UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

MODELOS MATEMÁTICOS DE OPTIMIZACIÓN

Álvaro Baíllo
Pedro Linares
Andrés Ramos
Pedro Sánchez
Ángel Sarabia
Begoña Vitoriano

Septiembre 2003

Alberto Aguilera 23 – E 28015 Madrid – Tel: 34 91 542 2800 – Fax: 34 91 541 1132 – www.doi.icai.upco.es

ÍNDICE

I.1. Optimización

I.1.1. Investigación operativa y optimización

> "In the last decade, new advances in algorithms have been as important as the impressive advances in computer technology" George L. Nemhauser (1994).

> "The technology improvements in algorithms, modeling languages, software, and hardware have made the methodology accessible, easy to use, and fast. So the Age of Optimization has arrived" George L. Nemhauser (1994).

Definir el término investigación operativa no es una tarea fácil ya que su evolución permanente hace que sea difícil dar con precisión una definición. La investigación operativa se puede definir como la aplicación de métodos científicos en la mejora de la efectividad en las operaciones, decisiones y gestión, ver [Robinson, 1999]. Otra definición más extensa es la siguiente: la investigación operativa es la aplicación, por grupos interdisciplinarios, del método científico a los problemas complejos producidos en la dirección y gestión de grandes sistemas de hombres, máquinas, etc. La principal característica consiste en construir un modelo científico del sistema del cual se pueden predecir y comparar los resultados de diversas estrategias, decisiones, incorporando medidas del azar y del riesgo. El objetivo es ayudar a los responsables a determinar su política y actuaciones en forma científica.

Los profesionales de la investigación operativa colaboran con los decisores en el diseño y mejora de las operaciones y decisiones, resuelven problemas y ayudan en las funciones de gestión, planificación o predicción, aportan conocimiento y ayuda en la toma de decisiones. Aplican las técnicas científicas más adecuadas seleccionadas de la matemática, ingeniería o cualquier ciencia social o de administración de empresas. Su trabajo normalmente consiste en recoger y analizar datos, desarrollar y probar modelos matemáticos, proponer soluciones o recomendaciones, interpretar la información y, en definitiva, ayudar a implantar acciones de mejora. Como resultado desarrollan e implantan aplicaciones informáticas, sistemas, servicios técnicos o productos.

La investigación operativa tiene sus orígenes en la Segunda Guerra Mundial, debido a la necesidad urgente de asignación de recursos escasos en las operaciones militares, en problemas tácticos y estratégicos. Estas mismas técnicas se han extendido con posterioridad a las empresas.

Disciplinas típicas de la investigación operativa son la optimización con sus múltiples sabores (lineal, no lineal, entera, estocástica, multiobjetivo), teoría de la decisión y de juegos, teoría de colas y simulación, teoría de grafos o flujos de redes. Otras disciplinas como algoritmos metaheurísticos y lógica borrosa, redes neuronales artificiales, reconocimiento de patrones y otras técnicas de inteligencia artificial, aunque conceptualmente se encuadran dentro de la investigación operativa, habitualmente se estudian dentro de otras disciplinas ligadas a la ingeniería informática como la inteligencia artificial. Los contenidos de algunas de estas últimas disciplinas también están muy ligados a la estadística.

La optimización es una parte relevante dentro de la investigación operativa. Tuvo un progreso algorítmico inicial muy rápido. Muchas técnicas – programación lineal (*linear programming*) LP, programación dinámica (*dynamic programming*) DP– son anteriores a 1960. Por ejemplo, el método Simplex[1] de programación lineal debido a Dantzig[2] es de 1947, el principio de optimalidad de Bellman base de la programación dinámica se formuló en 1957. En la última década se han producido avances significativos generados por el desarrollo en 1984 por parte de Karmarkar de un método de punto interior para programación lineal. Por ejemplo, en una nota técnica de ILOG se presenta que desde su optimizador CPLEX 3.0 en 1994 a CPLEX 7.0 en 2000 la reducción de tiempo de resolución ha sido de 28 veces en el método simplex dual para un problema lineal concreto. Para otro caso se observa una mejora global, de software y algorítmica, de 10000 veces entre la versión de CPLEX 1.0 de 1988 y la 7.0 del 2000. Como referencia, se estima que la mejora en el rendimiento del hardware ha sido del mismo orden de magnitud. Si tomamos conjuntamente ambas mejoras hoy se pueden resolver problemas en segundos que habrían tardado años en ser resueltos hace una docena de años. Estos avances han sido tan importantes como los realizados en el campo de la informática, según la opinión de George L. Nemhauser uno de los expertos actuales en programación entera, y se han producido acompasadamente con ellos. Hoy es posible resolver un problema LP de 200000 ecuaciones con 200000 variables y 1000000 de elementos no nulos en la matriz de restricciones en un PC con suficiente memoria principal. Aproximadamente, para un problema LP se puede decir que se requiere 1 MB de memoria principal por cada 1000 ecuaciones.

[1] En castellano la traducción de esta palabra es símplice pero no es habitual su uso para denominar este método de optimización lineal.

[2] En http://www.e-optimization.com/directory/trailblazers/dantzig/ se puede encontrar un resumen de sus logros así como una entrevista sobre diversos temas, incluyendo imágenes en vídeo.

El estilo de este documento es eminentemente aplicado, práctico, ingenieril, a caballo entre una visión matemática de los problemas y de los algoritmos y la visión económica o de gestión empresarial de algunas de sus aplicaciones. Este documento trata de explicar suficientemente los fundamentos matemáticos como para permitir desarrollar aplicaciones de optimización de manera rigurosa y precisa. Al mismo tiempo, se presentan algunas aplicaciones a problemas concretos de ingeniería.

Al final del capítulo se citan algunos libros generales o de referencia de investigación operativa que pueden servir de consulta o como texto para un nivel de pregrado y postgrado. Luego, en cada capítulo se indican además referencias específicas de los diferentes temas. Dentro de los libros generales, [Hillier y Lieberman, 2002] es un libro clásico de investigación operativa muy ampliamente utilizado que compendia numerosos temas y tiene una orientación ingenieril. [Taha, 1998] presenta los temas con una orientación más matemática mientras que [Winston, 1994] los presenta con una perspectiva más de administración de empresas. [Sarabia, 1996] da una base teórica suficiente para poder resolver una colección de problemas relacionados con el temario de investigación operativa.

Entre las revistas principales que tratan sobre optimización se pueden incluir: *Interfaces*, *Operations Research*, *Management Science, European Journal of Operational Research*, *Mathematics of Operations Research*, *OR/MS Today*, *Mathematical Programming*, *INFORMS Journal on Computing, Journal of the Operational Research Society, Omega, Journal of Optimization Theory and Applications*, *Transportation Science*, *Transportation Research.* Existe una enciclopedia de investigación operativa que puede servir como consulta inicial y referencia de un tema específico, ver [Gass, 2001]. Además se puede encontrar información sobre los temas de investigación operativa en las direcciones de la *Sociedad Española de Estadística e Investigación Operativa* (SEIO) www.seio.es, de la *Association of European Operational Research Societies* (EURO) http://www.euro-online.org/, de la *International Federation of Operational Research Societies* (IFORS) www.ifors.org y del *Institute for Operations Research and the Management Sciences* (INFORMS) www.informs.org.

La optimización consiste en la selección de una alternativa mejor, en algún sentido, que las demás alternativas posibles. Es un concepto inherente a toda la investigación operativa. Sin embargo, determinadas técnicas propias de la investigación operativa se recogen bajo el nombre de optimización o programación matemática.

Los problemas de optimización se componen generalmente de estos tres ingredientes:

- *función objetivo*
  Es la medida cuantitativa del funcionamiento del sistema que se desea optimizar (maximizar o minimizar). Como ejemplo de funciones objetivo se pueden mencionar: la minimización de los costes variables de operación de un sistema eléctrico, la maximización de los beneficios netos de venta de ciertos productos, la minimización del cuadrado de las desviaciones con respecto a unos valores observados, la minimización del material utilizado en la fabricación de un producto, etc.

- *variables*
  Representan las decisiones que se pueden tomar para afectar el valor de la función objetivo. Desde un punto de vista funcional se pueden clasificar en variables *independientes* o *principales* o *de control* y variables *dependientes* o *auxiliares* o *de estado*, aunque matemáticamente todas son iguales. En el caso de un sistema eléctrico serán los valores de producción de los grupos de generación o los flujos por las líneas. En el caso de la venta, la cantidad de cada producto fabricado y vendido. En el caso de la fabricación de un producto, sus dimensiones físicas.

- *restricciones*
  Representan el conjunto de relaciones (expresadas mediante ecuaciones e inecuaciones) que ciertas variables están obligadas a satisfacer. Por ejemplo, las potencias máxima y mínima de operación de un grupo de generación, la capacidad de producción de la fábrica para los diferentes productos, las dimensiones del material bruto del producto, etc.

