0  1]; beq2 = 0;
Aeq = [Aeq1;Aeq2];
beq = [beq1;beq2];
lb = zeros(numVars,1); % Bounds
ub = [10; 5; 15; 10; 10];
[x, fval] = linprog(f, [],[], Aeq, beq, lb, ub); % Solve the problem
% Recover max objective value
z = -fval;
% Output
disp('*** Solution ***')
fprintf('x_sA = %.2f\n', x(1));
fprintf('x_sB = %.2f\n', x(2));
fprintf('x_AB = %.2f\n', x(3));
fprintf('x_At = %.2f\n', x(4));
fprintf('x_Bt = %.2f\n', x(5));
fprintf('Maximum flow value: %.2f\n', z);
```

Código 7: Flujo Máximo

Por lo tanto, el grafo admite un flujo máximo de 15 unidades, utilizando la siguiente distribución:

Ejemplo

- Flujo en la arista sA: 10.0
- Flujo en la arista sB: 5.0
- Flujo en la arista AB: 0.0
- Flujo en la arista At: 10.0
- Flujo en la arista Bt: 5.0

Observación

En contextos de ingeniería telemática o de telecomunicaciones, resolver el problema de flujo máximo puede ayudar a optimizar la transmisión de datos en redes, garantizar una distribución eficiente del tráfico en infraestructuras críticas y modelar la dinámica del flujo de información en sistemas distribuidos. Comprender las limitaciones del flujo y los principios de conservación es esencial para diseñar redes seguras y eficientes.

Problema de flujo de costo mínimo

El problema de Flujo de Costo Mínimo es un modelo de optimización fundamental en la teoría de redes. Su objetivo es determinar la forma más eficiente en términos de coste para enviar una cierta cantidad de flujo a través de una red desde nodos de suministro hacia nodos de demanda, sujeto a restricciones de capacidad. Este problema tiene amplias aplicaciones en telecomunicaciones, como la optimización del enrutamiento de tráfico en redes IP, la asignación eficiente de recursos en redes ópticas o la minimización del coste en redes de transporte de datos. El modelo que se aplicará para la resolución de este tipo de problema incluye todos los modelos anteriores, incluso transporte y asignación.

Consideremos un grafo dirigido $G = (V, E)$, donde:

- V es el conjunto de nodos (que representan routers, estaciones base o puntos de conmutación).
- E es el conjunto de arcos (que representan enlaces de comunicación).
- Cada arco $(i,j) \in A$ tiene:
  - Una capacidad $u_{ij} \geq 0$.
  - Un coste por unidad de flujo c_{ij}.
- Cada nodo $i \in N$ tiene un valor de oferta/demanda b_i, donde:
  - $b_i > 0$: nodo de suministro (origen de tráfico).
  - $b_i < 0$: nodo de demanda (destino de tráfico).
  - $b_i = 0$: nodo de tránsito.

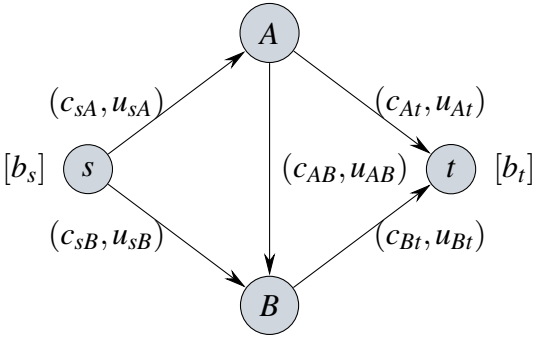


Figura 25: Grafo del problema de flujo de coste mínimo

Definición (Problema de flujo de costo mínimo)

El problema de flujo de coste mínimo puede formularse como:

$$\begin{aligned} \min \quad & \sum_{(i,j)\in A} c_{ij}x_{ij} \\ \text{s.a.} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i \quad \forall i \in N, \\ & 0 \le x_{ij} \le u_{ij} \quad \forall (i,j) \in E. \end{aligned}$$

Donde:

- x_{ij} es el flujo en el arco (i,j).
- La primera restricción asegura la conservación del flujo.
- La segunda restricción impone los límites de capacidad.

Teorema (Existencia de soluciones factibles)

Una condición necesaria para que un problema de costo mínimo tenga soluciones factibles es que la oferta y la demanda están equilibradas, esto es

$$\sum_{i=1}^{n} b_i = 0$$

Observación

En caso de $\sum_{i=1}^{n} b_i \neq 0$, se tendrán que ajustar las restricciones de flujo de manera acorde, en la dirección del desbalance, de manera que se pueda encontrar una solución. Generalmente, con ajustar las desigualdades en la dirección en la que hay exceso (oferta o demanda), será suficiente.

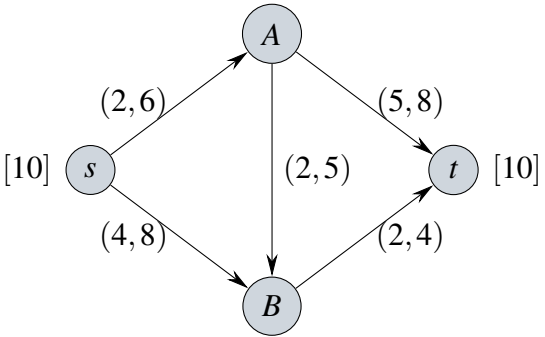


Figura 26: Grafo del problema de flujo de coste mínimo

Ejemplo (Flujo de coste mínimo)

Considere el grafo en la Figura 26. Vamos a encontrar el flujo de coste mínimo que se puede enviar desde el nodo s hasta t. Primero, escriba el conjunto de aristas y vértices.

$$V = \{s, A, B, t\}$$
$$E = \{sA, sB, AB, At, Bt\}$$

Declare las variables de decisión

$$x_{ij} : \text{unidades de flujo enviadas desde nodo } i \text{ and nodo } j \; \forall (i,j) \in E.$$

La función objetivo resulta

$$\min_{\mathbf{x}} z = \sum_{(i,j)\in E} c_{ij} x_{ij} = 2x_{sA} + 4x_{sB} + 2x_{AB} + 5x_{At} + 2x_{Bt}$$

Las restricciones de conservación de flujo son

Ejemplo

El modelo resulta

$$
\begin{aligned}
\min \quad z \quad & = 2x_{sA} + 4x_{sB} + 2x_{AB} + 5x_{At} + 2x_{Bt} \\
\text{s.t.} \quad & x_{sA} + x_{sB} = 10 \\
& -x_{sA} + x_{AB} + x_{At} = 0 \\
& -x_{sB} - x_{AB} + x_{Bt} = 0 \\
& -x_{At} - x_{Bt} = -10 \\
& x_{sA} \leq 6 \\
& x_{sB} \leq 8 \\
& x_{AB} \leq 5 \\
& x_{At} \leq 8 \\
& x_{Bt} \leq 8 \\
& x_{ij} \geq 0, \quad \forall (i,j) \in E
\end{aligned}
$$

Para resolver el problema se emplea el Código 8.

```
numVars = 5; % [x_sA, x_sB, x_AB, x_At, x_Bt]
f = [2; 4; 2; 10; 2]; % Objective function
% Equality constraints
Aeq1 = [1   1   0   0   0];   % x_sA + x_sB = 10
Aeq2 = [-1   0   1   1   0];   % -x_sA + x_AB + x_At = 0
Aeq3 = [0  -1  -1   0   1];   % -x_sB - x_AB + x_Bt = 0
Aeq4 = [0   0   0  -1   -1];    % -x_At + x_Bt = -10
Aeq = [Aeq1;Aeq2;Aeq3;Aeq4];
beq = [10; 0; 0; -10];
% Bounds
lb = zeros(numVars,1);          % x_ij >= 0
ub = [6; 8; 5; 8; 8];           % capacities
% Solve the problem
[x, fval, exitflag, output] = linprog(f, [], [], Aeq, beq, lb, ub);
% Output
disp('*** Solution ***')
fprintf('x_sA = %.4f\n', x(1));
fprintf('x_sB = %.4f\n', x(2));
fprintf('x_AB = %.4f\n', x(3));
fprintf('x_At = %.4f\n', x(4));
fprintf('x_Bt = %.4f\n', x(5));
fprintf('Minimum cost value: %.4f\n', fval);
```

Código 8: Flujo de coste mínimo

Ejemplo

Con solución $\mathbf{x}^* = [6,4,4,2,8]$ para $z^* = 72$. Por lo tanto, el flujo óptimo resulta, en unidades de flujo,

$$x_{sA} = 6$$
$$x_{sB} = 4$$
$$x_{AB} = 4$$
$$x_{At} = 2$$
$$x_{Bt} = 8$$

para un coste mínimo de 72 unidades.

Observación

En ingeniería de telecomunicaciones, el modelo del flujo de coste mínimo permite optimizar el enrutamiento de tráfico en redes de datos, minimizar costes en redes de transporte óptico y garantizar una asignación eficiente de recursos en redes multiservicio. Al reducir costes y mejorar la calidad del servicio, este enfoque impacta directamente en la eficiencia de la red y en la experiencia del usuario. Recordemos que este modelo engloba todas las aplicaciones vistas anteriormente.

§V. Administración de proyectos con PERT/CPM

- PERT: Program Evaluation and Review Technique .
- CPM: Critical Path Method.
- Manejo de incertidumbre.
- Trueque entre tiempo y coste. Crashing.

Para información más detallada, consultar [Hillier and Lieberman, 2015, Sec. 9.8].

Uno de los retos más importantes a la hora de dirigir proyectos de gran escala es la planificación. Es necesario conocer todas las dependencias, coordinar las partes y tener en cuenta posibles desviaciones. Para realizar este cometido, se utilizan las técnicas **PERT** (Program Evaluation and Review Technique, Técnicas de evaluación y revisión del programa) y **CPM** (Critical Path Method, método del camino crítico), que ayudan al director de proyectos a lidiar con todos los problemas de organización. Ambas técnicas están orientadas a la representación gráfica y análisis temporal de proyectos. Su creación se remonta a la década de los 50.

Definición

- **Actividad**: Tarea indivisible que consume tiempo y recursos.
- **Evento**: Punto en el tiempo que marca el inicio o fin de una o varias actividades.
- **Precedencia**: Una tarea i precede a una tarea j si i no puede ser completada hasta que se completa j.
- **Red de proyecto**: Representación gráfica mediante nodos (actividades) y arcos (relación de precedencia entre actividades) que describe la secuencia lógica del proyecto.
- **Ruta crítica**: Secuencia de actividades que determina la duración mínima del proyecto; cualquier retraso en esta ruta implica retraso en el proyecto completo.

Ejemplo (Reliable Construction Co)

La empresa Reliable construction ha ganado un proyecto de 5.4 millones de dólares para construir una nueva fábrica. El contratante necesita que la fábrica esté terminada en el plazo de un año (52 semanas). Por ello, el contrato incluye las siguientes cláusulas:

- Penalización de 300 mil dólares si se acaba en un plazo superior a 47 semanas.
- Gratificación de 150 mil dólares si se acaba en un plazo inferior a 40 semanas.

La Tabla 6 muestra las actividades y demás información necesaria para la realización del proyecto:

Tabla 6: Tabla de precedencias ejemplo prototipo

Actividad	Descripción	Precedencia	Duración estimada
A	Excavar	-	2 semanas
B	Cimentar	A	4 semanas
C	Levantar muros	B	10 semanas
D	Construir tejado	C	6 semanas
E	Instalar fontanería exterior	C	4 semanas
F	Instalar fontanería interior	E	5 semanas
G	Instalar revestimiento exterior	D	7 semanas
H	Pintar exterior	E,G	9 semanas
I	Instalación eléctrica	C	7 semanas
J	Panelar	F,I	8 semanas
K	Instalar el piso	J	4 semanas
L	Pintar interior	J	5 semanas
M	Instalar accesorios exteriores	H	2 semanas
N	Instalar accesorios interiores	K,L	6 semanas

Enfoque AON

Una forma muy eficiente de poder visualizar el proyecto, sus distintas tareas e información importantes es mediante el uso de grafos. Estos incluirán las actividades, relaciones de precedencia e información temporal. Se aplicará el enfoque AON (activity on node):

- Cada actividad se representa en un nodo
- Los arcos representan relaciones de precedencia.
- Los número en cada nodo representan la duración de la actividad.

En la Figura 27 se representa el resultado de aplicar el enfoque AON.
Para realizar esta organización necesitamos dos tipos de información:

- El orden de las actividades que deben realizarse.
- La duración estimada de cada actividad.

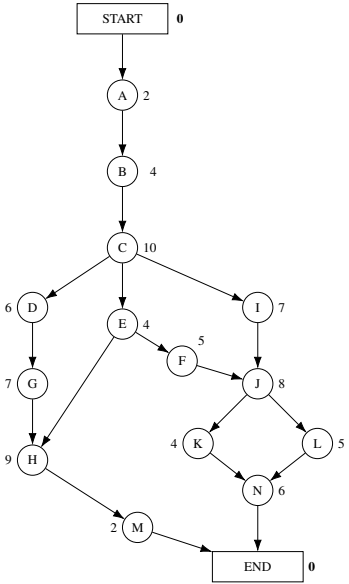


Figura 27: Representación AON correspondiente a Ejemplo 74

¿Cuánto tiempo va a tardar en realizarse el proyecto? Si sumamos la duración de todas las actividades, obtenemos un resultado de 79 semanas. Para calcular la duración del proyecto es necesario calcular la duración de todos los caminos que van desde origen hasta el fin. La duración estimada del proyecto será aquel camino que tenga una duración mayor. Este camino recibe el nombre de **camino crítico**. En la Tabla 7, se calcula la longitud de cada una de las rutas del nodo START al nodo END.

Tabla 7: Trayectorias y sus longitudes a través de la red de proyecto del Ejemplo 74, ver Figura 27

Trayectoria	Longitud (semanas)
START $\to A \to B \to C \to D \to G \to H \to M \to$ FIN	$2+4+10+6+7+9+2=40$
START $\to A \to B \to C \to E \to H \to M \to$ END	$2+4+10+4+9+2=31$
START $\to A \to B \to C \to E \to F \to / \to K \to N \to$ FIN	$2+4+10+4+5+8+4+6=43$
START $\to A \to B \to C \to E \to F \to J \to L \to N \to$ FIN	$2+4+10+4+5+8+5+6=44$
START $\to A \to B \to C \to l \to / \to K \to N \to$ FIN	$2+4+10+7+8+4+6=41$
START $\to A \to B \to C \to I \to J \to L \to N \to$ FIN	$2+4+10+7+8+5+6=42$

Luego, se concluye que en este proyecto

- Ruta crítica: START $\to A \to B \to C \to E \to F \to J \to L \to N \to$ FIN
- Duración estimada del proyecto 44 semanas.

Definición (Tiempo de inicio-finalización mínimo (ES, EF))

Al tiempo de inicio de una actividad , si no ocurre ningún retraso, se le conoce como **tiempo de inicio mínimo (ES - early start)**, de la misma forma, se define el tiempo de **finalización mínimo (EF - early finish)** tal que,

$$\text{EF} = \text{ES} + \text{duración}$$

Para mejorar la escalabilidad del problema, el *ES* de la primer actividad se fija en 0. Para ir rellenando el grafo, se utiliza la siguiente regla, el tiempo de inicio mínimo de una actividad es el tiempo de finalización mínimo mayor de entre sus predecesores.

En la Figura 28 se calculan los tiempos mínimos de inicio-finalización, de cada actividad del Ejemplo 74.

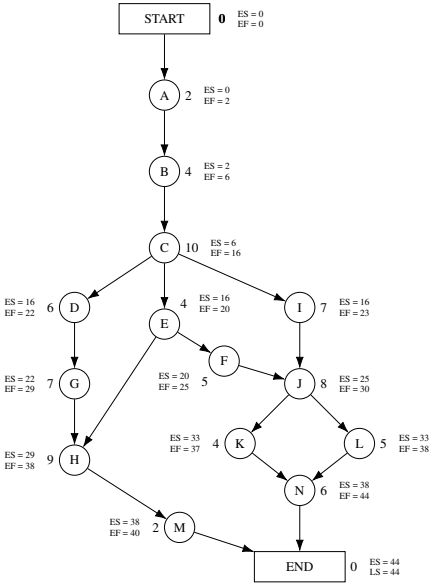


Figura 28: Tiempos mínimos de inicio-finalización de cada actividad del Ejemplo 74

Del mismo modo se pueden definir los tiempos de inicio-finalización máximos.

Definición (Tiempo de inicio-finalización máximos (LS, LF))

Los **tiempos máximos de finalización e inicio** (LF,LS), (last finish, last start) se definen como aquellos que, aun siendo mayores que los mínimos, no modifican el camino mínimo. Estos tiempo se puede definir como

$$LS = LF - \text{duración}$$

Como regla general, el tiempo máximo de finalización de una actividad es el menor tiempo de inicio máximo las actividades sucesoras.

En la Figura 29 se calculan los tiempos máximos de inicio-finalización, de cada actividad del Ejemplo 74.

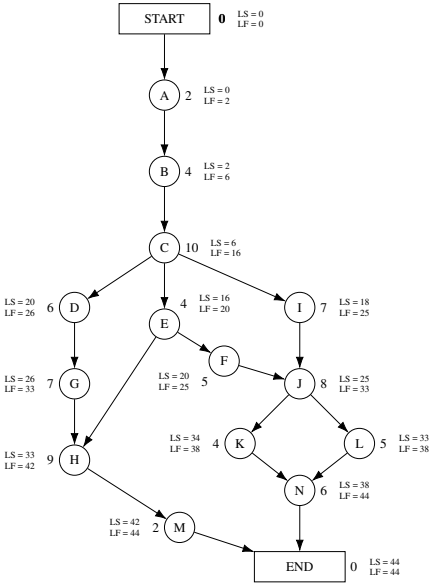


Figura 29: Tiempos máximos de inicio-finalización de cada actividad del Ejemplo 74

Observación

Por lo general para calcular los pares (ES, EF) se va de la tarea START a la tarea FIN, por otro lado, para calcular los pares (LS, LF), se va de la tarea FIN a la tarea START, una vez se conoce la duración mínima del proyecto.

Definición (Holgura)

Las holguras nos permiten saber qué actividades pueden retrasarse sin que eso revierta en un retraso del proyecto. Para calcularlas se utiliza la siguiente fórmula:

$$h = LF - EF$$

El camino crítico tiene holguras 0.

En la Figura 30, se ilustra el cálculo de holguras para cada actividad del Ejemplo 74.

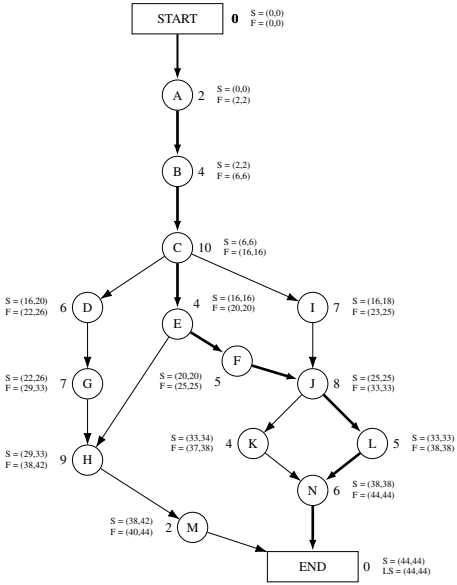


Figura 30: Ruta crítica Ejemplo 74

Luego se concluye que el camino crítico es

$$\text{START} \to A \to B \to C \to E \to F \to J \to L \to N \to \text{FINISH}$$

Tabla 8: Relación entre holguras y camino crítico

Actividad	Holgura (LF - EF)	Camino crítico
A	0	Sí
B	0	Sí
C	0	Sí
D	4	No
E	0	Sí
F	0	Sí
G	4	No
H	4	No
I	2	No
J	0	Sí
K	1	No
L	0	Sí
M	4	No
N	0	Sí

PERT introduce tres estimaciones para cada actividad, **optimista (o)**, **probable (m)** y **pesimista (p)**. Denotemos como

- μ_i, duración media de la actividad i. Reservamos μ_p para la duración del proyecto
- σ_i, la desviación típica de la duración de la actividad i. Reservamos σ_p, para la desviación típica de la duración del proyecto.

Definición (Camino crítico medio (CCM))

Definiremos el camino crítico medio (CCM) como el valor del camino crítico si cada actividad tiene una duración igual a su media.

Para calcular los tiempos de duración de un proyecto se siguen los siguientes supuestos

- **Supuesto 1**: El camino más largo es el CCM.
- **Supuesto 2**: Las actividades del camino crítico son estadísticamente independientes, por lo que

$$\mu_p = \sum_i^N \mu_i; \quad \sigma_p^2 = \sum_i^N \sigma_i^2; \quad \sigma_i^2 = \left(\frac{p_i - o_i}{6}\right)^2; \quad \mu_i = \frac{o_i + 4m_i + p_i}{6}$$

  Luego se asume que
  - μ_p: suma de las medias de las duraciones de las actividades del CCM.
  - σ_p^2: suma de las varianzas de las duraciones de las actividades en el CCM.

Ejemplo

Consultando a expertos se han podido estimar la media y varianza de cada una de las actividades del proyecto del Ejemplo 74. Ver Tabla 9.

Tabla 9: Aplicación de aproximación 2 a Ejemplo 74

Actividades en CCM	Media	Varianza
A	2	$\frac{1}{9}$
B	4	1
C	10	4
E	4	$\frac{4}{9}$
F	5	1
J	8	1
L	5	1
N	6	$\frac{4}{9}$
Duración del proyecto	$\mu_p = 44$	$\sigma_p^2 = 9$

- **Supuesto 3**: Se asume que la distribución es normal. De esta forma el cálculo de probabilidades vendría por la función de probabilidad acumulada de la distribución normal.

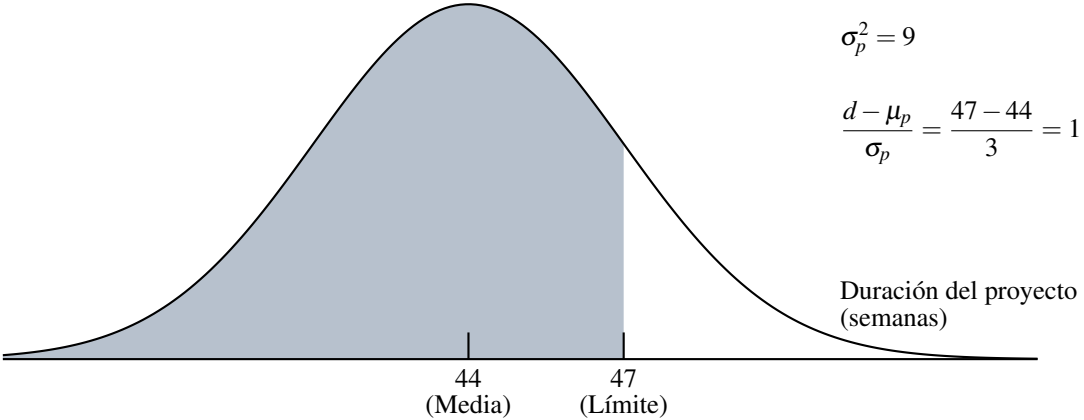


Figura 31: Estimación de la duración del proyecto a partir de los datos de la Tabla 9, empleando una aproximación normal.

Ejemplo

¿Cuál es la probabilidad de que el proyecto se complete en 47 semanas o menos?
Aplicamos la función de matlab `normcdf(x,`μ`,`σ`)`, que nos devuelve los valores de la función de distribución acumulada, evaluados en los valores en x para la distribución normal con media μ y desviación estándar σ, de donde se obtiene

$$P(X \leq 47) = 0.8413$$

Consideración del trueque entre tiempo y costo. Crashing.

CPM nos permite tener consideraciones en cuanto la reducción del tiempo de proyecto mediante la inyección de dinero a las actividades, en lo que se conoce como **crashing**. Este aporte extra de capital nos permite aplicar lo aprendido en el tema de programación lineal para encontrar soluciones óptimas, es decir, conseguir una reducción de la duración de la obra con la inversión mínima.
En primer lugar necesitaremos tener tabulados las posibles inversiones y su contrapartida en la reducción de la duración de las actividades. En caso de ser valores fijos, deberemos aplicar una interpolación lineal. Una vez realizado esto, debemos formular el modelo de programación matemática que quedará de la siguiente manera,

- Función objetivo: minimización de costes.
- Restricciones:
  - Reducción máxima.
  - No negatividad.
  - Tiempos de inicio, precedencia.
  - Duración del proyecto.

Ejemplo

Consideremos el proyecto presentado en el Ejemplo 74. La empresa matriz nos comunica que desea terminar el proyecto en 40 semanas, para lo que destinará recursos extra. En la Tabla 10, se recogen los costes estimados y semanas que se pueden ahorrar aplicando crashing en cada tarea.

Tabla 10: Costes de aplicar crashing al proyecto del Ejemplo 74

	Duración (semanas)		Coste ($)		Máxima reducción	Costo de crashing por semana
Actividad	Normal	Con Crash	Normal	Con Crash	en semanas	ahorrada
A	2	1	$180,000	$ 280,000	1	$100,000
B	4	2	$320,000	$ 420,000	2	$ 50,000
C	10	7	$620,000	$ 860,000	3	$ 80,000
D	6	4	$260,000	$ 340,000	2	$ 40,000
E	4	3	$410,000	$ 570,000	1	$160,000
F	5	3	$180,000	$ 260,000	2	$ 40,000
G	7	4	$900,000	$1,020,000	3	$ 40,000
H	9	6	$200,000	$ 380,000	3	$ 60,000
I	7	5	$210,000	$ 270,000	2	$ 30,000
J	8	6	$430,000	$ 490,000	2	$ 30,000
K	4	3	$160,000	$ 200,000	1	$ 40,000
L	5	3	$250,000	$ 350,000	2	$ 50,000
M	2	1	$100,000	$ 200,000	1	$100,000
N	6	3	$330,000	$ 510,000	3	$ 60,000

Con estos datos, ¿cuál es el coste mínimo que nos permite terminar el proyecto en 40 semanas?

Ejemplo

Primeramente consideremos las variables x_i, para $i = A, \ldots, N$, que representan la reducción en tiempo obtenida al realizar crashing en la actividad i. Notemos que $x_i \in \mathbb{Z}_+$.

Luego la función objetivo será $\boxed{\min z = 1000000x_A + 50000x_B + \ldots + 60000x_N}$

En las restricciones hay que incluir la precedencia. Para ello se introducen las variables auxiliares y_i, que representan el inicio de la actividad i. En cuanto a las restricciones se tendrá:

- Restricciones de crashing, no se puede superar el límite impuesto.

$$x_A \leq 1,\ x_B \leq 2, \ldots, x_N \leq 3$$

- Restricciones de no negatividad: no se pueden tener tiempo o inicios negativos, $y_i \geq 0$
- Restricciones de precedencia, se debe mantener el orden entre actividades

$$\begin{aligned} y_B &\geq 0 + 2 - x_A \\ y_H &\geq y_G + 7 - x_G \\ y_H &\geq y_E + 4 - x_E \end{aligned}$$

- Restricción de duración total del proyecto $y_{FIN} \leq 40$

El modelo matemático resulta

$$\begin{aligned}\min \quad & 100x_A + 50x_B + 80x_C + 40x_D + 160x_E + 40x_F + 40x_G + 60x_H \\ & + 30x_I + 30x_J + 40x_K + 50x_L + 100x_M + 60x_N\end{aligned}$$

s.t.

$$\begin{aligned}
y_B &\geq 2 - x_A \\
y_C &\geq y_B + 4 - x_B \\
y_D &\geq y_C + 10 - x_C \\
y_E &\geq y_C + 10 - x_C \\
y_F &\geq y_E + 4 - x_E \\
y_G &\geq y_D + 6 - x_D \\
y_H &\geq y_E + 4 - x_E \\
y_H &\geq y_G + 7 - x_G
\end{aligned}$$

$$\begin{aligned}
y_I &\geq y_C + 10 - x_C \\
y_J &\geq y_F + 5 - x_F \\
y_J &\geq y_I + 7 - x_I \\
y_K &\geq y_J + 8 - x_J \\
y_L &\geq y_J + 8 - x_J \\
y_M &\geq y_H + 9 - x_H \\
y_N &\geq y_K + 4 - x_K \\
y_N &\geq y_L + 5 - x_L
\end{aligned}$$

$$\begin{aligned}
y_{\text{FIN}} &\geq y_M + 2 - x_M \\
y_{\text{FIN}} &\geq y_N + 6 - x_N \\
0 \leq x_A &\leq 1, \\
0 \leq x_B &\leq 2, \\
0 \leq x_C &\leq 3, \\
0 \leq x_D &\leq 2 \\
0 \leq x_E &\leq 1, \\
0 \leq x_F &\leq 2,
\end{aligned}$$

$$\begin{aligned}
0 \leq x_G &\leq 3, \\
0 \leq x_H &\leq 3 \\
0 \leq x_I &\leq 2, \\
0 \leq x_J &\leq 2, \\
0 \leq x_K &\leq 1, \\
0 \leq x_L &\leq 2 \\
0 \leq x_M &\leq 1, \\
0 \leq x_N &\leq 3
\end{aligned}$$

$$y_B, y_C, y_D, y_E, y_F, y_G, y_H, y_I, y_J, y_K, y_L, y_M, y_N \geq 0$$

$$y_{\text{FIN}} \leq 40$$

$$x_A, \ldots, x_N \in \mathbb{Z}_+, \quad y_B, \ldots, y_N, y_{\text{FIN}} \in \mathbb{Z}_+$$

Para resolverlo empleamos el Código 9.

```
f = [100 50 80 40 160 40 40 60 30 30 40 50 100 60 0  0  0  0  0  0  0  0  0  0  0  0  0];
intcon = 1:length(f);
 A=[-1  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0;%B-A
     0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0  0  0  0  0  0  0  0;%C-B
     0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0  0  0  0  0  0  0;%D-C
     0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0 -1  0  0  0  0  0  0  0  0  0  0;%E-C
     0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0  0  0  0  0;%F-E
     0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0 -1  0  0  0  0  0  0  0  0;%G-D
     0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0 -1  0  0  0  0  0  0  0;%H-E
     0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0  0  0  0;%H-G
     0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0 -1  0  0  0  0  0  0;%I-C
     0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0 -1  0  0  0  0;%J-F
     0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0  0;%J-I
     0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1 -1  0  0  0  0;%K-J
     0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0 -1  0  0;%L-J
     0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 -1  0;%M-H
     0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0  0 -1  0;%N-K
     0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  1  0 -1  0;%N-L
     0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  1  0 -1;%Fin-M
     0  0  0  0  0  0  0  0  0  0  0  0  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  1 -1];%Fin-N
 b=-[2; 4; 10; 10; 4; 6; 4; 7; 10; 5; 7; 8; 8; 9; 4; 5; 2; 6];
Aeq = []; beq = [];
lb = zeros(1,length(f));
ub =[1  2  3  2  1  2  3  3  2  2  1  2  1  3   inf.*ones(1,13) 40];
[x val] = intlinprog(f,intcon,A,b,Aeq,beq,lb, ub)
cost = val + (180+320+620+260+410+180+900+200+210+430+160+250+100+330)
```

Código 9: PERT-CPM con crashing

Solución:
Se concluye que:

- Es posible acabar el proyecto en 40 semanas.
- El coste total ascendería a 4690000 dólares.
- Para ello se debe realizar un crashing sobre la
  - Tarea F de 2 semanas, y
  - Tarea J de 2 semanas.

§VI. Programación no lineal

- PERT: Program Evaluation and Review Technique .
- CPM: Critical Path Method.
- Manejo de incertidumbre.
- Trueque entre tiempo y coste. Crashing.

En este capítulo introducimos problemas más generales que involucran funciones no lineales. Está basado en el [Hillier and Lieberman, 2015, Cap. 12]

La optimización convexa desempeña un papel fundamental en la ingeniería debido a su tratabilidad matemática y su amplia aplicabilidad. Problemas como la reconstrucción de señales en imagen médica, el control óptimo de sistemas de administración de fármacos y los modelos de machine learning para herramientas de diagnóstico suelen basarse en formulaciones convexas para garantizar soluciones fiables y eficientes.
Los problemas convexos resultan especialmente atractivos porque cualquier mínimo local es también un mínimo global, y existen algoritmos potentes capaces de resolverlos de manera eficiente incluso en altas dimensiones.

Definición (Conjunto convexo)

Un conjunto $C \subseteq \mathbb{R}^n$ es *convexo* si, para cualesquiera dos puntos $x_1, x_2 \in C$, el segmento que los une queda completamente contenido en C:

$$\lambda x_1 + (1-\lambda)x_2 \in C \quad \text{para todo } \lambda \in [0,1]$$

Definición (Función convexa)

Una función $f : \mathbb{R}^n \to \mathbb{R}$ es *convexa* si su dominio es convexo y, para todo $x_1, x_2 \in \text{dom}(f)$ y $\lambda \in [0,1]$, se cumple:

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

Ejemplo (Función convexa)

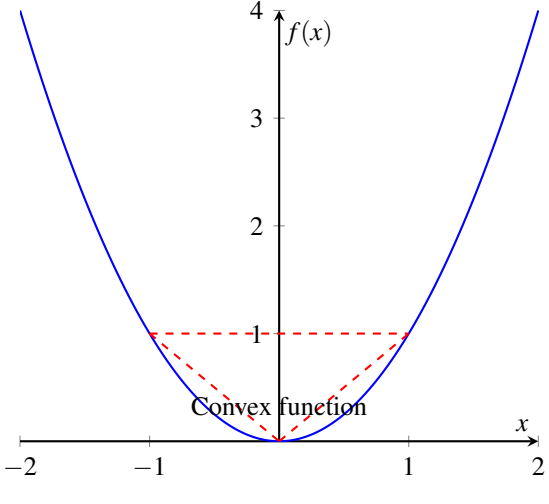
La función $f(x) = x^2$ es una función convexa. Notemos que $\text{dom}(f) = \mathbb{R}$, es trivialmente un conjunto convexo. Por otro lado, aplicamos la Definición 84, debemos probar que para todo $x_1, x_2 \in \mathbb{R}$, y $\lambda \in [0,1]$, se cumple

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) &= (\lambda x_1 + (1-\lambda)x_2)^2 \\ &\leq \lambda x_1^2 + (1-\lambda)x_2^2 = \lambda f(x_1) + (1-\lambda) f(x_2) \end{aligned}$$

$$\begin{aligned} \lambda x_1^2 + (1-\lambda)x_2^2 &\geq (\lambda x_1 + (1-\lambda)x_2)^2 \\ (1-\lambda)x_2^2 + \lambda x_1^2 &\geq \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1 x_2 + (1-\lambda)^2 x_2^2 \\ \lambda x_1^2 &\geq \lambda^2 x_1^2 + 2\lambda(1-\lambda)x_1 x_2 + \left[(1-\lambda)^2 - (1-\lambda)\right] x_2^2 \\ 0 &\geq \lambda(\lambda-1)x_1^2 - 2\lambda(\lambda-1)x_1 x_2 + \lambda(\lambda-1)x_2^2 \\ 0 &\geq \lambda(\lambda-1)(x_1 - x_2)^2 \end{aligned}$$

que se cumple para $\lambda \in [0,1]$.

Observe en la Figura 32 que geométricamente, el gráfico de una función convexa entre does puntos se encuentra por debajo del segmento que une ambos puntos.



Teorema (Funciones Convexas y Dos Veces Diferenciables)

Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función para la cual se cumple que $\nabla^2 f(\mathbf{x})$, la matriz Hessiana, es definida positiva para todo $\mathbf{x} \in \mathbb{R}^n$, entonces f es una función convexa.

Observación

Este teorema ofrece una condición fuerte para determinar si una función es convexa o no. Esto es equivalente, en una variable, a pedir que la segunda derivada sea positiva. Se recomienda encarecidamente revisar la definición de matriz Hessiana y de matriz definida positiva en el Apéndice B.

Ejemplo

La función $f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$ es convexa. Calculando la matriz hessiana de f,

$$\nabla^2 f(\mathbf{x}) = 2\mathrm{Id}_n$$

donde Id_n es la matriz identidad de tamaño $n \times n$. Como la matriz hessiana es definida positiva, aplicamos el Teorema 86 para concluir que f es una función convexa.

Definición (Función afín)

Se define $f : \mathbb{R}^n \to \mathbb{R}^p$ es una *función afín* si se puede escribir $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, donde $A \in \mathbb{R}^{n \times p}$ y $b \in \mathbb{R}^p$

Una función afín es la suma de una función lineal $A\mathbf{x}$ y una constante $\mathbf{b}$.

Definición (Máximo Local)

Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función, y sea $\mathbf{x}_0 \in \mathbb{R}^n$ un punto en el dominio de f. Decimos que f tiene un *máximo local* en $\mathbf{x}_0$ si existe un entorno U de $\mathbf{x}_0$ tal que:

$$f(\mathbf{x}) \leq f(\mathbf{x}_0) \quad \text{for all} \quad \mathbf{x} \in U.$$

En otras palabras, $f(\mathbf{x}_0)$ es mayor o igual a $f(\mathbf{x})$ para todo $\mathbf{x}$ cerca $\mathbf{x}_0$.