Resolver un problema de optimización consiste en encontrar el valor que deben tomar las variables para hacer óptima la función objetivo satisfaciendo el conjunto de restricciones.

Los métodos de optimización los podemos clasificar en: métodos *clásicos* (que son los algoritmos que habitualmente se explican en los libros de optimización) y métodos *metaheurísticos* (que aparecieron ligados a lo que se denominó inteligencia artificial e imitan fenómenos sencillos observados en la naturaleza). Dentro de los primeros se encuentra la optimización lineal, lineal entera mixta, no lineal, estocástica, dinámica, etc. que se explican en el documento. En el segundo grupo se incluyen los algoritmos evolutivos (genéticos entre otros), el método del recocido simulado (*simulated annealing*), las búsquedas heurísticas (método tabú, búsqueda aleatoria, avariciosa, etc.) o los sistemas multiagente. De forma muy general y aproximada se puede decir que los métodos clásicos buscan y garantizan un óptimo local mientras que los métodos metaheurísticos tienen mecanismos específicos para alcanzar un óptimo global aunque no garantizan su alcance.

En la siguiente tabla se muestran las expresiones matemáticas generales de algunos tipos de problemas de optimización dentro de los métodos clásicos. Los problemas se distinguen por el carácter de las funciones que intervienen (lineales o no lineales) y de las variables (reales/continuas o enteras/discretas).

Programación lineal (*linear programming*) LP	$\min_x c^T x$ $Ax = b$ $x \geq 0$ $x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$
Programación lineal entera mixta (*mixed integer programming*) MIP	$\min_x c^T x + d^T y$ $Ax + By = b$ $x, y \geq 0$ $x \in \mathbb{Z}^n, y \in \mathbb{R}^l, c \in \mathbb{R}^n, d \in \mathbb{R}^l$ $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times l}, b \in \mathbb{R}^m$
Programación cuadrática (*quadratic programming*) QP	$\min_x c^T x + \frac{1}{2} x^T Q x$ $Ax = b$ $x \geq 0$ $x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}$ $Q \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^m$
Programación no lineal (*non linear programming*) NLP	$\min_x f(x)$ $g(x) = 0$ $h(x) \leq 0$ $l \leq x \leq u$ $f : \mathbb{R}^n \to \mathbb{R}$ $g, h : \mathbb{R}^n \to \mathbb{R}^m$

Existen decisiones que no pueden ser representadas de forma adecuada mediante variables continuas. Por ejemplo, las decisiones de inversión son variables discretas (por ejemplo, planificación de la expansión de la generación o de la red, adquisición de equipos singulares, contratación de personas) o binarias (como localización de plantas o almacenes). Los problemas lineales con variables enteras se pueden clasificar en: programación entera pura PIP (*pure integer* *programming)* si todas las variables son enteras, programación entera binaria BIP (*binary integer programming)* si todas son binarias o programación lineal entera mixta MIP (*mixed integer programming*) si algunas son enteras o binarias y el resto continuas.

Un caso particular, pero muy frecuente, de variables *enteras* son las variables *binarias* (0/1), ya que permiten modelar condiciones de asignación o condiciones lógicas. Por otra parte, toda variable entera x se puede expresar como suma de variables binarias y_i, donde $x = \sum_{i=0}^{N} 2^i y_i$ siendo u una cota superior de x, $0 \leq x \leq u$, y estando u comprendida en el intervalo $2^N \leq u \leq 2^{N+1}$.

Existen algunos tipos de problemas de optimización que alteran ligeramente este esquema:

- *sistemas de ecuaciones lineales – no lineales*
  No existe una función objetivo como tal. Únicamente interesa encontrar una solución factible a un problema con un conjunto de restricciones.

- *optimización sin restricciones*
  Se trata de encontrar el conjunto de valores de las variables que determinan el mínimo/máximo de una función. Algunas de las técnicas que se verán en programación no lineal son para optimización sin restricciones.

- *optimización multiobjetivo*
  Existe más de una función objetivo. El problema que se plantea es cómo tratar varias funciones objetivo a la vez, teniendo en cuenta que el óptimo para un objetivo no lo es para otro, son objetivos en conflicto entre sí. Ésta se enmarca dentro de lo que se conoce de forma más general como decisión multicriterio (*multicriteria decision making* MCDM).

La formulación matemática de algunos problemas de optimización especiales por no incluir alguno de los componentes se presenta en la siguiente tabla.

Problema mixto complementario (*mixed complementarity problem*) MCP	$xF(x) = 0$ $x \in \mathbb{R}^n$ $F : \mathbb{R}^n \to \mathbb{R}^n$
Optimización no lineal sin restricciones	$\min_x f(x)$ $f : \mathbb{R}^n \to \mathbb{R}$
Ajuste no lineal mínimo cuadrático	

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Programación multiobjetivo (*multiobjective programming*)	$\min_x (f_1(x),...,f_k(x))$ $Ax = b$ $x \geq 0$ $x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ $f_i(x) : \mathbb{R}^n \to \mathbb{R}$

I.1.2. Referencias

Gass, S.L. and Harris, C.M. (eds.) (2001) *Encyclopedia of Operations Research and Management Science*. Centennial Edition. Kluwer Academic Publishers.

Hillier, F.S., Lieberman, G.J. (2002) *Investigación de Operaciones*. 7ª edición. McGraw Hill.

Robinson R. (1999) "Welcome to OR Territory" OR/MS Today pp. 40-43 August.

Sarabia, A. (1996) *La Investigación Operativa*. UPCO.

Taha, H.A. (1998) *Investigación de operaciones. Una introducción*. Prentice Hall.

Winston, W.L. (1994) *Investigación de Operaciones. Aplicaciones y Algoritmos*. Grupo Editorial Iberoamericana.

I.2. Modelos de optimización

I.2.1. Modelo y modelado

> **Modelo**. Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja (por ejemplo, la evolución económica de un país), que se elabora para facilitar su comprensión y el estudio de su comportamiento. DICCIONARIO DE LA LENGUA ESPAÑOLA. REAL ACADEMIA ESPAÑOLA.

Un modelo es una representación matemática simplificada de una realidad compleja. Modelar es la acción de construir un modelo, de encorsetar la realidad. Implica la relación entre dos figuras (no necesariamente encarnadas por personas únicas sino por equipos): el *modelador* (encargado de la especificación y desarrollo del modelo) y el *experto* sobre la realidad (conocedor del problema real). La mayoría de las veces, el desarrollo de un modelo puede involucrar a un equipo multidisciplinar compuesto por matemáticos, estadísticos, ingenieros, economistas, psicólogos, etc. que aportan diferentes perspectivas y conocimiento en la representación de la realidad. Un modelo debe equilibrar la necesidad de contemplar todos los detalles con la factibilidad de encontrar técnicas de solución adecuadas.

Un modelo es, en definitiva, una herramienta de ayuda a la toma de decisiones. Por esta razón, sus resultados deben ser inteligibles y útiles. Modelar se puede entender simultáneamente como *ciencia* y como *arte*. Es una ciencia pues se basa en un conjunto de procesos estructurados: análisis y detección de las relaciones entre los datos, establecimiento de suposiciones y aproximaciones en la representación de los problemas, desarrollo o uso de algoritmos específicos de solución. Es un arte porque materializa una visión o interpretación de la realidad no siempre de manera unívoca. Cada persona imprime su estilo en el modelo mismo y en la especificación, en el desarrollo y en la documentación. Características tales como elegancia o simplicidad pueden atribuirse a un modelo. El desarrollo de un modelo es una creación hecha con ayuda de ciencias básicas o herramientas de apoyo.

Entre los beneficios explícitos o implícitos, tanto para el modelador como para el experto, derivados del proceso de modelado además del modelo en sí mismo, se pueden mencionar:

- Ayuda a establecer un diálogo con intercambio de información entre el modelador y el experto

- Organiza los datos, la información disponible sobre el sistema
- Organiza, estructura y mejora la comprensión del sistema
- Internaliza la estructura organizativa de la empresa
- Permite compartir supuestos y resultados entre el modelador y el experto
- Proporciona un entorno ágil para el análisis y la sensibilidad
- Indica la dirección de mejora en las decisiones

En este capítulo se tratará exclusivamente de modelos de optimización, es decir, aquellos donde existe un conjunto de *variables* de decisión que deben maximizar/minimizar una *función objetivo* sometidas a un conjunto de *restricciones*. Los modelos de programación lineal son más utilizados que todos los otros tipos de optimización juntos y abarcan cualquier tipo de actividad humana como micro y macroeconomía, finanzas, marketing, economía de la energía, organización de la producción, planificación de la operación, selección de procesos, asignación de tareas, ingeniería química, forestal, agrónoma, comercio internacional, desarrollo económico, etc. Como referencias generales de modelado de problemas de optimización que se pueden utilizar en la enseñanza de pregrado o postgrado cabe citar a [Schrage, 1997] y [Williams, 1999].

I.2.2. Etapas en el desarrollo de un modelo

Las etapas que componen el *ciclo de vida* de un modelo son las siguientes:

Identificación del problema

Consiste en la recolección y análisis de la información relevante para el problema, en el intercambio de información entre el modelador y el experto, en establecer una relación simbiótica y una estrecha coordinación entre ambos.

Los problemas reales suelen estar definidos en términos vagos e imprecisos. Se debe hacer la tarea de traducción o interpretación en frases precisas, convertibles en ecuaciones matemáticas. En esta etapa se establecen y documentan los supuestos realizados que en etapas posteriores deberán ser validados.

Esta etapa es fundamental para que las soluciones proporcionadas, las conclusiones obtenidas sean útiles, las decisiones adoptadas sean correctas. Los datos suelen ser vitales para conseguir un realismo o aplicabilidad en las soluciones. A menudo representan el cuello de botella del proceso de modelado.

Especificación matemática y formulación

Escritura matemática del problema de optimización, definiendo sus variables, sus ecuaciones, su función objetivo, sus parámetros. En esta etapa se analiza el tamaño del problema, la estructura de la matriz de restricciones, su tipo (LP, MIP, NLP). Es una etapa de creación donde se debe prestar especial atención a la precisión en la formulación y a la escritura de las ecuaciones que describen el problema.

En LP la elección de una formulación de un problema, aunque importante, no afecta de manera significativa la resolución del mismo. Sin embargo, en NLP o MIP la elección de la formulación es crucial. Pueden existir diversas alternativas de modelado que afectan de manera fundamental en la resolución del mismo, existiendo un desarrollo cada vez mayor en la reformulación de problemas. En problemas MIP la calidad de una formulación se mide por la cercanía entre la envoltura convexa del poliedro de soluciones enteras factibles y la del poliedro del problema MIP relajado linealmente. En el apartado I.6.5.2 se explica en más detalle algunas técnicas de reformulación de problemas MIP.

La caracterización de un problema LP según su tamaño resulta difícil y ha sufrido un gran cambio desde los recientes desarrollos de algoritmos simplex mejorados y, sobre todo, desde la aparición de los métodos de punto interior. En la tabla 1.1 se propone una clasificación de tipos de problemas LP según su tamaño. Esta clasificación debe ser tomada como guía o referencia relativa actual pero téngase en cuenta que los tamaños relativos de los problemas cambiarán conforme evolucionen los códigos de optimización. Actualmente se puede afirmar que los códigos de optimización lineal implantan algoritmos muy eficientes, son fiables y numéricamente robustos y están ampliamente disponibles.

	Restricciones	Variables
Caso ejemplo	100	100
Tamaño medio	10000	10000
Gran tamaño	100000	100000
Muy gran tamaño	> 100000	> 100000

Tabla 1.1 Tipos de problemas LP según su tamaño.

En lo referente a MIP o NLP ni siquiera se pueden dar criterios generales de tamaño ya que la dificultad de resolución no tiene por qué estar ligada al tamaño del problema, puede ser incluso preferible reformular un problema aunque aumenten las dimensiones, para lograr una resolución más eficiente.

Resolución

Se trata de implantar un algoritmo de obtención de la solución numérica (muy próxima a la matemática) óptima o cuasióptima. El algoritmo puede ser de propósito general (método simplex) o específico. Puede haber diferentes métodos de solución de un problema o diferentes implantaciones de un mismo método. El tiempo de resolución de un problema también puede depender drásticamente de cómo esté formulado.

La solución óptima debe ser suficientemente satisfactoria, debe ser una guía de actuación para el experto.

Verificación, validación y refinamiento

Esta etapa conlleva la eliminación de los errores en la codificación, es decir, conseguir que el modelo haga lo que se ha especificado matemáticamente en la etapa anterior mediante su escritura en un lenguaje informático (depurar y verificar). Es necesario comprobar la validez de las simplificaciones realizadas a través de los resultados obtenidos, incluso contrastando éstos con situaciones reales ya transcurridas (validar) o comprobando que los resultados son coherentes con respecto a lo que sucedería en la realidad.

Esta etapa de verificación, validación y comprobación da lugar a nuevas necesidades de refinamiento en el modelado para mejorar la capacidad de representación del sistema. Por ejemplo, eliminar la linealidad y hacer el modelo no lineal o hacer el modelo estocástico si la realidad lo fuera. Además, también se puede abordar el refinamiento matemático en la formulación del problema para hacerla más eficaz.

Interpretación y análisis de los resultados

Esta etapa consiste en proponer soluciones. Permite conocer en detalle el comportamiento del modelo al hacer un análisis de sensibilidad en los parámetros de entrada, estudiar diferentes escenarios plausibles de los parámetros, detectar soluciones alternativas cuasióptimas pero suficientemente atractivas, comprobar la robustez de la solución óptima.

Implantación, documentación y mantenimiento

Ésta es una etapa fundamental del desarrollo de un modelo para garantizar su amplia difusión. La documentación ha de ser clara, precisa y completa. El manual de usuario debe incluir la especificación técnica funcional, matemática e informática. El propio código debe incluir una buena documentación para facilitar la tarea del mantenimiento. Piénsese que la mayor parte del ciclo de vida de un modelo no está en el desarrollo sino en la fase de uso y mantenimiento.

En esta etapa se incluye también la tarea de formación para los usuarios del modelo.

I.2.3. Referencias

Schrage, L. (1997) Optimization Modeling with LINDO. Duxbury Press.

Williams, H.P. (1999) Model Building in Mathematical Programming. 4th Edition. John Wiley and Sons.

I.3. Formulación de problemas de optimización

I.3.1. Modelos característicos de programación lineal

A continuación se presentan algunos problemas característicos de programación lineal y entera. Éstos se utilizan como referencia y clasificación para otros problemas. En particular, para los problemas enteros existen numerosas referencias de investigación dedicadas a la solución de los mismos. A pesar de la enorme atención que se ha dedicado a su solución su importancia práctica es limitada.

Problema de la dieta

El problema por excelencia de programación lineal es el de asignación óptima de recursos. Un caso particular de éste es el denominado problema de la dieta. Consiste en determinar la composición de la dieta de mínimo coste que satisface las necesidades específicas de nutrientes. Pongamos un caso particular muy sencillo de alimentación de ganado bovino.

Aprovechamos este ejemplo para seguir paso a paso las etapas en el desarrollo de un modelo.

- En primer lugar hay que ***identificar el problema***.
  Se ha determinado que las necesidades mínimas diarias en la alimentación de una ternera son de 700 g de proteínas, 28 g de calcio y 150 mg de vitaminas. Los alimentos disponibles son pienso y forraje con un coste unitario de 0.30 y 0.35 €/kg respectivamente. La composición nutritiva por kg de alimento se muestra en la siguiente tabla.

	Proteínas (g)	Calcio (g)	Vitaminas (mg)
Pienso	30	2	10
Forraje	45	1	5

  Se trata de determinar la cantidad diaria óptima de cada alimento para minimizar el coste total de alimentación.

- A continuación ***se especifica matemáticamente y se formula*** el problema.

Para ello analizamos y organizamos los **datos** del problema. Sean i los alimentos disponibles (pienso y forraje) y sean j los nutrientes (proteínas, calcio y vitaminas). Sea b_j la cantidad mínima diaria requerida de cada nutriente. Sea a_{ij} la cantidad de nutriente por kg de alimento correspondiente a los valores de la tabla dada. Sea c_i el coste unitario de cada alimento. A continuación definimos las **variables**. Sea x_i la cantidad diaria en kg de cada alimento i. Además indicamos la **función objetivo** y las **restricciones** del problema. La función objetivo es la minimización del coste diario de la dieta

$$\min_{x_i} \sum_i c_i x_i \tag{1.1}$$

Las restricciones corresponden a satisfacer con la mezcla de alimentos las necesidades mínimas diarias de cada nutriente y, por consiguiente, habrá tantas restricciones de este tipo como nutrientes.

$$\sum_i a_{ij} x_i \geq b_j \quad \forall j \tag{1.2}$$

Además hay que añadir la restricción natural de que la cantidad de cada alimento ha de ser no negativa.

$$x_i \geq 0 \tag{1.3}$$

Particularizando estas ecuaciones para los datos previos se obtiene.

$$\begin{aligned}
&\min_{x_1,x_2} 0.30x_1 + 0.35x_2 \\
&30x_1 + 45x_2 \geq 700 \\
&2x_1 + x_2 \geq 28 \\
&10x_1 + 5x_2 \geq 150 \\
&x_1 \geq 0 \\
&x_2 \geq 0
\end{aligned}$$

- Después viene la ***resolución***.
  Vamos a resolver gráficamente el problema. Para ello se dibujan las ecuaciones en forma de igualdad en el espacio de las variables y se indica la región factible del problema. Es decir, el conjunto de puntos que cumple todas las restricciones. Se traza la recta de la función objetivo para un valor cualquiera y se desplaza paralela a sí misma en el sentido de minimizar dicho valor hasta el último punto de la región factible. Dicho punto será el óptimo del problema.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

- Las etapas de ***verificación*** (comprobación de que el modelo es correcto) y ***validación*** (comprobación de que la realidad se representa adecuadamente) son inmediatas en un modelo tan sencillo como éste.