Definición (Mínimo Local)

Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función, y sea $\mathbf{x}_0 \in \mathbb{R}^n$ un punto en el dominio de f. Decimos que f tiene un *mínimo local* en $\mathbf{x}_0$ si existe un entorno U de $\mathbf{x}_0$ tal que:

$$f(\mathbf{x}) \geq f(\mathbf{x}_0) \quad \text{for all} \quad \mathbf{x} \in U.$$

En otras palabras, $f(\mathbf{x}_0)$ es menor o igual que $f(\mathbf{x})$ para todos los $\mathbf{x}$ cercanos a $\mathbf{x}_0$.

Un punto $\mathbf{x}_0$ donde f tiene un máximo o un mínimo local se denomina *punto crítico*. Estos puntos suelen aparecer donde el gradiente $\nabla f(\mathbf{x}_0)$ es cero, pero no todos los puntos críticos son extremos locales.

Ejemplo

Considera la gráfica de la siguiente función, $f(x) = x^4 - 17x^2 + x$, en la Figura 33. Los mínimos locales se pueden encontrar en

x^*	z^*
-1.11	-2.06
2.9	-69.34

y un máximo local en $x_3^* = 0.03$ para $z_3^* = 0.01$. Obsérvese que $x_1^* = -1.11$ es un mínimo global, porque es el punto más bajo que alcanza la función; sin embargo, x_3^* no es un máximo global, ya que esta función crece hasta infinito cuando el valor de x aumenta. Nótese que esta función no es convexa, la gráfica entre los mínimos locales es *cóncava*.

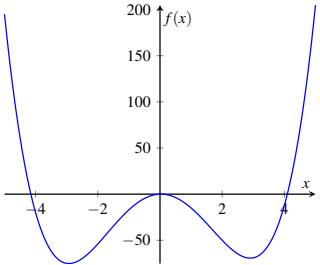


Figura 33: Función no convexa

Definición (Problema de optimización convexo)

Consideremos el problema de optimización en forma estándar

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{7}$$

decimos que (7) es un *problema de optimización convexa* si f y g_i son funciones convexas para $i = 1, \ldots, m$ y h_j son funciones afines para $j = 1, \ldots, p$. Esto hace que la región factible sea una región convexa.

Ejemplo (Convexo y no-convexo)

El problema

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s.t.} \quad & x_1^4 + e^{x_2} \leq 0 \\ & x_1 + x_2 = 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

es un problema convexo. Obsérvese que f y $x_1^4 + e^{x_2}$ son funciones convexas, y $x_1 + x_2$ es una función afín.
Sin embargo el problema

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = x_1^2 + x_2^3 \\ \text{s.t.} \quad & x_1^4 + e^{x_2} \leq 0 \\ & (x_1 + x_2)^2 = 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

no es un problema convexo porque f no es convexa y la restricción de igualdad no es una función afín, aunque puede transformarse en una.

Teorema (Optimalidad global en optimización convexa)

Cualquier óptimo local de un problema convexo es también un óptimo global

La programación no lineal sin restricciones se centra en encontrar el mínimo o el máximo de una función no lineal sin imponer restricciones sobre las variables. Esto resulta especialmente útil en aplicaciones donde la dinámica del sistema es inherentemente no lineal y compleja.

Definición (Problema Diferenciable No Lineal sin Restricciones)

Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función continuamente diferenciable. El Problema Diferenciable No Lineal sin Restricciones (DNLUP) es

$$\min_{\mathbf{x}} f(\mathbf{x})$$

donde $\mathbf{x} \in \mathbb{R}^n$.

La solución de este problema puede deducirse fácilmente a partir de las condiciones de primer y segundo orden estudiadas en Cálculo de varias variables.

Teorema (Solutiones del DNLUP)

Sea $\mathbf{x}^* \in \mathbb{R}^n$*, y* f *definidos anteriormente. Una condición necesaria para que* $\mathbf{x}^*$ *sea un mínimo local de* $f(x)$ *es que el gradiente se anule:*

$$\nabla f(\mathbf{x}^*) = 0$$

y la matriz hessiana $\nabla^2 f(\mathbf{x}^*)$ *sea positiva definida.* $\mathbf{x}^*$ *será el mínimo global, si* $z^* = f(\mathbf{x}^*) \leq f(\mathbf{x})$ *para todo* $\mathbf{x} \in \mathbb{R}^n$.

En otras palabras, para encontrar máximos o mínimos locales, primero localizamos los puntos críticos, donde el gradiente se anula.

Definición (Punto crítico)

Un punto $\mathbf{x}_0 \in \mathbb{R}^n$ se denomina *punto crítico* de una función diferenciable $f : \mathbb{R}^n \to \mathbb{R}$ si $\nabla f(\mathbf{x}_0) = \mathbf{0}$.

Observación

Los puntos críticos son candidatos a extremos locales, pero no garantizan necesariamente la existencia de máximos o mínimos.

Entonces, debemos aplicar la prueba de la segunda derivada, que ayuda a clasificar los puntos críticos como máximos locales, mínimos o puntos de silla.

Teorema (Test de la segunda derivada)

Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función dos veces diferenciable y sea $\mathbf{x}_0$ un punto crítico. Consideremos la matriz hessiana $H_f(\mathbf{x}_0)$:

- *Si $H_f(\mathbf{x}_0)$ es positiva definida, entonces $\mathbf{x}_0$ es un mínimo local.*
- *Si $H_f(\mathbf{x}_0)$ es definida negativa, entonces $\mathbf{x}_0$ es un máximo local.*
- *Si $H_f(\mathbf{x}_0)$ tiene autovalores positivos y negativos, entonces $\mathbf{x}_0$ es un punto de silla.*
- *Si $|H_f(\mathbf{x}_0)| = 0$ se está en presencia de un caso degenerado y no se puede deducir nada acerca de la función desde la matriz hessiana.*

Teorema (Test de la segunda derivada para funciones de dos variables)

Sea $f(x,y)$ una función de clase C^2 definida en un abierto U de $\mathbb{R}^2$. un punto (x_0,y_0) es un mínimo local de f si se cumplen las siguientes condiciones:

(i) $\frac{\partial f}{\partial x}(x_0,y_0) = \frac{\partial f}{\partial y}(x_0,y_0) = 0$

(ii) $\frac{\partial^2 f}{\partial x^2}(x_0,y_0) > 0$

(iii) $D = \left(\frac{\partial^2 f}{\partial x^2}\right)\left(\frac{\partial^2 f}{\partial y^2}\right) - \left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 > 0$ *en* (x_0,y_0)

(D es el determinante de la matriz hessiana.) Si en (ii) se tiene < 0 en vez de > 0 con la condición (iii), entonces se tendrá un máximo local. Si $D < 0$ (e.g., si $\frac{\partial^2 f}{\partial x^2}(x_0,y_0) = 0$ o $\frac{\partial^2 f}{\partial y^2}(x_0,y_0) = 0$, pero $\frac{\partial^2 f}{\partial x \partial y}(x_0,y_0) \neq 0$), entonces (x_0,y_0) es un punto de silla (no es máximo ni mínimo).

Ejemplo

Encuentre los máximos relativos, mínimos y puntos de silla de la función $f(x,y) = \log\left(x^2+y^2+1\right)$. Debemos primero localizar los puntos críticos de esta función; por lo tanto, de acuerdo con el Teorema 99, calculamos

$$\nabla f(x,y) = \frac{2x}{x^2+y^2+1}\mathbf{i} + \frac{2y}{x^2+y^2+1}\mathbf{j}$$

Así, $\nabla f(x,y) = \mathbf{0}$ si y solo si $(x,y) = (0,0)$, y por tanto el único punto crítico de f es $(0,0)$. Ahora debemos determinar si este es un máximo, un mínimo o un punto de silla. Las segundas derivadas parciales son

$$\frac{\partial^2 f}{\partial x^2} = \frac{2\left(x^2+y^2+1\right)-(2x)(2x)}{\left(x^2+y^2+1\right)^2}$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{2\left(x^2+y^2+1\right)-(2y)(2y)}{\left(x^2+y^2+1\right)^2}$$

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{-2x(2y)}{\left(x^2+y^2+1\right)^2}$$

Por tanto,

$$\frac{\partial^2 f}{\partial x^2}(0,0) = 2 = \frac{\partial^2 f}{\partial y^2}(0,0) \quad \text{y} \quad \frac{\partial^2 f}{\partial x \partial y}(0,0) = 0$$

lo que implica $D = 2 \cdot 2 = 4 > 0$. Como $\left(\partial^2 f/\partial x^2\right)(0,0) > 0$, concluimos que $(0,0)$ es un mínimo local. (¿Puedes demostrarlo únicamente a partir del hecho de que $\log t$ es una función creciente de $t > 0$?)

En programación lineal todas sus funciones (objetivo y de restricción) son lineales. Este supuesto se cumple en el caso de muchos problemas prácticos, pero no siempre es así. Para tratar con situaciones en las que la programación lineal no puede modular, vamos a tratar con funciones no lineales.

Definición

Problema de programación no lineal De manera general, el problema de programación no lineal consiste en encontrar $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ para

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & g_i(\mathbf{x}) \leq b_i, \text{ para } i = 1, 2, \ldots, m, \\ & \mathbf{x} \geq 0, \end{array}$$

donde $f(\mathbf{x})$ y $g_i(\mathbf{x})$ son funciones dadas de n variables de decisión.

Dada las diferentes formas que pueden tomar las funciones $f(x)$ y $g(x)$ existen diferentes tipos de problemas PNL. Cada uno de estos problemas tiene una resolución específica, por lo que tratarlos individualmente es una tarea compleja. Además, existen ciertas consideraciones que se deben tener en cuenta a la hora de abordar este tipo de problemas.

Ejemplo (Wyndor Glass formulación no lineal)

Considere el modelo del Ejemplo 6. Si se sustituye la segunda restricción por una restricción no lineal, $9x_1^2 + 5x_2^2 \leq 216$, la solución óptima sigue siendo $(x_1^*, x_2^*) = (2, 6)$, vea Figura 34, o Geogebra.

$$\begin{aligned} \max_{\mathbf{x}} \quad & z = 3x_1 + 5x_2 \\ \text{s.a.} \quad & x_1 \leq 4 \\ & 9x_1^2 + 5x_2^2 \leq 216 \\ & 3x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Notemos que en este ejemplo, la solución se encuentra sobre la frontera de la región factible, pero no es una **solución factible de vértice**.

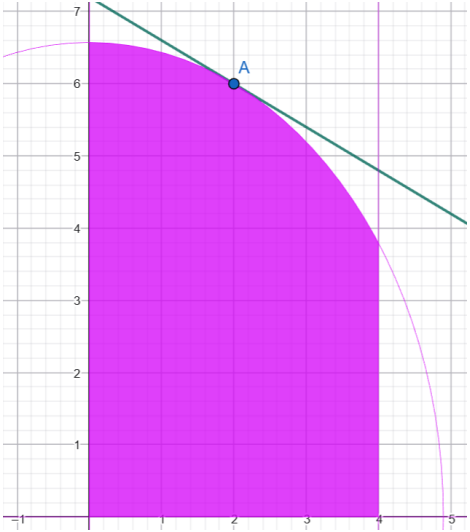


Figura 34: Modelo Ejemplo 6, con una restricción no lineal

Ejemplo

Ahora supongamos que las restricciones lineales se conservan, pero la función objetivo cambia, para hacerse no lineal. Si

$$\begin{aligned} \max_{\mathbf{x}} z \quad & = 126x_1 - 9x_1^2 + 182x_2 - 13x_2^2 \\ \text{s.a.} \quad & x_1 \leq 4 \\ & x_2 \leq 6 \\ & 3x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{aligned}$$

La solución óptima ahora resulta $(x_1^*, x_2^*) = \left(\frac{8}{3}, 5\right)$, con $z^* = 857$, vea Figura 35, o Geogebra.

Esta solución, nuevamente se encuentra sobre la frontera de la región factible, pero no es una **solución factible de vértice**.

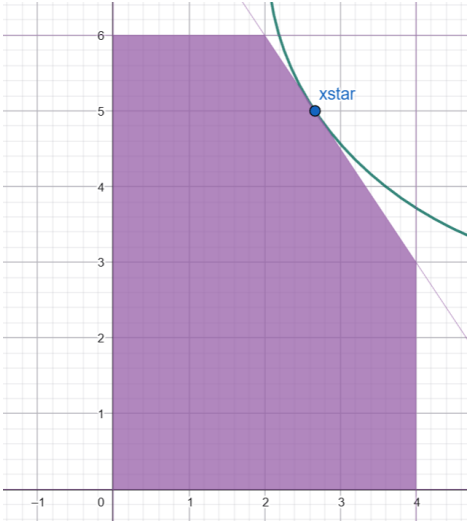


Figura 35: Modelo Ejemplo 6, con una función objetivo no lineal

Finalmente consideremos el siguiente caso

Ejemplo

Conservamos las restricciones lineales, pero la función objetivo cambia a $z = 54x_1 - 9x_1^2 + 78x_2 - 13x_2^2$, para hacerse no lineal. Si

$$\begin{aligned} \max_{\mathbf{x}} z \quad & = 54x_1 - 9x_1^2 + 78x_2 - 13x_2^2 \\ \text{s.a.} \quad & x_1 \leq 4 \\ & x_2 \leq 6 \\ & 3x_1 + 2x_2 \leq 18 \\ & x_1, x_2 \geq 0 \end{aligned}$$

La solución óptima ahora resulta $(x_1^*, x_2^*) = (3,3)$, con $z^* = 198$, vea Figura 36, o Geogebra.

Por lo tanto, es necesario que un algoritmo general para resolver problemas de este tipo tome en cuenta todas las soluciones en la región factible, y no sólo aquellas que están sobre la frontera.

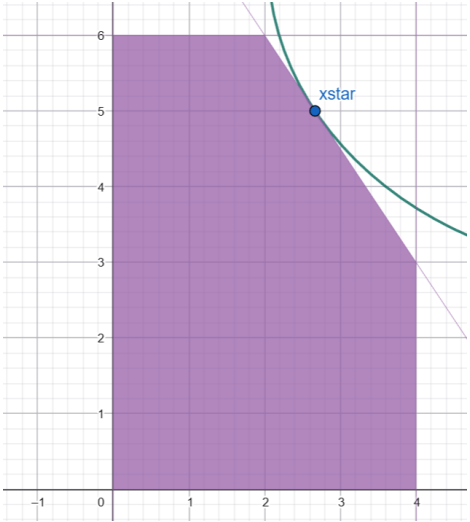


Figura 36: Modelo Ejemplo 6, con una función objetivo no lineal

Otra complicación que surge en programación no lineal es que un máximo local no necesariamente es un máximo global es decir, no es la solución óptima global, ver Figura 37.

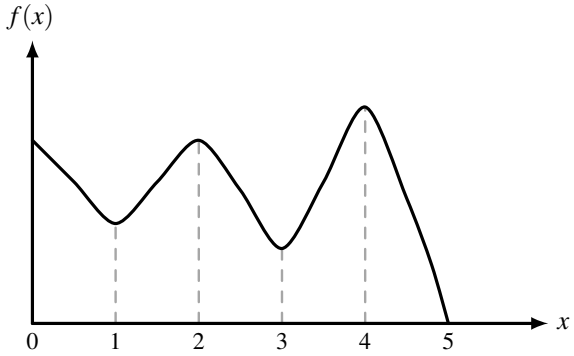


Figura 37: Función multimodal: varios óptimos locales

Los algoritmos de Programación no Lineal, en general, no pueden distinguir entre máximos y mínimos locales y globales.

Los problemas PNL presentan muchas formas distintas. Al contrario que en el caso lineal, no hay un método general de resolución. Existen pues distintos métodos específicos para cada una de las familias, tales como

- Optimización no restringida: este tipo de problemas no tiene restricciones, por lo que su función objetivo es del tipo

  Minimizar min o maximizar max $f(\mathbf{x})$, donde esta función es no lineal.

- Optimización restringida linealmente: en este tipo de problemas la única no linealidad aparece en la función objetivo, mientras las restricciones son lineales

  Minimizar min o maximizar max $f(\mathbf{x})$, donde esta función es no lineal, restringida a $g_i(\mathbf{x}) \leq b_i$, funciones lineales.

- **Programación cuadrática**: la función objetivo tiene forma cuadrática, es decir, hay polinomios de segundo grado, ya sea por variables al cuadrado o producto de variables.
- **Programación convexa**: aquí se deben cumplir dos condiciones:
  - $f(\mathbf{x})$ debe ser cóncava o convexa
  - Cada una de las funciones de restricción $g_i(\mathbf{x})$ han de ser convexas

  En este caso se puede asegurar que la solución es mínimo o máximo global.
- Programación no convexa

- **Programación separable**: es un tipo especial de programación convexa, donde además se ha de cumplir
  - $f(\mathbf{x})$ y $g_i(\mathbf{x})$ han de ser separables

  ¿Qué es una función separable? Aquella en la que cada uno de sus términos sólo contiene una variable

$$f(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j)$$

- **Programación no convexa**: este tipo de problemas recoge todos aquellos que no satisfacen los supuestos de la programación convexa. En este tipo de problemas no hay garantía de encontrar máximos globales.
- **Programación geométrica**: es un tipo especial de programación no convexa en el que las restricciones son de tipo polinomios.

$$\begin{aligned} g(\mathbf{x}) &= \sum_{i=1}^{N} c_i P_i(\mathbf{x}) \\ P_i(\mathbf{x}) &= x_1^{a_1 i}, x_2^{a_2 i}, \ldots, x_n^{a_n i} \end{aligned}$$

- **Programación fraccional**: en este tipo de problemas, la función objetivo tiene forma de fracción

Para optimización no lineal sin restricciones, `Matlab` tiene las funciones

- `fminunc`, basada en el gradiente y
- `fminsearch`, no basada en el gradiente.

Para más información consultar Apéndice G.

`fminunc` resuelve problemas del tipo $\min_x f(x)$

Ejemplo

Minimizar la función $2x^2 - 1$ empleando `fminunc`, para encontrar un mínimo alrededor del punto $x_0 = 1$

```
1 fun = @(x)x^2-1; %anonimus function
2 x0 = 1; % starting point
3 fminunc(fun,x0);
```

Código 10: Ejemplo de uso de `fminunc`

La función se debe escribir en modo anónimo `@(x)` para declarar la variable y se debe proveer de un punto de inicio para el algoritmo.
La solución es $x^* = 0$ para $z^* = -1$. Ver Código 10

Ejemplo

Tenemos que instalar una nueva antena que permita dar cobertura mediante radio-enlace a cada una de las sedes de nuestra empresa. Estas se encuentran distribuidas tal y como aparece en la figura 38.

- Las perdidas debidas a la propagación en espacio libre tienen la siguiente forma $FSPL = \left(\frac{4\pi d}{\lambda}\right)^2$
- La longitud de onda no va a cambiar para ninguna de las sedes a las que tenemos que darle cobertura, nuestro problema se convierte en la minimización de las distancias entre nuestra antena y nuestras bases.
- Si definimos la posición de nuestra antena como (x_a, y_a) podemos escribir la distancia hasta cada una de nuestras sedes como $d_i = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2}$
- De tal manera que la función a minimizar es

$$z = \sum_{i=1}^{N} d_i; \quad \text{donde N es el número de sedes}$$

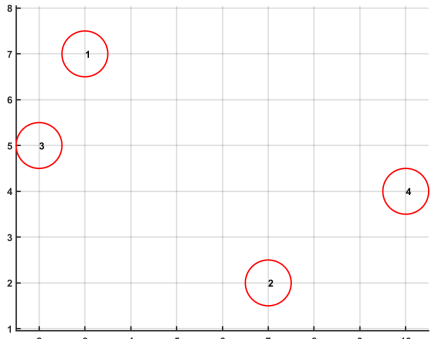


Figura 38: Mapa Ejemplo 106

Se desea que las perdidas debido a la propagación en espacio libre sea **mínima**. ¿En qué coordenadas deberemos instalar la antena? (asumir que las distancias están en km).

```
x0 = [1,1];
[x,fval] = fminunc(@fdist,x0);
function [d] = fdist(x)
pos_sedes = [3,7;
  7,2;
  2,5;
  10,4];
d = 0;
for i = 1:size(pos_sedes,1)
    d = d + sqrt((pos_sedes(i,1)-x(1))^2 +(pos_sedes(i,2)-x(2))^2);
end
end
```

Código 11: Ejemplo de uso de `fminunc`

Aplicando la función de *MATLAB*, ver Código 11, que hemos visto anteriormente para este problema obtenemos la siguiente solución:

- x = [4.5455,4.6818]
- fval = 15.2734

Ejemplo

Es decir, nuestra antena debe colocarse en las coordenadas $(4.5455, 4.6818)$ y la suma de distancias a todas las estaciones es de 15.2734 km, ver Figura 39.

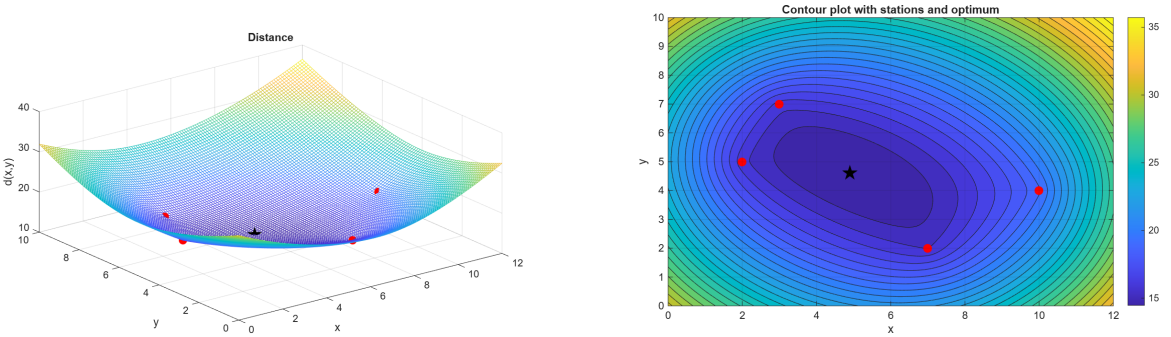


Figura 39: Solución Ejemplo 106. Izquierda, representación tridimensional del problema y su solución. Derecha: mapa de contornos.

Optimización no lineal con restricciones. Solución con Matlab

Para optimización no lineal con restricciones, emplearemos la función `fmincon` de matlab, basada en el gradiente. Para más información consultar Apéndice G.

Ejemplo

El problema 9 tiene como solución $x^* \approx (0.7071, 0.7071)$ for $z^* \approx 1.4142$, ver Código 12

$$\begin{aligned} \max z &= x_1 + x_2 \\ \text{s.a} x_1^2 + x_2^2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned} \tag{9}$$

```
1  A=[];b=[];
2  Aeq=[];beq=[];
3  lb = [0,0];
4  ub = [];
5  x0 = [1,0];
6  nonlcon = @circlecon;
7  [x,fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon);
8  fun = @(x)-(x(1)+x(2)); %function
9  function [c,ceq] = circlecon(x) %declaring the non-linear constraints
10 c = (x(1))^2 + (x(2))^2-1;
11 ceq = [];
12 end
```

Ejemplo (Ejemplo prototipo, uso de `fmincon`)

Tenemos que instalar una nueva antena que permita dar cobertura mediante radio-enlace a cada una de las sedes de nuestra empresa, ver Figura 38.
Se desea que las perdidas debido a la propagación en espacio libre sea mínima. ¿En qué coordenadas deberemos instalar la antena? (asumir que las distancias están en km).
Además queremos que ninguna de las sedes se encuentre a una **distancia superior a 5 km** ya que la banda de frecuencias que utilizamos está muy saturada y necesitamos que llegue un nivel de potencia determinado.

Solución: La longitud de onda no va a cambiar para ninguna de las sedes a las que tenemos que darle cobertura, nuestro problema se convierte en la minimización de las distancias entre nuestra antena y nuestras bases.

Si definimos la posición de nuestra antena como (x_a, y_a) podemos escribir la distancia hasta cada una de nuestras sedes como

$$d_i = \sqrt{(x_i - x_a)^2 + (y_i - y_a)^2}$$

De tal manera que la función a minimizar es

$$z = \sum_{i=1}^{N} d_i; \quad \text{donde N es el número de sedes}$$

$$st.d_i < 5 \quad \text{i} = 1, 2, 3, 4.$$

Aplicando la función de Matlab *fmincon*, ver Código 13, obtenemos la siguiente solución:

- x = [5.0321,4.5652]
- fval = 14.4675

Es decir, nuestra antena debe colocarse en las coordenadas $(5.0321, 4.5652)$ y la suma de distancias a todas las estaciones es de 14.4675 km, ver Figura 40

```
A=[];b=[];
Aeq=[];beq=[];
lb = [0,0];
ub = [];
x0 = [1,1];
[x,fval] = fmincon(@fdist,x0,A,b,Aeq,beq,lb,ub,@nonlcon);

function [d] = fdist(x) %distance function
pos_sedes = [3,7;
  7,2;
  2,5;
  10,4];
d = 0;
for i = 1:size(pos_sedes,1)
    d = d + sqrt((pos_sedes(i,1)-x(1))^2 +(pos_sedes(i,2)-x(2))^2);
end
end
function [c,ceq] = nonlcon(x) %nonlinear constraint -- distance
pos_sedes = [3,7;
  7,2;
  2,5;
  10,4];
c = zeros(size(pos_sedes,1),1);
for i = 1:size(pos_sedes,1)
    c(i) = sqrt((pos_sedes(i,1)-x(1))^2 + ...
                (pos_sedes(i,2)-x(2))^2) - 5;
end
ceq = [];
end
```

Código 13: Solución Ejemplo 109

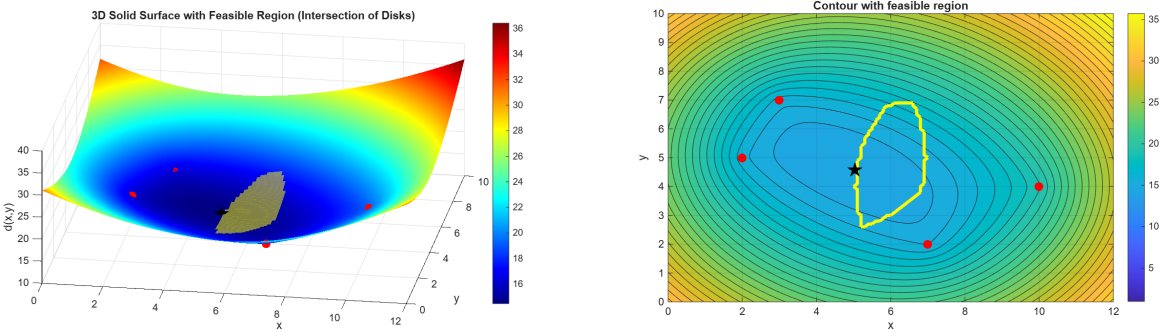


Figura 40: Solución Ejemplo 109. Izquierda, representación tridimensional del problema y su solución. Derecha: mapa de contornos. En amarillo la intersección de los discos inducida por las nuevas restricciones de distancia.

Muchos problemas de optimización en ingeniería implican relaciones no lineales y están sujetos a restricciones derivadas de consideraciones físicas, biológicas o de seguridad. Por ejemplo, optimizar la dosis de un fármaco debe respetar los límites fisiológicos, y el diseño de una prótesis debe ajustarse a las restricciones anatómicas. Estos problemas se modelan mediante *programación no lineal con restricciones*, donde tanto la función objetivo como las restricciones pueden ser no lineales.

Definición (Formulación canónica problema no lineal con restricciones)

Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función objetivo, y sean $g_i(x) \leq 0$ y $h_j(x) = 0$ las restricciones de desigualdad y de igualdad, respectivamente. La forma general de un problema de programación no lineal con restricciones es:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_j(x) = 0, \quad j = 1, \ldots, p \end{aligned}$$

Surge una pregunta natural: ¿existe alguna generalización de las anteriores condiciones (las dadas en la Sección 1) de punto crítico y de segundo orden para encontrar las soluciones óptimas? La respuesta es sí. La solución óptima debe satisfacer las condiciones de Karush-Kuhn-Tucker (KKT), que generalizan el método de los multiplicadores de Lagrange para manejar restricciones de desigualdad.

Las condiciones de Karush-Kuhn-Tucker (KKT) son un conjunto de condiciones necesarias para que una solución en programación no lineal sea óptima, dadas ciertas condiciones de regularidad. Estas condiciones generalizan el método de los multiplicadores de Lagrange para manejar restricciones de desigualdad y son fundamentales en problemas de optimización con restricciones.
Consideremos el siguiente problema de optimización:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{10}$$

Supongamos que f, g_i y h_j son diferenciables en la región factible $\mathscr{F} \subset \mathbb{R}^n$.

Definición (Función de Lagrange)

Definimos la función de Lagrange asociada a la ecuación (10) como

$$\mathscr{L}(\mathbf{x},\lambda,\mu) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g(\mathbf{x}) + \sum_{j=1}^{p} \mu_j h(\mathbf{x})$$

donde $\lambda = (\lambda_1, \ldots, \lambda_m) \in \mathbf{R}^m$ y $\mu = (\mu_1, \ldots, \mu_p) \in \mathbf{R}^p$ son vectores.

Teorema (KKT-conditions)

El puntox* es un mínimo local ***si y solo si****, existen multiplicadores de Lagrange $\lambda_i^* \geq 0$ and $\mu_j^* \in \mathbb{R}$ tales que:*

(Condición estacionaria)
$$\nabla \mathscr{L} = \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(x^*) \quad = 0$$

(Condición de factibilidad primal)
$$\begin{cases} g_i(x^*) & \leq 0, \quad i = 1, \ldots, m \\ h_j(x^*) & = 0, \quad j = 1, \ldots, p \end{cases}$$

(Condición de positividad/negatividad)
$$\lambda_i^* \geq 0 (\textit{mínimos}), \quad i = 1, \ldots, m$$
$$\lambda_i^* \leq 0 (\textit{máximos}), \quad i = 1, \ldots, m$$

(Condición de holgura)
$$\lambda_i^* g_i(x^*) = 0, \quad i = 1, \ldots, m$$

Si el problema es de maximización, como el único cambio es en los multiplicadores, estos serán un indicio para detectar la naturaleza del punto.

Observación

- Este mínimo local de x^* coincide con el mínimo $\ell^* = (x_1, \ldots, x_n, \lambda_1, \lambda_m, \mu_1, \ldots, \mu_p)$ de la función Lagrangiana $\mathscr{L}$.
- Las condiciones KKT transforman el proceso de optimización en una búsqueda de raíces de sistemas de ecuaciones.
- Si $p = 0$, es decir, no hay restricciones de igualdad, este resultado se reduce al caso habitual de los multiplicadores de Lagrange.
- La condición de estacionariedad se refiere a la condición de primer orden que hace que el gradiente de una función sea igual a cero.
- La factibilidad primal obliga a que la solución pertenezca a la región factible.
- La factibilidad dual fija el signo de los λ según la naturaleza de mínimo/máximo.
- La holgura complementaria fuerza una de las siguientes condiciones: $\lambda_i = 0$ o la restricción g_i está activa en la solución dada.

- Para poder asegurar que un punto cumple las condiciones necesarias de optimalidad es obligatorio que este punto cumpla alguna de las hipótesis de cualificación de restricciones.
- Demostrar estas hipótesis se sale fuera del alcance de la asignatura.
- Aplicaremos la siguiente: Un punto cumple las KKT si el conjunto del gradiente de las restricciones activas es linealmente independiente.

Ejemplo (Condiciones de KKT en matlab)

Dado el siguiente problema de optimización, encuentre todos los puntos críticos y clasifíquelos de acuerdo a las condiciones KKT de primer orden. Compruebe los puntos factibles obtenidos.

$$\begin{aligned} \min / \max_{(x,y,z)} \quad & x^2 - y^2 \\ \text{s.t.} \quad & x^2 + 2x + 1 + y^2 \leq 4 \\ & x^2 - 2x + 1 + y^2 \leq 4 \end{aligned}$$

Solución:

1. Obtener el lagrangiano

$$\mathscr{L} = f + \lambda_1 g_1 + \lambda_2 g_2$$

2. Aplicar la condición estacionaria. Calcular el gradiente del Lagrangiano e igualarlo a cero ($\nabla \mathscr{L} = 0$). En Matlab puedes utilizar el comando `gradient`.
3. Aplicar la condición de holgura. Igualar a cero las restricciones de desigualdad multiplicadas por λ_i, llamados multiplicadores de Lagrange.
4. Resolver el sistema formado por las cuatro ecuaciones anteriores. En Matlab puedes utilizar el comando `solve`.

5. Los puntos críticos (x, y) obtenidos son, ver Tabla 11.

Tabla 11: Puntos crítcos Ejemplo 113

x	y	λ_1	λ_2
0	0	0	0
1	0	-1/2	0
-1	0	0	-1/2
-3	0	-3/2	0
3	0	0	-3/2
0	$\sqrt{3}$	1/2	1/2
0	$-\sqrt{3}$	1/2	1/2
$-1/2$	$-\sqrt{15}/2$	1	0
$-1/2$	$\sqrt{15}/2$	1	0
1/2	$-\sqrt{15}/2$	0	1
1/2	$\sqrt{15}/2$	0	1

6. Las condiciones KKT de primer grado son las siguientes
   - **Condición estacionaria y de holgura**: se cumplen a la hora de resolver el sistema de ecuaciones
   - **Condición de factibilidad**: Comprobar que se cumplen las restricciones

$$h_i(x^*) = 0 \qquad i = 1, \ldots, m$$
$$g_j(x^*) = 0 \qquad j = 1, \ldots, p$$

   - **Condición de positividad/negatividad**: El signo de λ_i indica el tipo de punto: $\lambda_i \leq 0$ indica máximo, $\lambda_i \geq 0$ indica mínimo, ver Tabla 12.
7. Comprobar los puntos factibles. Utilizamos `subs` para sustituir los puntos obtenidos en las restricciones y comprobaremos si se cumplen `subs(g1, [x,y], [sol.x, sol.y])`

8. Comprobar los valores obtenidos para los puntos factibles. Utilizaremos la función `fmincon`, ya que vamos a optimizar una función no lineal con restricciones para cada punto, ver Código 14.

El flujo de trabajo completo de este ejemplo se puede reproducir a partir del Código 15.

```
x0 =[sol.x,sol.y];
A=[]; b=[]; Aeq = []; beq = []; lb =[]; ub = [];
fun = @(x) x(1)^2 - x(2)^2;
[x,fval]=fmincon(fun, x0, A,b,Aeq,beq,lb,ub,@nonlcon);

function[c, ceq] = nonlcon(x)
    c = [ x(1)^2 + 2*x(2) - 3 + x (2)^2 , x(1)^2 - 2 *x(2) - 3 + x(2) ^2 ];
    ceq = [];
end
```

Código 14: Aplicación de `fmincon`

Tabla 12: Clasificación de puntos críticos Ejemplo 113

x	y	λ_1	λ_2	Factible	Máx/Mín	Valor
0	0	0	0	Sí	¿?	0
1	0	-1/2	0	Sí	Máx	1
-1	0	0	-1/2	Sí	Máx	1
-3	0	-3/2	0	No	-	-
3	0	0	-3/2	No	-	-
0	$\sqrt{3}$	1/2	1/2	Sí	Mín	-3
0	$-\sqrt{3}$	1/2	1/2	Sí	Mín	-3
$-1/2$	$-\sqrt{15}/2$	1	0	No	-	-
$-1/2$	$\sqrt{15}/2$	1	0	No	-	-
1/2	$-\sqrt{15}/2$	0	1	No	-	-
1/2	$\sqrt{15}/2$	0	1	No	-	-

```
syms  x y m1 m2
Z = x^2-y^2;
g1 = x^2+2*x-3+y^2;
g2 = x^2-2*x-3+y^2;

L = Z + m1*g1 + m2*g2;
nLx = gradient(L,x);
nLy = gradient(L,y);
eq1 = nLx == 0;
eq2 = nLy == 0;
eq3 = m1*g1 == 0;
eq4 = m2*g2 == 0;
sist = [eq1;eq2;eq3;eq4];
sol = solve(sist);

for i = 1:11
    Fac(i) = subs(g1,[x,y],[double(sol.x(i)),double(sol.y(i))])<=0 && subs(g2,[x,y],[double(sol.x(i)),
        double(sol.y(i))])<=0;
    mx(i) = sol.m1(i)<=0 && sol.m2(i)<=0;
    mn(i) = sol.m1(i)>=0 && sol.m2(i)>=0;
end

for i = 1:11
    x0 = double([sol.x(i),sol.y(i)]);
    if Fac(i) && mn(i)
        disp('min')
        i, x0
        fun = @(x) x(1)^2-x(2)^2;
        [x,fval] = fmincon(fun,x0,[],[],[],[],[],[],@nonlcon)
    end
    if Fac(i) && mx(i)
        disp('max')
        i, x0
        fun = @(x) -x(1)^2+x(2)^2;
        [x,fval] = fmincon(fun,x0,[],[],[],[],[],[],@nonlcon)
    end
end
%

function [c,ceq] = nonlcon(x)
c = [x(1)^2+2*x(1)-3+x(2)^2, x(1)^2-2*x(1)-3+x(2)^2];
ceq = [];
end
```

Se debe tener en cuenta que

- Las condiciones KKT son expresiones de primer orden que pueden dar lugar a máximos, mínimos o puntos de silla.
- Si aplicamos las condiciones KKT de segundo orden podemos, conocer la verdadera naturaleza de los puntos obtenidos. Además, este tipo de condiciones son suficientes para la optimalidad.