- Seguidamente se realiza la ***interpretación*** y ***análisis*** de los resultados.
  Los resultados indican que la decisión óptima es comprar 18.83 kg de pienso y 8.33 kg de forraje cada día. Con estas decisiones el coste diario de los alimentos es de 6.1667 €.
  Al ganadero le ha llegado una oferta de otro fabricante de piensos a un precio de 0.25 €/kg pero con menor contenido en calcio, 1.5 g de calcio por kg de pienso, y tiene interés en analizar si le interesa comprar o no a dicho fabricante. Para ello planteamos este nuevo problema de optimización

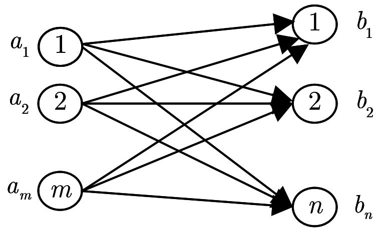
$$
\begin{aligned}
&\min_{x_1,x_2} 0.25x_1 + 0.35x_2 \\
&30x_1 + 45x_2 \geq 700 \\
&1.5x_1 + x_2 \geq 28 \\
&10x_1 + 5x_2 \geq 150 \\
&x_1 \geq 0 \\
&x_2 \geq 0
\end{aligned}
$$

  La solución óptima para este nuevo problema es comprar 14.93 kg de pienso y 5.6 kg de forraje diariamente con un coste de 5.6933 €. Luego, esta oferta es atractiva económicamente.

- La etapa de ***implantación, documentación y mantenimiento*** se da por satisfecha en este modelo sencillo con este apartado donde se explica el modelo.

Problema de transporte

Se trata de minimizar el coste total de transporte de un cierto producto desde los diferentes orígenes a los destinos, satisfaciendo la demanda de cada destino sin superar la oferta disponible en cada origen (ver ejemplo en I.4.2.1 Ejemplo de transporte). Se supone que todos los m orígenes están conectados con los todos los n destinos. Sea a_i la oferta de producto en el origen i, b_j la demanda de producto en el destino j y c_{ij} el coste unitario de transporte desde el origen i al destino j.



El problema de optimización consiste en determinar las unidades de producto $x_{ij} \geq 0$ transportadas desde i hasta j, $\forall i, j$, que minimizan los costes de transporte sujeto a las restricciones de oferta disponible en cada origen i (m restricciones de oferta) y demanda en cada destino j (n restricciones de demanda)

$$
\begin{aligned}
&\min_{x_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\
&\sum_{j=1}^{n} x_{ij} = a_i \quad i = 1, \ldots, m \\
&\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, \ldots, n \\
&x_{ij} \geq 0
\end{aligned}
\tag{1.4}
$$

Implícitamente en esta formulación, se supone que la oferta del producto es igual a la demanda del mismo $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. Si $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ se añade un sumidero universal con coste nulo. Si $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ se añade una fuente universal conectada con todos los destinos con coste muy elevado.

La estructura que presenta la matriz de restricciones del problema tiene el siguiente aspecto.

$$
\begin{array}{c|cccc|cccc|c|cccc|}
 & x_{11} & x_{12} & \ldots & x_{1n} & x_{21} & x_{22} & \ldots & x_{2n} & \ldots & x_{m1} & x_{m2} & \ldots & x_{mn} \\
1 & 1 & 1 & \ldots & 1 & & & & & & & & & \\
2 & & & & & 1 & 1 & \ldots & 1 & & & & & \\
\vdots & & & & & & & & & \ddots & & & & \\
m & & & & & & & & & & 1 & 1 & \ldots & 1 \\
1 & 1 & & & & 1 & & & & \ldots & 1 & & & \\
2 & & 1 & & & & 1 & & & \ldots & & 1 & & \\
\vdots & & & \ddots & & & & \ddots & & \ldots & & & \ddots & \\
n & & & & 1 & & & & 1 & \ldots & & & & 1
\end{array}
$$

Si tanto las ofertas a_i como las demandas de los productos b_j son números enteros, entonces el valor óptimo de x_{ij} es entero por ser la matriz totalmente unimodular[3], por lo que no se necesita recurrir a métodos específicos de resolución de problemas de programación entera.

Problema de transbordo

Consiste en determinar en una red con n nodos las cantidades óptimas para llevar unidades de un producto desde sus orígenes a sus destinos pasando por puntos de transbordo intermedios.

Cada *origen* genera $b_i > 0$ unidades, cada *destino* consume $b_i < 0$ unidades y cada *transbordo* ni genera ni consume unidades $b_i = 0$. El coste unitario de transporte desde el origen i hasta el destino j en dicho sentido es c_{ij}.

Hay que determinar las unidades de producto transportadas desde i a j, $x_{ij} \geq 0$, $\forall i, j$, que minimizan los costes de transporte teniendo en cuenta la restricción de balance o conservación del flujo en cada nudo i.

$$\begin{aligned}
&\min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
&\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \quad i = 1, \ldots, n \\
&x_{ij} \geq 0
\end{aligned} \tag{1.5}$$

Implícitamente en esta formulación, se supone que la oferta es igual a la demanda del producto, es decir, $\sum_{i=1}^{n} b_i = 0$.

Esta matriz también es totalmente unimodular por lo que el problema también puede ser resuelto mediante programación lineal.

Problema de asignación

Se trata de asignar la realización de n tareas a n personas (máquinas, etc.). Este problema es un caso particular del problema de transporte. Por consiguiente las variables toman valores enteros sin exigir esta condición en la formulación del problema.

Consiste en minimizar el coste total de realizar las tareas sabiendo que cada tarea i debe ser hecha por una sola persona y cada persona j debe realizar una única tarea, siendo c_{ij} el coste de realizar la tarea i por la persona j. Las variables del problema son $x_{ij} = \begin{cases} 1 & \text{si se asigna la tarea } i \text{ a la persona } j \\ 0 & \text{en cualquier otro caso} \end{cases}$, $\forall i, j$.

[3] Una matriz es *totalmente unimodular* si toda submatriz cuadrada tiene determinante 0, 1 ó –1. Si la matriz de un problema lineal es totalmente unimodular y las cotas de las restricciones son enteras, entonces todos los puntos extremos del poliedro tienen coordenadas enteras (se denomina *politopo entero*).

$$\begin{aligned}
&\min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
&\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1,\ldots,n \\
&\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1,\ldots,n \\
&x_{ij} \geq 0
\end{aligned} \tag{1.6}$$

I.3.2. Modelos característicos de programación entera

Los problemas de programación entera surgen en numerosos ámbitos de decisión.

- Por ejemplo, cuando se necesita representar número entero de productos o unidades enteras o indivisibles de recursos (aviones, personas, máquinas, etc.).
- También cuando se quieren imponer restricciones lógicas (si se fabrica el producto A también debe fabricarse el B y el C).
- Otro tercer ámbito son los denominados problemas combinatoriales, como por ejemplo los de secuenciación de tareas en máquinas, equilibrado de líneas de producción, asignación de tripulaciones, localización de almacenes o factorías, programación temporal de actividades.
- Así mismo surgen cuando hay funciones de naturaleza no lineal, como las de coste fijo o no convexas en general.
- Por último, también aparecen problemas enteros en teoría de grafos, por ejemplo el problema del tetracoloreado de un mapa.

En este apartado se presentan algunas técnicas de modelado que facilitan la formulación de problemas de optimización con variables enteras. No existe una manera sistemática de formular este tipo de problemas y plantear un "buen" modelado es, a menudo, un arte. Por esta razón, el método de aprendizaje se basa en la realización de ejemplos que muestren las posibles formulaciones de problemas característicos que aparecen con más frecuencia de lo que podría aparecer por su nombre. Algunos de los mencionados anteriormente se describen en este apartado y otro aparecen como problemas al final del capítulo.

En los problemas que siguen, la envoltura convexa definida para el conjunto de soluciones ya no tiene soluciones enteras en los vértices como sucedía en los del apartado anterior, por lo que para su resolución hay que acudir a procedimientos específicos de programación entera que se exponen en el apartado I.6.

Problema de la mochila (*knapsack*)

Se trata de maximizar el valor total de la elección de un conjunto de n proyectos sin sobrepasar el presupuesto b disponible, siendo v_j y c_j el valor y coste de cada proyecto j respectivamente. El nombre procede de la decisión que toma un montañero que trata de maximizar el valor de lo que introduce en su mochila con una restricción de máximo peso admisible. Las variables del problema son $x_j = \begin{cases} 1 & \text{si se elije el proyecto } j \\ 0 & \text{en cualquier otro caso} \end{cases}$. Ésta es una utilización habitual de las variables binarias como forma de seleccionar una alternativa, un proyecto en este caso. La formulación del problema es la siguiente

$$\begin{aligned} &\max_{x_j} \sum_{j=1}^{n} v_j x_j \\ &\sum_{j=1}^{n} c_j x_j \le b \\ &x_j \in \{0,1\} \end{aligned} \tag{1.7}$$

Problema de recubrimiento (*set covering*)

Existen m características y n combinaciones (subconjuntos) de dichas características. La elección de una combinación implica realizar todas las características de la misma. Se trata de minimizar el coste total de las combinaciones elegidas de manera que se cubra o posea cada característica i *al menos* una vez. Los datos son c_j el coste de elegir la combinación j y la matriz de pertenencia de cada característica i a cada combinación j, $a_{ij} = \begin{cases} 1 & \text{si } i \text{ pertenece a } j \\ 0 & \text{si no pertenece} \end{cases}$. Denominamos las variables $x_j = \begin{cases} 1 & \text{si se elige la combinación } j \\ 0 & \text{en cualquier otro caso} \end{cases}$. El problema se formula de la siguiente manera

$$\begin{aligned} &\min_{x_j} \sum_{j=1}^{n} c_j x_j \\ &\sum_{j=1}^{n} a_{ij} x_j \ge 1 \quad i = 1, \dots, m \\ &x_j \in \{0,1\} \end{aligned} \tag{1.8}$$

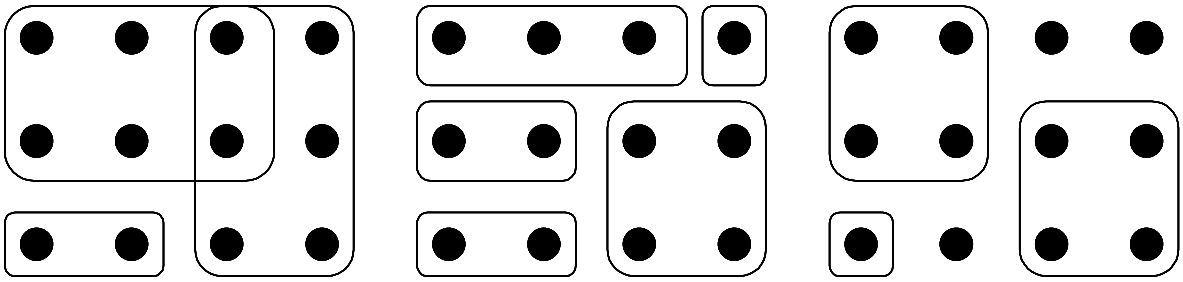


Figura 1.1 Representación gráfica de un recubrimiento, una partición y un empaquetado, respectivamente.

Veamos a continuación un ejemplo de recubrimiento: *asignación de tripulaciones*, tomado de [Hillier y Lieberman, 2002]. Una compañía aérea necesita asignar sus tripulaciones para cubrir todos sus vuelos. En particular, quiere resolver el problema de asignar tres tripulaciones con base en San Francisco a los vuelos listados en la primera columna de la tabla. Las otras columnas muestran las 12 secuencias factibles de vuelos para una tripulación cualesquiera. Los números de cada columna indican el orden de los vuelos. Se necesita elegir tres secuencias (una por tripulación) de manera que se cubran todos los vuelos. Se permite tener más de una tripulación en un vuelo, donde la/s tripulación/es extra viajan como pasajeros, pero por convenio laboral la tripulación extra cobra como si estuviera trabajando. El coste de asignación de una tripulación a cada secuencia de vuelos se da en millones de euros en la última fila. El objetivo es minimizar el coste total de asignación de las tres tripulaciones para cubrir todos los vuelos. Resolver el mismo problema para el caso en que no se permite el vuelo de una tripulación fuera de servicio en un vuelo.

	Secuencias factibles											
	1	2	3	4	5	6	7	8	9	10	11	12
SF – LA	1			1			1			1		
SF – Denver		1			1			1			1	
SF – Seattle			1			1			1			1
LA – Chicago				2			2		3	2		3
LA – SF	2					3				5	5	
Chicago – Denver				3	3				4			
Chicago – Seattle							3	3		3	3	4
Denver – SF		2		4	4				5			
Denver – Chicago					2			2			2	
Seattle – SF			2				4	4				5

Seattle – LA						2			2	4	4	2
Coste (M€)	2	3	4	6	7	5	7	8	9	9	8	9

Se definen las variables del problema como $x_j = \begin{cases} 1 & \text{si se asigna la secuencia } j \\ 0 & \text{en cualquier otro caso} \end{cases}$, $x_j \in \{0,1\} \quad j = 1,\ldots,12$

La función objetivo será

$$\min 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12}$$

Cobertura de cada vuelo al menos una vez

$$\begin{gathered} x_1 + x_4 + x_7 + x_{10} \geq 1 \quad \text{(SF-LA)} \\ x_2 + x_5 + x_8 + x_{11} \geq 1 \quad \text{(SF-Denver)} \\ x_3 + x_6 + x_9 + x_{12} \geq 1 \quad \text{(SF-Seatlle)} \\ \vdots \end{gathered}$$

Asignación de las tres tripulaciones

$$\sum_{j=1}^{12} x_j = 3$$

Las soluciones óptimas son $x_3 = x_4 = x_{11} = 1$ y el resto 0 ó $x_1 = x_5 = x_{12} = 1$ y el resto 0, ambas con coste 18 millones de €.

Si no se permite que una tripulación fuera de servicio vuele en un avión las restricciones de cobertura de mayor o igual pasan a ser de igualdad. Luego, se trata de un problema de partición, cuya formulación se verá a continuación.

Problema de empaquetado (*set packing*)

Se tienen que realizar m proyectos divididos en n paquetes. La elección de un paquete implica realizar todos los proyectos del mismo. Se trata de maximizar el beneficio total de manera que cada proyecto i del conjunto de todos los paquetes que lo incluyen no pueda ser elegido más de una vez. c_j es el beneficio de elegir el paquete j, la matriz de pertenencia de cada proyecto i a cada paquete j es $a_{ij} \begin{cases} 1 & \text{si } i \text{ pertenece a } j \\ 0 & \text{si no pertenece} \end{cases}$. Las variables del problema son $x_j \begin{cases} 1 & \text{si se elige el paquete } j \\ 0 & \text{en cualquier otro caso} \end{cases}$. La formulación del problema es la siguiente

$$\max_{x_j} \sum_{j=1}^{n} c_j x_j$$
$$\sum_{j=1}^{n} a_{ij} x_j \leq 1 \quad i = 1,\ldots,m \tag{1.9}$$
$$x_j \in \{0,1\}$$

Problema de partición (*set partitioning*)

La formulación es similar al problema anterior pero en este caso exactamente una característica (proyecto) del conjunto de combinaciones (paquetes) que la contienen debe ser elegida.

$$\max_{x_j} \sum_{j=1}^{n} c_j x_j$$
$$\sum_{j=1}^{n} a_{ij} x_j = 1 \quad i = 1,\ldots,m \tag{1.10}$$
$$x_j \in \{0,1\}$$

Problema del viajante de comercio (*Traveling Salesman Problem TSP*)

El problema consiste en hacer un recorrido que pase por n ciudades sin repetir ninguna y volviendo a la ciudad de partida de manera que la distancia (o tiempo o coste) total sea mínima. Es un problema de asignación pero con la condición de que la asignación sea un *ciclo*. Es uno de los problemas más importantes en la historia de la programación matemática por todas las investigaciones a las que ha dado lugar y por todas las aplicaciones que tiene, tanto directamente o apareciendo como subproblema dentro de otros más complejos. En una noticia de *Optima* (*Mathematical Programming Society Newsletter*) de junio de 1998, mencionaba que se había conseguido resolver un problema del viajante con 13509 ciudades. Los problemas de enrutamiento de vehículos (expedición o recogida de mercancías) pueden ser formulados de esta manera.

Una de las características más interesantes de este problema es que existen muchas formulaciones conocidas para el mismo, ver [Williams, 1999] y [Nemhauser, 1999]. Dos de ellas se presentan a continuación, una tercera se presenta en el apartado I.4.2.4 y se formulan en GAMS. Sea c_{ij} la distancia entre las ciudades i y j.