- Estas condiciones vienen marcadas por condiciones sobre el Hessiano:

$$\mathbf{z}^{\mathbf{T}} \mathscr{H}\mathbf{x}^{*2} \mathscr{L}(\mathbf{x}^*, \lambda^*, \mu^*)\mathbf{z} > 0$$
$$\forall z \begin{bmatrix} \nabla \mathbf{h}(\mathbf{x}^*) \\ \nabla \mathbf{g}_{act}(\mathbf{x}^*) \end{bmatrix} \mathbf{z} = 0$$

Ejemplo

Dado el siguiente problema de optimización, encuentre todos los puntos críticos y clasifíquelos de acuerdo a las condiciones KKT de primer orden. Compruebe los puntos factibles obtenidos.

$$\begin{aligned} \min / \max_{(x,y,z)} \quad & x^2 + y^2 \\ \text{s.t.} \quad & x^2 + 2x - 1 + y^2 \leq 4 \\ & x^2 - 2x + 1 + y^2 \leq 3 \end{aligned}$$

Los puntos críticos obtenidos están en la Tabla 13.

Tabla 13: Clasificación puntos críticos Ejemplo 114

x	y	μ_1	μ_2	Valor	Mín/Máx
0	0	0	0	0	Mín/Máx
$1-\sqrt{3}$	0	0	$\sqrt{3}/3-1$	0.5359	Máx
$-1-\sqrt{6}$	0	$-\sqrt{6}/6-1$	0	11.899	-
3/4	$-\sqrt{47}/4$	-1/2	-1/2	12.3125	Máx
3/4	$\sqrt{47}/4$	-1/2	-1/2	12.3125	Máx
$1+\sqrt{3}$	0	$-1/2$	$-\sqrt{3}/3-1$	7.4641	
$-1+\sqrt{6}$	0	$\sqrt{6}/6-1$	0	2.1010	Máx

Para resolver un problema no lineal en el solver de Windows solo seleccione la opción `Problem: non-linear` y escriba el problema teniendo en cuenta que las potencias deben escribirse usando `**`. Véase el Apéndice B para más información.

§VII. Teoría de colas

- PERT: Program Evaluation and Review Technique .
- CPM: Critical Path Method.
- Manejo de incertidumbre.
- Trueque entre tiempo y coste. Crashing.

Referencia

Para este capítulo se emplea como referencia [Miranda, 2005]. Para más detalles acerca de la formulación de los modelos esta referencia es sumamente útil.

La teoría de colas estudia el comportamiento de sistemas donde existe un conjunto limitado de recursos para atender las peticiones generadas por los usuarios, de tal manera que cuando un usuario envía una tarea al sistema, ésta podrá tener que esperar para ser atendida por algún recurso, o, incluso, podrá ser rechazada si el sistema no tiene capacidad suficiente para almacenarla en espera de ser atendida. El estudio de estos sistemas implicará el modelado no sólo del sistema en sí, sino también del comportamiento aleatorio del tráfico ofrecido por los usuarios al sistema. Este tráfico ofrecido por los usuarios se modela mediante dos procesos estocásticos: proceso de llegadas de las tareas y proceso de servicio demandado por las mismas, usualmente considerados independientes entre sí.

El conjunto más amplio de resultados analíticos en teoría de colas se encuentran para los procesos de llegada de Poisson y proceso de servicio demandado aleatorio puro modelado por variables aleatorias con distribución exponencial, que caracterizan sistemas de espera en los que tanto las llegadas de tareas de usuarios como la demanda que realizan son completamente aleatorias, en el sentido de que la evolución del sistema sólo depende de su condición presente y no de su pasado. Si bien en la práctica surgen otros modelos de colas con caracterización del tráfico mediante otros tipos de procesos, su análisis matemático es más complejo y no se dispone, en la actualidad, de una solución general para todos los casos posibles de interés.

Para el estudio de sistemas modelados mediante modelos de colas cuya solución analítica se desconoce, o bien es muy difícil de obtener, se suele recurrir a técnicas de simulación que, mediante el empleo de programas de computación, permiten obtener estimaciones de los parámetros de interés del comportamiento del sistema.

El objetivo de este capítulo es el estudio de esta disciplina enmarcada dentro de la investigación de operaciones por lo que en los siguientes apartados nos centraremos en el estudio de un conjunto de modelos de colas con solución analítica conocida.

La teoría de colas estudia aquellos sistemas en los que existe contienda por los recursos, lo que hace que en un momento dado pueda haber más demanda de recursos que recursos disponibles, por lo que algunas de estas solicitudes de servicio deberán esperar a ser atendidas, formándose así una cola de acceso de las tareas a los recursos. e incluso podrán ser rechazadas por falta de capacidad de almacenamiento.

La mayoría de resultados de la teoría de colas se obtienen suponiendo que los sistemas estudiados son conservativos. Un sistema de espera es conservativo si los recursos no están desocupados cuando hay tareas esperando a ser servidas, y si la demanda de servicio de dichas tareas no depende del orden en que son servidas.

En los modelos de colas existentes, las tareas generadas por los usuarios de una población llegan al sistema en un instante de tiempo aleatorio. Estas tareas demandan una cantidad de servicio también aleatoria, y se sirven una a una o por grupos. El orden en el que se sirven las tareas que ocupan la cola del sistema lo determina una disciplina de gestión de cola.

En los siguientes apartados se describen detalladamente los elementos necesarios para la caracterización de un modelo de colas cualquiera, tanto el modelado del tráfico ofrecido por los usuarios, como el modelado del sistema encargado de aceptar y atender sus peticiones de servicio.

Importante

Todos los resultados obtenidos en este curso parten de la suposición de que nuestros sistemas son **conservativos**, es decir, no se puede dar el caso de servidores inactivos cuando hay tareas por realizar.

La **Teoría de Colas** estudia el comportamiento de los sistemas donde existe un conjunto limitado de recursos para atender las peticiones generadas por los usuarios.
Un modelo genérico de colas puede representarse como la Figura 41. El principio básico de funcionamiento es el siguiente: un cliente **llega** a través de *Llegadas*, **ingresa** o no al sistema, representado por líneas discontinuas. una vez dentro del sistema, respeta una disciplina de cola, para llegar al canal de servicio. Una vez atendido, **egresa** o sale del sistema.

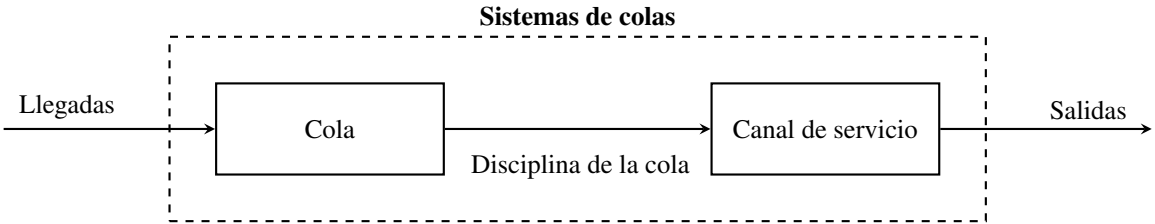


Figura 41: Estructura genérica de una cola

Los modelos de colas cuentan con varias partes básicas que hay que tener en cuenta, siendo estas:

- **Población**: La población representa el conjunto de clientes potenciales que pueden requerir el servicio. Puede ser finita, cuando el número de posibles clientes está limitado (por ejemplo, máquinas en una fábrica), o infinita, cuando no existe un límite práctico al número de clientes que pueden llegar al sistema, como ocurre en muchos servicios abiertos al público.
- **Llegadas**: Las llegadas describen la forma en que los clientes acceden al sistema. Se caracterizan por la frecuencia y regularidad con que ocurren y suelen modelarse mediante procesos aleatorios, siendo el proceso de Poisson uno de los más utilizados en teoría de colas.
- **Impaciencia**: La impaciencia se refiere al comportamiento de los clientes cuando el tiempo de espera es excesivo. Algunos clientes pueden abandonar la cola antes de ser atendidos (abandono) o decidir no entrar al sistema al observar una cola demasiado larga (rechazo).
- **Capacidad**: La capacidad del sistema indica el número máximo de clientes que pueden estar simultáneamente en la cola y en servicio. Cuando esta capacidad es limitada, los clientes que llegan al sistema lleno no pueden ingresar y se pierden.
- **Modo de llegadas**: El modo de llegadas especifica si los clientes llegan individualmente o en lotes, así como si las llegadas dependen o no del estado del sistema. Este aspecto influye directamente en la congestión y el desempeño del sistema.

- **Disciplina de cola**: La disciplina de cola define el orden en que los clientes son atendidos. La más común es primero en llegar, primero en ser atendido, aunque también existen otras como prioridad, atención aleatoria o último en llegar, primero en ser atendido.
- **Duración del servicio**: La duración del servicio corresponde al tiempo necesario para atender a un cliente. Este tiempo puede ser constante o aleatorio y se modela mediante distintas distribuciones de probabilidad, dependiendo de la naturaleza del sistema.
- **Modo de atención**: El modo de atención describe la forma en que se presta el servicio, incluyendo el número de servidores disponibles y si estos trabajan de manera independiente o coordinada. Puede tratarse de atención individual, en paralelo o en etapas sucesivas.
- **Ingresos y salidas de clientes**: Este aspecto considera el flujo de clientes dentro del sistema, desde su ingreso hasta su salida tras recibir el servicio o abandonar la cola. El análisis de estos flujos permite evaluar el desempeño del sistema en términos de tiempos de espera, longitud de cola y nivel de servicio.

Ejemplos de sistemas de colas reales. Aplicaciones

URJC

Sistemas	Ejemplo	Clientes	Canales
Transporte	Puertos	Barcos	Muelles
	Aeropuertos	Aviones	Pistas
	Autopistas	Coches	Casillas de peaje
	Terminales	Buses	Plataforma
	Carga y descarga	Camiones	Islas
Computación	Procesamiento	Trabajos	CPU
	Impresión	Trabajos	Impresoras
Telefonía	Centrales	Llamados	Líneas
Producción	Elaboración	Productos	Máquinas
	Reparación	Solicitudes	Operarios
	Control de calidad	Productos	Inspectores
Atención al público	Bancos	Personas	Cajeros
	Supermercados	Personas	Cajeros
	Estaciones de servicio	Coches	Surtidores
	Negocios	Personas	Empleados
	Atención médicos	Pacientes	Ambulancias
	Alquiler de coches	Personas	Empleados
Administración	Dpto. compras	Solicitudes	Compradores
	Juzgados	Causas	Jueces

El **proceso de nacimiento y muerte** es un tipo particular de proceso estocástico en tiempo continuo que modela la evolución del número de elementos en un sistema mediante transiciones entre estados consecutivos. Este tipo de proceso constituye la base matemática de muchos modelos clásicos de teoría de colas.
Sea $\{N(t), t \geq 0\}$ el número de clientes en el sistema en el instante t. El proceso es de nacimiento y muerte si sólo son posibles las siguientes transiciones:

$$n \longrightarrow n+1 \quad \text{(nacimiento)}$$

$$n \longrightarrow n-1 \quad \text{(muerte)}$$

donde:

- La tasa de nacimiento en el estado n es λ_n.
- La tasa de muerte en el estado n es μ_n.

Estas tasas representan, respectivamente, la llegada de nuevos clientes al sistema y la finalización del servicio. Este proceso es una **cadena de Markov en tiempo continuo**, ya que el futuro del sistema depende únicamente del estado actual y no de la historia previa.

En régimen estacionario, i.e., bajo estabilización de las variables del sistema; las probabilidades de que hayan n clientes en el sistema, $p(n) = \lim_{t\to\infty} \mathbb{P}(N(t) = n)$ satisfacen las ecuaciones de equilibrio:

$$\lambda_{n-1}p(n-1) + \mu_{n+1}p(n) = (\lambda_n + \mu_n)p(n)$$

para $n \geq 1$, junto con:

$$\mu_1 p(1) = \lambda_0 p(0).$$

A partir de estas relaciones se obtiene de forma recursiva:

$$p(n) = \left(\prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}} \right) p(0).$$

El valor de $p(0)$ se determina mediante la condición de normalización:

$$\sum_{n=0}^{\infty} p(n) = 1.$$

Para que exista régimen estacionario es necesario que la serie anterior converja. Para este fin, o bien se limita el largo de la cola, o en muchos modelos prácticos esto se traduce en la condición $\rho < 1$ donde ρ representa la utilización del sistema (menos clientes entrantes que la capacidad de procesamiento).
Muchos de los modelos fundamentales de teoría de colas pueden formularse como procesos de nacimiento y muerte, debido a que el número de clientes en el sistema cambia únicamente por llegadas y salidas.

En el un modelo con llegadas y salidas que se comporten como un proceso de Poisson, con una sola estación de servicio (modelo $M/M/1$), se tendrá que

- Las llegadas siguen un proceso de Poisson con tasa constante λ, por lo que $\lambda_n = \lambda$.
- El tiempo de servicio es exponencial con tasa μ, por lo que $\mu_n = \mu$ para $n \geq 1$.

Las probabilidades estacionarias resultan:

$$P_n = (1-\rho)\rho^n, \quad \text{donde } \rho = \frac{\lambda}{\mu}.$$

Por otro lado, cuando existen m servidores $(M/M/m)$, la tasa de servicio depende del número de clientes en el sistema:

$$\mu_n = \begin{cases} n\mu, & n < m, \\ m\mu, & n \geq m. \end{cases}$$

Este modelo describe sistemas con múltiples canales de servicio, típicos en centrales telefónicas, servidores de red o estaciones base.
Si bajo las condiciones anteriores, la capacidad del sistema está limitada a K clientes, $(M/M/m/K)$, se tiene:

$$\lambda_K = 0,$$

lo que significa que nuevas llegadas son rechazadas cuando el sistema está lleno. Este modelo es especialmente relevante para el dimensionamiento de buffers en routers y dispositivos de red.

Una gran parte del espectro de los sistemas de colas no pueden resolverse de manera exacta, en dichos casos se recurre a la simulación. Sin embargo un subgrupo, bajo ciertas hipótesis, tiene solución exacta. En este capítulo se estudiarán la mayoría de modelos de colas cuya solución analítica es conocida. Para ellos se definirán los principales parámetros y variables de un sistema de colas.

La disciplina de cola se encarga de establecer el acceso de las tareas a los diferentes recursos. Los tipos de disciplinas más utilizados son:

- FIFO (*First-In-First-Out*): Los clientes se atienden de acuerdo a su orden de llegada a la cola.
- LIFO (*Last-In-First-Out*): Los clientes se atienden en orden inverso al que llegan a la cola.
- SIRO (Service In Random Order): Los clientes son atendidos de forma aleatoria.
- SIFO (*Shortest-In, First-Out*): Los clientes con servicios más cortos se atienden primero.
- RR (*Round Robin*): Se atiende a varios clientes a la vez segmentando sus tareas.

Definición (Notación de Kendall)

La forma más usual de representar los sistemas de colas es mediante la notación de Kendall

$$A/B/m/K/N/Z$$

- A: Distribución de la variable aleatoria tiempo entre llegadas (población finita).
- B: Distribución de la variable aleatoria tiempo de servicio.
- m: Número de servidores en el sistema.
- K: Capacidad del sistema, ∞ si no se especifica.
- N: Tamaño de la población, ∞ si no se especifica.
- Z: Disciplina de gestión de cola, FIFO si no se especifica.

Papel de las distribuciones exponenciales:

Las principales distribuciones que se emplean en el modelado de sistemas de colas son:

- M: Distribución exponencial, ver Figura 42.

$$f(x) = \lambda e^{-\lambda x}$$

- D: Distribución determinista (constante).

$$f(x) = \delta(x-a)$$

- U: Distribución uniforme, ver Figura 43.

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{resto} \end{cases}$$

- E_k: Distribución Erlang de parámetro k, ver Figura 44.

$$f(x) = \frac{\lambda^k x^{k-1} e^{-\lambda k}}{(k-1)!}$$

- H_k: Distribución hiperexponencial de parámetro k, ver Figura 45.

$$f(x) = \sum_{i=1}^{k} \alpha_i \lambda_i e^{-\lambda_i x}, \quad \sum_{i}^{k} \alpha_i = 1$$

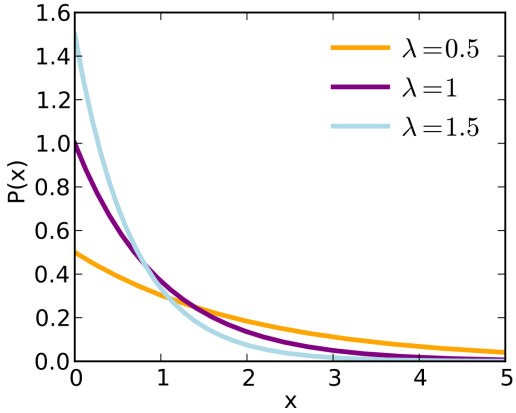


Figura 42: Distribución Exponencial

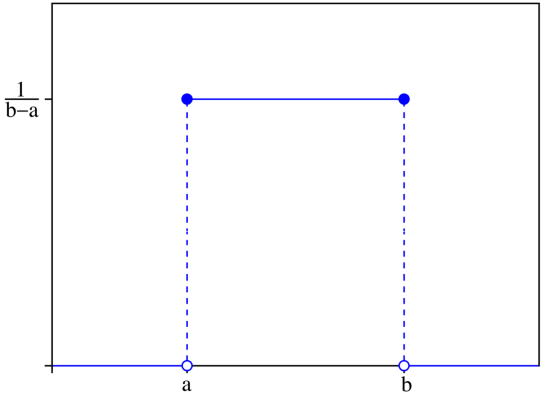


Figura 43: Distribución Uniforme

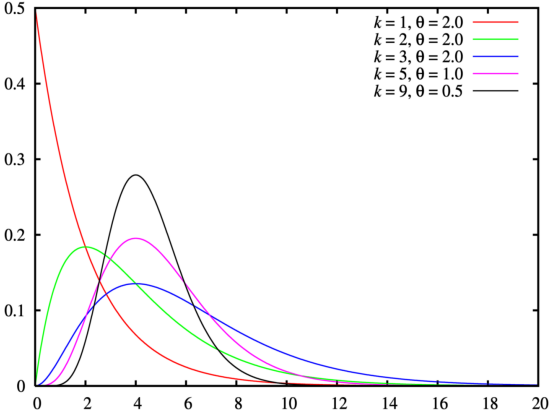


Figura 44: Distribución Erlang

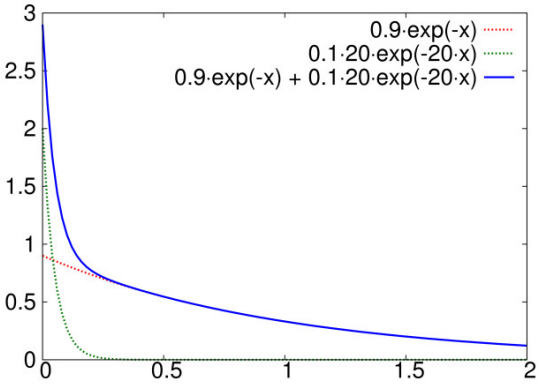


Figura 45: Distribución Hiperexponencial

Importante (Régimen de trabajo)

El régimen de trabajo se refiere a la fase en la cual se encuentra el sistema. Existen dos regímenes de trabajo:

- **Régimen transitorio:** Cuando el valor de las variables de un sistema depende de las condiciones iniciales del mismo, decimos que nos encontramos en **régimen transitorio**, ver Figura 46.
- **Régimen permanente**: En este curso nos **centraremos** en el **régimen permanente**, es decir, cuando las variables han alcanzado un estado estable, ver Figura 46.

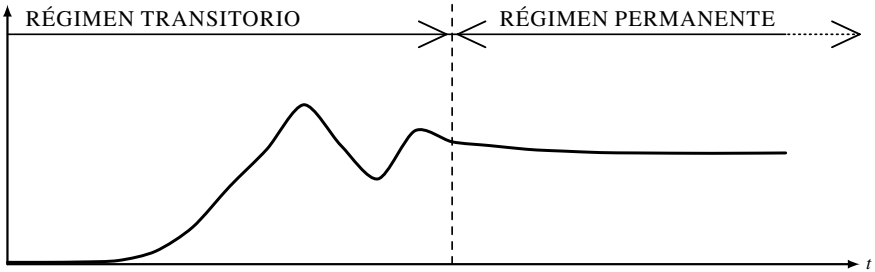


Figura 46: Diferencia de regímenes

Nomenclatura

En general vamos a emplear la siguiente nomenclatura

- λ : Número promedio de clientes que arriban al sistema por unidad de tiempo (tasa promedio de arribos)
- T_a : Intervalo promedio entre arribos de clientes (llamado también tiempo promedio de arribos)

$$T_a = \frac{1}{\lambda}$$

- μ : Número promedio de clientes que puede atender un canal por unidad de tiempo (velocidad promedio de atención).
- T_s : Duración promedio del servicio (o Tiempo promedio de atención)

$$T_s = \frac{1}{\mu}$$

- ρ : Factor de tránsito (o de tráfico)

$$\rho = \frac{\lambda}{\mu}$$

- n: Estado del sistema (es la cantidad de clientes que hay dentro del sistema en un momento determinado).
- $p(n)$: Probabilidad de que el sistema se encuentre en el estado n.

- K: Capacidad del sistema (cantidad máxima de clientes que puede haber en el sistema, tanto en los canales como en la cola).
- m: Cantidad de canales dispuestos en paralelo en la unidad operativa de atención.
- H: Número promedio de canales activos
- $\mathrm{L_C}$: Longitud promedio de la cola (número esperado de clientes esperando ser atendidos).

$$\mathrm{L_C} = \sum_{n=\mathrm{m}+1}^{\mathrm{K}} (\mathrm{n}-\mathrm{m}) \cdot \mathrm{p(n)}$$

- L: Longitud promedio del sistema (número esperado de clientes dentro del sistema).

$$\mathrm{L} = \sum_{n=0}^{K} \mathrm{n} \cdot \mathrm{p(n)}$$

Además: $\mathrm{L} = \mathrm{L_C} + \mathrm{H}$

- λ_n : Tasa promedio de ingreso de clientes al sistema cuando el estado del mismo es n.
En términos generales:

$$\lambda_\mathrm{n} = \lambda \cdot \mathrm{p(i/n)}$$

- μ_n : Tasa promedio de egreso de clientes del sistema cuando el estado del mismo es n.
- $\bar{\lambda}$: Tasa promedio de ingreso de clientes al sistema. En forma general, tendremos que

$$\bar{\lambda} = \sum_{n=0}^{K} \lambda_n p(n)$$

- $\bar{\mu}$: Tasa promedio de egreso de clientes del sistema. En forma general:

$$\bar{\mu} = \sum_{n=0}^{K} \mu_n p(n)$$

También: $\bar{\mu} = \mu \cdot H$

- PA: Porcentaje de actividad del canal

$$\mathrm{PA} = \frac{\mathrm{H}}{\mathrm{M}}$$

- $\mathrm{W_C}$: Tiempo promedio de espera en cola de un cliente.

$$\mathrm{W_C} = \frac{\mathrm{L_C}}{\bar{\lambda}}$$

- W: Tiempo promedio de permanencia en el sistema de un cliente.

$$\mathrm{W} = \mathrm{W_C} + \mathrm{T_s}$$

o también:

$$\mathrm{W} = \frac{\mathrm{L}}{\bar{\lambda}}$$

La expresión anterior es la fórmula de Little, y establece que el tiempo promedio de residencia en el sistema está dado por la relación entre el número promedio de clientes que permanece en el mismo y el flujo de circulación de clientes que pasa por él.

- $\omega_C(t)$: Probabilidad de que un cliente tenga que esperar un tiempo superior a t en la cola

$$\omega_{\mathrm{C}}(\mathrm{t}) = \mathrm{p}\,(\mathrm{w_C} > \mathrm{t})$$

- $\omega(\mathrm{t})$: Probabilidad de que un cliente tenga que permanecer un tiempo superior a " t " en el sistema.

$$\omega(\mathrm{t}) = \mathrm{p}(\ \mathrm{W} > \mathrm{t})$$

- $\overline{\mathrm{R}}$: Tasa promedio de clientes rechazados (por capacidad finita o por impaciencia).

$$\overline{\mathrm{R}} = \lambda - \bar{\lambda}$$

- PR: Porcentaje de rechazos del sistema

$$\mathrm{PR} = \frac{\overline{\mathrm{R}}}{\lambda}$$

- $\overline{\mathrm{A}}$: Tasa promedio de clientes que abandonan el sistema sin recibir el servicio.

$$\bar{\lambda} = \bar{\mu} + \overline{\mathrm{A}}$$

- PAB: Porcentaje de abandonos del sistema

$$\mathrm{PAB} = \frac{\overline{\mathrm{A}}}{\bar{\lambda}}$$

- λ_{r} : Número promedio de veces por unidad de tiempo que un mismo cliente requiere un servicio en el sistema (aplicable a sistemas de población finita).
- $\mathrm{U}\left(\mathrm{oT}_{\mathrm{r}}\right)$: Tiempo promedio que transcurre para que un cliente solicite un servicio desde que sale del sistema (aplicable a sistemas de población finita).

$$\mathrm{U} = \frac{1}{\lambda_{\mathrm{r}}}$$

Nomenclatura

- N: Tamaño de la población
- J: Número promedio de clientes fuera del sistema (aplicable a población finita)
- D: Probabilidad de que un cliente que arriba al sistema tenga que esperar para recibir el servicio.
- x: Factor de servicio de un sistema de población finita

$$\mathrm{x} = \frac{\mathrm{T_S}}{\mathrm{T_S} + \mathrm{U}}$$

- F: Factor de eficiencia de un sistema de población finita

$$\mathrm{F} = \frac{\mathrm{T_S} + \mathrm{U}}{\mathrm{T_S} + \mathrm{U} + \mathrm{W_C}}$$

- α : Coeficiente de impaciencia promedio de la población de clientes del sistema.

Parámetros económicos

Para el análisis económico denotaremos

- c_e como el coste derivado de esperar en el sistema.
- c_s costo de atención de un canal a velocidad unitaria.
- c_L como el coste de cada lugar en la cola.
- u como el precio de cada servicio
- c_e como el coste derivado de mantener una unidad en el sistema, y
- c_c como el coste por unidad de tiempo cada canal.

En esta sección se estudian modelos con un solo servidor/canal $m = 1$.

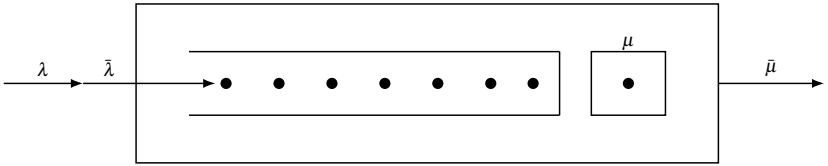


Figura 47: Modelo M/M/1

El primer modelo a estudiar es el M/M/1, bajo las siguientes **hipótesis:**

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de un solo canal de servicio.
- La capacidad del sistema es infinita.
- Los clientes no presentan impaciencia.
- La población es infinita.

- $p(n)$ probabilidad de que en el sistema haya n clientes (estado n del sistema).
- λ: tasa promedio de llegadas al sistema, el cliente llega, a espera de entrar o no.
- $T_a = \frac{1}{\lambda}$ intervalo medio entre llegadas
- μ: tasa promedio de salidas del sistema.
- $T_s = \frac{1}{\mu}$ duración promedio de servicios
- $m = 1$: un solo canal de atención (servidor).
- $\rho = \frac{\lambda}{\mu}$: factor de tránsito

Modelo M/M/1: Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes (estado n del sistema).
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- $\overline{\lambda}$: tasa promedio de ingresos de clientes al sistema, clientes entran al sistema.
- μ: tasa promedio de egresos de clientes del sistema, clientes salen del sistema.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de permanencia en el sistema.
- $\omega_c(t)$: probabilidad de espera mayor a t en la cola.
- $\omega(t)$: probabilidad de espera mayor a t en el sistema.

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \left(\frac{\lambda}{\mu}\right)^n p(0) = \rho^n p(0) = \rho^n(1-\rho)$$
$$p(n > k) = \rho^{k+1}$$
$$p(n \geq k) = \rho^k$$
$$\rho < 1 \leftarrow \lambda < \mu$$

- Determinación de L: la longitud del sistema es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = E[p(n)] = \sum_{n=0}^{\infty} n\rho^n(1-\rho) = \frac{\lambda}{\mu - \lambda}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-1$ clientes en cola.

$$L_c = L - \rho = \frac{\overline{\lambda}^2}{(\overline{\mu} - \overline{\lambda})\overline{\mu}}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= \rho \\ L &= L_c + H \\ PA &= \frac{H}{m} = \rho \text{ (Porcentaje de actividad)} \end{aligned}$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$ o $\overline{\mu}$ (Fórmula de Little)

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{\overline{\lambda}}{\overline{\mu}(\overline{\mu} - \overline{\lambda})}$$

- Determinación de W: el tiempo de permanencia en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención. Además, verifica la Fórmula de Little.

$$W = W_c + T_s$$
$$W = \frac{L}{\overline{\lambda}}$$

- Determinación de $\omega(t)$: la probabilidad de que un cliente tenga que permanecer en el sistema un tiempo superior a t, es decir, $\omega(t) = p(W > t)$

$$\omega(t) = p(W > t) = e^{-(\overline{\mu}-\overline{\lambda})t}$$

- Determinación de $\omega_c(t)$: la probabilidad de que un cliente tenga que permanecer en la cola un tiempo superior a t, es decir, $\omega_c(t) = p(W_c > t)$

$$\omega_c(t) = p(W_c > t) = \rho e^{-(\overline{\mu}-\overline{\lambda})t}$$

- La velocidad de media de atención, $\overline{\mu}$, es una variable de decisión.
- Podemos modificarla para obtener un balance entre el coste de mejorar el servicio y las pérdidas por mal funcionamiento.
- Definimos c_e como el coste derivado de esperar en el sistema, y
- c_s costo de atención de un canal a velocidad unitaria.
- Podemos definir una función de costes tal que:

$$Z = c_e L + c_s \overline{\mu}$$

- Minimizando la función anterior, obtenemos la velocidad de servicio óptima como:

$$\mu_0 = \overline{\lambda} + \sqrt{\overline{\lambda} \frac{c_e}{c_s}}$$

Ejemplo (Modelo M/M/1)

A la ventanilla de un banco llegan usuarios según un proceso de Poisson de tasa 5 usuarios por hora. Sabiendo que estos consumen un tiempo aleatorio cuya distribución puede suponerse exponencial de media $\frac{1}{6}$ de hora y que la disciplina de atención es FIFO. Se pide:

a) Calcular los valores medios de clientes en el sistema y en la cola.
Notemos que $\overline{\lambda} = 5$ usuarios/hora, y $T_s = \frac{1}{6} = \frac{1}{\overline{\mu}}$, por tanto $\overline{\mu} = 6$ usuarios/hora. Luego, se tendrá

$$L = \frac{\lambda}{\mu - \lambda} = 5$$

Por otro lado,

$$L_c = L - \rho = \frac{25}{6} \approx 4.1$$

Se concluye que el sistema tendrá un número medio de usuarios de 5 y una longitud de cola media de 4.1 usuarios.

b) ¿Qué porcentaje de los usuarios llega al sistema y lo encuentra desocupado?

$$p(0) = 1 - \rho = 1 - \frac{5}{6} = \frac{1}{6}$$

Aproximadamente **un sexto** de los usuarios encuentran desocupado el sistema.

c) Calcular el tiempo medio total de estancia en el centro de un usuario

$$W = \frac{L}{\lambda} = 1$$

Luego, el tiempo medio de un usuario en el sistema es de **una hora**.

d) Si en la sala de espera hay 4 sillas, ¿cuál es la probabilidad de que un usuario llegue al sistema y tenga que estar de pie?
Esto es equivalente a preguntar la probabilidad de que haya estrictamente más de 4 usuarios en el sistema, esto es

$$p(n > 4) = \rho^{k+1} = \rho^5 \approx 0.402$$

La probabilidad de que un usuario llegue al sistema y tenga que estar de pie es de aproximadamente un 40%.

e) Calcular los valores medios del tiempo de espera en cola y del tiempo de respuesta.

$$W_c = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{5}{6}$$

Por lo que se concluye que el tiempo de espera en cola medio es de 50 minutos, y el tiempo de respuesta medio es de 1 hora ($W = 1$).

f) ¿Cuál es la probabilidad de que un usuario espere más de una hora en ser atendido?

$$\omega(t)|_{t=1} = p(W_c > t) = \rho e^{-(\mu-\lambda)t} \approx 0.301$$

La probabilidad de que un usuario tenga que esperar más de una hora es de aproximadamente un 30%.

g) ¿Cuántas sillas se necesitarían para que un usuario al llegar al sistema tenga una probabilidad menor del 10% de esperar de pie?

En esta pregunta se nos indica calcular la cantidad de sillas k, necesarias para que la probabilidad de tener más usuarios que $k+2$ (k sillas en la cola, 1 usuario que llega y 1 usuario siendo atendido) sea menor que el 10%. Para ello empleamos

$$p(n > k+1) = \rho^{k+2} \leq 0.1$$

$$k \leq -\frac{1}{\log \rho} - 2$$

$$k \geq 10.6$$

Por lo tanto se necesitarían 11 sillas para que la probabilidad de un cliente de pie sea menor que el 10%

El siguiente modelo a estudiar es el M/M/1/K, ver figura 48, bajo las siguientes hipótesis.

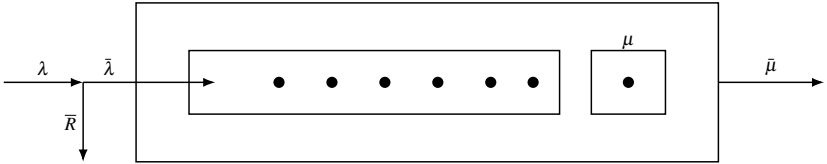


Figura 48: representación gráfica de un modelo M/M/1/K

- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes que llegan forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de un solo canal de servicio.
- **La capacidad del sistema está limitada a un valor** K (cola finita).
- Los clientes no presentan impaciencia.
- La población de clientes potenciales es infinita.

- λ: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = \frac{1}{\lambda}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- $m = 1$: número de servidores, en este caso 1.
- K: **capacidad del sistema**. Esto significa que no pueden haber más clientes que K, si hay más clientes, estos serán rechazados.

Modelo M/M/1/K: Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema (número esperado de clientes esperando dentro del sistema).
- L_c: longitud promedio de la cola (número esperado de clientes esperando ser atendidos).
- $\overline{\lambda}$: tasa promedio de ingreso de clientes al sistema.
- $\overline{\mu}$: tasa promedio de egreso de clientes del sistema.
- W_c: tiempo promedio de espera en cola.
- W: tiempo promedio de espera en el sistema.
- H: promedio de canales activos.
- $\overline{R}$: tasa promedio de clientes rechazados.
- PR: porcentaje de rechazo.

Importante

Para varias de las variables se cumplirá que la fórmula presenta una indeterminación del tipo $0/0$ en el caso $\rho = 1$. Para resolverla se aplicará la regla de L'Hòpital, con respecto a ρ.

- $p(n)$: el **estado de sistema** es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} \rho^n \frac{1}{K+1} & \text{Si } \rho = 1 \\ \rho^n \frac{1-\rho}{1-\rho^{K+1}} & \text{si no} \end{cases}$$

- L: la **longitud del sistema** representa el número esperado de clientes en el sistema, se deduce como el valor esperado de la v.a. número de clientes en el sistema

$$L = \begin{cases} \rho \frac{1-(K+1)\rho^K + K\rho^{K+1}}{(1-\rho)(1-\rho^{K+1})} & \rho \neq 1 \\ \frac{K}{2} & \rho = 1 \end{cases}$$

- **Longitud esperada de la cola** L_c: representa el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-1$ clientes en cola, clientes esperando ser atendidos.

$$L_c = L - (1 - p(0)) = \rho^2 \frac{1 - K\rho^{K-1} + K\rho^K}{(1-\rho)(1-\rho^{K+1})}$$

- H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= 1 - p(0) = L - L_c \\ L &= L_c + H \end{aligned}$$

- Porcentaje de actividad: $PA = \frac{H}{m} = 1 - p(0)$
- Tasa promedio de egresos $\overline{\mu} = \mu[1 - p(0)] = \mu \cdot H$.
  Los egresos se producen solo en los estados $n \neq 0$.
- Tasa promedio de ingresos $\overline{\lambda} = \lambda \cdot [1 - p(K)]$
  Se producirán ingresos al sistema mientras no haya K clientes, esto es $n \leq K$.
  Bajo condiciones de **equilibrio**, se tendrá que $\overline{\lambda} = \overline{\mu}$.

- W_c: El tiempo medio de espera en cola viene dado por la relación entre L_c y $\overline{\lambda}$ (o $\overline{\mu}$) (Fórmula de Little).

$$W_c = \frac{L_c}{\overline{\mu}} = \frac{L_c}{\mu\left(1 - \frac{1-\rho}{1-\rho^{K+1}}\right)} W_c \qquad = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{\lambda \cdot [1 - p(K)]} = \frac{L_c}{\lambda \cdot [1 - \rho^k p(0)]}$$

- W: El tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

- $\overline{R}$: el número promedio de clientes rechazados por unidad de tiempo es la tasa promedio de llegadas multiplicada por la probabilidad de no ingresar.