Formulación 1 (clásica): Se definen las variables

$$x_{ij} = \begin{cases} 1 & \text{si se va de la ciudad } i \text{ a la ciudad } j \\ 0 & \text{en otro caso} \end{cases}$$

La formulación del problema es:

$$\begin{aligned}
&\min_{x_{ij}} \sum_{i,j} c_{ij} x_{ij} \\
&\sum_i x_{ij} = 1 \quad \forall j \\
&\sum_j x_{ij} = 1 \quad \forall i \\
&\sum_{i,j \in U} x_{ij} \leq \operatorname{card}(U) - 1 \qquad \forall U \subset \{1,\ldots,n\}, 2 \leq \operatorname{card}(U) \leq n-2 \\
&x_{ij} \in \{0,1\}
\end{aligned} \tag{1.11}$$

La primera restricción indica que a una ciudad j sólo se puede llegar una vez desde cualquier ciudad i. La segunda dice que desde una ciudad i sólo se puede salir una vez a cualquier otra ciudad j.

Con esta formulación el problema se resuelve iterativamente. Primero se ignora el tercer tipo de restricciones. Se analiza la solución y se determina si contiene subciclos. En tal caso, se introducen restricciones para evitar subciclos de longitud U con la estructura del tercer bloque y se resuelve de nuevo el problema. Estos dos últimos pasos se repiten hasta encontrar la solución óptima.

El número de restricciones del tercer tipo crece exponencialmente ya que el número de subciclos de longitud U es $C_{m,n} = \binom{n}{\operatorname{card}(U)}$ y siempre que aparece un subciclo de longitud U aparece otro de longitud $n-U$. Por esta razón, su adición en la formulación se hace específicamente para los subciclos que van apareciendo y, usualmente, se necesita añadir solamente una pequeña fracción de las mismas.

Formulación 2: Se definen las variables

$$x_{ijk} = \begin{cases} 1 & \text{si se va de la ciudad } i \text{ a la ciudad } j \text{ en la etapa } k \text{ de recorrido} \\ 0 & \text{en otro caso} \end{cases}$$

El problema queda

$$
\begin{aligned}
&\min_{x_{ijk}} \sum_{i,j,k} c_{ij} x_{ijk} \\
&\sum_{j,k} x_{ijk} = 1 \quad \forall i \\
&\sum_{i,k} x_{ijk} = 1 \quad \forall j \\
&\sum_{i,j} x_{ijk} = 1 \quad \forall k \\
&\sum_{i} x_{ijk} = \sum_{r} x_{jrk+1} \quad \forall j,k \\
&x_{ijk} \in \{0,1\}
\end{aligned}
\tag{1.12}
$$

La primera restricción indica que desde una ciudad i sólo se puede salir una vez a cualquier otra ciudad j en cualquier etapa k. La segunda dice que a una ciudad j sólo se puede llegar una vez desde cualquier ciudad i en cualquier etapa k. La tercera muestra que en la etapa k sólo se puede ir una vez de una ciudad cualquiera i a otra cualquiera j. La última indica que si en la etapa k estamos en la ciudad j habiendo venido de otra ciudad cualquiera i, en la siguiente etapa $k+1$ saldremos de la ciudad j a otra ciudad cualquiera r. Esta última restricción es necesaria para establecer la continuidad espacial entre etapas sucesivas.

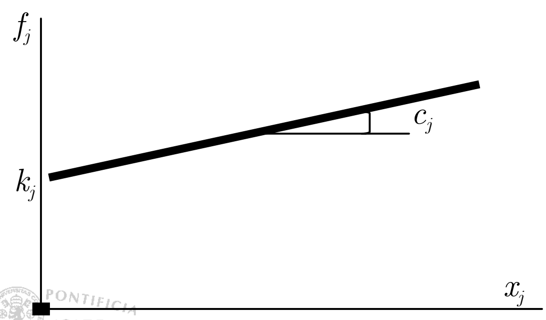
La pregunta que cabe plantearse, es cuál de las dos formulaciones es mejor. En este caso, las formulaciones desde un aspecto teórico no son comparables, aunque en la práctica resulta preferible la primera ya que tiene menos variables (la primera tiene del orden de n^2 y la segunda n^3), aunque tiene muchas más restricciones.

Las comparaciones teóricas de distintas formulaciones se presentan más adelante, dentro de lo que se denomina reformulación de problemas en programación lineal entera.

Problema de coste fijo

Los problemas de coste fijo aparecen cuando el coste de una variable tiene un término fijo con valor diferente de 0 si la variable toma un valor estrictamente positivo. Es una función no lineal y discontinua.

$$
f_j(x_j) = \begin{cases} 0 & x_j = 0 \\ k_j + c_j x_j & x_j > 0 \end{cases}
$$



Este coste se puede modelar con ayuda de una variable binaria auxiliar $y_j \in \{0,1\}$ definida como $y_j = \begin{cases} 1 & x_j > 0 \\ 0 & x_j = 0 \end{cases}$, que indica la realización de la actividad x_j. Introduciendo la condición $x_j \le My_j$, $j = 1,\ldots,n$, siendo M una constante, cota superior de x_j, cuyo valor dependerá del problema, se distingue entre no realizar la actividad y realizarla al menos infinitesimalmente. El valor de la constante M debe ser el menor posible ya que esto es computacionalmente beneficioso. El problema lineal entero se formula como sigue

$$\begin{aligned}
&\min_{x_j,y_j} \sum_{j=1}^{n} f_j(x_j) = \sum_{j=1}^{n} \left(k_j y_j + c_j x_j\right) \\
&x_j \le My_j \\
&x_j \ge 0 \\
&y_j \in \{0,1\}
\end{aligned}$$

I.3.3. Modelado de restricciones con variables binarias

I.3.3.1. Modelado de algunas restricciones especiales

Supongamos que necesitamos considerar en un problema la condición de que si se produce el producto A también se debe producir el producto B. La condición de producción de un producto j la representamos por la restricción $x_j \ge 1$. Entonces, la implicación es

$$x_A \ge 1 \rightarrow x_B \ge 1$$

Esta condición no se puede introducir directamente en un problema lineal porque hace que la estructura del problema (el que se considere o no una restricción más $x_B \ge 1$) depende de que se cumpla otra ($x_A \ge 1$) y esto sólo se conoce una vez que se ha determinado la solución óptima. Un problema de optimización no se puede redefinir endógenamente, es decir, en función de los propios valores que toman las variables del problema.

En este apartado se van a modelar en un problema de optimización algunas condiciones especiales (las restricciones lógicas entre ellas) que requieren el uso de variables binarias para detectar o forzar el cumplimiento de restricciones.

Disyunciones

Las disyunciones implican una pareja de restricciones donde sólo una (cualquiera de las dos) debe satisfacerse, mientras que la otra no es necesario que se cumpla. Debe cumplirse una al menos pero no necesariamente las dos.

$$f(x) \le 0 \text{ ó } g(x) \le 0$$

Supongamos el ejemplo de esta disyunción

$$3x_1 + 2x_2 - 18 \le 0 \text{ ó } x_1 + 4x_2 - 16 \le 0$$

Veamos cómo estas restricciones se pueden incorporar en un problema de optimización. Añadir una constante de valor elevado M a una restricción es equivalente a eliminar (relajar) dicha restricción (se supone que las variables son positivas en estas restricciones), dado que los coeficientes de las variables son también positivos.

$$\begin{matrix} 3x_1 + 2x_2 - 18 \le 0 \\ x_1 + 4x_2 - 16 \le M \end{matrix} \text{ ó } \begin{matrix} 3x_1 + 2x_2 - 18 \le M \\ x_1 + 4x_2 - 16 \le 0 \end{matrix}$$

Se define la variable binaria auxiliar y que selecciona la ecuación correspondiente, $y = \begin{cases} 1 & \text{se relaja la ecuación 1} \\ 0 & \text{se relaja la ecuación 2} \end{cases}$. Luego las restricciones disyuntivas se modelan en un problema de optimización como

$$\begin{aligned} 3x_1 + 2x_2 - 18 &\le My \\ x_1 + 4x_2 - 16 &\le M(1-y) \end{aligned}$$

Si $y = 1$ se relaja la restricción 1 pero se obliga a cumplir la 2 y viceversa para $y = 0$.

Algunas implicaciones son un caso semejante a las restricciones disyuntivas

$$f(x) > 0 \to g(x) \le 0$$

es equivalente a

$$f(x) \le 0 \text{ ó } g(x) \le 0$$

ya que $P \to Q$ es equivalente a (No P) ó Q.

Cumplir k de N ecuaciones

Se tiene un conjunto de N ecuaciones de las cuales se han de satisfacer al menos k, siendo $k < N$. Las disyunciones son un caso particular de éste para $k = 1$ y $N = 2$.

Sea el conjunto de N ecuaciones

$$f_1(x_1,\dots,x_n)\le 0$$
$$f_2(x_1,\dots,x_n)\le 0$$
$$f_N(x_1,\dots,x_n)\le 0$$

añadiendo una constante M y una variable binaria y_i para cada ecuación tenemos

$$f_1(x_1,\dots,x_n)\le My_1$$
$$f_2(x_1,\dots,x_n)\le My_2$$
$$f_N(x_1,\dots,x_n)\le My_N$$

donde además se impone la condición de seleccionar solamente k ecuaciones.

$$\sum_{i=1}^{N} y_i = N-k$$

$$y_i \in \{0,1\} \quad i=1,\dots,N$$

Seleccionar entre *N* valores

Sea una función con múltiples posibles valores y se desea elegir uno de ellos.

$$f(x_1,\dots,x_n)=\begin{cases} d_1 \\ d_2 \\ \vdots \\ d_N \end{cases}$$

La manera de modelarlo es introduciendo una variable binaria auxiliar y_i por cada valor y la condición de elección única.

$$f(x_1,\dots,x_n)=\sum_{i=1}^{N} d_i y_i$$
$$\sum_{i=1}^{N} y_i = 1$$
$$y_i \in \{0,1\} \quad i=1,\dots,N$$

I.3.3.2. Modelado de implicaciones lógicas

Las variables binarias se utilizan para indicar que el cumplimiento de una restricción implica el cumplimiento de otra.

Implicaciones sencillas

Retomemos el ejemplo de la restricción que aparecía en el problema de coste fijo

$$x \leq M\delta$$

siendo M una cota superior positiva de x (por ejemplo, 10^6), $m \leq x \leq M$ y δ la variable binaria. Por claridad en la explicación en este apartado se utiliza la letra griega δ para denominar a la variable binaria auxiliar.

Si $\delta = 1$ la restricción no obliga a nada ya que $x \leq M$ se cumple por definición. Si $\delta = 0$ entonces $x \leq 0$. Luego esta restricción permite modelar la implicación

$$\delta = 0 \rightarrow x \leq 0 \text{ (si } \delta = 0 \text{ entonces se cumple que } x \leq 0\text{)}$$

Por otra parte, si $x > 0$ entonces $\delta = 1$. Si $x \leq 0$ la restricción no obliga a nada.

$$x > 0 \rightarrow \delta = 1 \text{ (si } x > 0 \text{ entonces se cumple que } \delta = 1\text{)}$$

Ambas son implicaciones equivalentes puesto que $P \rightarrow Q$ es equivalente a No $Q \rightarrow$ No P. Luego, la restricción lineal $x \leq M\delta$ nos permite representar dichas implicaciones en un problema lineal.

De forma análoga veamos la restricción

$$x \geq m\delta$$

siendo m una cota inferior negativa de x (por ejemplo, -10^6), $m \leq x \leq M$ y δ la variable binaria.

Si $\delta = 1$ la restricción no obliga a nada ya que $x \geq m$ se cumple por definición. Si $\delta = 0$ entonces $x \geq 0$. Luego esta restricción permite modelar la implicación

$$\delta = 0 \rightarrow x \geq 0 \text{ (si } \delta = 0 \text{ entonces se cumple que } x \geq 0\text{)}$$

Por otra parte, si $x < 0$ entonces $\delta = 1$. Si $x \geq 0$ la restricción no obliga a nada.

$$x < 0 \rightarrow \delta = 1 \text{ (si } x < 0 \text{ entonces se cumple que } \delta = 1\text{)}$$

Nuevamente ambas son implicaciones equivalentes puesto que $P \rightarrow Q$ es equivalente a No $Q \rightarrow$ No P.

En resumen, hasta ahora hemos visto la representación en un problema lineal de las siguientes implicaciones

$$\left.\begin{array}{r}\delta = 0 \to x \le 0 \\ x > 0 \to \delta = 1\end{array}\right\} x \le M\delta$$

$$\left.\begin{array}{r}\delta = 0 \to x \ge 0 \\ x < 0 \to \delta = 1\end{array}\right\} x \ge m\delta$$

A continuación, vamos a generalizar la representación de implicaciones para cualquier tipo de restricción genérica.

Implicaciones de una restricción $\le$

La implicación

$$\delta = 1 \to \sum_j a_j x_j \le b$$

es equivalente a

$$\sum_j a_j x_j \le b + M(1-\delta)$$

siendo M una cota superior de la restricción para cualquier valor de cualquier x_j, $\sum_j a_j x_j - b \le M$. Efectivamente de manera directa se deduce que si $\delta = 1$ se impone la restricción original y si $\delta = 0$ no implica nada (se relaja la restricción original). Análogamente al caso anterior esta restricción también representa la implicación

$$\sum_j a_j x_j > b \to \delta = 0$$

La implicación

$$\sum_j a_j x_j \le b \to \delta = 1$$

se puede transformar en $\delta = 0 \to \sum_j a_j x_j > b$ o bien en $\delta = 0 \to \sum_j a_j x_j \ge b + \varepsilon$ que es equivalente a

$$\sum_j a_j x_j \ge b + \varepsilon + (m - \varepsilon)\delta$$

siendo m una cota inferior de la restricción para cualquier valor de cualquier x_j, $\sum_j a_j x_j - b \ge m$.

Implicaciones de una restricción $\geq$

De manera simétrica se pueden representar las implicaciones con restricciones de tipo mayor o igual.

La implicación

$$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$$

es equivalente a

$$\sum_j a_j x_j \geq b + m(1-\delta)$$

siendo m una cota inferior de la restricción para cualquier valor de cualquier x_j, $\sum_j a_j x_j - b \geq m$. Efectivamente de manera directa se deduce que si $\delta = 1$ se impone la restricción original y si $\delta = 0$ no implica nada (se relaja la restricción original). Análogamente al caso anterior esta restricción también representa la implicación

$$\sum_j a_j x_j < b \rightarrow \delta = 0$$

La implicación

$$\sum_j a_j x_j \geq b \rightarrow \delta = 1$$

se puede transformar en $\delta = 0 \rightarrow \sum_j a_j x_j < b$ o bien en $\delta = 0 \rightarrow \sum_j a_j x_j \leq b - \varepsilon$ que es equivalente a

$$\sum_j a_j x_j \leq b - \varepsilon + (M + \varepsilon)\delta$$

siendo M una cota superior de la restricción para cualquier valor de cualquier x_j, $\sum_j a_j x_j - b \leq M$.

Implicaciones de una restricción $=$

Para deducir las implicaciones de restricciones igualdad se transforman en ecuaciones de tipo mayor o igual y menor o igual simultáneamente.

La implicación

$$\delta = 1 \rightarrow \sum_j a_j x_j = b$$

es equivalente a

$$\delta = 1 \rightarrow \sum_j a_j x_j \leq b$$
$$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$$

Luego se representa por las ecuaciones

$$\sum_j a_j x_j \leq b + M(1-\delta)$$
$$\sum_j a_j x_j \geq b + m(1-\delta)$$

Efectivamente para $\delta = 1$ se cumplen ambas restricciones y para $\delta = 0$ ambas se relajan.

La implicación

$$\sum_j a_j x_j = b \rightarrow \delta = 1$$

es una combinación de los casos anteriores simultáneamente

$$\begin{matrix} \sum_j a_j x_j \leq b \rightarrow \delta' = 1 \\ \sum_j a_j x_j \geq b \rightarrow \delta'' = 1 \end{matrix} \text{ y además } \delta' = 1 \text{ y } \delta'' = 1 \rightarrow \delta = 1$$

que se modela con las restricciones

$$\sum_j a_j x_j \geq b + \varepsilon + (m - \varepsilon)\delta'$$
$$\sum_j a_j x_j \leq b - \varepsilon + (M + \varepsilon)\delta''$$

y la restricción adicional que indica el cumplimiento de ambas.

$$\delta' + \delta'' - \delta \leq 1$$

Implicaciones dobles

Para formular implicaciones dobles éstas se desdoblan en las implicaciones unidireccionales correspondientes.

$$\delta = 1 \leftrightarrow \sum_j a_j x_j \leq b \text{ es equivalente a } \begin{cases} \delta = 1 \rightarrow \sum_j a_j x_j \leq b \\ \sum_j a_j x_j \leq b \rightarrow \delta = 1 \end{cases}$$

y lo mismo para los otros tipos de restricciones.

Como resumen se presentan en la siguiente tabla todas las implicaciones lógicas de cumplimiento de cualquier tipo de restricción y su variable binaria asociada y su equivalencia en restricciones lineales.