$$\overline{R} = \lambda p(K) = \lambda \rho^K \frac{(1-\rho)}{1-\rho^{K+1}}$$

$$\lambda = \overline{\lambda} + \overline{R}$$

- PR: el porcentaje de rechazos, serán aquellas llegadas que serán rechazadas por encontrarse el sistema ocupado se puede escribir como:

$$PR = \frac{\overline{R}}{\overline{\lambda}}$$

- En este tipo de sistemas se intentan balancear las perdidas por ingresos no percibidos y el coste asociado a la infraestructura de espera.
- Definimos c_L como el coste de cada lugar en la cola, y u como el precio de cada servicio.
- Variable de desición K, lugares habilitados para cola.
- Podemos definir una función de costes tal que:

$$Z = c_L(K-1) + u\overline{R} = c_L(K-1) + u\overline{\lambda}\rho^K \frac{1-\rho}{1-\rho^{K+1}}$$

- Problema de minimización en enteros.

Ejemplo (Ejemplo 2.4 de [Miranda, 2005] – Modelo M/M/1/10)

A un sistema M/M/1/10 llega un ciudadano cada 5 minutos, en promedio. Si no hay sillas vacías, el ciudadano se va. La duración media del servicio es de 6 minutos. Calcule:

a) Probabilidad de que un ciudadano no tenga que esperar para recibir el servicio.
Primeramente calculemos los parámetros del sistema

$$\lambda = \frac{1}{5}\text{cl/min} = 12\text{cl/h}$$
$$\mu = \frac{1}{6}\text{cl/min} = 10\text{cl/h}$$
$$\rho = \frac{\lambda}{\mu} = 1.2$$

Por lo que se tiene que la probabilidad de llegar y no tener que esperar es de

$$p(0) = \frac{1-\rho}{1-\rho^n} \approx 0.0310$$

b) Porcentaje de ocupación.
El porcentaje de ocupación del canal se calcula como

$$\text{PA} = \frac{H}{m} = H = 1 - p(0) \approx 0.9689 = 96.89\%$$

El porcentaje de ocupación será aproximadamente del 96.89%

c) Probabilidad de encontrar 3 o más sillas ocupadas.

$$p(n > 3) = 1 - p(0) - p(1) - p(2) \approx 0.8872$$
$$p(1) = \rho p(0) \approx 0.0372$$
$$p(2) = \rho^2 p(0) \approx 0.0446$$

Por tanto, la probabilidad de encontrar 3 o más sillas ocupadas es aproximadamente 4.46%

d) Probabilidad de que haya menos de 2 sillas ocupadas.

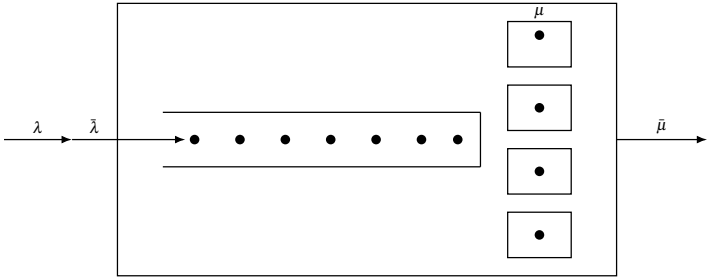
$$p(n \leq 1) = p(0) + p(1) \approx 0.0682$$

Por tanto, la probabilidad de que hayan menos de 2 clientes en el sistema es aproximadamente 6.82%

Modelo M/M/1/K: Ejemplo de aplicación

Sistemas de varios canales en paralelo: Modelo M/M/m

En esta sección se estudian modelos con más de un servidor/canal $m \geq 1$. El siguiente modelo a estudiar es el **M/M/m**, bajo las siguientes hipótesis.



- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- **Se dispone de m canales de servicio**, en paralelo, con la misma velocidad de atención.
- La capacidad del sistema es infinita.
- Los clientes no presentan impaciencia.
- La población es infinita.

Modelo M/M/m: Parámetros

- λ: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = \frac{1}{\lambda}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- m: número de servidores.

Modelo M/M/m: Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de espera en el sistema.
- $\omega_c(t)$: probabilidad de espera mayor a t en la cola.
- $\omega(t)$: probabilidad de espera mayor a t en el sistema.

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} \frac{\rho^n}{n!} p(0), & \text{si } n < m \\ \left(\frac{\rho}{m}\right)^n \frac{m^m}{m!} p(0), & \text{si } n \geq m \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} \frac{\rho^n}{n!} + \frac{\rho^m}{(m-1)!(m-\rho)} \right]^{-1}$$

$$p(n \geq k) = \frac{\rho^k}{(k-1)!(k-\rho)} p(0)$$

  Luego se debe cumplir que $\frac{\rho}{m} < 1$, para que el sistema esté en equilibrio y funcione sin colapsar.

- Determinación de L: la longitud del sistemas es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = E[p(n)] = \frac{\rho^m \mu \lambda}{(m-1)!(m\mu - \lambda)^2} p(0) + \rho$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-m$ clientes en cola.

$$L_c = L - \rho = \frac{\rho^m \mu \lambda}{(m-1)!(m\mu - \lambda)^2} p(0)$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= \rho \\ L &= L_c + H = L_c + \rho \\ PA &= \frac{H}{m} = \frac{\rho}{m} \text{ (Porcentaje de actividad de cada canal)} \end{aligned}$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y λ

$$W_c = \frac{L_c}{\lambda}$$

- Determinación de W: el tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención. Además, verifica la Fórmula de Little.

$$W = W_c + T_s$$
$$W = \frac{L}{\lambda}$$

- Determinación de $\overline{\mu}$, $\overline{\lambda}$: tasas promedio de ingresos y egresos del sistema.

$$\overline{\mu} = \mu \cdot H,$$
$$\overline{\lambda} = \lambda$$

- Determinación de $\omega(t)$: la probabilidad de que un cliente tenga que permanecer en el sistema un tiempo superior a t, es decir, $\omega(t) = p(W > t)$

$$\omega(t) = p(W > t) = e^{-\mu t}\left[\frac{1 + (m\rho)^m p(0)[1 - e^{\mu(m-1-m\rho)t}]}{m!(1-\rho)(m-1-m\rho)}\right]$$

- Determinación de $\omega_c(t)$: la probabilidad de que un cliente tenga que permanecer en la cola un tiempo superior a t, es decir, $\omega_c(t) = p(W_c > t)$

$$-\mu(m-\rho)t \quad m \quad (\rho)$$

- En este tipo de sistemas, el número de servidores m es la variable de decisión.
- Podemos modificarla para minimizar el coste operativo del sistema.
- c_e como el coste derivado de mantener una unidad en el sistema, y
- c_c como el coste por unidad de tiempo cada canal.
- Podemos definir una función de costes tal que:

$$z = c_e L + c_c m$$

- Problema de minimización en enteros.

Ejemplo (Ejemplo 3.1 de [Miranda, 2005] – Modelo M/M/4)

Un taller de reparaciones cuenta con cuatro técnicos (sistema M/M/4). En promedio, llegan pedidos cada 5 minutos y la duración media de las reparaciones es de 10 minutos. Determinar:

a) El número mínimo de operarios para que el sistema funcione.
Cálculo de parámetros

$$\lambda = \frac{1}{5}\frac{\text{cl}}{\text{min}} \cdot 60 \cdot \frac{\text{min}}{h} = 12\frac{\text{cl}}{h}$$
$$\mu = \frac{1}{10}\frac{\text{cl}}{\text{min}} \cdot 60 \cdot \frac{\text{min}}{h} = 6\frac{\text{cl}}{h}$$
$$\rho = \frac{\lambda}{\mu} = 2$$

Nótese que para que el sistema no colapse y entre en estado de equilibrio se tiene que dar que $\frac{\rho}{m} < 1$, por lo tanto a partir de 3 operarios serán suficientes para que el sistema funcione.

b) La probabilidad de que una pieza sea reparada conforme llega.
La probabilidad de que una pieza sea reparada al llegar es la probabilidad de tener 3 o menos clientes en el sistema, esto es

$$p(n \leq 3) = p(0) + p(1) + p(2) + p(3) = 1 - p(n \geq 4) \approx 0.8261$$

Por tanto, la probabilidad de que una pieza sea reparada al llegar es aproximadamente de 82.61%.

c) Porcentaje de ocupación de los técnicos.
El porcentaje de ocupación de los técnico es el porcentaje de actividad del canal, esto es

$$PA = \frac{H}{m} = \frac{\rho}{m} = \frac{1}{2}$$

Por tanto, el porcentaje de ocupación de los técnicos es del 50%.

d) Probabilidad de que haya menos de dos piezas en el sistema.

$$p(n < 2) = p(0) + p(1) = (1 + \rho)p(0) \approx 0.3912$$

Por tanto, la probabilidad de que haya menos de dos piezas en el sistema es aproximadamente 39.12%.

e) Número medio de piezas en el taller.

$$L \approx 2.1739$$

Por tanto la media de piezas en el sistema es de aproximadamente 2.17.

f) Número promedio de piezas a la espera de ser reparadas.

$$L_c = L - \rho = 0.1739$$

Por tanto el promedio de piezas a la espera de ser reparadas es de 0.1739.

g) Ganancias esperadas, si cada reparación se cobra a 50 €.
Los ingresos esperados por hora serán el promedio de ingresos (egresos) por hora multiplicados por la ganancia, esto es

$$\text{Ingreso} = u \cdot \overline{\lambda} = 50 \cdot 12 = 600€/h$$

Por tanto los ingresos esperados por hora son de 600 €.

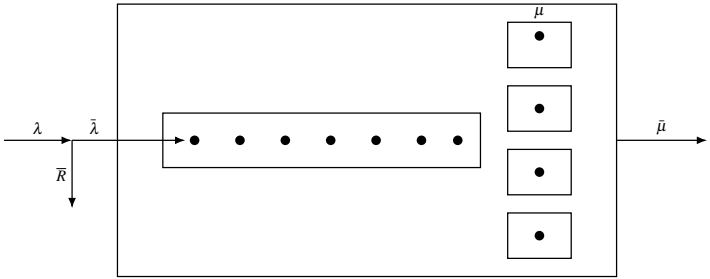
h) Tiempo promedio de permanencia de una pieza en el taller.
El tiempo promedio de permanencia de la pieza en el taller se calcula como

$$W = W_c + T_s = \frac{L_c}{\lambda} + \frac{1}{\mu} \approx 0.1812h$$

Por tanto, el tiempo promedio de permanencia de una pieza en el taller es de aproximadamente 0.1812 horas.

Modelo M/M/m/K

El siguiente modelo a estudiar es el **M/M/m/K**, bajo las siguientes hipótesis.



- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- **Se dispone de m canales de servicio**, en paralelo, con la misma velocidad de atención.
- **La capacidad del sistema está limitada a un valor K.**
- Los clientes no presentan impaciencia.
- La población es infinita.

Modelo M/M/m/K: Parámetros

- λ: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = \frac{1}{\lambda}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- m: número de servidores.
- K: capacidad del sistema.

Modelo M/M/m/K: Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de espera en el sistema.
- $\overline{R}$: tasa promedio de clientes rechazados.

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} \frac{\rho^n}{n!} p(0) & \text{Si } n < m \\ \left(\frac{\rho}{m}\right)^n \frac{m^m}{m!} p(0) & \text{Si } m \leq n \leq K \end{cases}$$

$$p(0) = \left[\frac{\rho^m}{(m-1)!} \left(1 - \left[\frac{\rho}{m}\right]^{K-m+1} \right) \frac{1}{(m-\rho)} + \sum_{n=0}^{m-1} \frac{\rho^n}{n!} \right]^{-1}$$

- Determinación de L: la longitud del sistema es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$\begin{aligned} L &= \sum_{n=0}^{K} n p(n) \\ &= \frac{\frac{\rho}{m} \cdot \rho^m \cdot p(0)}{m! \cdot \left(1 - \frac{\rho}{m}\right)^2} \cdot \left[1 - \left(\frac{\rho}{m}\right)^{K-m} - (K-m) \cdot \left(\frac{\rho}{m}\right)^{K-m} \cdot \left(1 - \frac{\rho}{m}\right) \right] + \rho \cdot \left[1 - \left(\frac{\rho}{m}\right)^K \cdot \frac{K^K}{m!} \cdot p(0) \right] \end{aligned}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-m$ clientes en cola.

$$\begin{aligned} L_c &= \sum_{n=m}^{K} (n-m)p(n) \\ &= \frac{\frac{\rho}{m} \cdot \rho^m}{m! \cdot \left(1-\frac{\rho}{m}\right)^2} \cdot \left[1 - \left(\frac{\rho}{m}\right)^{K-m} - (K-m) \cdot \left(\frac{\rho}{m}\right)^{K-m} \cdot \left(1-\frac{\rho}{m}\right)\right] \cdot p(0) \end{aligned}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$\begin{aligned} H &= \rho\,[1-p(K)] \\ PA &= \frac{H}{m} \text{ (Porcentaje de actividad de cada canal)} \end{aligned}$$

- Determinación de $\overline{\mu}$, $\overline{\lambda}$: tasas promedio de ingresos/egresos del sistema.

$$\overline{\mu} = \mu H,$$
$$\overline{\lambda} = \lambda[1 - p(K)]$$

- Determinación de W_c: El tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$ o $\overline{\mu}$.

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{\lambda(1 - p(K))}$$

- Determinación de W: El tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

- Determinación de $\overline{R}$: el número promedio de clientes rechazados por unidad de tiempo es la tasa promedio de llegadas multiplicada por la probabilidad de no ingresar.

$$\overline{R} = \lambda p(K) = \lambda - \overline{\lambda}$$

Modelo M/M/m/K: Análisis Económico

- Si elegimos el número de canales disponibles m como variable a optimizar, definimos
  - c_e como el coste de esperar una unidad de tiempo en el sistema.
  - c_c el coste por unidad de tiempo de cada canal.

$$\min(Z) = c_e L + c_c m$$

- Si elegimos el tamaño de la cola $K - m$ como variable a optimizar, definimos
  - c_L como el coste de cada plaza de la cola.
  - u precio de una unidad de servicio.

$$\min(Z) = c_L(K - m) + u\overline{R}$$

- Si elegimos la velocidad de servicio μ como variable a optimizar
  - c_s costo de atención de un canal a velocidad unitaria

$$\min(Z) = c_s M\mu + u\overline{R}$$

Ejemplo (Ejemplo 3.4 de [Miranda, 2005] – Sistema M/M/2/4)

Un taller mecánico cuenta con 2 técnicos y una capacidad de 4 coches, cada 5 minutos llega un coche. Si no hay huecos vacíos, el coche se va. La duración media de las consultas es de 10 minutos. Calcule:

a) Probabilidad de que un coche no tenga que esperar para ser atendido.

$$p(n \leq 1) = p(0) + p(1) = \frac{1}{3}$$
$$p(0) = \frac{1}{9}$$
$$p(1) = \rho p(0) = \frac{2}{9}$$

Por tanto, la probabilidad de no tener que esperar para ser atendido es de aproximadamente un 33%.
Nota: al calcular $p(0)$, algunas calculadoras pueden dar 0 (lo que es imposible en este caso). La razón es que $m = \rho$ produce una indeterminación. Para calcular $p(0)$ hay que resolverla analíticamente.

b) Porcentaje de ocupación.

$$PA = \frac{H}{m} = \frac{\rho[1-p(K)]}{m} = \frac{7}{9}$$

Por lo que se concluye que el porcentaje de ocupación es de aproximadamente 77%.

c) Probabilidad de que un coche no pueda ingresar.
Esto es, la probabilidad de que el sistema esté lleno, $n = K$

$$p(4) = \frac{2}{9}$$

Por tanto, la probabilidad de que un coche no pueda entrar es de aproximadamente el 22%.

d) Tasa promedio de coches que no son atendidos.
Esto es la tasa de rechazos $\overline{R}$

$$\overline{R} = \lambda p(K) = \frac{8}{3} \text{ cl/h}$$

Por tanto, la tasa de rechazos será de aproximadamente $2,666$ coches por hora.

e) Número promedio de clientes esperando ser atendidos.

$$L_c = \sum_{n=m}^{K} (n-m)p(n) = p(3) + 2p(4) = \frac{2}{3} \text{ coches}$$

Por tanto, el número promedio de coches esperando ser atendidos es de aproximadamente 0.666.

f) Número promedio de coches en el taller.

$$L = L_c + H = \frac{2}{3} + 2[1 - p(4)] = \frac{20}{9} \text{ coches}$$

Por tanto, el número de clientes siendo atendidos es de aproximadamente $2,22$.

g) Facturación esperada, si cada servicio se cobra a 50 €.

$$\text{Ingreso} = u \cdot \overline{\lambda} = u\lambda[1 - p(K)] \approx 466,68 \text{ €}/h$$

Por tanto la facturación esperada es de aproximadamente $466,68$ €$/h$.

h) Tiempo promedio de espera en cola.

$$W_c = \frac{L_c}{\overline{\lambda}} \approx 0.0714\ h$$

i) Si el cliente es aceptado por el taller a las 9h, ¿a qué hora esperaría irse con su coche reparado?

$$W = W_c + T_s \approx 0.2381\ h = 14.286\ \text{min}$$

Por tanto, el cliente espera poder irse aproximadamente a las $9:14$ horas.

j) Pérdida de facturación, si cada reparación se factura a 50 €.

$$\text{Loss} = u \cdot \overline{R} \approx 133.33\ €/h$$

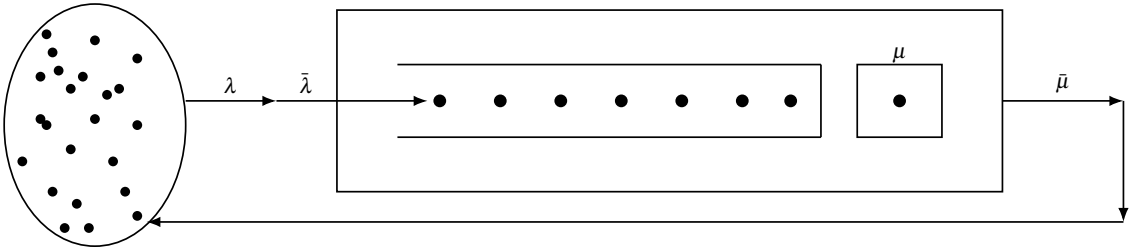
Si la población fuente de clientes de un sistema de colas es finita pero lo suficientemente grande como para que la tasa media de arribos no se vea afectada de manera apreciable por la entrada o salida de un cliente del sistema, el modelo puede analizarse, en la práctica, como si la población fuese infinita.
En cambio, cuando la fuente de usuarios es reducida y la llegada de un cliente o la finalización de un servicio modifica sensiblemente la probabilidad de arribos futuros, se trata de un sistema con población finita, que requiere un tratamiento específico.
En estos sistemas, los clientes atendidos no abandonan definitivamente el conjunto de potenciales usuarios, sino que regresan a la población fuente del sistema.
Para el análisis de sistemas con población finita, la observación no se centra en el flujo global de llegadas, sino en el comportamiento individual de cada cliente. Con este fin se introduce el parámetro T_r (frecuentemente denotado por la letra "U"), que representa el tiempo medio entre dos arribos consecutivos de un mismo cliente al sistema; es decir, el tiempo medio que un cliente permanece fuera del sistema.

Se define entonces λ_r como el inverso de T_r. Este parámetro expresa la frecuencia con la cual un usuario solicita el servicio, es decir, el número medio de veces por unidad de tiempo que un cliente, estando fuera del sistema, ingresa al mismo para recibir atención.
Si se denota por N el tamaño total de la población (número de clientes tanto dentro como fuera del sistema), todo arribo necesariamente proviene de los $N - n$ clientes que no se encuentran en el sistema en ese instante.
A continuación se formulará el modelo correspondiente al caso en que no existen restricciones de capacidad. No obstante, el mismo esquema de modelización puede adaptarse a sistemas con capacidad finita, limitando el número de estados posibles, de forma análoga a lo estudiado previamente para esos sistemas.

Los sistemas de población finita se definen sobre los siguientes supuestos.

- Si la población de un sistema es lo suficientemente grande, la tasa de llegadas no se ve afectada cuando aparece una llegada o una salida, como en los casos anteriores.
- En caso contrario, hay que tener en cuenta las siguientes consideraciones:
  - Las observaciones se hace sobre cada cliente, en vez de sobre la población.
  - Definiremos el parámetro T_r o U, como el tiempo entre llegadas para un mismo cliente.
  - Llamaremos N al tamaño de la población, es decir, al número de clientes dentro y fuera del sistema.

Un sistema **M/M/1/-/N**, se basa en las siguientes hipótesis.



- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de **un solo canal de servicio**.
- La **capacidad del sistema es infinita**.
- Los clientes no presentan impaciencia.
- La **población es finita y relativamente pequeña**.

Modelo M/M/1/-/N: Parámetros

- λ_r: frecuencia de requerimientos de servicios por el cliente. También podemos referirnos como intervalo medio entre requerimientos: $T_r = U = \frac{1}{\lambda_r}$.
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas: $T_s = \frac{1}{\mu}$.
- m: número de servidores, en este caso 1.
- N: número total de clientes, tamaño de la población.

Modelo M/M/1/-/N: Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- P_{in}: porcentaje de clientes que se encuentran en el sistema.
- L_c: longitud promedio de la cola.
- P_q: porcentaje de clientes que se encuentran en la cola.
- H: promedio de canales activos.
- P_{active}: porcentaje de clientes que están siendo atendidos.
- P_{out}: porcentaje de clientes que están fuera del sistema.
- W: tiempo promedio de espera en el sistema.
- W_c: tiempo promedio de espera en cola.
- J: número promedio de clientes fuera del sistema.

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \frac{N!}{(N-n)!}\rho_r^n p(0)$$
$$p(0) = \frac{1}{\sum_{n=0}^{N} \frac{N!}{(N-n)!}\rho_r^n}$$

- Determinación de L: la longitud del sistemas es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = E[p(n)] = \sum_{0}^{\infty} np(n) = N - \frac{1-p(0)}{\rho_r}$$

- El porcentaje de clientes que se encuentran en el sistema viene dado por

$$P_{in} = \frac{L}{N}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n-1$ clientes en cola.

$$L_c = N - \frac{(1+\rho_r)[1-p(0)]}{\rho_r}$$

- El porcentaje de clientes que se encuentran esperando en la cola viene dado por

$$P_{queue} = \frac{L_c}{N}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media representa el número de clientes atendiéndose en el sistema.

$$H = 1 - p(0)$$
$$P_{active} = \frac{H}{N} \text{ Porcentaje de clientes siendo atendidos}$$
$$PA = \frac{H}{m} = 1 - p(0) \text{(Porcentaje de actividad)}$$

- Determinación de J: el número promedio de usuarios que se encuentran fuera del sistema, viene dado por:

$$J = N - L$$

$$J = \frac{H}{\rho_r}$$

- El porcentaje de clientes que se encuentran fuera del sistema viene dado por la expresión

$$P_{out} = \frac{J}{N}$$

- Determinación de $\overline{\mu}/\overline{\lambda}$: tasas promedio de ingresos/egresos del sistema.

$$\overline{\mu} = \mu[1 - p(0)] = \overline{\lambda} = \lambda_r J$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$(o $\overline{\mu}$).

$$W_c = \frac{L_c}{\overline{\lambda}} = \frac{L_c}{J\lambda_r} = \frac{L_c}{\mu[1 - p(0)]} = \frac{L_c}{\overline{\mu}}$$

- Determinación de W: el tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

- Como hay un solo canal, la velocidad de media de atención, $\overline{\mu}$, se escoge usualmente como variable de decisión.
- Podemos modificarla para obtener un balance entre el coste de mejorar el servicio y las perdidas por mal funcionamiento.
- Definimos c_e como el coste derivado de esperar en el sistema, y c_s como el coste por atender a una velocidad unitaria.
- Podemos definir una función de costes tal que:

$$\min(Z) = c_e L + c_s \overline{\mu}$$

Ejemplo (Ejemplo 4.1 de [Miranda, 2005] – modelo M/M/1/-/5)

En un sistema M/M/1/-/5, cada cliente permanece en promedio 0.5 horas fuera del sistema. La duración media del sistema es de 20 minutos. Determinar:

a) Probabilidad de que el sistema se encuentre ocupado.
Cálculo de parámetros

$$U = 0.5 \text{ h/pedido} \Rightarrow \lambda_r = 2 \text{ pedidos}/h$$
$$\mu = 3 \text{ pedidos}/h$$
$$\rho_r = \frac{2}{3}$$

Las probabilidades de cada estado resultan:
$p(0) = 0.0142, \quad p(1) = 0.0473, \quad p(2) = 0.1261, \quad p(3) = 0.2524, \quad p(4) = 0.3362, \quad p(5) = 0.2241$
Por tanto la probabilidad de que el sistema esté ocupado es de

$$p(1) + p(2) + p(3) + p(4) + p(5) = 1 - p(0) \approx 0.9858$$

aproximadamente un 98%.

b) Porcentaje de clientes que se encuentran fuera del sistema.

$$P_{out} = \frac{J}{N} = \frac{N-L}{N} = 1 - \frac{1}{N}\left(N - \frac{1-p(0)}{\rho_r}\right) \approx 0.2958$$

o usando $J = 1.4789$, se concluye igualmente que el porcentaje de clientes fuera del sistema es de aproximadamente 30%.

c) Cantidad de clientes que se atienden en el sistema por hora, en promedio.

$$\overline{mu} = \mu \cdot H = \mu(1-p(0)) \approx 2.957$$

Por tanto, se atienden de media aproximadamente 3 clientes por hora.

d) Longitud promedio de la cola.

$$L_c = N - \frac{(1+\rho_r)(1-p(0))}{\rho_r} \approx 2.5353$$

Por tanto, la longitud media de la cola es de aproximadamente 2.5353 clientes.

e) Tiempo de espera promedio de un cliente en cola.

$$W_c = \frac{L_c}{J \cdot \lambda_r} \approx 0.8573\ h$$

Por tanto, el tiempo de espera promedio de un cliente en cola es de $0.8573\ h$.

f) Tiempo promedio de permanencia de un cliente dentro del sistema.

$$W = W_c + T_s \approx 1.19\ h$$

Por tanto, el tiempo promedio de permanencia de un cliente en cola es de $1.19\ h$.

Tarea (Ejemplo 4.2 de [Miranda, 2005] – modelo M/M/1/-/5)

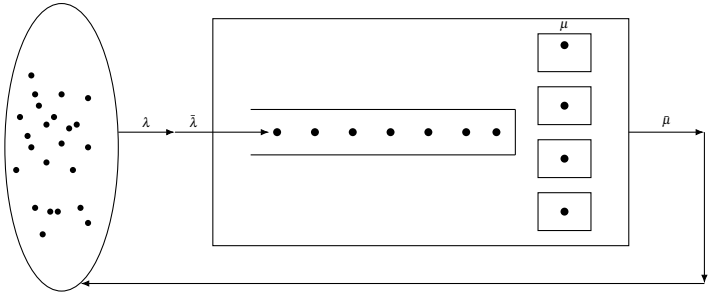
Un CPD tiene disponibles 4 racks de servidores idénticos, de los cuales están en funcionamiento 3 y el cuarto está de backup, es decir, entra en funcionamiento cuando uno de los anteriores sufre algún problema. Sabiendo que los servidores fallan, en promedio, cada 12 horas de funcionamiento y que existe un equipo de reparación que tardas 3 horas en arreglar los desperfectos. Tenga en cuenta que si hay más de dos servidores fuera de servicio el CPD debe desconectarse.

a) Deducir y calcular las probabilidades asociadas a cada uno de los estados posibles.

b) Determinar la longitud promedio de cola de servidores averiados en esperas de ser reparados .

c) Sabiendo que el beneficio que produce una máquina en funcionamiento es de 1500€, calcular las pérdidas esperadas.

Un sistema M/M/m/-/N, se basa en las siguientes hipótesis.



- El proceso de llegadas es de tipo Poisson.
- El proceso de salidas es de tipo Poisson.
- El sistema se encuentra en régimen permanente.
- Los clientes forman una cola simple.
- La disciplina de cola es FIFO.
- Se dispone de m **canales de servicio** en paralelo, con la misma velocidad de atención.
- La capacidad del sistema es infinita (cola infinita).
- Los clientes no presentan impaciencia.
- La **población es finita** y relativamente pequeña.

Modelo M/M/m/-/N: Parámetros

- λ_r: tasa promedio de llegadas al sistema. También podemos referirnos como intervalo medio entre llegadas $T_a = U = \frac{1}{\lambda_r}$
- μ: tasa promedio de salidas del sistema. También podemos referirnos como intervalo medio entre salidas $T_s = \frac{1}{\mu}$
- m: número de servidores.
- N: número total de clientes.

Modelo M/M/m/-/N: Variables

- $p(n)$: probabilidad de que en el sistema haya n clientes.
- L: longitud promedio del sistema.
- L_c: longitud promedio de la cola.
- W_c: tiempo promedio de espera en cola.
- H: promedio de canales activos.
- W: tiempo promedio de espera en el sistema.
- J: número promedio de clientes fuera del sistema
- D: probabilidad de esperar para recibir servicio.
- X: factor de servicio.
- F: factor de eficiencia.

- Determinación de $p(n)$: el estado de sistema es una V.A que representa la probabilidad de que en el sistema haya n clientes.

$$p(n) = \begin{cases} C_N^n \rho_r^n p(0) = \frac{N!}{(N-n)!n!} \rho_r^n p(0) \text{ Si } n < m \\ \frac{n! m^m}{m! m^n} C_N^n \rho_r^n p(0) \text{ Si } m \leq n \leq N \end{cases}$$

$$p(0) = \left[\sum_{n=0}^{m-1} C_N^n \rho_r^n + \sum_{n=m}^{N} C_N^n \frac{n! m^m}{m! m^n} \rho_r^n \right]^{-1}$$

Recordemos que C_N^n denota el número combinatorio

$$C_N^n = \frac{N!}{n!(N-n)!}$$

- Determinación de L: la longitud del sistemas es una V.A cuya media representa el valor esperado de que en el sistema haya n clientes.

$$L = \sum_{n=0}^{m-1} n C_N^n \rho_r^n p(0) + \sum_{n=m}^{N} n C_N^n \frac{n! m^m}{m! m^n} \rho_r^n p(0)$$

- El porcentaje de clientes que se encuentran en el sistema viene dado por

$$P_{in} = \frac{L}{N}$$

- Determinación de L_c: el estado de sistema es una V.A cuya media representa el valor esperado de que en el sistema haya $n - m$ clientes en cola.

$$L_c = \sum_{n=m}^{N} (n-m)p(n) = \sum_{n=m}^{N} (n-m) \cdot C_N^n \cdot \frac{n!}{m!} \cdot \frac{m^m}{m^n} \cdot \rho_r^n \cdot p(0)$$

- El porcentaje de clientes que se encuentran esperando en la cola viene dado por

$$P_{queue} = \frac{L_c}{N}$$

- Determinación de H: el número promedio de canales activos es una V.A cuya media es el número de clientes atendiéndose en el sistema.

$$H = \sum_{n=1}^{m-1} np(n) + \sum_{n=m}^{N} mp(n)$$

$$P_{active} = \frac{H}{N}$$ Porcentaje de clientes siendo atendidos

$$PA = \frac{H}{m}$$ Porcentaje de actividad

- Determinación de J: el número promedio de usuarios que se encuentran fuera del sistema, viene dado por:

$$J = N - L$$

$$J = \frac{H}{\rho_r}$$

- El porcentaje de clientes que se encuentran fuera del sistema viene dado por la expresión

$$P_{out} = \frac{J}{N}$$

- Determinación de W_c: el tiempo medio de espera en cola viene dado por la relación entre la L_c y $\overline{\lambda}$ (o $\overline{\mu}$).

$$W_c = \frac{L_c}{\overline{\overline{\lambda}}} = \frac{L_c}{J\lambda_r}$$

- Determinación de $\overline{\mu}/\overline{\lambda}$: tasas promedio de ingresos y egresos del sistema.

$$\overline{\mu} = \mu \cdot H = \overline{\lambda} = \lambda_r J$$

- Determinación de W: el tiempo de espera en el sistema es igual al tiempo de espera en cola más el tiempo promedio de atención.

$$W = W_c + T_s$$

$$W = \frac{L}{\overline{\overline{\lambda}}} = \frac{L}{J\overline{\lambda_r}}$$

- Si el tamaño de N es demasiado grande ($N > 10$), es conveniente usar las aproximaciones enunciadas por Peck y Hazelwood. Para ello se definen el factor de servicio X y el factor de eficiencia F, que cumplen

$$
\begin{aligned}
X &= \frac{T_s}{T_s + U} \\
F &= \frac{T_s + U}{T_s + U + W_c} \\
H &= FNX \\
J &= FN(1 - X) \\
L_c &= (1 - F)N
\end{aligned}
$$

- En este tipo de sistemas, el número de servidores es la variable de decisión.
- Podemos modificarla para minimizar el coste operativo del sistema.
- Definimos c_e como el coste derivado de esperar en el sistema, y c_c como el coste de atención de un servidor.
- Podemos definir una función de costes tal que:

$$Z = c_e L + c_c m$$

- Problema en enteros positivos.

Ejemplo (Ejemplo 4.4 de [Miranda, 2005] – M/M/m/-/N)

Una empresa cuenta con 5 máquinas idénticas que necesitan supervisión, en promedio, cada 2 horas. Las operaciones de supervisión tienen una duración promedio de 1 hora. La empresa cuenta con dos operarios para realizar esta actividad. Cada máquina proporciona una ganancia esperada de 500€ por hora.

a) Deducir y calcular las probabilidades asociadas a cada uno de los estados posibles.
Calculamos los parámetros

$$\rho_r = \frac{\lambda_r}{\mu} = \frac{0.5}{1} = 0.5$$

Para calcular las probabilidades empleamos Matlab, para obtener

$$p(1) = 0.2763$$
$$p(2) = 0.2763$$
$$p(3) = 0.2073$$
$$p(4) = 0.1036$$
$$p(5) = 0.0260$$

b) Determinar, L, L_c, H y J.

$$L = 0 \cdot p(0) + 1 \cdot p(1) + 2 \cdot p(2) + 3 \cdot p(3) + 4 \cdot p(4) + 5 \cdot p(5) = 1.9948$$
$$L_c = 1 \cdot p(3) + 2 \cdot p(4) + 3 \cdot p(5) = 0.4922$$
$$H = L - Lc = 1,5026$$
$$J = N' - L = 3,0052$$

c) Determinar el factor de servicio, el factor de eficiencia y el porcentaje de actividad de cada trabajador.

$$X = \frac{T_S}{T_S + U} = 0.3333$$
$$\bar{\mu} = \mu \cdot H = 1.5026$$

$$W_C = \frac{L_C}{\bar{\mu}} = 0.3276$$
$$F = \frac{T_S + U}{T_S + U + W_C} = 0.9016$$
$$PA = \frac{H}{M} = 0.7513$$

d) Calcular las pérdidas.

$$\mathrm{LC} = 500 \cdot \ \mathrm{L} = 997,41 \frac{€}{\mathrm{h}}$$

En los sistemas de colas analizados hasta el momento se ha supuesto que cada sistema ofrecía un único tipo de servicio, prestado en un único centro de atención compuesto por uno o varios canales dispuestos en paralelo, con determinadas características operativas tales como capacidad, modalidad de atención y disciplina de servicio.

No obstante, en numerosos sistemas reales coexisten varios centros de atención dentro de un mismo sistema, en los cuales se brindan distintos servicios que, en muchos casos, se encuentran vinculados de manera secuencial. En estas configuraciones, un cliente que ingresa al sistema recibe un primer servicio en un centro de atención (o subsistema) y, una vez finalizado, se dirige a un segundo centro para recibir un nuevo servicio, continuando este proceso hasta completar la secuencia y abandonar el sistema.
Esta organización de centros de atención conectados secuencialmente se denomina disposición **en serie** o **en cascada**.

Dentro de las redes en serie podemos diferenciar dos tipos: sistemas secuenciales, ver Figura 49 y sistemas paralelos, ver Figura, 50.



Figura 49: Sistema en serie

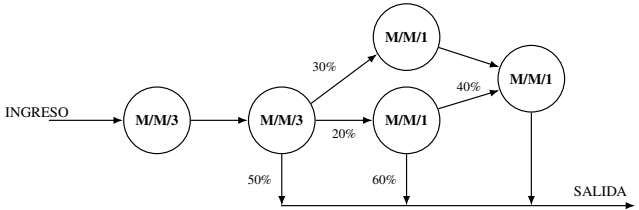


Figura 50: Sistema en paralelo

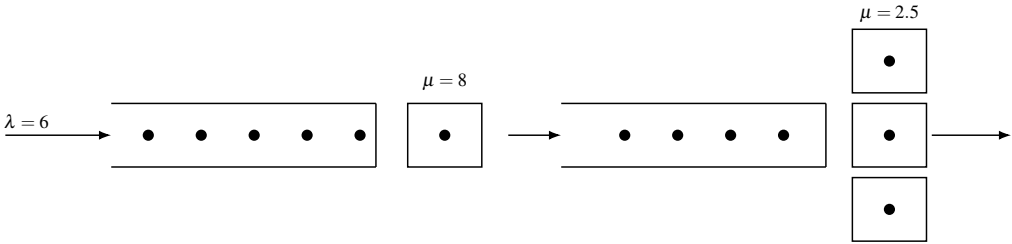


Figura 51: Recepción de pacientes

Ejemplo (Ejemplo aplicación redes secuenciales)

Supongamos una sala de urgencias, en donde los pacientes deben formar una cola única a la espera de recibir asignación en uno de 3consultorios, ver Figura 51. En promedio la llegada de pacientes se realiza a razón de 6 por hora. El recepcionista tarda una 0.125 horas en llamar a un paciente, y el tiempo medio de tratamiento en cada uno de los consultorios es de 0.4 horas. Obtener los parámetros básicos del sistema.

- Subsistema I

$$\rho_1 = 0.75 \qquad L_1 = 3$$
$$H_1 = 0.75 \qquad W_{c1} = 0.375$$
$$p(0)_1 = 0.25 \qquad W_1 = 0.5$$
$$L_{c1} = 2.25$$

Para el segundo subsistema, la tasa de llegadas es igual a la tasa efectiva de salidas del primer subsistema, es decir, $\lambda_2 = \overline{\mu_1} = \mu_1 \cdot H = 6$

- Subsistema II

$$\rho_2 = 2.4 \qquad L_2 = 4.99$$
$$H_2 = 2.4 \qquad W_{c2} = 0.4315$$
$$p(0)_2 = 0.0562 \qquad W_2 = 0.8315$$
$$L_{c2} = 2.59 \qquad \overline{\mu_2} = \mu_2 H_2 = 6$$

- Sistema completo

$$H = H_1 + H_2 = 3.15 \qquad L = L_1 + L_2 = 7.99$$
$$p(0) = p(0)_1 p(0)_2 = 0.01405 \qquad W_c = W_{c1} + W_{c2} = 0.8065$$
$$L_c = L_{c1} + L_{c2} = 4.84 \qquad W = W_1 + W_2 = 1.3315$$

El trabajo con sistemas en paralelo se basa en los siguientes principios.

Teorema (Teorema de Burke)

La división de un flujo de clientes de un proceso de Poisson, probabilísticamente lleva a procesos que también son de tipo Poisson, ver Figura 52.

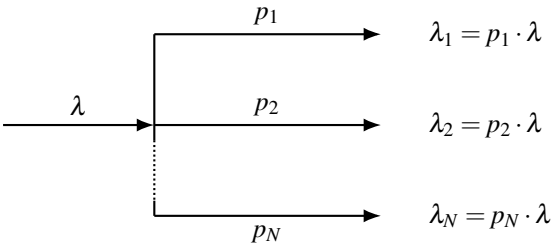


Figura 52: División de flujos de Poisson.

Proposición

La combinación de flujos independientes de clientes de procesos de Poisson lleva a un proceso de naturaleza poissoniana, ver Figura 53.

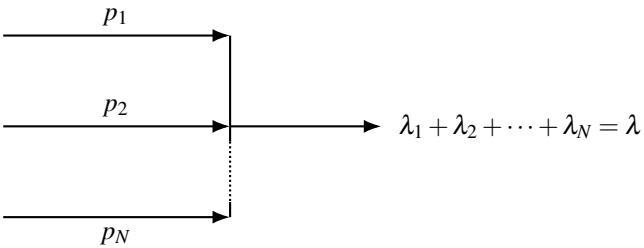


Figura 53: Combinación de flujos de Poisson.

Teorema (Teorema de Jackson)

Para un sistema con m centros de atención, la probabilidad de que el sistema se encuentre en el estado $n_1, n_2, \ldots, n_m$ viene dada por:

$$p(n_1, n_2, \ldots, n_m) = \prod_{i=1}^{m} p_i(n_i)$$

Proposición (Equilibrio entre ingresos y egresos)

En condiciones de equilibrio, las tasa efectivas de ingresos y egresos son iguales ($\overline{\mu} = \overline{\lambda}$). Por tanto, se cumple:

$$\overline{\lambda} = \mu \cdot H = \frac{1}{T_s} \cdot H$$

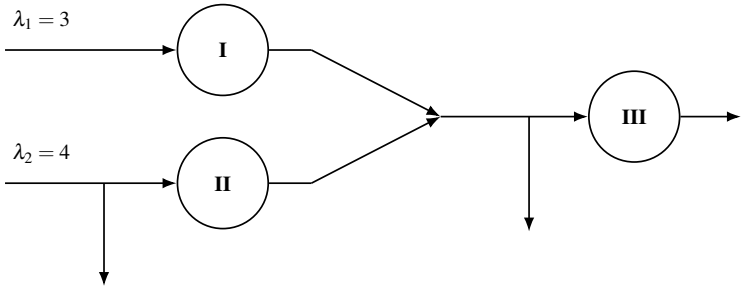
$$T_s = \frac{H}{\overline{\lambda}}$$

Teorema (Fórmula de Little)

Los sistemas de colas cumplen la fórmula de Little

$$W = \frac{L}{\overline{\lambda}}$$

$$W_c = \frac{L_c}{\overline{\lambda}}$$



Ejemplo

Considere el sistema anterior, donde la estación 1 es M/M/1 con $\mu_1 = 6$, la estación 2 es M/M/2/2 con $\mu_2 = 2$ y la estación 3 es M/M/1/2 con $\mu_3 = 5$, estando los valores de λ y μ expresados en usuarios por día, sabiendo que cada servicio ofrecido en la estación III se cobran 100€ y que cada canal tiene un coste de 50€ día, calcular la ganancia esperada diaria.

Para resolver este problema tenemos que encontrar la tasa efectiva de clientes que dejan el sistema $\overline{\mu_3}$, lo cual indicará la facturación, luego como el sistema tiene en total $m = m_1 + m_2 + m_3 = 4$ canales, tendrá un coste operativo de 200 €/día.

Notar que como el nodo 3 es M/M/1/2, se deduce que $\overline{\mu_3} = \overline{\lambda_3} = \lambda_3 \cdot [1 - p_3(2)]$. Luego, la estructura del problema indica que se tendrá que encontrar $\lambda_3 = \overline{\mu_1} + \overline{\mu_2}$ y $p_3(2)$.

- Estación 1 − M/M/1, $\lambda_1 = 3$, $\mu_1 = 6$.
  - Dado que no hay restricciones de capacidad $\overline{\mu_1} = \overline{\lambda_1} = \lambda_1 = 3$
- Estación 2 − M/M/2/2, $\mu_2 = 2$.
  - $p_2(0) = 0.2$, $p_2(1) = 0.4$ y $p_2(2) = 0.4$.
  - $\overline{\mu_2} = \mu_2 \cdot [1 - p_2(2)] = 2.4$
- Estación 3 − M/M/1/2, $\mu_3 = 5$.
  - $\lambda_3 = \overline{\mu_1} + \overline{\mu_2} = 5.4$.
  - $p_3(0) = 0.308$, $p_3(1) = 0.3326$ y $p_3(2) = 0.3593$.
  - $\overline{\mu_3} = 3.4598$
- Luego se concluye que la tasa de clientes atendidos en el sistema 3, y que por tanto, dejan el sistema es, $\overline{\mu_3} = 3.4598$.
  - Beneficio $B = 100€/\text{día} \cdot \overline{\mu_3} - 50€/\text{día} \cdot m \cdot = 146$ €/día.

Las redes con realimentación (también denominado como redes con reciclaje) responden a la siguiente situación, ver Figura 54:

- En algunos sistemas reales ocurre que parte de los clientes atendidos en una determinada unidad del sistema tienen que retornar a una unidad anterior, para ser atendidos nuevamente.
- En estos casos, es necesario tener en cuenta el flujo de clientes que van por la rama realimentada.
- Es importante notar que, siendo los procesos de llegadas de tipo Poisson, el flujo resultante se comporta como si fuera de Poisson, aunque no lo es estrictamente hablando.

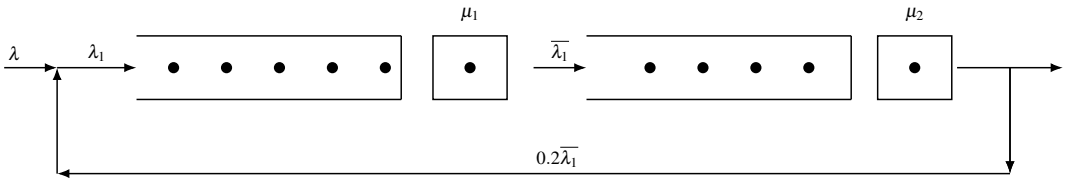


Figura 54: Ejemplo de red con retroalimentación

Ejemplo (Red simple con realimentación)

Por ejemplo, supongamos un sistema de elaboración de un producto en una máquina, al que llegan piezas con una tasa $\overline{\lambda}$ para ser procesadas a una velocidad $\overline{\mu_1}$. Una vez fabricadas, las piezas son sometidas a un control de calidad, que se realiza a una velocidad promedio de $\overline{\mu_2}$. Un 20% de las piezas inspeccionadas requiere ser reprocesadas, por lo que vuelven a la cola de la máquina 1, ver Figura 54.

La filosofía para la resolución de este tipo de sistemas es considerar que la tasa de entrada a los sistemas que reciben la realimentación es una combinación de la tasa media con el flujo que se recibe de otros nodos. En este ejemplo específico se trata con dos sistemas $M/M/1$, por lo que se tendrá que

$$\overline{\lambda_1} = \overline{\mu_1} = \lambda_1.$$

Luego para el segundo subsistema, se tendrá de manera análoga:

$$\overline{\lambda_2} = \overline{\mu_2} = \overline{\lambda_1} = \lambda_1.$$

- Tasa efectiva de entrada al primer subsistema

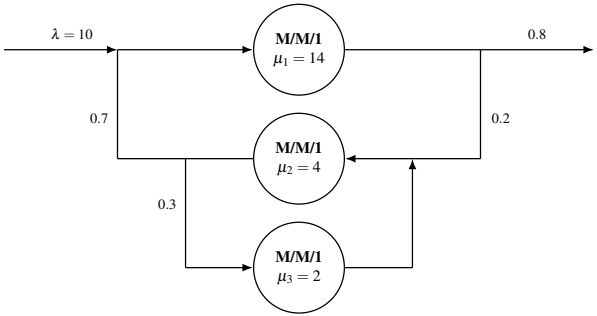
$$\lambda_1 = \lambda + 0.2\lambda$$
$$\lambda_1 = \frac{\lambda}{0.8}$$

- Centro 1

$$\rho_1 = \frac{\lambda_1}{\mu_1} \qquad L_1 = \frac{\overline{\lambda}}{\mu_1 - \overline{\lambda}} \qquad H_1 = \rho_1$$
$$W_{c1} = \frac{L_{c1}}{\lambda_1}$$
$$p(n)_1 = \rho_1^n(1-\rho_1) \qquad L_{c1} = \frac{\lambda^2}{\overline{\mu_1}(\overline{\mu_1} - \lambda_1)} \qquad W_1 = W_{c1} + T_{s1}$$

- Centro 2

$$\rho_2 = \frac{\lambda_1}{\overline{\mu_2}} \qquad L_2 = \frac{\lambda_1}{\mu_2 - \lambda_1} \qquad H_2 = \rho_2$$
$$W_{c2} = \frac{L_{c2}}{\lambda_1}$$
$$p(n)_2 = \rho_2^n(1-\rho_2) \qquad L_{c2} = \frac{\lambda_1^2}{\mu_2 \cdot (\mu_2 - \lambda_1)} \qquad W_2 = W_{c1} + T_{s2}$$



Ejemplo (Sistema con realimentación en paralelo)

Se tienen tres subsistemas

- Subsistema 1: M/M/1 – $\mu_1 = 14$
- Subsistema 2: MM1 – $\mu_1 = 4$
- Subsistema 3: M/M/1 – $\mu_3 = 2$

Los datos están expresados en clientes/hora. Determinar cuántos clientes salen del sistema, calcular la longitud promedio de clientes en cada centro, y encontrar la probabilidad de que el sistema esté vacío.

Solución: Al sistema ingresan en media $\lambda = 10$ clientes por hora. De acuerdo a los flujos de clientes, se tendrá

Centro 1:

$$\lambda_1 = \lambda + 0.7$$

$$\overline{\lambda_1} = \overline{\mu_1} = \lambda_1$$

Centro 2:

$$\lambda_2 = \overline{\mu_3} + 0.2\overline{\mu_1}$$

$$\overline{\lambda_2} = \overline{\mu_2} = \lambda_2$$

Centro 3:

$$\lambda_3 = 0.3\overline{\mu_2}$$

$$\overline{\lambda_3} = \overline{\mu_3} = \lambda_3$$

Por tanto las ecuaciones de **balance de flujo**, resultan

$$\begin{cases} \lambda_1 = \lambda + 0.7\lambda_2 \\ \lambda_2 = 0.2\lambda_1 + \lambda_3 \\ \lambda_3 = 0.3\lambda_2 \end{cases}$$

Resolviendo este sistema se obtiene:

$$\lambda_1 = 12.5, \qquad \lambda_2 = 3.5714, \qquad \lambda_3 = 1.0714$$

Luego se tiene que, del sistema salen

$$\overline{\mu} = 0.8\overline{\mu_1} = 10$$

La **longitud promedio de cada subsistema** será

$$L_1 = 8.3333, \qquad L_2 = 8.3327, \qquad L_3 = 1.1538$$

clientes.
La **probablidad de que el sistema esté vacío** se obtiene aplicando el Teorema de Jackson,

$$p(0,0,0) = p(0)_1 \cdot p(0)_2 \cdot p(0)_3 = (1-\rho_1)(1-\rho_2)(1-\rho_3) = 0.0053$$

donde se ha empleado que

$$\rho_1 = 0.8929, \qquad \rho_2 = 0.8929, \qquad \rho_3 = 0.5357$$

En resumen:

- $\overline{\mu} = 10$ clientes/hora,
- $L_1 = 8.333$, $L_2 = 8.3327$ y $L_3 = 1.1538$ clientes.
- $p(vacio) = 0.0053$.

En el estudio clásico de la teoría de colas, gran parte de los resultados analíticos se apoyan en la suposición de **llegadas y tiempos de servicio exponenciales**. Esta hipótesis conduce a modelos matemáticamente manejables (como $M/M/1$, $M/M/m$, etc.) gracias a la propiedad de **falta de memoria** de la distribución exponencial, que permite modelar el sistema como una **cadena de Markov**.

Sin embargo, en muchos sistemas reales de telecomunicaciones, esta suposición no es realista.

El tráfico de datos, la duración de las conexiones, el tamaño de los paquetes, el comportamiento de los usuarios y los tiempos de procesamiento en dispositivos de red presentan con frecuencia **variabilidad, asimetría y dependencia temporal** que no pueden describirse adecuadamente mediante distribuciones exponenciales.

Por ello, es fundamental extender el análisis hacia **modelos con distribuciones generales**, que permiten capturar de manera más fiel el comportamiento de los sistemas reales.

En redes de telecomunicaciones se observan fenómenos como:

- Tráfico **autocorrelacionado** (por ejemplo, tráfico de video o descargas masivas).
- Tiempos de servicio con **baja variabilidad** (procesamiento casi determinista en routers).
- Tiempos de servicio con **alta variabilidad** (procesamiento variable en servidores, retransmisiones, colisiones).
- Distribuciones **pesadas en la cola** (heavy–tailed), típicas en tamaños de archivos o sesiones de usuario.

Estos comportamientos producen desviaciones significativas respecto a las predicciones de los modelos $M/M/1$ o $M/M/c$. En particular, la variabilidad de los tiempos de servicio impacta directamente en:

- El tiempo medio de espera.
- La longitud media de la cola.
- La probabilidad de congestión.

Notación general: modelos G/G/1

Para describir sistemas con distribuciones arbitrarias se utiliza la notación de Kendall extendida:

$$A/B/c$$

donde:

- A describe la distribución de los tiempos entre llegadas.
- B describe la distribución de los tiempos de servicio.
- c es el número de servidores.

La letra G (General) indica que no se asume ninguna forma específica para la distribución.
Así, el modelo más general de un solo servidor es:

$$G/G/1$$

que representa un sistema con llegadas y servicios de distribución arbitraria.

Importancia del coeficiente de variación

En modelos no exponenciales, la forma exacta de la distribución es menos relevante que su **media** y su **variabilidad**. Esta última se mide mediante el **coeficiente de variación**:

$$C_X = \frac{\sigma_X}{\mathbb{E}[X]}$$

donde X puede representar tiempos entre llegadas o tiempos de servicio.

- Para la exponencial: $C_X = 1$.
- Para la determinista: $C_X = 0$.
- Para distribuciones muy variables: $C_X > 1$.

Este parámetro resulta clave porque el rendimiento del sistema depende fuertemente de la variabilidad, incluso cuando la media se mantiene constante.

Dado que el modelo $G/G/1$ no admite una solución exacta sencilla, se utilizan aproximaciones. La más importante es la **fórmula de Kingman** para el tiempo medio de espera en cola:

$$W_q \approx \frac{\rho}{1-\rho} \cdot \frac{C_A^2 + C_S^2}{2} \cdot \mathbb{E}[S]$$

donde:

- $\rho = \lambda \mathbb{E}[S]$ es la utilización del sistema.
- C_A es el coeficiente de variación de los tiempos entre llegadas.
- C_S es el coeficiente de variación de los tiempos de servicio.
- $\mathbb{E}[S]$ es el tiempo medio de servicio.

Esta expresión muestra de forma explícita que **la variabilidad influye tanto como la carga del sistema**.

Casos particulares relevantes I

Algunos modelos intermedios muy utilizados en telecomunicaciones son:

- $M/G/1$: llegadas Poisson y servicios generales.
- $G/M/1$: llegadas generales y servicios exponenciales.
- $D/G/1$: llegadas deterministas y servicios generales.
- $G/D/1$: servicios deterministas.

Por ejemplo, en un router que procesa paquetes a velocidad casi constante, el modelo $M/D/1$ describe mejor el comportamiento que el $M/M/1$.
El uso de modelos no exponenciales permite:

- Dimensionar buffers con mayor precisión.
- Estimar retardos reales en enlaces y routers.
- Analizar efectos de tráfico multimedia y tráfico bursty.
- Evaluar impacto de protocolos que introducen variabilidad (retransmisiones, backoff, etc.).

Casos particulares relevantes II

En muchos casos, ignorar la variabilidad conduce a **subestimar los retardos y la congestión**, lo que resulta crítico en aplicaciones sensibles al tiempo como VoIP, videoconferencia y servicios en tiempo real.
Los modelos con distribuciones no exponenciales representan una herramienta esencial para analizar sistemas de telecomunicaciones modernos. Aunque su tratamiento matemático es más complejo, el uso de métricas como el coeficiente de variación y aproximaciones como la fórmula de Kingman permite obtener resultados prácticos y aplicables en el diseño y evaluación de redes reales.

Referencias I

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004).
Convex Optimization.
Cambridge University Press, Cambridge.
https://web.stanford.edu/~boyd/cvxbook/.

[Hillier and Lieberman, 2015] Hillier, F. S. and Lieberman, G. J. (2015).
Introducción a la investigación de operaciones.
McGraw-Hill.

[Miranda, 2005] Miranda, M. (2005).
Teoría de colas.
Primera Edición, Educa.

[Taha, 2012] Taha, H. A. (2012).
Investigación de operaciones.
Pearson Educación, México, 9 edition.

Investigación Operativa

Grado En Ingeniería en Sistemas de Telecomunicación
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Telemática

Javier Quintero

Slides – Material docente en abierto de la Universidad Rey Juan Carlos
Repositorio Institucional de la Universidad Rey Juan Carlos (BURJC Digital)
https://burjcdigital.urjc.es/home

2025 – 2026 v1.02.00

Este trabajo se distribuye bajo la licencia "Creative Commons **Attribution – Share alike**"
diponible en https://creativecommons.org/licenses/by-sa/4.0/deed.es

Colección de problemas Investigación Operativa

Grado En Ingeniería en Sistemas de Telecomunicación
Grado en Ingeniería en Tecnologías de la Telecomunicación
Grado en Ingeniería en Telemática

Material docente en abierto de la Universidad Rey Juan Carlos
Disponible en el Repositorio Institucional de la Universidad Rey Juan Carlos (BURJC Digital)

`https://burjcdigital.urjc.es/home`

Javier Quintero

2025 – 2026 v1.02.00

Hojas de Problemas

1 Programación lineal en variable continua **2**

2 Programación lineal con variable entera **6**

3 Transporte y Asignación **14**

4 Programación en redes **18**

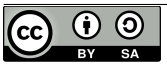
5 PERT-CPM **24**

6 Programación no lineal **25**

7 Teoría de colas **27**

8 Exámenes resueltos **30**
8.1 Primer Examen Parcial 30
8.2 Segundo Examen Parcial 33

9 Soluciones seleccionadas **38**
Soluciones de problemas de Programación Lineal 38
Soluciones problemas de programación lineal en enteros 48
Soluciones problemas de programación lineal en enteros 48



Copyright © 2026. Algunos derechos reservados. Este trabajo se distribuye bajo la licencia "Creative Commons **Attribution – Share alike"** avaible at `https://creativecommons.org/licenses/by-sa/4.0/deed.es`.

1. Programación lineal en variable continua

Problema 1.1. Dado el siguiente problema de programación lineal, hallar el máximo y el mínimo de la función objetivo:

1.
$$\begin{array}{lll} Z = & 2x_1 + 2x_2 & \\ \text{s.t.} & x_1 + 2x_2 & \geq 4 \\ & 6x_1 + 5x_2 & \leq 30 \\ & x_1, x_2 & \geq 0 \end{array}$$

2.
$$\begin{array}{lll} Z = & 5x_1 - 3x_2 & \\ \text{s.t.} & x_1 \quad + x_2 & \leq 3 \\ & 2x_1 \quad + x_2 & \geq 8 \\ & x_1, x_2 & \geq 0 \end{array}$$

Problema 1.2.
Resuelve los siguientes problemas de programación lineal de forma analítica, gráfica y utilizando el solver de Matlab. (las soluciones deben ser iguales).

1.
$$\begin{array}{lll} \max & x_1 + x_2 & \\ \text{s.t.} & x_1 + x_2 & \leq 2 \\ & x_1, x_2 \geq 0 & \end{array}$$

2.
$$\begin{array}{lll} \max & 2x_1 + 5x_2 & \\ \text{s.t.} & 4x_1 + 52x_2 & \leq 20 \\ & x_1, x_2 \geq 0 & \end{array}$$

3.
$$\begin{array}{lll} \max & x_1 + x_2 & \\ \text{s.t.} & x_1 & \leq 4 \\ & x_2 - 4 & \leq 0 \\ & x_1, x_2 \geq 0 & \end{array}$$

4.
$$\begin{array}{lll} \max & 8x_1 + 10x_2 & \\ \text{s.t.} & x_1 + 2x_2 & \leq 20 \\ & x_1 + x_2 & \leq 15 \\ & x_1, x_2 \geq 0 & \end{array}$$

5.
$$\begin{array}{lll} \max & 3x_1 + 5x_2 & \\ \text{s.t.} & x_1 - 2x_2 & \leq 3 \\ & x_1 - 7x_2 & \geq 0 \\ & x_1, x_2 \geq 0 & \end{array}$$

6.
$$\begin{array}{lll} \max & x_1 - 2x_2 & \\ \text{s.t.} & x_1 - 7x_2 & \geq 0 \\ & 2x_1 - 4x_2 & \leq 6 \\ & x_1, x_2 \geq 0 & \end{array}$$

Problema 1.3. Dado el siguiente problema de programación lineal

$$\begin{array}{rrl} \min Z = & 4x_1 + x_2 & \\ \text{s.t.} & x_1 + 2x_2 & \leq 3 \\ & 4x_1 + 3x_2 & \geq 6 \\ & 3x_1 + x_2 & = 3 \\ & x_1, x_2 \geq 0 & \end{array}$$

1. Dibuja la región factible con geogebra, y halla la solución de forma gráfica.
2. Aplica el solver de Matlab correspondiente y hallar la solución.

Problema 1.4. Usar el método gráfico para encontrar todas las soluciones óptimas del siguiente modelo:

$$\begin{array}{lll} \max Z = & 500x_1 + 300x_2 & \\ \text{s.t.} & 15x_1 + 5x_2 & \leq 300 \\ & 10x_1 + 6x_2 & \leq 240 \\ & 8x_1 + 12x_2 & \leq 450 \\ & x_1, x_2 \geq 0 & \end{array}$$

Problema 1.5. Utilizar el método gráfico para demostrar que el siguiente modelo no tiene soluciones factibles:

$$\begin{array}{rrl} \max Z & = 5x_1 + 7x_2 & \\ \text{s.t.} & 2x_1 - x_2 & \leq -1 \\ & -x_1 + 2x_2 & \leq -1 \\ & x_1, x_2 \geq 0 & \end{array}$$

Problema 1.6. Considérese el siguiente problema, donde el valor de k todavía no se ha establecido.

$$\begin{array}{lll} \max & x_1 + 2x_2 & \\ \text{s.t.} & -x_1 + x_2 & \leq 2 \\ & -x_2 & \leq 3 \\ & kx_1 + x_2 & \leq 2k + 3, \text{where } k \geq 0 \\ & x_1, x_2 \geq 0 & \end{array}$$

La solución que se usa por ahora es $x_1 = 2$, $x_2 = 3$. Utilizar el análisis gráfico para determinar los valores de k tales que esta solución sea de hecho óptima.

Problema 1.7. Para fabricar dos tipos de caja de cartón una fábrica vende a 0.12€ la unidad, la caja de tipo cuadrada y a 0.08€ la de base rectangular. Para el primer tipo se necesita emplear 2 m de cinta adhesiva y 0.5 m de rollo de cartón. Para las de base rectangular se emplean 4 m de cinta adhesiva y 0.25 m de rollo de cartón. Si la fábrica dispone de 440 m de cinta adhesiva y de 65 m de rollo de cartón, se pide:

1. Elaborar una tabla con la información.
2. Hallar el número de cajas de cada tipo que deben fabricarse para que el valor de la producción sea máximo. Representar a escala la región factible del problema y resolverlo usando el método de las rectas de nivel.

Problema 1.8. Se dispone de 600 gr de un determinado fármaco para elaborar pastillas grandes y pequeñas. Las grandes pesan 40 gr y las pequeñas 30 gr. Se necesitan al menos tres pastillas grandes, y al menos el doble de pequeñas que de las grandes. Cada pastilla grande proporciona un beneficio de 2 € y la pequeña de 1 €.

1. ¿Cuántas pastillas se han de elaborar de cada clase para que el beneficio sea máximo?

2. Plantear el modelo de optimización e indicar todos los elementos que intervienen. Representar gráficamente.

Problema 1.9. Un ganadero debe suministrar un mínimo diario de 4 mg de vitamina A y 6 mg de vitamina B en el pienso que da a sus reses. Para ello, dispone de dos tipos de pienso P_1 y P_2, cuyos contenidos vitamínicos en mg por kg son los siguientes:

	A	B
P_1	2	6
P_2	4	3

Si el pienso P_1 cuesta 4 céntimos por 100 g y el pienso P_2 cuesta 6 céntimos por 100 g. ¿Cómo deben mezclarse los piensos para suministrar las vitaminas requeridas a un coste mínimo?

Problema 1.10. Un comerciante acude a cierto mercado a comprar naranjas con 500€. Le ofrecen 2 tipos: las de tipo A a 0.5€/kg y las de tipo B a 0.8€/kg. Sabemos que solo dispone en su furgoneta de espacio para transportar 700 kg de naranjas como máximo y que piensa vender el kg de naranjas de tipo A a 0.58€/kg y el de tipo B a 0.9€/kg. ¿Cuántos kg de naranjas de cada tipo deberá comprar para obtener el máximo beneficio?

Problema 1.11. Una refinería de petróleo va a producir un nuevo tipo de gasolina mezclando cuatro tipos de gasolina que están compuestos por tres aditivos A, B, C. La siguiente tabla contiene el porcentaje de dichos aditivos en cada tipo de gasolina y el precio de compra unitario de cada tipo de gasolina:

	Porciento de A	Porciento de B	Porciento de C	Precio
1	80	10	10	43
2	30	30	40	31
3	70	10	20	47
4	40	50	10	37

Debido a las exigencias del mercado, la nueva gasolina deberá estar compuesta por al menos un 60% de aditivo A y un máximo de 30% de aditivo C. Se pide

1. Modelizar el problema a resolver para que la empresa consiga minimizar los costes de gasolina comprada.

2. Resolver con Matlab

Problema 1.12.
Dado el siguiente problema en $\mathbb{R}^2$:

$$\begin{array}{lll} \max & \mathbf{c^t}x & \\ \text{s.t.} & Ax & \leq \mathbf{b} \\ & \mathbf{x} \geq 0 & \end{array}$$

donde:

$$\mathbf{c} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}; \mathbf{A} = \begin{pmatrix} 3 & 5 \\ 5 & 2 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} 15 \\ 10 \end{pmatrix}$$

Se pide:

1. Resolverlo gráficamente dibujando la región factible y las curvas de nivel de la función objetivo.

2. Resolverlo analíticamente indicando todos los puntos extremos.

Problema 1.13. Una compañía que manufactura tres productos A, B y C, para lo que necesita dos recursos (mano de obra y materiales) quiere determinar la estrategia de producción óptima que maximiza el beneficio total. Para ello se formula el siguiente problema de programación lineal:

$$\begin{array}{lll} \max & 3x+y+5z & \\ \text{s.t.} & 6x+3y+5z & \geq 45 \quad \text{(labor)} \\ & 3x+4y+5z & \leq 30 \quad \text{(materials)} \\ & x,y,z \geq 0 & \end{array}$$

1. Determinar los niveles óptimos de producción y el beneficio máximo.
2. Supóngase que pueden obtenerse 15 unidades adicionales de material con un coste global de 10 unidades monetarias. ¿Conviene invertir en la adquisición de material?

Problema 1.14. Un artesano fabrica collares y pulseras. Hacer un collar le lleva 2 horas y hacer una pulsera 1 hora. El material del que dispone no le permite hacer más de 50 piezas. Como mucho, el artesano puede dedicar al trabajo 80 horas. Por cada collar gana 5 € y por cada pulsera 4 €. El artesano desea determinar el número de collares y pulseras que debe fabricar para optimizar sus beneficios. Se pide:

1. Modelizar matemáticamente el problema indicando todos los elementos del modelo.
2. Representar gráficamente la región factible.
3. Obtener el número de collares y pulseras para maximizar el beneficio.

Problema 1.15. Un pintor necesita pintura para pintar como mínimo una superficie de $480m^2$. Puede comprar la pintura a dos proveedores, A y B. El proveedor A le ofrece una pintura con un rendimiento de $6m^2$ por kg y un precio de 1 € por kg. La pintura del proveedor B tiene un precio de $1,2$ € por kg y un rendimiento de $8m^2$ por kg. Ningún proveedor le puede proporcionar más de 75 kg y el presupuesto máximo del pintor es de 120 €.

1. Plantear el problema de PL para determinar la cantidad de pintura que el pintor tiene que comprar a cada proveedor para obtener el mínimo coste.
2. Resolver el problema utilizando el método gráfico.

Problema 1.16. BagCo produce sacos y bolsas de piel. Para un saco se requieren $8m^2$ de piel, y en una bolsa sólo se usan $2m^2$. Las necesidades de mano de obra para los dos productos son 12 y 5 horas, respectivamente. Los suministros semanales actuales de piel y de mano de obra se limitan a $1200m^2$ y a 1850 horas, respectivamente. La empresa vende los sacos y las bolsas a 350 € y a 120 €, respectivamente. Suponiendo que BagCo puede vender a otras empresas unidades incompletas o puede guardarlas para futuras producciones sin perder beneficios.

1. Determinar el programa de producción que maximice los ingresos netos.
2. BagCo planea un aumento de producción: ¿Cuál es el precio máximo de compra que debería pagar la empresa por unidad de piel adicional? ¿Y por unidad de mano de obra?
3. ¿Qué ingresos netos obtendría la empresa si, una vez llevado a cabo el plan de producción óptimo, decide vender los sacos 10 € más caros?

Problema 1.17. Un comerciante acude a cierto mercado a comprar naranjas con 500 €. Le ofrecen 2 tipos: las de tipo A a 0,5€/kg, y las de tipo B a 0,8€/kg. Sabemos que solo dispone en su furgoneta de espacio para transportar 700 kg de naranjas como máximo y que piensa vender el kg de naranjas de tipo A a 0,58 €/kg y el de tipo B a 0,9 €/kg.

1. ¿Cuántos kg de naranjas de cada tipo deberá comprar para obtener el máximo beneficio?
2. Si decide cambiar su furgoneta por una con espacio para transportar 900 kg, ¿Qué beneficio obtendría?
3. ¿Cuánto aumentará su beneficio si aumenta 40 e su presupuesto inicial?
4. ¿Cuál será el beneficio total si vende las naranjas de tipo B a 0,92 €/kg?

Observación: Para a) se pide plantear y resolver el modelo de PL correspondiente al problema. Para b), c) y d) responder sin resolver de nuevo el problema con los nuevos parámetros. Pista: los precios sombra pueden ayudar.

Problema 1.18. Para vender un nuevo producto, una empresa dispone de 100000 €/semana. Los clientes potenciales captados por cada anuncio y semana, el coste semanal (en euros) de un anuncio y los límites semanales sobre los anuncios a contratar, se dan en la siguiente tabla:

	TV	Periódicos	Radio	Internet
Clientes potenciales captados	5,000	8,500	2,400	2,800
Costo Semanal (euros)	600	925	290	364
Límite semanal (anuncios)	30	60	60	80

1. ¿Cómo ha de repartir la empresa el presupuesto semanal en publicidad para conseguir la mayor audiencia posible? Formular y resolver el modelo de programación lineal asociado.
2. ¿Cuánto conseguiría aumentar la audiencia si el presupuesto, en lugar de 100.000 €/semana, fuese de 120.000 € semanales? Responde sin volver a ejecutar el solver.

2. Programación lineal con variable entera

Problema 2.1. Resolver, comparando las soluciones con sus respectivos problemas relajados, los siguientes problemas de programación lineal entera:

1.
$$\begin{array}{lll} \max & x_1 + x_2 & \\ \text{s.t.} & x_1 + x_2 & \leq 2 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

2.
$$\begin{array}{lll} \max & 2x_1 + 5x_2 & \\ \text{s.t.} & 4x_1 + 52x_2 & \leq 20 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

3.
$$\begin{array}{lll} \max & x_1 + x_2 & \\ \text{s.t.} & x_1 & \leq 4 \\ & x_2 - 4 & \leq 0 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

4.

$$\begin{array}{lll} \max & 8x_1 + 10x_2 & \\ \text{s.a.} & x_1 + 2x_2 & \leq 20 \\ & x_1 + x_2 & \leq 15 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

5.

$$\begin{array}{lll} \min & 3x_1 + 5x_2 & \\ \text{s.t.} & x_1 - 2x_2 & \leq 3 \\ & x_1 - 7x_2 & \geq 0 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

6.

$$\begin{array}{lll} \max & x_1 - 2x_2 & \\ \text{s.t.} & x_1 - 7x_2 & \geq 0 \\ & 2x_1 - 4x_2 & \leq 6 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

Problema 2.2. Dado el siguiente problema de programación lineal entera:

$$\begin{array}{lll} & (IP) & \\ \max z^{IP} & = x_1 + 0.64x_2 & \\ \text{s.t.} & 50x_1 + 31x_2 & \leq 250 \\ & -3x_1 + 2x_2 & \leq 4 \\ & x_1, x_2 & \geq 0 \\ & x_1, x_2 \in \mathbb{Z} & \end{array}$$

a) Plantear el problema relajado (PL) del problema (PE)

b) Resolver gráficamente ambos problemas, obteniendo las soluciones óptimas x^{PE} y x^{PL}.

c) Comparar los valores obtenidos en la función objetivo z^{PE} y z^{PL}.

d) ¿Qué se puede concluir a la vista de estos resultados?

Problema 2.3. Dado el siguiente problema de programación lineal entera:

$$\begin{array}{lll} \max z & = 5x_1 + x_2 & \\ \text{s.t.} & -x_1 + 2x_2 & \leq 4 \\ & x_1 - x_2 & \leq 1 \\ & 4x_1 + x_2 & \leq 12 \\ & x_1, x_2 \in \mathbb{Z}^+ & \end{array}$$

a) Resuelva este problema de manera gráfica.

b) Resuelva el relajamiento de PL de manera gráfica. Redondee esta solución a la solución entera más cercana y verifique si es factible. Después enumere todas las soluciones redondeadas. Para ello redondee esta solución del relajamiento de PL de todas las maneras posibles (es decir, redondee cada valor no entero tanto hacia arriba como hacia abajo). Verifique la factibilidad de cada solución redondeada y, si es factible, calcule z. ¿Alguna de estas soluciones redondeadas es óptima para el problema de PE?

Problema 2.4. Se dispone de 5 tipos distintos de mercancías y de un único envase en el que transportarlas. Cada unidad de mercancía es indivisible. La siguiente tabla contiene el peso (en Kg), el volumen (en litros) y el valor unitario (en euros) de cada tipo de mercancía.

	Tipo de mercancía				
	1	2	3	4	5
Peso (Kg)	5	8	3	2	7
Volumen (l)	1	8	6	5	4
Valor (€)	4	7	5	6	4

Sabiendo que los límites del envase son 0.112 toneladas y de 109 litros:

a) Modelizar el problema a resolver para determinar qué cantidad ha de transportarse de cada tipo de mercancía para maximizar el valor total de la carga.

b) ¿Cuánto cambiaría el valor total de la carga si las unidades de cada mercancía fueran divisibles?

Problema 2.5. Tienen que transportarse sacos con alimentos mediante tres tipos de aviones A1, A2, A3, desde un aeropuerto y arrojarse en las aldeas V1, V2, V3, V4, V5, afectadas por inundaciones. La cantidad de alimentos (en unidades adecuadas) que cada avión puede transportar a cada aldea en cada viaje, se da en la siguiente tabla:

	V1	V2	V3	V4	V5	
A1	10	8	6	9	12	50
A2	5	3	8	4	10	90
A3	7	9	6	10	4	60
	100	80	70	40	20	

El número de viajes que puede hacer cada avión se da en la última columna y el número máximo de aviones que puede recibir diariamente cada aldea en la última fila.

Encontrar el número de viajes que deberá hacer cada avión a cada aldea de forma que se maximice la cantidad de alimento distribuido por día.

Problema 2.6. Una empresa puede fabricar 4 productos diferentes utilizando en su elaboración 5 tipos de materias primas. Las unidades requeridas de cada materia prima por cada unidad de cada producto se recogen en la tabla siguiente, así como el precio de venta de cada producto y la disponibilidad de cada materia prima:

	M1	M2	M3	M4	M5	Precio venta
P1	1	1		1		15
P2		1		1	1	20
P3	1	1	1			15
P4	1		1			10
Disponibilidad	200	150	150	200	100	

El precio unitario de cada materia prima es de 2, 3, 4, 5 y 5, respectivamente, pero hay un coste adicional por hacer un pedido de cada materia prima, independiente de la cantidad que se pida, de 20, 25, 20, 25, 25 unidades, respectivamente. Determinar el plan de producción óptimo para maximizar los beneficios de la empresa.

Problema 2.7. Una compañía analiza la posibilidad de invertir en cinco proyectos durante los próximos tres años. La realización de cada proyecto necesita determinada cantidad de dinero cada uno de los tres años, y cada proyecto tiene un beneficio asociado al final de los tres años. Todas estas cantidades (medidas en millones de euros) se detallan en la siguiente tabla:

Proyecto	Inversión			Ganacia
	Año 1	Año 2	Año 3	
1	3	0	3	15
2	0	8	3	14
3	5	1	4	10
4	4	5	5	20
5	5	0	0	16

La compañía dispone de 9 millones de euros cada año, y tiene las siguientes restricciones adicionales:

a) Si ejecuta el proyecto 3, debe ejecutar también los proyectos 1 y 2.

b) Si decide ejecutar alguno de los proyectos 1 ó 4, entonces debe de ejecutar al menos dos de los proyectos 2, 3 ó 5. (Indicación: $A \Rightarrow B$ es equivalente a $\neg A \vee B$)

c) Si decide ejecutar el proyecto 5, entonces no puede ejecutar el proyecto 1.

Formular y resolver el problema de programación lineal para maximizar el beneficio de la compañía.

Problema 2.8. Un senderista planea salir de ruta por la montaña. Tiene disponible cinco objetos que desea llevar consigo, pero no puede cargar con el peso total de todos ellos, pues su mochila tiene una capacidad máxima de 60 Kg. Para ayudarse en la selección ha asignado un valor a cada artículo en orden ascendente de importancia:

Artículo	O1	O2	O3	O4	O5
Peso (Kg)	42	23	21	15	7
Valor (€)	100	60	70	15	15

- Los objetos O2 y O4 son excluyentes, de forma que no puede llevar ambos simultaneamente.
- Debido a las características especiales del objeto O5, solo tiene sentido llevarlo si se lleva alguno de los objetos O1 u O4.

Formular y resolver el problema de programación matemática que permite determinar qué objetos debe elegir el senderista para maximizar el valor de la mochila.

Problema 2.9. Una compañía de investigación tiene un presupuesto de 5500 € para proyectos y se presentan los siguientes:

Proyecto	1	2	3	4
Ingresos (€)	2500	1000	2600	3000
Coste (€)	2000	350	1500	2000

Se consideran las siguientes condiciones:

- Se tienen que aceptar al menos 2 de los 3 primeros proyectos.
- El proyecto 4 solo se puede aceptar si se ha aceptado el proyecto 2.

Encontrar la solución óptima del problema de maximizar la rentabilidad que se le presenta a la compañía, planteando y resolviendo el modelo de optimización matemática correspondiente.

Problema 2.10. Una persona desea invitar a una fiesta a cuatro amigos A, B, C y D, pero estas personas han impuesto determinadas condiciones para asistir:

- A y C no están dispuestos a coincidir en la fiesta,

- B acepta coincidir con C o con D, pero no con ambos a la vez,
- D no está dispuesto a coincidir con A, salvo que B asista a la fiesta.

Formular y resolver el problema de programación que se debe considerar para determinar el mayor número de personas que podrían asistir a la fiesta.

Problema 2.11. En una misión pacífica de las Naciones Unidas se dispone de 5 aviadores para formar las tripulaciones de dos aviones biplaza. Estos aviadores son de distintas nacionalidades: Español, Francés, Italiano, Griego y Portugués. Como en toda cuestión diplomática las relaciones internacionales son de gran peso, cada una de las distintas composiciones de las tripulaciones conlleva un beneficio, siendo estos:

	Francés	Italiano	Griego	Portugués
Español	2	5	4	3
Francés		4	4	2
Italiano			5	4
Griego				3

Por otra parte, estas mismas relaciones internacionales hacen que si una tripulación está formada por el aviador español y el italiano la otra ha de estar formada por el aviador francés y el griego. Formular el problema de programación lineal entera.

Problema 2.12. La empresa Record-a-Song Company ha contratado una estrella en ascenso para grabar ocho canciones. La duración de esas canciones es de 8, 3, 5, 5, 9, 6, 7 y 12 minutos, respectivamente. Record-a-Song usa cinta de ida y vuelta para la grabación. Cada lado tiene 30 minutos de capacidad. La empresa desea distribuir las canciones en los dos lados, en forma equilibrada. Eso quiere decir que la duración de las canciones en cada lado debe ser más o menos la misma, hasta donde sea posible. Formule el problema como programa lineal entero, y determine la solución óptima.

Problema 2.13. Una universidad se encuentra en un proceso de formar una comisión. Diez personas han sido nominadas: A, B, C, D, E, F, G, H, I y J. El reglamento obliga a que sean incluidos en dicha comisión al menos una mujer, un hombre, un estudiante, un administrativo y un profesor. Además, el número de mujeres debe ser igual que el de hombres y el número de profesores no debe de ser inferior al de administrativos. La mezcla de los nominados en las siguientes categorías es como sigue:

Categoría	**Personas**
Mujeres	A, B, C, D, E
Hombres	F, G, H, I, J
Estudiantes	A, B, C, J
Administrativos	E, F
Profesores	D, G, H, I

1. Modelizar el problema para conseguir que la comisión sea lo más reducida posible.
2. Encontrar la comisión más reducida posible teniendo en cuenta todas las restricciones.
3. Si además, nos enteramos a posteriori de que por problemas personales, I renunciará si A y J forman parte de la comisión, ¿Cómo se puede incluir esta condición en el modelo? ¿Cuál sería la nueva comisión conociendo esta incompatibilidad?

Autoevaluación Programación Lineal

Problema 2.14. Una empresa fabrica 4 tipos de materiales para la construcción. Para su fabricación se han impuesto las siguientes restricciones:

1. Se tiene que elegir como máximo tres de los 4 tipos de material.
2. La producción se llevará a cabo en una de las dos plantas existentes para la producción.

La ganancia unitaria por cada unidad de los 4 tipos de producto es de 300, 100,500 , y 100 euros respectivamente. Y los tiempos de producción son 2,5 , 2 , y 2 horas respectivamente en la planta 1 ; y $4, 3, 3$ y 1 horas en la planta 2. Las horas disponibles semanales por planta son 45 para la planta 1 y 53 para la planta 2. Y sus ventas potenciales son de $5, 8, 4$ y 11 unidades por semana.

1. Elaborar una tabla que recoja toda la información.
2. Determinar el modelo de programación matemática que permita resolver el problema descrito.
3. Resolver el modelo anterior con Matlab y comentar los resultados obtenidos.

Nota: Supóngase un problema de programación entera mixta. Explicar detalladamente el procedimiento de resolución de los ejercicios.

Problema 2.15. Una empresa está estudiando 10 posibles proyectos en los cuales invertir un capital total de 22 millones de EUR. Cada proyecto requiere un capital inicial y tiene unas ganancias netas estimadas que se indican en millones de EUR en la tabla siguiente
Existen las siguientes restricciones sobre las inversiones.

Proyecto	1	2	3	4	5	6	7	8	9	10
Capital inicial (millones de EUR)	8	10	9	4	7	5	6	2	1	3
Ganancias netas estimadas (millones de EUR)	16	20	18	17	13	15	10	14	12	11

1. Si se invierte en el proyecto 9 , hay que invertir en el proyecto 5 .
2. Si se invierte en los proyectos 7 y 8 , hay que invertir en el proyecto 3 .
3. De los proyectos: $8, 9, 10$, se puede invertir solo en uno.
4. Hay que invertir al menos en uno de los siguientes dos proyectos: 3,4 .
5. Si se invierte en el proyecto 10, hay que invertir también en al menos uno de entre los proyectos 2 y 3 .

El objetivo es elegir la combinación de inversiones que maximice la ganancia total estimada.

1. Determinar un modelo de programación matemática que permita resolver el problema.
2. Resolver el modelo anterior con Matlab y comentar el resultado obtenido.

Problema 2.16. Nuestra empresa debe alquilar la infraestructura de red necesaria para su correcto funcionamiento. Hemos estado preguntado a los principales operadores y sus respuestas están recogidas en la tabla.
El objetivo es obtener la mejor combinación de operadores que nos asegure un flujo total de 20 TB por el coste mínimo para nuestra empresa.

1. Determinar un modelo de programación matemática que permita resolver el problema.

2. Resolver el modelo anterior con Matlab y comentar el resultado obtenido.

Sede	Coste instalación (€)	Coste trabajo (/TB)	Máximos TB ofertables
Operador 1	8000	400	10
Operador 2	7000	500	8
Operador 3	5000	900	2
Operador 4	5000	800	4
Operador 5	6000	600	6

Problema 2.17. Dado el siguiente problema de programación lineal

$$\begin{array}{ll} \min & Z = -4x_1 + x_2 \\ \text{s.t.} & x_1 + 3x_2 \geq 3 \\ & 6x_1 + 3x_2 \leq 6 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

1. representa a escala la región factible del problema.
2. Determina el mínimo de la función objetivo usando el método de las rectas de nivel.
3. Determina los precios sombra mediante consideraciones geométricas.

Problema 2.18. Una empresa metalúrgica produce cuatro tipos de productos en secuencia en dos máquinas. Los detalles de la producción se detallan en la siguiente tabla:

		Production time				
Máquina	coste/minuto	Producto				Capacidad diaria
	\$/min	1	2	3	4	(min)
1	10	2	3	4	2	500
2	5	3	2	1	2	380
	ganancia/unidad (\$)	65	70	55	45	

1. Construir un modelo de programación lineal que permita obtener la mayor ganancia posible.
2. Resuelve con Matlab el modelo formulado en el punto anterior y comenta el resultado obtenido.

Sol: 28 unids del producto 1, 148 producto 2. Ganancia de 5280 dólares.

Problema 2.19. Nuestra empresa se dedica a la fabricación de circuitos impresos de alta calidad. Tras varios años en el negocio, estamos en disposición de conseguir que una gran marca nos contrate como fabricante de sus GPU's. Nuestro departamento de inversiones ha estado recabando información y en la siguientes tablas se recogen los costes de creación de las nuevas infraestructuras necesarias, así como, el beneficio unitario y la capacidad de cada una de nuestras sedes.

GPU	Coste de infraestructura (€)	Beneficio unitario (€)	
Titán RTX	4500000	1200	
RTX 3090	7600000	1600	
Fábrica	Horas disponibles	Titán RTX	RTX 3090
1	480 horas	52 unidades/hora	38 unidades/hora
2	720 horas	42 unidades/hora	23 unidades/hora

el objetivo es elegir la combinación de fábricas y productos de tal manera que nuestro beneficio sea el más alto posible.

1. Formular el modelo de programación matemática que permita resolver el problema descrito.

2. Resolver el modelo con Matlab y comentar el resultado obtenido.

Sol: 24960 Titán RTX en fábrica 1. 30240 Titán RTX en fábrica 2. Beneficios de 57240000 euros.

Problema 2.20. Dado el siguiente problema de programación lineal con función objetivo Z sometida a las siguientes restricciones

$$\begin{aligned} Z &= 2x_1 + x_2 \\ 0 &\leq x_1 \leq 6 \\ 0 &\leq x_2 \leq 10 \\ 8 &\leq 2x_1 + x_2 \leq 16 \end{aligned}$$

1. Representar a escala la región factible del problema y resolverlo (buscar el máximo y el mínimo de la función objetivo Z) mediante el método de las rectas de nivel. Indicar el tipo de solución.

2. ¿Cómo cambia la solución si x_1 y x_2 sólo pueden tomar valores enteros? Calcula el valor de la función objetivo en la solución. Indicar el tipo de solución.

Nota: Explicar detalladamente el procedimiento de resolución de los ejercicios.

Problema 2.21. Sea el problema de programación lineal:

$$\begin{array}{lll} \max Z = & 18x_1 + 6x_2 & \\ \text{s.t.} & 3x_1 + x_2 & \leq 12, \\ & x_1 + 2x_2 & \leq 16 \\ & x_1 & \leq 3.75 \\ \text{con} & x_1, x_2 \geq 0 & \end{array}$$

1. Representa a escala la región factible del problema.

2. Determina todas las soluciones continuas.

3. Determina todas las soluciones enteras.

4. Indica el máximo de la medida global de desempeño.

5. Determina de forma gráfica los precios sombra de los recursos disponibles.

6. Obtén la solución de los apartados d y e mediante Matlab.

Problema 2.22. En una central eléctrica hay tres generadores y hay que decidir cuáles activar en los períodos diurno y nocturno de modo que se garantice una producción de por lo menos 4000 megavatios de día y 2800 megavatios de noche. Los costes de activación en EUR de los tres generadores en los dos periodos se presentan en la tabla siguiente, junto con el coste en EUR por megavatio producido en cada generador y la capacidad máxima de producción de cada generador.

	Coste activación dia	Coste activación noche	Coste por megavatio	Capacidad máxima
Generador 1	750	1000	3	2000
Generador 2	600	900	5	1700
Generador 3	800	1100	6	2500

1. Determina el modelo de programación matemática que permita establecer qué generadores activar de día y de noche y a qué nivel de actividad para minimizar el coste total de generación de energía satisfaciendo la demanda.

2. Resuelve el modelo obtenido en el apartado a. con Matlab y comenta el resultado obtenido.

Problema 2.23. En la siguiente tabla se enumeran las necesidades de los pilotos capaces de volar un A320 durante los siguientes seis meses. El costo del salario de un piloto es de 8000 por mes. Al comienzo del mes 1 , la aerolínea tiene una plantilla de 20 pilotos, pero esta plantilla se puede ajustar cada mes.

Mes	1	2	3	4	5	6
Pilotos requeridos	30	60	55	40	45	50

Necesidades de pilotos para los siguientes seis meses.
Los pilotos se pueden contratar y despedir a principios de cada mes. Los pilotos recién contratados pueden comenzar a trabajar el mismo mes y los pilotos despedidos dejan de trabajar el mismo día que son despedidos. El costo de despedir a un piloto es de 10000 y el costo de contratación es de 5000 por piloto. Si es conveniente, la aerolínea puede tener una plantilla de pilotos más grande que las necesidades reales.

1. Definir un modelo de programación lineal para obtener los pilotos a contratar y despedir cada mes para minimizar el costo total del personal piloto (costos de salario más costos de contratación y despido).

2. Modifique el modelo lineal para incluir la restricción de que la aerolínea no puede despedir pilotos si ha contratado pilotos el mes anterior.

Resuelve cada modelo obtenido con Matlab y comenta los resultados obtenidos.

3. Transporte y Asignación

Importante: Antes de resolver cada problema indique que tipo de modelo se utilizará para la solución, i.e., que tipo de problema es: transporte o asignación.

Problema 3.1. Una empresa multinacional de componentes de automoción dispone de 3 fábricas situadas en Portugal, Grecia e Irlanda. Debido al lanzamiento de un nuevo automóvil, que se fabrica en Alemania y Francia, estos países solicitan 1400 y 2300 respectivamente, de las componentes que fabrica dicha empresa. La capacidad de producción de las factorías de Portugal, Grecia e Irlanda es 1000, 1500 y 1200 componentes por año, respectivamente. El transporte de cada componente implica un coste que depende del país donde se fabrica y el país de destino. La matriz de costes es la siguiente:

	Desitno	
Factoría	Alemania	Francia
Portugal	6.0	16.0
Grecia	7.5	8.0
Irlanda	7.5	5.0

Se pide:

1. Dibujar la red del problema (nodos origen, nodos destino, arcos con costes, suministros y demandas).

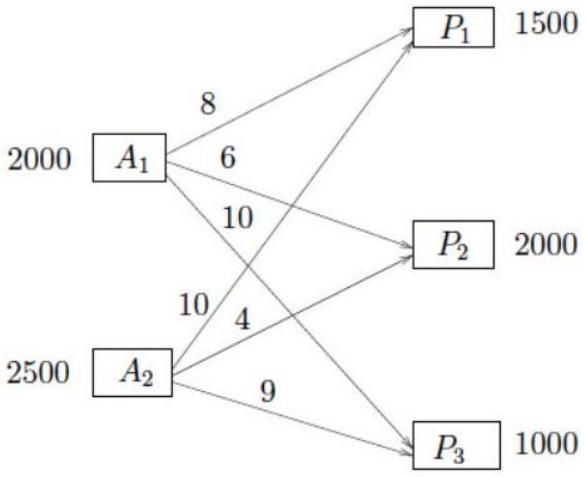
2. Plantear y resolver el modelo de programación lineal que minimiza el coste del transporte de las componentes.

Problema 3.2. Cuatro contratistas concursan para conseguir la construcción de cuatro edificios, debiendo ser asignado cada edificio a un único contratista. El tiempo que cada contratista necesita para la construcción de cada edificio viene dado en la siguiente tabla:

	B1	B2	B3	B4
C1	58	58	60	54
C2	66	70	70	78
C3	106	104	100	95
C4	52	54	64	54

Calcular la asignación para que la suma total del tiempo empleado en la construcción de los cuatro edificios sea mínima.

Problema 3.3. Una empresa productora de barras de pan tiene dos almacenes A1 y A2 desde los cuales debe enviar pan a tres panaderías P1, P2 y P3. Las ofertas, las demandas y los costes de envío se dan en el siguiente grafo:



Encontrar la manera óptima de realizar los envíos para minimizar el coste.

Problema 3.4. Una empresa produce un único artículo en tres plantas, A1, A2 y A3. La capacidad de producción mensual de la empresa está limitada a 1500 unidades mensuales en cada una de las plantas. La empresa tiene cuatro clientes mayoristas cuyas demandas mensuales son 1000, 1200, 1400 y 900 unidades respectivamente. El beneficio unitario que le proporciona su producto, considerados los costes de producción y el precio de venta, es de 110 unidades. Los costes de envío a los 4 clientes mayoristas que la empresa tiene vienen dados por la siguiente tabla:

	C1	C2	C3	C4
A1	30	10	25	20
A2	15	25	30	10
A3	20	30	15	20

El objetivo de la empresa es organizar la producción en cada uno de los meses para obtener el máximo beneficio. ¿Cómo puede hacerlo?

Problema 3.5. Una empresa convoca unas pruebas de selección para cubrir las vacantes que hay en 3 puestos de trabajo $(V_1, V_2,$ and $V_3)$. Realizadas las pruebas la empresa asigna a las 5 personas que se han presentado una puntuación entre 1 y 10. Las puntuaciones se recogen en la tabla:

	V 1	V 2	V 3
P1	2	7	8
P2	4	7	6
P3	10	5	7
P4	3	6	4
P5	6	4	9

Encontrar la asignación que maximice la adecuación total de las tres personas elegidas para las 3 vacantes.

Problema 3.6. Para desplegar la infraestructura de wifi del nuevo edifico se dispone de un número máximo de tres tipos de dispositivos. Cada tipo de dispositivo permite un número máximo de conexiones. Se nos indica que cada una de las plantas va a necesitar un número de conexiones distintas. En ciertas plantas va a ser necesario realizar obras, lo cual aumenta el precio de instalación de los puntos de acceso. En la tabla siguiente se recogen los costes, el número de usuarios permitidos por punto de acceso y el número mínimo de usuarios a los que hay que dar servicio.

Dependencia–Punto Acceso	Tipo G	Tipo N	Tipo AC	Nº de conexiones necesarias
Cafetería (coste/dispositivo €)	200	250	225	200
Aulario (coste/dispositivo €)	150	100	175	600
Biblioteca (coste/dispositivo €)	200	150	250	300
Nº dispositivos disponibles	25	10	6	
Nº de conexiones/dispositivo	20	30	50	

El objetivo es elegir la combinación de dispositivo que minimice el coste de instalación y permita dar cobertura a los usuarios indicados.

1. Determinar un modelo de programación matemática que permita resolver el problema.
2. Resolver el modelo anterior con `Matlab` y comentar el resultado obtenido.
3. Indicar el coste de instalación de cada una de las dependencias del nuevo edificio.

Problema 3.7. El jefe del departamento de Estadística de una facultad universitaria está planificando la docencia de 5 asignaturas (A1, A2, A3, A4 y A5) para el próximo curso. Para ello, revisa las valoraciones docentes históricas de los 5 posibles profesores (P1, P2, P3, P4 y P5) que han impartido las asignaturas y aún tienen disponibilidad. Las valoraciones, en una escala de 0 a 5, se recogen en la siguiente tabla:

	A_1	A_2	A_3	A_4	A_5
P_1	4.7	3.5	2.6	4.8	3.5
P_2	4.5	2.9	3.5	2.1	3.2
P_3	3.9	2.1	3.0	4.6	2.9
P_4	4.6	3.1	2.7	2.4	3.1
P_5	3.3	2.3	2.8	3.4	4.2

Cada profesor solo puede ser asignado a una asignatura y cada asignatura solo puede ser impartida por un profesor. Además, el profesor P4 no puede impartir bajo ningún concepto la asignatura A1 puesto que su hija formará parte de la clase el próximo año. Por la misma razón, el profesor P2 tampoco podrá encargarse de la asignatura A3. Encontrar la asignación que maximice la valoración total de los cinco profesores elegidos para las cinco asignaturas.

Problema 3.8. Una empresa ha desarrollado un transpondendor en banda L con una figura de ruido realmente baja. Diferentes agencias espaciales están interesadas en la compra de este nuevo dispositivo. La

Sede–Agencia	ESA	NASA	ROSCOSMOS	CNSA	Nº de unidades fabricadas
Europa (coste unitario miles €)	30	50	40	50	35
EEUU (coste unitario miles €)	40	25	65	35	30
China (coste unitario miles €)	20	35	15	10	20
Nº de Transpondedores necesarios	60	50	40	60	

empresa cuenta con tres sedes en todo el mundo, la tabla siguiente se recogen los costes de la fabricación y envío para cada una de las sedes,
Además existen ciertas restricciones sobre la distribución derivadas de los acuerdos con las distintas agencias.

1. Dado que nuestra empresa ha recibido fondos europeos, la ESA nos exige que como mínimo debemos cubrir el 75% de sus necesidades.
2. Debido a la guerra comercial entre EEUU y China, nuestras sedes de esos países solo pueden servir un máximo del 10% de su producción entre ellas.
3. El requerimiento mínimo de la NASA es del 20% de sus necesidades.
4. El requerimiento mínimo de ROSCOSMOS es del 30% de sus necesidades.
5. El requerimiento mínimo de CNSA es del 10% de sus necesidades.

El objetivo es elegir la distribución del producto desde las sedes hasta las cadenas de montaje de cada una de las agencias espaciales con un gasto mínimo, de manera que se cubran las exigencias propuestas.

1. Determinar un modelo de programación matemática que permita resolver el problema.
2. Resolver el modelo anterior con `Matlab` y comentar el resultado obtenido.
3. Determinar el coste total de los equipos enviados a cada una de las sedes.

Problema 3.9. Una CPU consta de 4 ALU optimizadas para realizar un tipo determinado de tareas. Queremos procesar un programa que consta de 4 rutinas de la forma más eficiente posible, la tabla siguiente muestra el coste en tiempo de ejecución de cada una de las ALU's para cada una de las rutinas

ALU–Rutina	Suma	Multiplicación	Potencia	Op lógicas
1 (tiempo de ejecución en μs)	15	5	10	6
2 (tiempo de ejecución en μs)	3	15	8	6
3 (tiempo de ejecución en μs)	9	11	4	13
4 (tiempo de ejecución en μs)	4	8	7	11

El objetivo es elegir la combinación ALU–rutina que minimize el tiempo de ejecución, sabiendo que cada ALU solo puede tener asignada una rutina y que todas las rutinas tienen que ser asignadas.

1. Determinar un modelo de programación matemática que permita resolver el problema.
2. Resolver el modelo anterior con `Matlab` y comentar el resultado obtenido.

Problema 3.10. Una empresa necesita asignar 4 trabajos a 4 trabajadores. El coste de realizar un trabajo es una función de las habilidades de los trabajadores. La tabla resume el coste de las asignaciones. El trabajador 1 no puede realizar el trabajo 3, y el trabajador 3 no puede realizar el trabajo 4.

Trabajador/tarea	1	2	3	4
1	50	50	–	20
2	70	40	20	30
3	90	30	50	–
4	70	20	60	70

El objetivo es elegir la combinación trabajador–tarea que minimice el coste, sabiendo que cada trabajador solo puede tener asignada una tarea y que todas las tareas tienen que ser asignadas.

Problema 3.11. Una empresa se encarga de construir cuatro tipos distintos de antenas para cada una de sus sedes de acuerdo con la tabla siguiente,

Fábrica	Tipos de antenas que se fabrican
1	1, 4
2	2, 3
3	3, 4
4	1, 2, 3

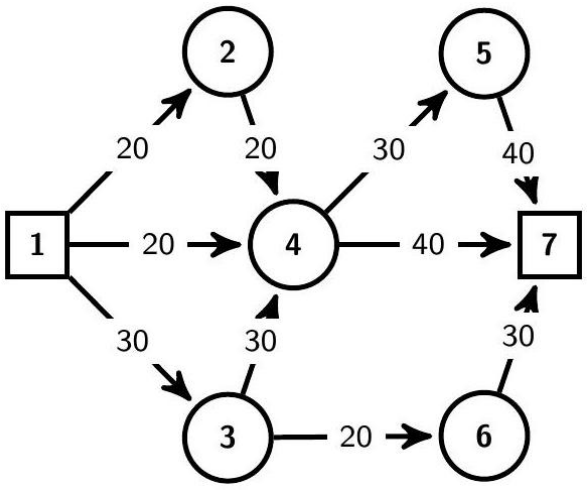
Las fábricas puede llegar a fabricar 300, 180, 100 y 250 antenas al día, mientras que las demandas diarias de los cuatro tipos de antenas son de 200, 150 350 y 100 unidades. De esta manera, el objetivo es elegir la combinación de fábricas y antenas de tal manera que se puedan fabricar el máximo de antenas posibles.

1. Determinar el grafo que define al problema.
2. Resolver el problema anterior con Matlab.

4. Programación en redes

Importante: Antes de resolver cada problema indique que tipo de modelo es utilizará para la solución, i.e., que tipo de problema es: flujo máximo, camino más corto, árbol de soporte mínimo, etc.

Problema 4.1. Dado el siguiente grafo determinar cual es la ruta más corta para llegar de 1 hasta 7 , sabiendo que el número que acompaña a cada arco está expresado en km.



Problema 4.2. Queremos llegar desde la estación de Sol hasta el campus de Fuenlabrada. Para ello, tenemos varias opciones de viajes y transbordos, cuyo coste en € se recoge en la tabla siguiente.

	Sol	Pza de España	Puerta del Sur	Atocha	Getafe	Fuenlabrada	Campus
Sol	-	1.4	-	1.8	2.6	-	-
Plaza de España	1.4	-	1.4	-	-	-	-
Puerta del sur	-	1.4	-	-	1.8	2.6	1.4
Atocha	1.8	-	-	-	2.6	2.6	-
Getafe	2.6	-	1.8	2.6	-	1.4	1.4
Fuenlabrada	-	-	2.6	2.6	1.4	-	1.4
Campus	-	-	1.4	-	1.4	1.4	-

El objetivo es obtener la forma más barata de llegar desde Sol hasta el Campus.

1. Determina el grafo que modeliza este problema.
2. Determinar un modelo de programación matemática que permita resolver el problema.
3. Resolver el modelo mediante el uso de Matlab.

Problema 4.3. Somos los encargados del mantenimiento de la red de comunicaciones interna de la URJC. Nuestro jefe nos ha encargado conocer el tráfico máximo que puede cursarse por nuestra red para conocer si podemos añadir nuevos servicios de multimedia. Tras una etapa de investigación y análisis de la estructura de red llegamos a la siguiente tabla de enrutamiento y capacidades.

Origen	Destino	Tráfico (Gbps)
Móstoles	Aranjuez	30
Móstoles	Vicálvaro	35
Aranjuez	Alcorcón	15
Vicálvaro	Alcorcón	20
Vicálvaro	Fuenlabrada	15
Alcorcón	Fuenlabrada	25

El objetivo es conocer el volumen total de tráfico que podemos cursar en nuestra red desde Móstoles hasta Fuenlabrada.

1. Determinar el grafo que define al problema.
2. Determinar un modelo de programación matemática que permita resolver el problema. ¿A qué tipo de problema de redes corresponde el modelo?
3. Resolver el modelo mediante el uso de Matlab.

Problema 4.4. Los centros de procesamiento de datos (CPDs) de las universidades públicas madrileñas han de ser integrados dentro de una red que permita compartir de manera fácil y rápida la información que contiene cada uno. Para realizar este despliegue se ha pensado en utilizar una red de fibra óptica, la cual tiene un coste de 300 € /km. En la tabla siguiente se muestran las distintas sedes y las distancia entre ellas.
a. Determinar las conectividades entre los distintos CPDs para reducir el gasto en fibra óptica.
b. Determinar el coste total de la instalación de la red de fibra.

(km)	UAH	UAM	UC3M	UCM	UPM	URJC
UAH	-	38	38	33	32	53
UAM	38	-	33	20	19	29
UC3M	38	33	-	20	22	21
UCM	33	20	20	-	1	24
UPM	32	19	22	1	-	25
URJC	53	29	21	24	25	-

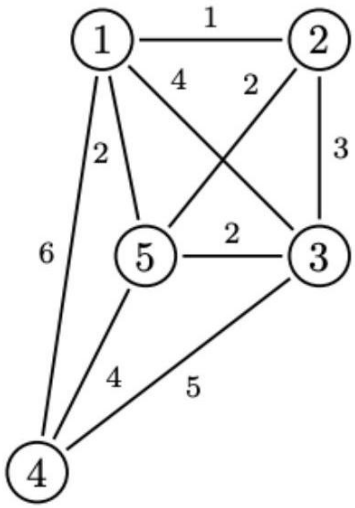
Problema 4.5. Nuestra empresa se dedica a dar soporte técnico a dispositivos de red en CPD's. Nuestro equipo se divide la jornada laboral en 4 franjas en las que pueden atender hasta a 3 clientes. En caso de que entren más clientes se les puede retrasar a la siguiente franjas pero esto repercute en un coste de 100 € más 10 por franja que se haya retrasado, en caso contrario no se tienen en cuenta los costes. Además, todos los clientes que no se hayan atendido serán atendidos por la tarde realizando horas extras, con el mismo coste que las anteriores.

Franja	Clientes posibles	Clientes entrantes
8:00-10:00	3	5
10:00-12:00	3	6
12:00-14:00	3	3
14:00-16:00	3	2
Tardes	4	-

El objetivo es asignar los trabajos a las distintas franjas horarias para minimizar el coste.

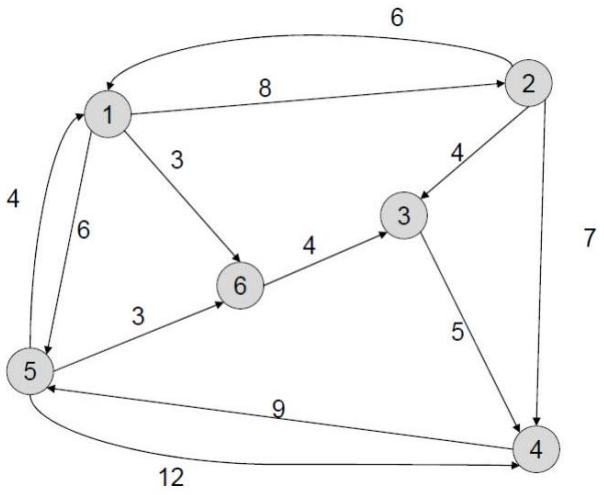
1. Determinar el grafo que define al problema.
2. Determinar un modelo de programación matemática que permita resolver el problema.
3. Resolver el modelo mediante el uso de Matlab.

Problema 4.6. Una pequeña escuela dispone de 5 ordenadores. La distancia entre cada equipo se muestra en la siguiente figura.

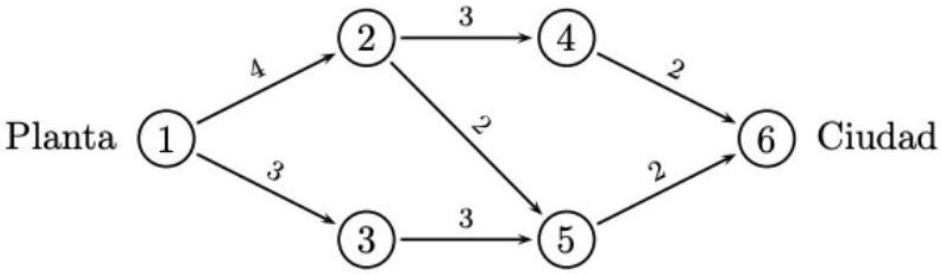


¿Cuál es la longitud mínima de cable requerida para conectar todos los ordenadores ?

Problema 4.7. En la siguiente red, encuentre el camino más corto desde el nodo 2 al resto de nodos.



Problema 4.8. Supongamos que se envía energía eléctrica desde una planta (nodo 1) a una cierta ciudad (nodo 6) a través de una serie de subestaciones (nodos 2 al 5). Las distancias entre las combinaciones factibles se muestran en la siguiente red:



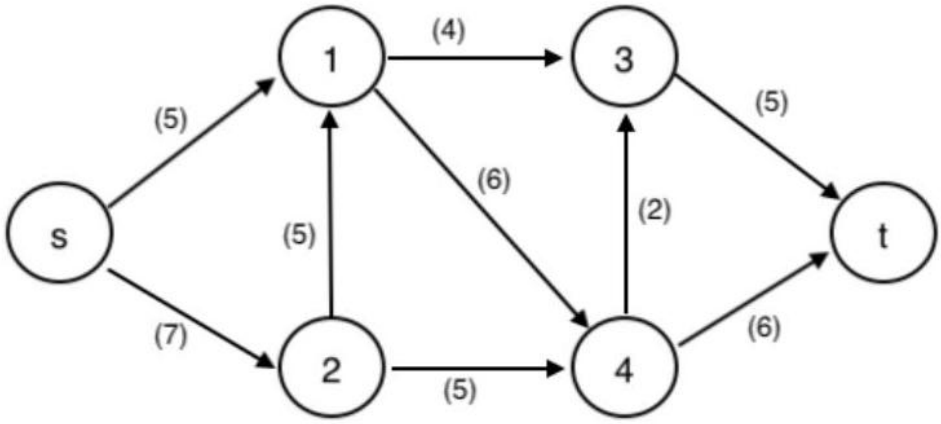
Determine la forma de enviar la energía de modo que se minimice la distancia a recorrer.

Problema 4.9. Telefónica se está planteando trazar una red de televisión por cable para dar servicio a seis nuevas áreas recientemente urbanizadas. La siguiente figura representa los enlaces que podrían establecerse entre las seis zonas. El coste asociado a cada arista indica los kilómetros de cable que serían necesarios.

Origin/ Destination	V 1	V 2	V 3	V 4	V 5	V 6
V 1		1	5	7	9	
V 2			6	4	3	
V 3				5		10
V 4					8	3
V 5						
V 6						

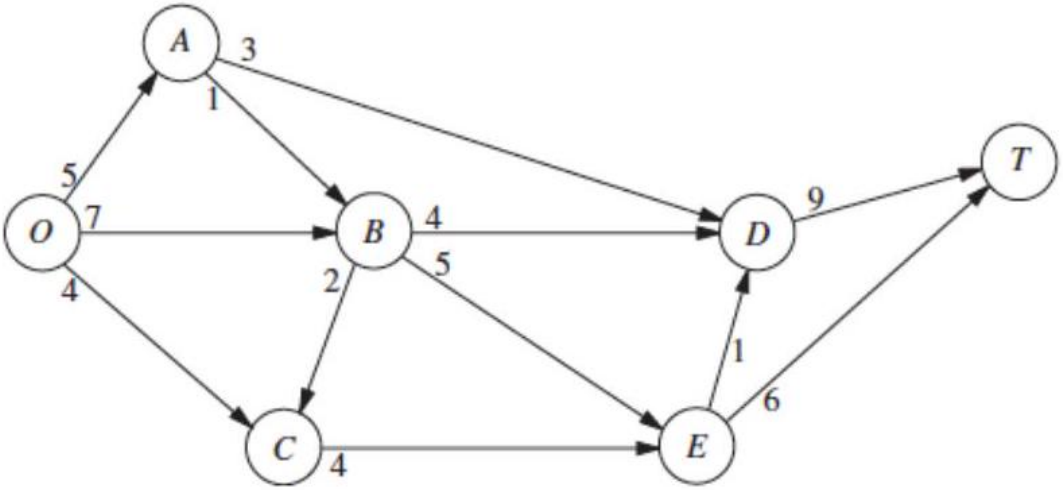
La empresa desea encontrar el trazado de red de coste mínimo. Dibujar el grafo asociado a la red y encontrar dicho trazado.

Problema 4.10. Dada la siguiente red, en la que se muestra entre paréntesis la capacidad de cada arco,



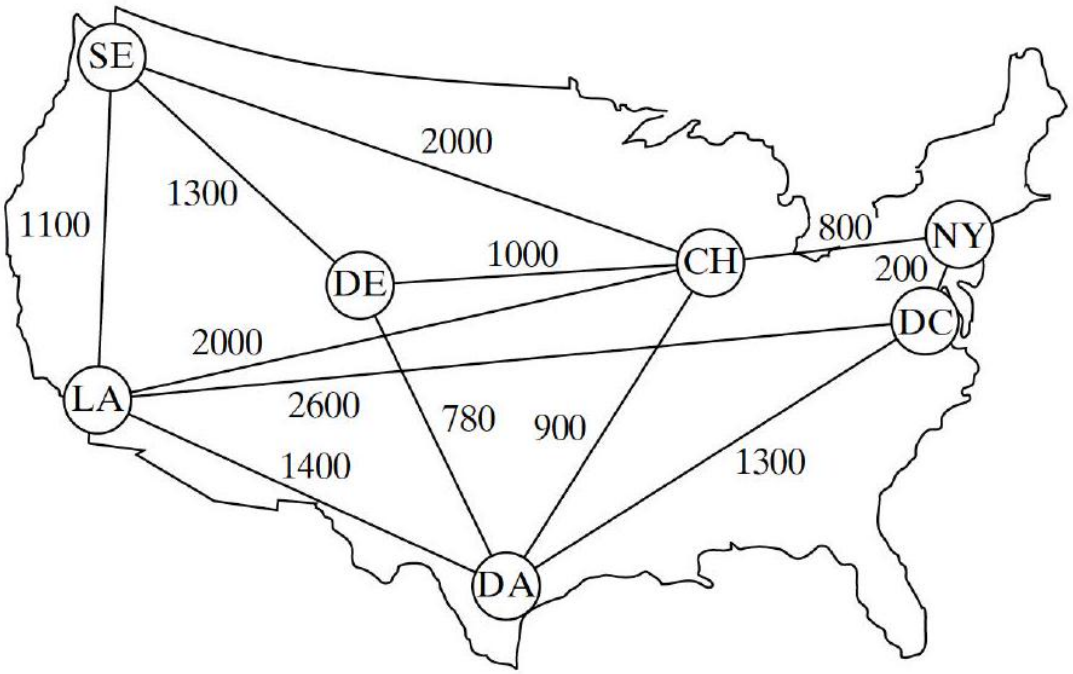
encontrar el flujo máximo compatible.

Problema 4.11. Dada la siguiente red, en la que se muestra la capacidad de cada arco,



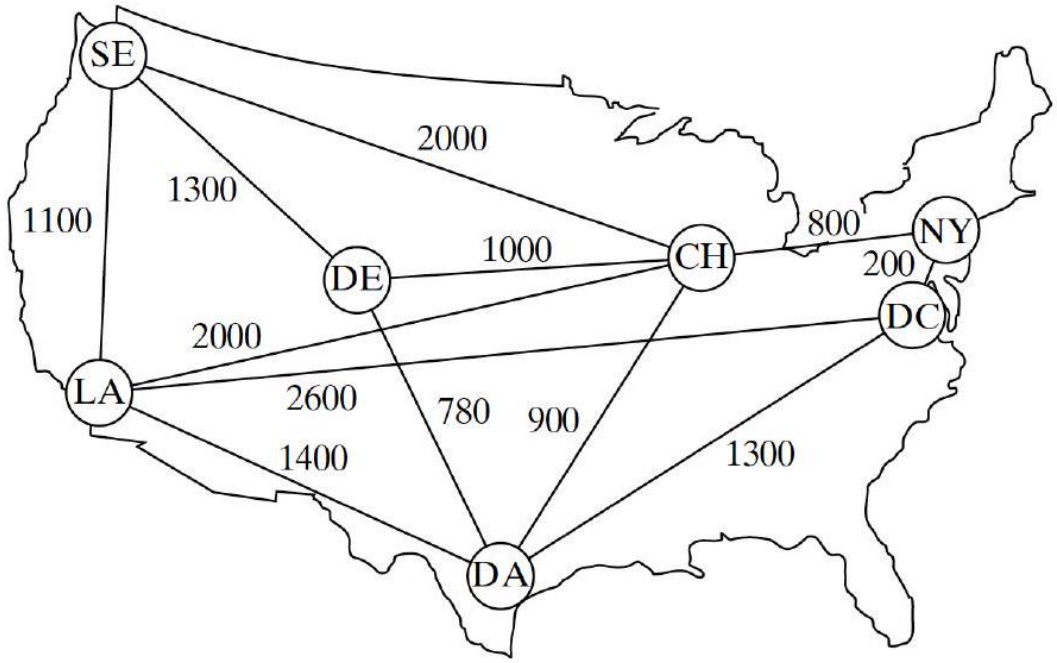
encontrar el flujo máximo compatible.

Problema 4.12. En la figura adjunta se muestra la ubicación de las principales terminales y vías de ferrocarril con su longitud (en millas) existentes en Estados Unidos:



El nuevo presidente del país, tras ganar las elecciones, tiene como objetivo decidir qué vías se deben "revitalizar" para manejar el tráfico ferroviario. Si solamente se quiere favorecer a los ciudadanos de Los ángeles permitiéndoles ir a cualquier terminal por el trayecto más corto posible, ¿cuántas millas del trazado habría que revitalizar?

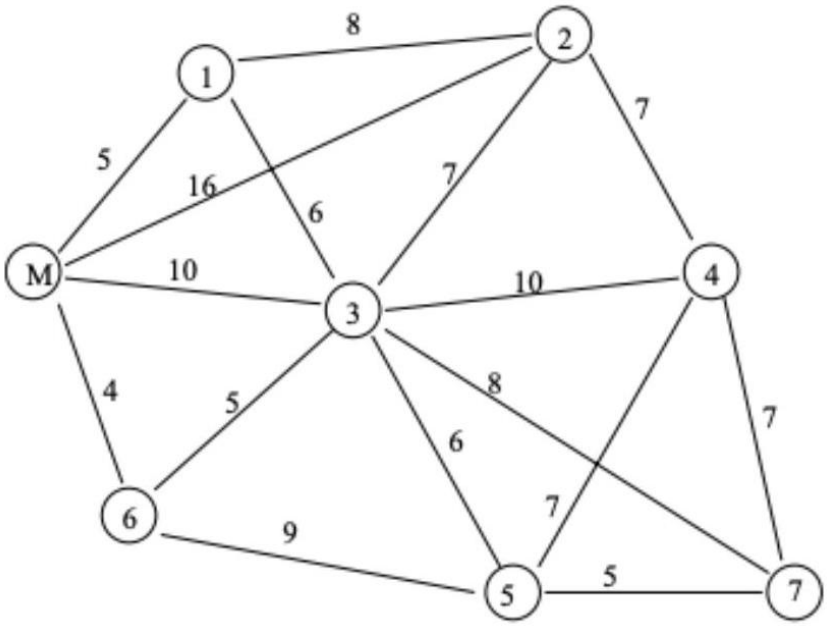
Problema 4.13. En la figura adjunta se muestra la ubicación de las principales terminales y vías de ferrocarril con su longitud (en millas) existentes en Estados Unidos:



El nuevo presidente del país, tras ganar las elecciones, tiene como objetivo decidir qué vías se deben "revitalizar" para manejar el tráfico ferroviario. En especial, se debe unir la terminal de Los Ángeles (LA) de forma directa con la de Chicago (CH) para dar cabida al intenso tráfico esperado. Por otra parte, todas las terminales restantes se deben enlazar, de forma directa o indirecta, de tal modo que se minimice

la longitud total de las vías seleccionadas. ¿Qué segmentos de vías de ferrocarril se deben incluir en el programa de revitalización?

Problema 4.14. La mayoría de los vecinos de cierto municipio trabaja en alguno de los siete pozos que una compañía minera explota cerca del municipio. El municipio, los pozos y las vías que los conectan (en km) están descritos en el gráfico siguiente:



Antes de las elecciones el actual alcalde prometió a todos los vecinos que pavimentaría algunos caminos de forma que cada trabajador tuviera pavimentado el camino más corto desde el municipio hasta su mina. ¿Cuántos kilómetros, como mínimo, tiene que pavimentar el alcalde para cumplir su promesa?

5. PERT-CPM

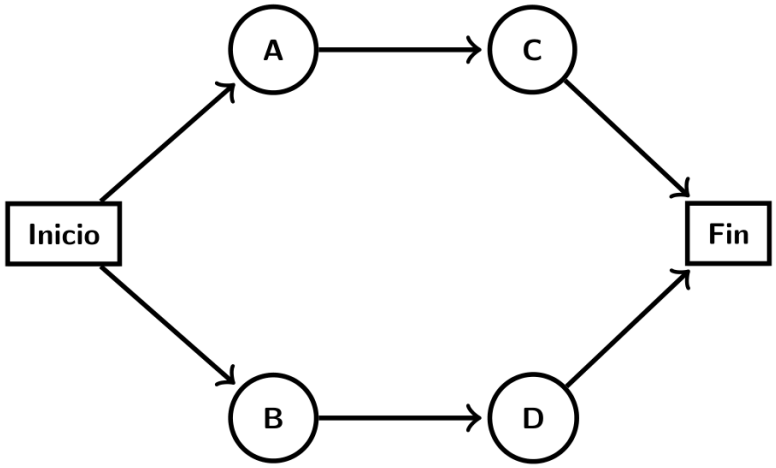
Problema 5.1. Para la cena que tenemos esta noche vamos a preparar una lasaña. Los diferentes procesos que tenemos que llevar a cabo, así como la duración y precedencia se muestran en la siguiente tabla.

Tarea	Descripción	Precedencia	Duración (min)
A	Comprar queso		30
B	Rallar el queso	A	5
C	Batir huevos		2
D	Mezclar el ricota con los huevos	C	3
E	Cortar la guarnición		7
F	Preparar la salsa	E	25
G	Hervir agua		15
H	Hervir la pasta	G	10
I	Escurrir la pasta	H	2
J	Mezclar ingredientes	I, F, D, B	10
K	Precalentar horno		15
L	Hornear la lasaña	J, K	30

1. Construya la red que modela el problema.

2. Encuentre las holguras de cada tarea y el camino crítico.

Problema 5.2. Nuestra empresa quiere realizar un proyecto en 12 meses. Este proyecto consta de 4 actividades (A, B, C y D) cuyas dependencias se muestran en la siguiente red, ver Figura . Minimice el coste de realizar el proyecto en 12 meses.



Tarea	Duración normal (meses)	Duración Crashing (meses)	Coste normal ()	Coste Crashing ()
A	8	5	25000	40000
B	9	7	20000	30000
C	6	4	16000	24000
D	7	4	27000	45000

6. Programación no lineal

Problema 6.1. Para cada uno de los siguientes problemas de programación no lineal:

1.
$$\begin{array}{ll} \min & 3x_1+2x_2 \\ \text{s.t.} & \left(\frac{x_1-1}{3}\right)^2+\left(\frac{x_2-3}{2}\right)^2 \leq 1 \\ & 2x_1+4x_2 \leq 17 \\ & x_1,x_2 \in \mathbb{R} \end{array}$$

2.
$$\begin{array}{ll} \max & 3x_1+2x_2 \\ \text{s.t.} & \left(\frac{2x_1+1}{3}\right)^2-3x_2 \geq 1 \\ & 4x_1+7x_2 \leq 30 \\ & x_1,x_2 \in \mathbb{R}^+ \end{array}$$

3.
$$\begin{array}{ll} \min & x_1-3x_2 \\ \text{s.t.} & 2x_1+5x_2 \geq -1 \\ & 5e^{\frac{x+1}{2}}-4x_2 \geq -4 \\ & x_1 \leq 0, x_2 \geq 0 \end{array}$$

4.

$$\begin{array}{lll} \max & x_1 + \frac{x_2}{2} & \\ \text{s.t.} & x_2 - \sqrt{3x_1+4} & \leq 0 \\ & x_2 - x_1^2 & \geq -2 \\ & x_1, x_2 \geq 0 & \end{array}$$

5.

$$\begin{array}{lll} \max & 20x_1 + 10x_2 & \\ \text{s.t.} & x_1^2 + x_2^2 & \leq 1 \\ & x_1 + 2x_2 & \leq 2 \\ & x_1, x_2 \in \mathbb{R} & \end{array}$$

Se pide:

a.) Esbozar la región factible y las curvas de nivel de la función objetivo.

b.) Formular las ecuaciones de Karush-Kuhn-Tucker.

c.) Resolver el problema empleando el solver adecuado.

Problema 6.2. Consideremos los siguientes problemas no lineales en $\mathbb{R}^2$

$$\begin{array}{lll} \min z & = 3x_1 + 4x_2 & \\ \text{s.t.} & \left(\frac{x_1}{3}\right)^2 + \left(\frac{x_2}{2}\right)^2 & \leq 4 \\ & (x_1-3)^2 + (x_2-3)^2 & \leq 6 \\ & 2x_1 - 3x_2 & \geq -5 \\ & x_1, x_2 \in \mathbb{R} & \end{array}$$

1. Formular las ecuaciones de Karush-Kuhn-Tucker.
2. Resolverlos aplicando el solver adecuando.
3. Plotear la solución óptima y la región factible.
4. Calcular el valor de los multiplicadores u_i^* en el óptimo.

Problema 6.3. Cierto tipo de cáncer requiere para su tratamiento dos tipos de haces de radiación, R_1 y R_2. Sin embargo la exposición a esta radiación cause daño colateral en los tejidos, de acuerdo con la función $x_1^4 + 2x_1^2 + 4x_2^2$, donde x_i representa las unidades de radiación de cada tipo R_i $(\forall i \in \{1,2\})$.

Para que el tratamiento sea efectivo el tumor debe recibir al menos 10 unidades de radiación de tipo A y al menos 10 unidades de radiación de tipo B. Cada unidad del haz R_1 entrega 2 unidades de A y 1 unidad de B, mientras que cada unidad del haz R_2 entrega 1 unidad de A y 2 unidades de B.

1. Determinar cuántas unidades de cada haz de radiación deben aplicarse para minimizar el daño colateral al tejido.
2. Formular las condiciones de Karush–Kuhn–Tucker (KKT) para este modelo.
3. Calcular los valores de los multiplicadores u_i^* en el óptimo.
4. Para mejorar el rendimiento, el manual sugiere que el número de unidades de radiación debe ser discreto. ¿Cuál sería la nueva solución? Pista: Utilice `ga` como solver en Matlab.

Problema 6.4. Resuelva los siguientes problemas aplicando el solver adecuado.

1.
$$\max_{x_1,x_2} f(x_1,x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$$

2.
$$\begin{aligned} \min_{x_1,x_2} \quad & f(x_1,x_2) = x_1^2 - x_2 \\ \text{s.t.} \quad & x_1 + x_2 = 2 \\ & x_1^2 + x_2^2 \leq 4 \\ & -x_1 + 1 \leq 0 \end{aligned}$$

3.
$$\begin{aligned} \max f(x_1,x_2) &= -x_1^2 + x_2 \\ \text{s.t. } & x_1^2 + x_2 \leq 10 \\ & x_1 \geq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

4.
$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & f(x_1,x_2,x_3) = x_1^2 + x_2^2 + x_3^2 \\ \text{s.t.} \quad & 2x_1 + x_2 - 5 \leq 0 \\ & x_1 + x_3 - 2 \leq 0 \\ & 1 - x_1 \leq 0 \\ & 2 - x_2 \leq 0 \\ & -x_3 \leq 0 \end{aligned}$$

Problema 6.5. Encuentre el máximo y el mínimo de los siguientes problemas aplicando las condiciones de KKT.

1.
$$\begin{aligned} f(x_1,x_2) \quad & = -2(x_1 - 2)^2 - x_2^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 25 \\ & x_1 \geq 0 \end{aligned}$$

2.
$$\begin{aligned} f(x_1,x_2) &= \log(x_1 + 1) + x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0 \end{aligned}$$

7. Teoría de colas

Problema 7.1. Una empresa de electrónica tiene su propio taller de reparación. El centro puede reparar a razón de un único ordenador cada vez, pero puede disponer de hasta 8 técnicos para esta tarea. El tiempo promedio de reparación depende de la cantidad de técnicos que intervengan y corresponde a $T_s = 0.5e^{-0.1n}$ horas, donde n es el número de técnicos asignados. La tasa promedio de llegada de ordenadores es de 2.5 por hora. Se sabe que el coste de tener ordenadores no operativos es de 30 €/h y el salario de cada técnico es de 15€/h.

1. Determinar la cantidad mínima de técnicos requerida.
2. Determinar la cantidad óptima de técnicos

Problema 7.2. A un sistema de colas, con capacidad limitada para un máximo de 4 clientes, llegan usuarios a una tasa media de 6 cl/h, siendo la velocidad de atención, en promedio, 8 cl/h. Si cada servicio se abona a 100€.

1. Determinar las pérdidas producidas por los clientes que no entran en la cola.
2. Determinar la longitud promedio del sistema y de la cola.
3. Determinar el tiempo medio de permanencia en el sistema

Problema 7.3. Una empresa de transporte tiene plataformas de carga. A la empresa llega en promedio 5 camiones por hora y cada plataforma puede cargar, en promedio, 2 camiones por hora. Se estima que el lucro cesante por cada camión a la espera de ser atendido es de 50€/h. Cuando el camión está siendo cargado se considera operativo por lo que no cuenta para el lucro cesante. Por su parte, el coste de tener activa una plataforma es de 30€. Determinar el número óptimo de plataformas de carga

Problema 7.4. En un taller mecánico llegan vehículos para una puesta a punto antes de pasar la ITV, las llegadas siguen un proceso de Poisson de promedio 18 vehículos/hora. Las dimensiones del taller sólo permiten que haya 4 vehículos, y las ordenanzas municipales no permiten esperar en la vía pública. El taller despacha un promedio de 6 vehículos/hora de acuerdo con una distribución exponencial. Se pide:

1. Probabilidad de que no haya ningún vehículo en el taller. Calcular las probabilidades
2. Promedio de vehículos en el taller
3. ¿Cuánto tiempo pasa por término medio un vehículo en el taller?
4. ¿Cuánto tiempo esperan por término medio los vehículos en la cola?
5. ¿Cuál es la longitud media de la cola?

Problema 7.5. Un asesor fiscal dispone de un local para atender a sus clientes, los cuales se concentran mayoritariamente entre los meses de mayo y junio. El local tiene una capacidad máxima de 8 asientos en espera, el cliente se va si no encuentra un asiento libre, y el tiempo entre llegada de clientes se puede considerar distribuido exponencialmente según un parámetro $\lambda = 20$ clientes por hora en período punta. El tiempo de una consulta esta distribuido exponencialmente con una media de 12 minutos. **Nota**: Es un sistema con $K = 9$, ya que son 8 asientos en espera más el que está siendo atendido.

1. ¿Cuántas consultas por hora realizará en promedio?
2. ¿Cuál es el tiempo medio de permanencia en el local?

Problema 7.6. Una estación de lavado tiene 3 máquinas que lavan los autos. El tiempo medio de servicio a cada cliente es 2.5 minutos, mientras que el promedio de autos que llegan a la estación es de 65 automóviles/hora, y esperan en una sola cola a que los atiendan. Suponiendo que los tiempos entre llegada y los tiempos de servicio se distribuyen de forma exponencial, se pide:

1. Identifique el sistema de colas que puede aplicarse a la estación de lavado.
2. Número medio de autos en el sistema.
3. Porcentaje de ocupación del sistema.
4. Calcule el tiempo medio de espera en cola.
5. Probabilidad de que un auto que llega a la estación no tenga que esperar.

Problema 7.7. Una sucursal bancaria cuenta con dos empleados de oficina igualmente eficientes, capaces de atender un promedio de 60 operaciones por hora, con un tiempo de servicio distribuido exponencialmente. Los clientes llegan a la sucursal siguiendo un proceso de Poisson con una tasa media de 100 clientes por hora. Determine:

1. Probabilidad de que haya estrictamente más de 3 clientes simultáneamente en la sucursal.
2. Probabilidad de que al menos uno de los empleados esté inactivo.

Problema 7.8. Una asesoría cuenta con 5 asesores. En promedio, llegan clientes cada 10 minutos y la duración media de las consultas es de 20 minutos. Determinar:

1. La probabilidad de que un cliente no tenga que esperar.
2. Porcentaje de ocupación de los asesores.
3. Número medio de clientes en la asesoría.
4. Ganancias esperadas, si cada consulta se cobra a 25 €.
5. Tiempo promedio de permanencia de cliente en la asesoría.

Problema 7.9. En un sistema M/M/3/-/10, cada cliente permanece en promedio 4 horas fuera del sistema. La duración media del servicio es de 30 minutos. Determinar

1. Probabilidad de que todos los canales se encuentren ocupados.
2. Porcentaje de clientes fuera del sistema.
3. Cantidad de clientes que se atienden por hora, en promedio.
4. Longitud promedio de la cola.
5. Tiempo de espera promedio de un cliente en cola.
6. Tiempo promedio de permanencia dentro del sistema

Problema 7.10. Supongamos que el sistema mostrado en la Fig. 1 cuenta con 4 unidades operativas de la forma M/M/1 sin impaciencia, con velocidades promedio de atención de 15, 20, 4 y 20 cl/hora para los centros I, II, III, IV, respectivamente. Determinar

1. Tasa de clientes que salen atendidos de los centros II y IV.
2. Longitud promedio de clientes para cada estación.
3. Longitud promedio de clientes para el sistema completo.
4. Tiempos promedio de espera en cola y de permanencia en cada centro.
5. Tiempo de espera en cola y de permanencia global.
6. Tiempos de espera en cola y permanencia para los clientes que salen de II y IV.

Problema 7.11. Dado el sistema mostrado en la Fig. 2 siguiente determine:

1. Probabilidad de que el sistema se encuentre vacío.
2. Tiempo promedio de permanencia de los clientes en el sistema.
3. Tiempo promedio de permanencia de los clientes que salen de B y de C.
4. Tiempo promedio de atención de un cliente.

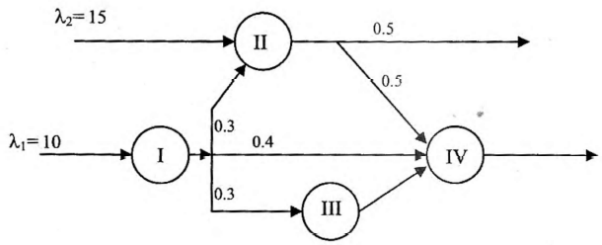


Figure 1: Sistema de 4 unidades operativas M/M/1 sin impaciencia.

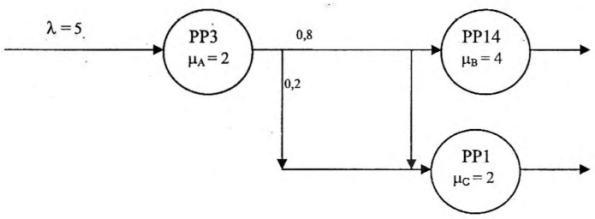


Figure 2: Sistema de red con 3 subsistemas con 1 o más canales.

8. Exámenes resueltos

En esta sección se resuelven exámenes de años anteriores.

8.1. Primer Examen Parcial

Problema 8.1.1. Dado el siguiente problema de programación lineal en $\mathbb{R}^2$:

$$\begin{aligned} \min\ & z = x_1 - x_2 \\ \text{s.a.}\ & 2x_1 - x_2 \leq 4 \\ & 3x_2 + 2x_1 \leq 12 \\ & 2 - x_2 \geq 0 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Se pide

1. Representar el problema de forma canónica (matricial) indicando cada uno de los elementos.

   Solución. Escribir problema en forma canónica

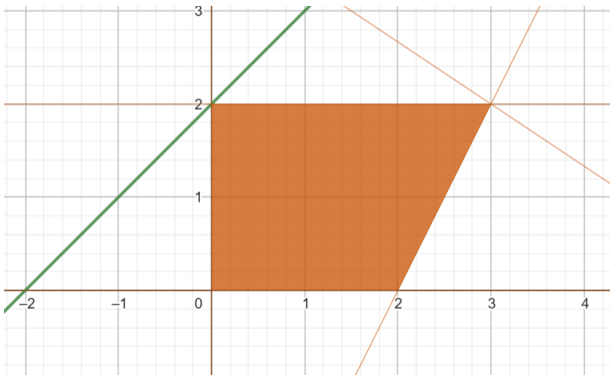
$$\begin{aligned} \min\ & z = x_1 - x_2 \\ \text{s.a.}\ & 2x_1 - x_2 \leq 4 \\ & 2x_1 + 3x_2 \leq 12 \\ & x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

   Luego $\mathbf{x} = [x_1\ x_2]^t$, $c = f = [1\ -1]^t$, $A = \begin{bmatrix} 1 & -1 \\ 2 & 3 \\ 0 & 1 \end{bmatrix}$ y $b = [4\ 12\ 2]^t$ EL problema resulta en

$$\begin{aligned} \min\ & z = c^t \mathbf{x} \\ \text{s.a.}\ & A\mathbf{x} \leq b \\ & \mathbf{x} \geq 0 \end{aligned}$$

2. Representar la región factible.

   La representación gráfica resulta



3. Resolver el problema utilizando el método analítico. Para calcular las intersecciones puede auxiliarse de Geogebra.

   La región factible queda determinada por restricciones. El óptimo en un problema de programación lineal está en los vértices de la región factible. En este caso los vértices factibles y su valor son

x	z
$(3,2)$	1
$(2,0)$	2
$(0,0)$	0
$(0,2)$	-2

Se observa que el valor mínimo se obtiene en $x^* = (0,2)$ para $z^* = -2$.

Problema 8.1.2. Una compañía constructora necesita recibir piedras para completar tres obras. Para ello contrata a una empresa de transporte y compra la piedra a dos canteras. Cada cantera vende la tonelada de piedra a 300 y 420 dólares la tonelada, respectivamente. La compañía de transporte cobra diferente para cada ruta por tonelada de piedra transportada, los precios (siempre por tonelada de piedra), están reflejados en la siguiente tabla.

	Obra 1	**Obra 2**	**Obra 3**	**Cantidad disponible** (ton)	**Precio de compra**
Cantera 1	\$100	\$90	\$160	18	\$300
Cantera 2	\$180	\$110	\$140	14	\$420
Cantidad requerida (ton)	10	5	10		

Table 1: Precio por tonelada

1. Formula el modelo de programación lineal adecuado para minimizar el costo total de adquirir y transportar la piedra, respetando la oferta de cada cantera y la demanda de cada obra.

2. Resuelve el modelo anterior empleando Matlab.

3. La cantera 2 decide aumentar su producción a 15 toneladas de piedra. Como influye dicha decisión en el coste total de la transportación. Justifique adecuadamente sin volver a calcular la solución del problema.

4. Después de presentar el plan, la compañía de transporte informa que cierra la ruta entre la cantera 2 y la obra 3. ¿Cómo incorporaríamos esta condición al modelo? ¿Cuál sería la planificación y el coste total de la operación en este caso?

Solución

Este problema consiste en un problema de planificación de transporte en el que hay que minimizar costes

1.
   - Índices de trabajo: canteras $I = \{1,2\}$ y obras $J = \{1,2,3\}$
   - Variables de decisión x_{ij} toneladas de piedra que se transportan de cantera i a obra j, $x_{ij} \geq 0$ para todo $(i,j) \in I \times J$.
   - El coste de mover piedra de una cantera i a una obra j será la suma entre el coste de transportación más el coste de compra en cada cantera $c_{ij} + p_i$, luego la función de costes resulta

     $$\min z = \sum_{i=1}^{2} \sum_{j=1}^{3} (c_{ij} + p_i) x_{ij}$$

     en dólares, donde c_{ij} es el coste por tonelada transportada de cantera i a obra i y p_i es el precio del material en la cantera i.
   - Restricciones de demanda

     $$\sum_{i=1}^{2} x_{ij} \geq d_j, \quad \forall j \in J$$

     donde d_j es la demanda de la obra j.
   - Restricciones de oferta

     $$\sum_{j=1}^{3} x_{ij} \leq o_i, \quad \forall i \in I$$

     donde d_j es la demanda de la obra j.
   - No negatividad $x_{ij} \geq 0$ para todo $(i,j) \in I \times J$.

   El modelo resulta

   $$\begin{aligned} \min z &= \sum_{i=1}^{2} \sum_{j=1}^{3} (c_{ij} + p_i) x_{ij} \\ \text{s.a.} \sum_{i=1}^{2} x_{ij} &\geq d_j, \quad \forall j \in J \\ \sum_{j=1}^{3} x_{ij} &\leq o_i, \quad \forall i \in I \\ x_{ij} &\geq 0 \end{aligned}$$

2. Resultado

   $$\begin{aligned} x^* &= [10\ 5\ 3\ 0\ 0\ 7]^t \\ z^* &= 11250 \end{aligned}$$

   Conclusión: Para un gasto mínimo de 11250 dólares el plan es:

   - 10 toneladas en cantera 1 para obra 1
   - 5 toneladas en cantera 1 para obra 2

- 3 toneladas en cantera 1 para obra 3
- 7 toneladas en cantera 2 para obra 3

3. El precio sombra de la restricción asociada a la oferta de la cantera 2 es cero, por lo que aumentar la disponibilidad de este recurso no tendrá influencia en el valor de la función objetivo. Por lo que concluimos que el coste de transportación seguirá siendo el mismo.

4. Para incorporar la restricción al modelo se puede, o bien fijar $x_{23} = 0$ o tomar $c_{23} = M = 1000$. La solución será

$$x^* = [8,\ 0,\ 10,\ 2,\ 5,\ 0]^t$$
$$z^* = 11650$$

Problema 8.1.3. Determine si las siguientes proposiciones son verdaderas o falsas. Corrija las falsas.

1. _____ Si un problema de programación lineal (PL) tiene una solución óptima, entonces existe una solución óptima en un vértice (punto extremo) del poliedro factible. **Verdadero**

2. _____ Todo problema de programación lineal con región factible no vacía tiene solución. **Falso.** Considere el siguiente contraejemplo $\max x$.

3. _____ El relajado continuo de un problema de programación lineal entera proporciona una cota inferior para problemas de maximización y una cota superior para problemas de minimización. **Falso**. Cota inferior para minimización y superior para maximización.

4. _____ Si en un problema de programación lineal entera la solución óptima del relajado continuo es entera, entonces coincide con la solución del problema entero. **Verdadero**

5. _____ La solución de un problema de programación lineal en variable continua siempre está exclusivamente en los vértices del poliedro que forma la región factible. **Verdadero**

8.2. Segundo Examen Parcial

Problema 8.2.1. Se nos encarga el diseño de una red de fibra para conectar ocho nodos de transmisión. La instalación de 1 km de fibra cuesta 1000€. El siguiente grafo en figura 3 representa en cada vértice un nodo, cada arista representa una posible conexión. El peso de la arista representa la cantidad de kilómetros de fibra necesarios para conectar los nodos que definen la arista.

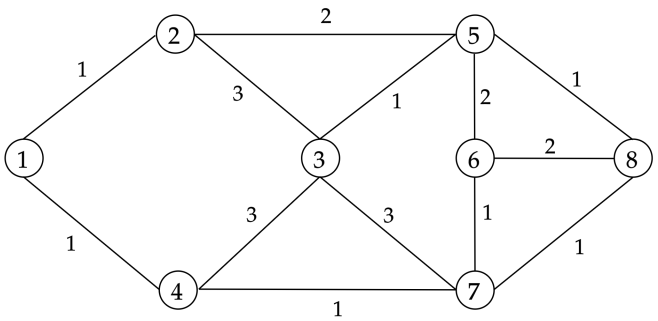
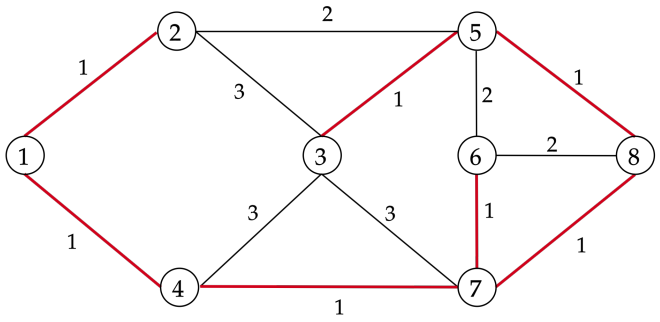


Figure 3: Red fibra ocho nodos

Se considera que dos nodos están conectados si existe al menos un camino entre ellos.

1. Diseñe la red de coste mínimo que conecta todos los nodos.
2. ¿Cuál es el coste de instalación de la red?

Solución



- Aplicar algoritmo de Prim para encontrar árbol de expansión mínimo, ver Figura 8.2.
- 7000€

Problema 8.2.2. Se debe hacer un viaje en automóvil desde la ciudad **Origen**, denotada con O, hasta la ciudad **Destino**, denotada como T. La siguiente tabla almacena las distancias en kilómetros entre cada una de las ciudades. Se omite repetición debajo de la diagonal. El guión indica que no hay conexión directa entre las ciudades sin pasar por otras.

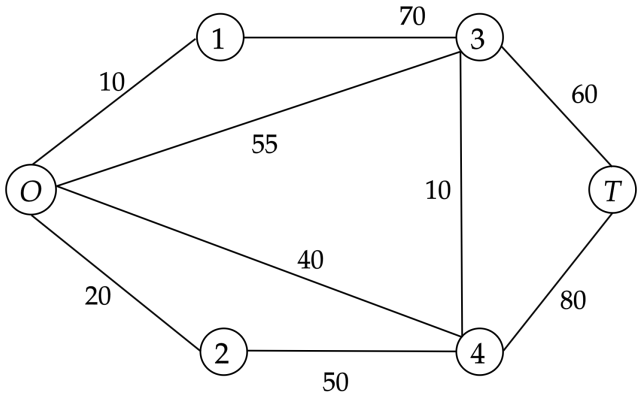
Ciudad	1	2	3	4	T
O	10	20	55	40	–
1		–	70	–	–
2			–	50	–
3				10	60
4					80

1. Determinar el grafo que modela el problema.
2. Determinar el modelo matemático que permite resolver el problema de encontrar el camino más corto entre origen y destino.
3. Resolver el modelo mediante el uso de Matlab.

Solución.

1. Grafo del problema, ver Figura 1.
2. El objetivo de este problema es encontrar la ruta más corta entre la ciudad origen y destino. Debemos plantearlo como un problema de flujo de costo mínimo enviando un flujo unitario desde el nodo origen hasta el nodo destino.

   Se define el conjunto de índices $i, j \in \{O, 1, 2, 3, 4, T\}$.



Variables binarias $x_{ij} = \begin{cases} 1 & \text{el camino i-j se incluye en la ruta} \\ 0 & \text{en otro caso} \end{cases}$, por lo que presentan las siguientes restricciones de caja $0 \leq x_{ij} \leq 1$. De acuerdo con la tabla de distancias se tendrán en cuenta las siguientes 10 variables

$$x_{O1}, x_{O2}, x_{O3}, x_{O4}, x_{13}, x_{24}, x_{34}, x_{3T}, x_{43}, x_{4T}$$

Nótese que la arista 34 se considera bidireccional.

Se define el siguiente vector de costes $c = [c_{ij}]$, donde c_{ij} representa la distancia de viajar de la ciudad i a la j.

$$c = [10\ 20\ 55\ 40\ 70\ 50\ 10\ 60\ 10\ 80]$$

Luego la función objetivo es

$$\min z = \sum_{i,j} c_{ij} x_{ij} = 10x_{O1} + 20x_{O2} + 55x_{O3} + 40x_{O4} + 70x_{13} + 50x_{24} + 10x_{34} + 60x_{3T} + 10x_{43} + 80x_{4T}$$

que será la distancia total de la ruta escogida en kilómetros.

Restricciones sobre origen: se envía una unidad de flujo

$$x_{O1} + x_{O2} + x_{O3} + x_{O4} = 1$$

Restricciones sobre nodos de tránsito, todo flujo recibido se envía

$$\begin{aligned} -x_{O1} + x_{13} &= 0 \\ -x_{O2} + x_{24} &= 0 \\ -x_{O3} - x_{13} + x_{34} - x_{43} &= 0 \\ -x_{O4} - x_{24} - x_{34} + x_{43} + x_{41} &= 0 \end{aligned}$$

Restricciones sobre nodo destino: se recibe una unidad de flujo

$$-x_{3T} - x_{4T} = -1$$

Las unidades de medida de la función objetivo están dadas en kilómetros.

Table 2: Clasificación de soluciones del sistema KKT

x	y	λ_1	λ_2	Factible	Máx/Mín	Valor
3	0	0	0	Sí	Máx	0
0	0	0	18	Sí	–	-27
5	0	6/5	0	Sí	Mín	-12
0	-5	1	18	Sí	Mín	-52
0	5	1	18	Sí	Mín	-52
-5	0	24/5	0	No	–	–
9/2	$-\sqrt{19}/2$	1	0	Sí	No	-11.5
9/2	$\sqrt{19}/2$	1	0	Sí	No	-11.5

3. [1] **Función predefinida de Matlab**

   Para resolver este problema se ha empleado la función `intlinprog` de Matlab que permite resolver problemas de programación lineal entera y entera mixta. La función está programada para buscar el mínimo y para restricciones de tipo menor o igual.

   [2] **Comentar la solución obtenida**

   Tras introducir el modela matemático en Matlab se obtiene que la ruta más corta conecta origen con ciudad 4, luego con ciudad 3 y finalmente de ciudad 3 se viaja a destino. $O43T$

   [3] **Comentar el valor de la función objetivo**

   De esta manera el trayecto es de 110 km en total.

Problema 8.2.3. Encuentre los máximos y los mínimos utilizando las condiciones KKT.

$$\begin{aligned} f(x_1,x_2) &= -3(x-3)^2 - y^2, \\ \text{s.t.} \quad & x^2+y^2 \leq 25, \\ & x \geq 0. \end{aligned}$$

Solución:
Las ecuaciones de KKT son:

$$\begin{aligned} 2\lambda_1 * x - 6x - \lambda_2 + 18 &= 0 \\ 2\lambda_1 y - 2y &= 0 \\ \lambda_1(x^2+y^2-25) &= 0 \\ -\lambda_2 x &= 0 \end{aligned}$$

La tabla de soluciones del sistema es, ver Tabla 2

Problema 8.2.4. Dado el sistema mostrado en la Fig 4, con tasas de salida de 20, 20 y 7 para los subsistemas A, B y C, respectivamente, hallar

1. Tasa de clientes que entran en cada subsistema.
2. Probabilidad de que el sistema esté vacío.
3. Longitud promedio de clientes para el sistema completo.
4. Tiempo promedio de permanencia de los clientes en el sistema.

Solución

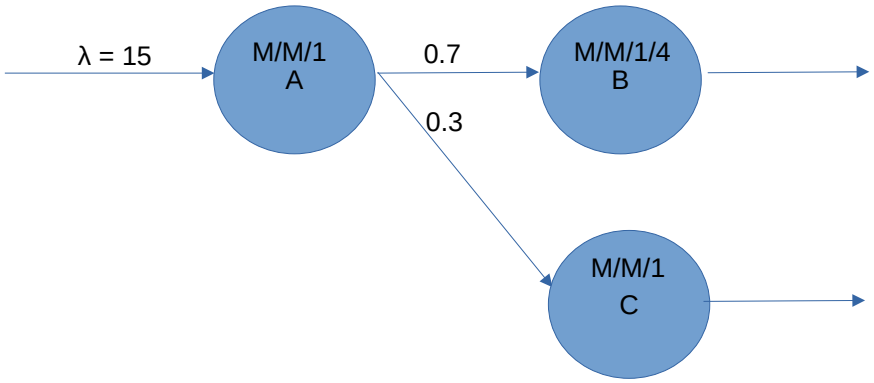


Figure 4: Sistema de colas

1. $\overline{\lambda_A} = 15$, $\overline{\lambda_B} = 10.11$ y $\overline{\lambda_C} = 4.5$
2. $p(0) = p_A(0) \cdot p_B(0) \cdot p_C(0) = 0.25 \cdot 0.4947 \cdot 0.3571 = 0.0442$
3. $L = L_A + L_B + L_C = 5.6976$
4. $W = 0.6888$

9. Soluciones seleccionadas

Soluciones de problemas de Programación Lineal

Problema 1.1. Given the following linear programming problem, find the maximum and minimum of the objective function:

1. Maximum: $x^* = (0,6)$ for $z^* = 12$.

   Minimum: $x^* = (0,2)$ for $z^*_{\min} = 4$.

   See Geogebra-solution

2. Unfeasible region see Geogebra-solution

Problema 1.2. Solve the following Linear Programming Problems analytically, graphically and using Matlab solver. (solutions must be the same)

1. Analytically

x	z
$(0,0)$	0
$(2,0)$	2
$(0,2)$	2

   Infinite solutions case, $z^* = 2$ in the segment $x^*(t) = (2,0)t + (0,2)(1-t)$ for $t \in [0,1]$.

   Graphical solution, see Geogebra-solution

2. Analytically

x	z
$(0,0)$	0
$(0,5/13)$	$25/13$
$(0,5)$	10

   Maximum: $x^* = (0,5)$ for $z^* = 10$.

   Graphical solution, see Geogebra-solution

3. Analytically

x	z
$(0,0)$	0
$(0,4)$	4
$(4,0)$	4
$(4,4)$	8

   Maximum: $x^* = (4,4)$ for $z^* = 8$.

   Graphical solution, see Geogebra-solution

4. Analytically

x	z
$(0,0)$	0
$(0,10)$	100
$(15,0)$	120
$(10,5)$	130

Maximum: $x^* = (10,5)$ for $z^* = 130$.

Graphical solution, see Geogebra-solution

5. Analytically

x	z
$(0,0)$	0
$(3,0)$	9
$(4.2,0.6)$	15.6

Maximum: $x^* = (4.2,0.6)$ for $z^* = 15.6$.

Graphical solution, see Geogebra-solution

6. Analytically

x	z
$(0,0)$	0
$(4.2,0.6)$	3
$(3,0)$	3

Maximum: $z^* = 3$ with infinite solutions in the segment $x^*(t) = (3,0)t + (1-t)(4.2,0.6)$.

Graphical solution, see Geogebra-solution

Problema 1.3.

1. Graphical solution, see Geogebra-solution. Minimum at $x^* = (1.8,0.6)$ for $z^* = 7.8$
2. Use the following python code

Problema 1.4. Graphical solution, see Geogebra-solution. Note that the level curve of the objective function is parallel to one of the boundaries joining the points $r_1 \cap r_2$ and $r_2 \cap r_3$, where r_i are

$$\begin{aligned} r_1 &: 15x_1 + 5x_2 \leq 300 \\ r_2 &: 10x_1 + 6x_2 \leq 240 \\ r_3 &: 8x_1 + 12x_2 \leq 450 \end{aligned}$$

Note that $r_1 \cap r_2 = (15,15)$ and $r_2 \cap r_3 = (2.5,35.83)$, thus this problem has infinite solutions of the form

$$\ell(t) = (15,15)t + (2.5,35.83)(1-t)$$

where $t \in [0,1]$.

Problema 1.5. When drawing the region, points in the intersection between the first two constraints get negative values for the variables, thus making the feasible region empty. See Geogebra-solution

Problema 1.6. See the solution at Geogebra-solution. Given that the solution is $(2,3)$, set the level curve for the objective function at 8. We note that $(2,3)$ must be at one of the vertices of the feasible region or on the border, in case of infinite solutions. See by moving the slider for k, that we are in the case of infinite solutions for $k = 0.5$. Note that this value of k makes the constraint parallel to the objective function.

Thus we conclude that $k = 0.5$.

Problema 1.7.

1. Using columns refer to the decision variables and rows to the resources we have the following table

	Square	Rectangle	Available
Tape (m)	2	4	440
Polymer (m)	0.5	0.25	65
Revenue (€/m^2)	0.12	0.08	

2. First, write the model, taking x_1 : square meters of coating for square containers needed, and x_2 : square meters of coating for rectangle containers.

$$\begin{aligned} \max z = \quad & 0.12x_1 + 0.08x_2 & & \\ \text{s.t.} \quad & 2x_1 + 4x_2 & \leq 440 & \text{(tape used)} \\ & 0.5x_1 + 0.25x_2 & \leq 65 & \text{(polymer used)} \\ & x_1, x_2 & \geq 0 & \text{(used material must be positive)} \end{aligned}$$

The graphical solution can seen at Geogebra-solution

The mathematical solution yields $x^* = (100,60)$ for $z^* = 16.8$ which means that

Natural Language solution: To maximize the total production value, the laboratory must produce 100 m^2 of coating for square containers and 60 m^2 of coating for rectangular containers for a maximum value of 16.8 euros.

Problema 1.8. Let us begin by building a table summarizing the data in the problem

	Large	Small	Available
Active ingredient (g)	40	30	600
Revenue (€/tablet)	2	1	

Additionally, we are required to produce at least twice small-dose pills than the number of large pills.

Set the decision variables to be x_1: number of large pills to produce, and x_2: number of small pills to produce. Note that $x_1, x_2 \geq 0$, because we must produce a positive number of pills.

- **Objective function**: profit $\max z = 2x_1 + x_2$ in euros.
- **Active ingredient constraint**: $40x_1 + 30x_2 \leq 600$ in grams.
- **Production constraint:** $2x_1 \leq x_2$, small pills are at least twice the number of large pills

The graphical solution yields Geogebra-solution

Mathematical solution: $x^* = (6,12)$ and $z^* = 24$

Natural language solution: To maximize profit the laboratory must produce 6 large pills and 12 small pills for a total profit of 24 €

Problema 1.9.

	P_1 (Kg)	P_2 (Kg)	Requirements
Vitamin A (mg/Kg)	2	4	4
Vitamin B (mg/Kg)	6	3	6
Cost (cent/kg)	40	60	

- **Decision variables**: x_i : kilograms of supplement P_i, $i = 1,2$, where $x_i \geq 0$.
- **Objective function**: $\min z = \min_{\mathbf{x}} 40x_1 + 60x_2$ in cents.
- **Vitamin A requirements**: $2x_1 + 4x_2 \geq 4$,
- **Vitamin B requirements**: $6x_1 + 3x_2 \geq 6$,

The graphical solution yields Geogebra-solution
Mathematical solution: $x^* = (2/3, 2/3)$ and $z^* = 66.67$
Natural language solution: To minimize cost the nutricionist must mix $2/3$ Kg of each product for a total cost of 67 cents.

Problema 1.10.

	Type A	Type B
Cost (€/Kg)	0.5	0.8
Sell (€/Kg)	0.58	0.9
Profit (€/Kg)	0.08	0.1

- **Decision variables**: x_i : kilograms of each material to buy, $i = 1,2$, where $x_i \geq 0$.
- **Objective function**: Maximize profit = Sell price - cost price $\max z = \max_{\mathbf{x}} 0.08x_1 + 0.1x_2$ in euros.
- **Storage constraint**: $x_1 + x_2 \leq 700$,
- **Budget constraint**: $0.5x_1 + 0.8x_2 \leq 500$

The model yields

$$\begin{aligned} \max z = \quad & 0.08x_1 + 0.1x_2 & \\ \text{s.t.} \quad & x_1 + x_2 & \leq 700 \\ & 0.5x_1 + 0.8x_2 & \leq 500 \\ & x_1, x_2 \geq 0 & \end{aligned}$$

Mathematical solution: $x^* = (200, 500)$ and $z^* = 66$
Natural language solution: To maximize profit the laboratory must purchase 200 kilograms of material A and 500 Kg of material B for a profit of 66 euros.

Problema 1.11.

1. This is a model on proportions (percent).
   - **Decision variables**: x_i: proportion of polymer i, where $i = 1,2,3,4$ to be mixed. Note that since x_i is a percent, we have $0 \leq x_i \leq 1$.
   - **Objective function**: Minimize cost = $\min z = 43x_1 + 31x_2 + 47 * x_3 + 37x_4$ in monetary units.
   - **Mix constraint**: $x_1 + x_2 + x_3 + x_4 = 1$ (the mix is only made of these components),
   - **Additive A constraint**: $0.8x_1 + 0.3x_2 + 0.7x_3 + 0.4x_4 \geq 0.6$

- **Additive B constraint**: $0.1x_1 + 0.4x_2 + 0.2x_3 + 0.1x_4 \leq 0.3$

The model yields

$$\begin{aligned} \max z = \quad & 43x_1 + 31x_2 + 47x_3 + 37x_4 & & \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 & = 1 & \\ & 0.8x_1 + 0.3x_2 + 0.7x_3 + 0.4x_4 & \geq 0.6 & \\ & 0.1x_1 + 0.4x_2 + 0.2x_3 + 0.1x_4 & \leq 0.3 & \\ & 0 \leq x_i & \leq 1, \qquad & i = 1,2,3,4 \end{aligned}$$

2. **Mathematical solution:** $x^* = (0.6, 0.4, 0, 0)$ and $z^* = 38.2$

   Natural language solution: To minimize the cost of the blend, the company must mix 60% of polymer 1 with 40% of polymer 2 with a minimum cost of 38.2 monetary units per unit produced.

3. Perform a sensitivity analysis on additive prices and right-hand side (RHS) constraints.

Table 3: Variable summary from the solver output

	Solution	Reduced Cost	Objective Coefficient	Increase Margin	Decrease Margin
x_1^*	0.6	0	43	8	12
x_2^*	0.4	0	31	4.5	∞
x_3^*	0	6.4	47	∞	6.4
x_4^*	0	3.6	37	∞	3.6

Table 4: Constraint summary from the solver output

	Final Value	Shadow Price	RHS	Increase Margin	Decrease Margin
R1 L.H.S	1	23.8	1.0	0.1379	0.25
R2 L.H.S	0.6	24.0	0.6	0.2	0.1333
R3 L.H.S	0.22	0.0	0.3	∞	0.08

Important takeouts: The reduced cost of c_3 and c_4 indicate that if we want the components 3 and 4 to be part of the mixture, we should decrease their price by 6.4 and 3.6, respectively. Also, from the shadow price we conclude that the limit on additive C is not reached. On the other hand, the solution will be the same under the increase/decrease margins indicated in the first table, and it still be feasible inside the margins of the second table.

Problema 1.12.

1. See solution at Geogebra-solution

2. The extreme points are

x	z
$(0,0)$	0
$(0,3)$	9
$(2,0)$	10
$(1.05, 12.37)$	12.37

3. Perform a sensitivity analysis of the parameters b_1 and b_2 considering the attached table

Name	Final Value	Shadow Price	RHS	Increase Margin	Decrease Margin
R1 LHS	15	0.26315789	15	10	9
R2 LHS	10	0.84210526	10	15	4

Important takeouts: the shadow prices points that both of the constraints are active, also that an increase of one unit in the value of b_1 or b_2 will transform into an increase of 0.26315789 or 0.84210526 in the objective function, respectively. The coefficients of **b** can move inside the increase/decrease margins, without losing the feasibility of the current solution.

Problema 1.13.

1. Using matlab (or pyomo/GUI-solver) we find that the optimal solution is

$$\mathbf{x}^* = (5,0,3)^t \qquad \text{for} \qquad z^* = 30$$

This implies that to maximize profit the company must produce 5 units of product 1 and 3 of product 2 for a profit of 30 monetary units.

2. Consider now the situation of a decrease on the profit obtained from product 1, from 3 to 2. This is $\hat{c_1} = c_1 - 1$. Is c_1 allowed to decrease 1 unit without changing the optimal solution? The table of the post-optimal analysis associated with this solution yields, see Table 5,

	Solution	Reduced cost	Coefficient in Z	Increase Margin	Decrease Margin
x_1^*	5	0	3	0	∞
x_2^*	0	-3	1	3	∞
x_3^*	3	0	5	∞	0

Table 5: Post-optimal report of Example (1.13)

Note that c_1 is allowed to decrease infinitely, therefore, the optimal solution does not change under this reduction. On the other hand, the profit, that is, the optimal value would change, because $\hat{z^*} = \hat{c_1}x_1 + c_2x_2 + c_3x_3$, which gives

$$\hat{z^*} = 2(5) + 5(3) = 25$$

Thus the optimal solution would be $x^* = (5,0,3)^t$ for $\hat{z}^* = 25$.

3. In this part we deal with a problem where $\hat{b}_2 = b_2 + \Delta b_2 = 30 + 15$, from the acquisition of 15 additional units of sterile material. The cost of these units is $10 \cdot 15 = 150$ monetary units. The sterile material according to Table 6

Name	Final Value	Shadow Price	RHS	Increase Margin	Decrease Margin
C1 L.H.S	45	0	45	15	15
C2 L.H.S	30	1	30	15	7.5

Table 6: Post-optimal report of problem (1.13)

is a critical material, note that constraint 2 is active, $s_2^* = 1 > 0$. Thus, increasing the availability of sterile material changes the optimal value. Also note that, if the change is below 15, the shadow price is valid and the optimal solution still feasible, that is $\hat{z^*} = z^* + s_2 \cdot \Delta b_2$.

To know if the company should invest, the difference in profit, must be above the cost, that is $\hat{z^*} - z^* > 10$. The shadow price of b_2 is $s_2 = 1$, this means that for each unit of increase of b_2, z^* increases 1 unit, thus $\hat{z^*} - z^* = 5$. Thus, the profit increase beyond the cost of the new material, and it should be bought.

The company must invest in acquiring new material.

Problema 1.14.

1. - **Decision variables**: x_i: number of devices, $i = 1$ for diagnostic test and $i = 2$ for wristbands. Note that $x_i \geq 0$.
   - **Objective function**: Maximize profit = $\max z = 5x_1 + 4x_2$ euros.
   - **Time constraints**: $2x_1 + x_2 \leq 80$
   - **Material constraints**: $x_1 + x_2 \leq 50$

   The model yields

$$\begin{aligned} \max z = \quad & 5x_1 + 4x_2 & \\ \text{s.t.} \quad & 2x_1 + x_2 & \leq 80 \\ & x_1 + x_2 & \leq 50 \\ & x_1, x_2 \geq 0 & \end{aligned}$$

2. The region can be seen here, Geogebra-solution

3. **Mathematical solution**: $\mathbf{x}^* = (30, 20)$ and $z^* = 230$

   Natural Language solution: For a maximum profit of 230€the company must produce 30 test kits, and 20 wristbands.

Problema 1.15.

Suppliers	A	B
Coverage (m^2/L)	6	8
Price (€/L)	1	1.2

1. **Decision variables**: x_i : Litters of disinfectant of supplier $i = 1, 2$, for supplier A and B, respectively, to be purchased. $x_i \geq 0$.

   Objective function: Minimize cost = $\min z = \min_{\mathbf{x}} x_1 + 1.2x_2$ in euros.

- **Budget constraint**: $x_1 + 1.2x_2 \leq 120$,
- **Area requirement**: $6x_1 + 8x_2 \geq 480$,
- **Purchase requirement**: $x_i \leq 75$,

The model yields

$$\begin{aligned} \min z = \quad & x_1 + 1.2x_2 & \\ \text{s.t.} \quad & x_1 + 1.2x_2 & \leq 120 \\ & 6x_1 + 8x_2 & \geq 480 \\ & 0 \leq x_1, x_2 \leq 75 & \end{aligned}$$

2. See the graphical solution at Geogebra-solution

   Mathematical solution: $x^* = (0, 60)$ and $z^* = 72$

   Natural language solution: To minimize cost the hospital must purchase 60 litters of disinfectant from supplier B for a minimum cost of 72€

3. Considering the attached table corresponding to the optimal solution, what will be the total cost if Supplier B increases the price of its disinfectant to €1.3/L?

	Reduced cost	Coeficient in Z	Increase Margin	Decrease Margin
X_A^*	0.1	1	∞	0.1
X_B^*	0	1.2	0.13	1.2

According to the sensitivity report provided, b_2 is allowed to increase up to 0.13 units without changing the optimal solution. Therefore, if supplier B increase the price increases by 10 cents, that is $\hat{b}_2 = b_2 + 0.1$, then the cost of purchasing $x_2 = 60$ litters of disinfectant ascend to $\hat{z}^* = b_1 x_1^* + \hat{b}_2 x_2^* = 1.3 \cdot 60 = 78$ €.

Problema 1.16.

Type of kit	S	P	Weekly limits
Wrapping (m^2)	8	2	1200
Labor (hours)	12	5	1850
Profit (€)	350	120	

1. **Decision variables**: x_i : number of devices of type $i = 1,2$, for S and P, respectively, to be produced. $x_i \geq 0$.

   Objective function: maximize profit = $\max z = \max_{\mathbf{x}} 350x_1 + 120x_2$ in euros.

   - **Material constraint**: $8x_1 + 2x_2 \leq 1200$,
   - **Labor constraint**: $12x_1 + 5x_2 \leq 1850$,

   The model yields

$$\begin{aligned} \max z = \quad & 350x_1 + 120x_2 & \\ \text{s.t.} \quad & 8x_1 + 2x_2 & \leq 1200 \\ & 12x_1 + 5x_2 & \leq 1850 \\ & x_1, x_2 & \leq 0 \end{aligned}$$

2. **Mathematical solution:** $x^* = (143.75, 25)$ and $z^* = 53321.5$

   Natural language solution: For a maximize profit of 53321.5€, the company must produce 143.5 standard kits and 25 portable kits.

3. If the company plans to increase production, what is the maximum price it should pay for each additional unit of sterile wrapping material? And for each additional hour of labor?

   This part of the problem refers to the shadow price. Thus, obtaining the sensitivity report we have

Name	Final Value	Shadow Price	RHS	Increase Margin	Decrease Margin
R1 L.H.S	1200	19.375	1200	33.33	460
R2 L.H.S	1850	16.25	1850	1150	50

   Note that for the shadow price s_i, refers to the increase in profit if the corresponding b_i is increased in 1 unit. That is if $\hat{b}_i = b_i + 1$, then $\hat{z}^* = z^* + s_i = 53321.5 + 19.375 = 53340.875$. This means that for each unit of sterile wrapping material the company could pay up to 19.375 euros, and still being profitable. Note that this is valid to an increase up to 33.33 hours.

4. This item refers to changes in the coefficients of the objective function. Thus, the sensitivity report yields

	Solution	Reduced cost	Coefficient in Z	Increase Margin	Decrease Margin
x_1^*	143.75	0	350	130	62
x_2^*	25	0	120	25.83333333	32.5

We can easily see that an increase of sale price c_1, of 10 €, in the Type S kits, does not changes the optimal solution, because, this price can be increased up to 130 units without changing the optimal solution. Therefore,

$$\hat{z}^* = \hat{c}_1 x_1^* + c_2 x_2^* = (c_1 + 10)x_1^* + c_2 x_2^* = (360 + 10)143.75 + 120(25) = 56187.5$$

Problema 1.17.

Suppliers	A	B
Buy price (€/L)	0.5	0.8
Sell price (€/L)	0.58	0.9
Profit (€/L)	0.08	0.1

1. **Decision variables**: x_i : Litters of disinfectant $i = 1, 2$, for supplier A and B, respectively, to be purchased. $x_i \geq 0$.

   Objective function: maximize profit = $\max z = \max_{\mathbf{x}} 0.08x_1 + 0.1x_2$ in euros.

   - **Budget constraint**: $0.5x_1 + 0.8x_2 \leq 500$,
   - **Volume constraint**: $x_1 + x_2 \leq 700$,

   The model yields

$$\begin{aligned} \max z = \quad & 0.08x_1 + 0.1x_2 & \\ \text{s.t.} \quad & 0.5x_1 + 0.8x_2 & \leq 500 \\ & 6x_1 + x_2 & \leq 700 \\ & x_1, x_2 \geq 0 & \end{aligned}$$

2. Using matlab/Pyomo/GUI/Geogebra we obtain **Mathematical solution:** $x^* = (13.95, 616.28)$ and $z^* = 62.74$

   Natural language solution: To minimize cost the supplier must purchase 13.95 litters of disinfectant of type A and 616.28 of type B for a maximum profit of 62.74€

3. This part refers to the shadow price s_2. Consider the post-optimal report below

Name	Final Value	Shadow Price	RHS	Increase Margin	Decrease Margin
R1 L.H.S	500	0.120930233	500	60	441.6666667
R2 L.H.S	700	0.003255814	700	5300	75

   Note that $s_2 = 0.003255814$ and an increase of 200 litters is inside the margin of allowed variability. Thus, an increase of 200 units for b_2 would imply that

$$\hat{z}^* = z^* + 200s_2 = 62.74 + 200 \cdot 0.003255814 \approx 63.39$$

   Therefore, the expected profit after this change would be 63.39€.

4. To evaluate the impact of an increase of budget we may refer to $s_1 = 0.120930233$. The proposed increase of 40€, is inside the allowed limits. Therefore,

$$\hat{z}^* = z^* + 40s_1 = 62.74 + 40 \cdot 0.120930233 \approx 67.58$$

5. Last, to evaluate the effects of an increase in profit if the sell price of disinfectant B is increased to $0.92e/L$, we may refer to the first table of the sensitivity report, that is

	Solution	Reduced Cost	Coefficient in Z	Increase Margin	Decrease Margin
x_1^*	13.95348	0	0.08	0.52	0.0175
x_2^*	616.2790	0	0.10	0.028	0.086666667

Note that increasing the sell price, would increase the profit coefficient to $\hat{c}_2 = c_2 + 0.02$, which is allowed within the limits of this optimal solution (that is, this change, doesn't affect x^*). Therefore, the new total profit is computed as

$$z^* = c_1 x_1^* + \hat{c}_2 x_2^* = 0.08(13.95) + (0.12)(616.28) = 75.06$$

Therefore, the profit would increase to 75.06€.

Problema 1.18.

1. - **Decision variables**: x_i : number of ads of each type the company buys a week, where $i = 1,2,3,4$, for TV, Journal, Radio and Online, respectively, type of ad. $x_i \geq 0$.
   - **Objective function**: maximize audience = $\max z = \max_{\mathbf{x}} 5000x_1 + 8500x_2 + 2400x_3 + 2800x_4$ professional are reached.
   - **Budget constraint**: $600x_1 + 925x_2 + 290x_3 + 364x_4 \leq 100,000$,
   - **weekly limit of adds**: $x_1 \leq 30$, $x_2 \leq 60$, $x_3 \leq 60$, $x_4 \leq 80$

   The model yields

$$\begin{aligned} \max z = \quad & 5000x_1 + 8500x_2 + 2400x_3 + 2800x_4 & \\ \text{s.t.} \quad & 600x_1 + 925x_2 + 290x_3 + 364x_4 & \leq 100,000 \\ & 0 \leq x_1 & \leq 30 \\ & 0 \leq x_2 & \leq 60 \\ & 0 \leq x_3 & \leq 60 \\ & 0 \leq x_4 & \leq 80 \end{aligned}$$

2. Using the code below we obtain **Mathematical solution:** $x^* = (30,60,60,25)$ and $z^* = 874000$

   Natural language solution: To maximize the number of professionals reached the company must buy

   - 30 TV ads
   - 60 Journal ads
   - 60 Radio ads, and
   - 25 Online ads

   to potentially reach $874,000$ professionals.

3. According to the sensitivity report, the budget is allowed to increase up to 20020, without losing feasibility. Thus, to compute the increase in audience, we must compute the new value, taking into account the increase, $\Delta b_1 = \hat{b}_1 - b_1 = 120000 - 100000$

$$\hat{z}^* = z^* + \Delta b_1 s_1 == 874000 + 20000 \cdot 7.6923 = 1027846$$

   Which means an increase of $20000 \cdot 7.6923 = 153846$ professionals reached.

4. The number of professional reached by online add is c_4. Thus, according to the first table of the sensitivity report, this coefficient can increase up to 212.41 units, without losing optimality on the solution. In this limit, the audience would increase

$$\hat{z}^* = z^* + \Delta c_4 x_4 = 874000 + 212.41 \cdot 25 = 879310.25$$

   where $\Delta c_4 = \hat{c}_4 - c_4$. Thus, the total audience would be of 879310.25 professionals at the limit of Online Ads increase efficiency.

Soluciones problemas de programación lineal en enteros

Problema 2.20. The goal of this problem is to schedule the production to maximize profit. Thus, we must decide: **if producing a GPU**, **how many unit of the GPU** and **where to produce it**. In this direction let us consider the variables

- x_{ij}: units of GPU i produced in factory j, where $i, j \in I = \{1,2\}$, $i = 1$ represents Titan RTX and $i = 2$ represents RTX 3090. $x_{ij} \geq 0$ and $x_{ij} \in \mathbb{Z}$, non-negative integers
- $y_i = \begin{cases} 1, & \text{if production } i \text{ is actvated} \\ 0, & \text{otherwise} \end{cases}$, for $i = 1,2$.

Thus, the objective function is a combination of the unitary benefit and the infrastructure cost, thus

$$\max z = 1200(x_{11}+x_{12}) + 1600(x_{21}+x_{22}) - 4\,500\,000y_1 - 7\,600\,000y_2 \qquad \text{(in euros)}$$

The constraints are

- $x_{11}+x_{12} \leq My_1$, this constraint represents the dependency between the decision of producing and the units produced, of $y_i = 0$, then the respective x_{ij} must be 0 as well. M is choose accordingly to make the constraint trivial in case the production is activated. (for example $M = 10^{10}$, to avoid the border of the feasible region)
- $x_{21}+x_{22} \leq My_2$
- $\frac{x_{11}}{52} + \frac{x_{21}}{38} \leq 480$, note that every 52 units of GPU 1, an hour of factory 1 is consumed, apply the same reasoning for the rest of the production slots to model the time availability for each factory.
- $\frac{x_{12}}{42} + \frac{x_{22}}{27} \leq 720$

The mathematical model yields

$$\begin{aligned} \max z \quad & = 1200(x_{11}+x_{12}) + 1600(x_{21}+x_{22}) - 4\,500\,000y_1 - 7\,600\,000y_2 \\ \text{s.t} \quad & x_{11}+x_{12} \leq My_1 \\ & x_{21}+x_{22} \leq My_2 \\ & \frac{x_{11}}{52} + \frac{x_{21}}{38} \leq 480 \\ & \frac{x_{12}}{42} + \frac{x_{22}}{27} \leq 720 \\ & x_{ij} \geq 0, x_{ij} \in \mathbb{Z}, y_i \in 0,1, \text{ for } i,j \in I \times I \end{aligned}$$

Solution:

$$x^* = [x_{11}^*, x_{12}^*, x_{21}^*, x_{22}^*, y_1^*, y_2^*] = [24960, 30240, 0, 0, 1, 0]$$
$$z^* = 61\,740\,000$$

Thus, to maximize total profit we must produce 24 960 and 30 240 units of Titan RTX in factories 1 and 2, respectively for a total profit of 61 740 000 euros.

Soluciones problemas de programación lineal en enteros

Problema 6.1. Given the following linear programming problem, find the maximum and minimum of the objective function:

6.1.1) For parts a and c, see Geogebra-solution.

Solution: $x_1^* \approx -1.74$, $x_2^* \approx 2.19$ for $z^* \approx -0.849$.

6.1.2) For parts a and c, see Geogebra-solution.

Solution: $x_1^* = 7.5 \quad , x_2^* = 0$ for $z^* = 22.5$.

KKT conditions

6.1.3) For parts a and c, see Geogebra-solution.

Solution: $x_1^* = 0 \quad , x_2^* \approx 3.06$ for $z^* \approx -9.183$.

6.1.4) For parts a and c, see Geogebra-solution.

Solution: $x_1^* = 2.3 \quad , x_2^* \approx 3.3$ for $z^* \approx 3.954$.

6.1.5) For parts a and c, see Geogebra-solution.

Solution: $x_1^* = 0.894 \quad , x_2^* \approx 0.447$ for $z^* \approx 22.36$.

Problema 6.2. First, write the problem in the standard canonical KKT form as

$$\begin{array}{lll} \min z & f(x_1,x_2) = 3x_1 + 4x_2 & \\ \text{s.t.} & g_1 : \left(\frac{x_1}{3}\right)^2 + \left(\frac{x_2}{2}\right)^2 - 4 & \leq 0 \\ & g_2 : (x_1-3)^2 + (x_2-3)^2 - 6 & \leq 0 \\ & g_3 : -2x_1 + 3x_2 - 5 & \leq 0 \\ & x_1, x_2 \in \mathbb{R} & \end{array}$$

1. To compute the KKT conditions, first note that the Lagrangian is

$$\begin{aligned} \mathscr{L}(x_1,x_2;\lambda_1,\lambda_2,\lambda_3) := (3x_1+4x_2) + \lambda_1 \left(\left(\frac{x_1}{3}\right)^2 + \left(\frac{x_2}{2}\right)^2 - 4 \right) \\ + \lambda_2 \left((x_1-3)^2 + (x_2-3)^2 - 6 \right) + \lambda_3 \left(-2x_1 + 3x_2 - 5\right) \end{aligned}$$

- Thus, stationarity, $\nabla \mathscr{L} = 0$, yields

$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial x_1} &= 3 + \frac{2}{3}\lambda_1 x_1 + 2\lambda_1(x_1-3) - 2\lambda_3 = 0 \\ \frac{\partial \mathscr{L}}{\partial x_2} &= 4 + \lambda_1 x_2 + 2\lambda_2(x_2-3) + 3\lambda_3 = 0 \end{aligned}$$

- On the other hand, primal feasibility yields, $g_1, g_2, g_3 \leq 0$

$$\begin{aligned} \left(\frac{x_1}{3}\right)^2 + \left(\frac{x_2}{2}\right)^2 &\leq 4 \\ (x_1-3)^2 + (x_2-3)^2 &\leq 6 \\ -2x_1 + 3x_2 &\leq 5 \end{aligned}$$

- Dual feasibility: $\lambda_i \geq 0$ for $i = 1,2,3$, and
- Complementary slackness, $\lambda_i g_i = 0$ for $i = 1,2,3$,

$$\begin{aligned} \lambda_1 \left(\left(\frac{x_1}{3}\right)^2 + \left(\frac{x_2}{2}\right)^2 - 4 \right) &= 0 \\ \lambda_2 \left((x_1-3)^2 + (x_2-3)^2 - 6 \right) &= 0 \\ \lambda_3 \left(-2x_1 + 3x_2 - 5\right) &= 0 \end{aligned}$$

2. Using pyomo the solution yields $x_1^* \approx 1.53$, $x_2^* \approx 1.04$ for $z^* \approx 8.753$.

3. See Geogebra-solution.

4. After substituting the value of the solution in the given equations we find that the multipliers λ_i at the optimum are
$$\lambda_1^* = \lambda_3 = 0, \qquad \lambda_2^* \approx 1.02$$

Problema 6.3.

1. The model yields
$$\begin{aligned} \min_{x_1,x_2} \quad & x_1^4 + 2x_1^2 + 4x_2^2 \\ \text{s.t.} \quad & 2x_1 + x_2 \geq 10 \\ & x_1 + 2x_2 \geq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$
With solution $x_1^* \approx 2.567$, $x_2^* \approx 4.867$, for $z^* \approx 151.312$.

Therefore, there should be administered approximately 2.567 units of beam 1 and 4.867 units of beam 2 for a minimum collateral tissue damage that approximately 151.312 units.

2. First, write the problem in the standard canonical KKT form as
$$\begin{aligned} \min_{x_1,x_2} \quad & f(x_1,x_2) = x_1^4 + 2x_1^2 + 4x_2^2 \\ \text{s.t.} \quad & g_1 : -2x_1 - x_2 + 10 \leq 0 \\ & g_2 : -x_1 - 2x_2 + 10 \leq 10 \\ & g_3 : -x_1 \leq 0 \\ & g_4 : -x_2 \leq 0 \\ & x_1, x_2 \in \mathbb{R} \end{aligned}$$
To compute the KKT conditions, first note that the Lagrangian is
$$\begin{aligned} \mathscr{L}(x_1,x_2;\lambda_1,\lambda_2,\lambda_3,\lambda_4) := & (x_1^4 + 2x_1^2 + 4x_2^2) + \lambda_1(-2x_1 - x_2 + 10) \\ & + \lambda_2(-x_1 - 2x_2 + 10) - \lambda_3 x_1 - \lambda_4 x_2 \end{aligned}$$

- Thus, **stationarity**, $\nabla \mathscr{L} = 0$, yields
$$\begin{aligned} \frac{\partial \mathscr{L}}{\partial x_1} &= 4x_1^3 + 4x_1 - 2\lambda_1 - \lambda_2 - \lambda_3 = 0 \\ \frac{\partial \mathscr{L}}{\partial x_2} &= 8x_2 - \lambda_1 - 2\lambda_2 - \lambda_4 = 0 \end{aligned}$$

- On the other hand, **primal feasibility** yields, $g_1, g_2, g_3, g_4 \leq 0$
$$\begin{aligned} -2x_1 - x_2 + 10 &\leq 0 \\ -x_1 - 2x_2 + 10 &\leq 10 \\ -x_1 &\leq 0 \\ -x_2 &\leq 0 \end{aligned}$$

- **Dual feasibility**: $\lambda_i \geq 0$ for $i = 1,2,3$, and

- **Complementary slackness**, $\lambda_i g_i = 0$ for $i = 1, 2, 3$,

$$\begin{aligned} \lambda_1 (-2x_1 - x_2 + 10) &= 0 \\ \lambda_2 (-x_1 - 2x_2 + 10) &= 0 \\ \lambda_3 x_1 &= 0 \\ \lambda_4 x_2 &= 0 \end{aligned}$$

3. The values of the multipliers at the optimum are

$$\lambda_1 \approx 38.938, \quad \lambda_2 = \lambda_3 = \lambda_4 = 0.$$

4. Finally, if the number of radiation units must be discrete, use the solver `'mindtpy'` to obtain $x_1^* = 3$, $x_2^* = 4$, for $z^* = 163$.