$\delta = 1 \rightarrow \sum_j a_j x_j \leq b$	$\sum_j a_j x_j \leq b + M(1-\delta)$
$\sum_j a_j x_j \leq b \rightarrow \delta = 1$	$\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta$
$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$	$\sum_j a_j x_j \geq b + m(1-\delta)$
$\sum_j a_j x_j \geq b \rightarrow \delta = 1$	$\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta$
$\delta = 1 \rightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \leq b + M(1-\delta)$ $\sum_j a_j x_j \geq b + m(1-\delta)$
$\sum_j a_j x_j = b \rightarrow \delta = 1$	$\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta'$ $\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta''$ $\delta' + \delta'' - \delta \leq 1$
$\delta = 1 \leftrightarrow \sum_j a_j x_j \leq b$	$\sum_j a_j x_j \leq b + M(1-\delta)$ $\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta$
$\delta = 1 \leftrightarrow \sum_j a_j x_j \geq b$	$\sum_j a_j x_j \geq b + m(1-\delta)$ $\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta$
$\delta = 1 \leftrightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \leq b + M(1-\delta)$ $\sum_j a_j x_j \geq b + m(1-\delta)$ $\sum_j a_j x_j \geq b + \varepsilon + (m-\varepsilon)\delta'$ $\sum_j a_j x_j \leq b - \varepsilon + (M+\varepsilon)\delta''$ $\delta' + \delta'' - \delta \leq 1$

donde M y m son la menor constante superior y la mayor constante inferior de la restricción que cumplen $\sum_j a_j x_j - b \leq M$ y $\sum_j a_j x_j - b \geq m$ para cualquier valor de cualquier x_j. ε es una constante de valor muy pequeño, que en el caso de restricciones con todas las variables binarias será siempre 1.

I.3.3.3. Modelado de proposiciones condicionales y/o compuestas

Hasta ahora se han modelado disyunciones entre restricciones o implicaciones del tipo si $\delta = 1$ entonces se debe verificar tal restricción o viceversa, si se verifica esta restricción entonces $\delta = 1$, o la doble implicación. Se puede necesitar el modelado de proposiciones condicionales y/o compuestas más complejas que las simples implicaciones o disyunciones anteriores. Por ejemplo, si se fabrica el producto A o B (o ambos) entonces debe fabricarse también al menos uno de los productos C, D o E. Este tipo de restricciones también se modela con la ayuda de variables binarias.

Antes de entrar en el modelado de estas restricciones conviene recordar algunas operaciones lógicas que pueden utilizarse para transformar las proposiciones en otras cuyo modelado pueda resultar más sencillo. Aquí se muestra una tabla de equivalencias.

P → Q	no P o Q
P → (Q y R)	(P → Q) y (P → R)
P → (Q o R)	(P → Q) o (P → R)
(P y Q) → R	(P→ R) o (Q → R)
(P o Q) → R	(P→ R) y (Q → R)
no (P o Q)	no P y no Q
no (P y Q)	no P o no Q

Existen algunas proposiciones condicionales y/o compuestas que se transforman de manera sencilla en restricciones, mediante el uso de variables binarias. Si denominamos X_i al cumplimiento de la restricción i y $\delta_i = \begin{cases} 1 \text{ si se cumple la restricción } i \\ 0 \text{ si no se cumple} \end{cases}$ a la variable binaria auxiliar indicadora de su cumplimiento, esta tabla se puede interpretar fácilmente.

X_1 o X_2	$\delta_1 + \delta_2 \geq 1$
X_1 y X_2	$\delta_1 = 1$, $\delta_2 = 1$
no X_1	$\delta_1 = 0$
$X_1 \rightarrow X_2$	$\delta_1 - \delta_2 \leq 0$
$X_1 \leftrightarrow X_2$	$\delta_1 - \delta_2 = 0$

La primera fila dice que se debe cumplir la restricción 1 o la 2 (o ambas), luego efectivamente al menos una de las dos variables δ_1 y δ_2 debe tomar valor 1 y la forma de expresarlo con una ecuación lineal es $\delta_1 + \delta_2 \geq 1$. Además tiene que haber una restricción que diga que si se satisface la restricción i entonces $\delta_i = 1$, $x_i > 0 \rightarrow \delta_i = 1$. Esa condición ha surgido ya para el problema de coste fijo y su modelado como restricción lineal es $x_i \leq M\delta_i$. Ya se ha visto en el apartado anterior la manera generalizada de obtener las restricciones lineales asociadas a las implicaciones de cumplimiento de una restricción de cualquier tipo y su variable binaria asociada.

Volviendo al ejemplo anterior. Si X_i representa la fabricación del producto i (por ejemplo, controlado mediante el consumo de energía en una máquina por encima de un cierto umbral) y δ_i la variable binaria de cumplimiento de dicha condición de fabricación, la implicación lógica

$$(X_A \text{ o } X_B) \rightarrow (X_C \text{ o } X_D \text{ o } X_E)$$

se puede representar como

$$\delta_A + \delta_B \geq 1 \rightarrow \delta_C + \delta_D + \delta_E \geq 1$$

Para poder modelar estas implicaciones lógicas, por complejas que sean, de manera automática la implicación se separa en dos bloques

$$\delta_A + \delta_B \geq 1 \rightarrow \delta = 1$$

$$\delta = 1 \rightarrow \delta_C + \delta_D + \delta_E \geq 1$$

donde δ es la variable binaria auxiliar indicadora del cumplimiento de la primera restricción.

Aplicando este procedimiento al ejemplo anterior, la implicación

$$\delta_A + \delta_B \geq 1 \rightarrow \delta = 1 \text{ equivale a } \delta_A + \delta_B \leq 2\delta$$

ya que $\varepsilon = 1$ y $M = 1$ y

$$\delta = 1 \rightarrow \delta_C + \delta_D + \delta_E \geq 1 \text{ equivale a } \delta_C + \delta_D + \delta_E \geq \delta$$

siendo $m = -1$.

Luego las restricciones lineales a introducir en el problema de optimización son

$$\begin{cases} \delta_A + \delta_B \leq 2\delta \\ \delta_C + \delta_D + \delta_E \geq \delta \end{cases}$$

Alternativamente, la implicación original

$$(X_A \text{ o } X_B) \rightarrow (X_C \text{ o } X_D \text{ o } X_E)$$

se podía haber transformado en ésta

$$[X_A \rightarrow (X_C \text{ o } X_D \text{ o } X_E)] \text{ y } [X_B \rightarrow (X_C \text{ o } X_D \text{ o } X_E)]$$

y el conjunto de restricciones resultantes habría sido

$$\begin{cases} \delta_A - \delta \leq 0 \\ \delta_B - \delta \leq 0 \\ \delta_C + \delta_D + \delta_E \geq \delta \end{cases}$$

que también es equivalente a

$$\begin{cases} \delta_C + \delta_D + \delta_E \geq \delta_A \\ \delta_C + \delta_D + \delta_E \geq \delta_B \end{cases}$$

Para esta implicación se han encontrado tres formulaciones matemáticamente equivalentes.

Veamos a continuación un ejemplo donde aparecen este tipo de implicaciones. Un entrenador de baloncesto tiene 9 jugadores, a los que ha evaluado de 1 a 3 de acuerdo con su manejo de pelota, tiro, rebote y defensa, según se indica en la tabla adjunta.

Jugador	Posiciones	Manejo de pelota	Tiro	Rebote	Defensa
1	Pivot	2	1	3	3
2	Base	3	3	1	2
3	Pivot, Alero	2	3	2	2
4	Alero, Base	1	3	3	1
5	Pivot, Alero	1	3	1	2
6	Alero, Base	3	1	2	3
7	Pivot, Alero	3	2	2	1
8	Pivot	2	1	3	2
9	Alero	3	3	1	3

El equipo titular de 5 jugadores debe tener la máxima capacidad defensiva y satisfacer las siguientes condiciones:

- Por los menos dos jugadores deben estar en disposición de actuar de pivot, al menos dos de alero y por lo menos uno de base.
- Su nivel medio, tanto en el manejo de pelota como de tiro y rebote, debe ser no inferior a 2.

- Si juega el jugador 3, entonces el jugador 6 no puede estar en pista.
- Si el jugador 1 está en el equipo titular, también deberá estar el 4 ó el 5, pero en este caso no los dos a la vez. Si el jugador 1 no está en el equipo titular, 4 y 5 pueden hacerlo, si interesa.
- El jugador 8 ó el 9, pero no los dos a la vez, deben formar parte del equipo.

Formular un programa lineal que facilite la selección del equipo titular.

En primer lugar se definen las variables de decisión que se van a utilizar. Éstas son

$$x_j = \begin{cases} 1 & \text{si se incluye el jugador } j \text{ en el equipo} \\ 0 & \text{en otro caso} \end{cases} \qquad j = 1,\ldots,9$$

$$x_{jk} = \begin{cases} 1 & \text{si se incluye el jugador } j \text{ en posición } k \\ 0 & \text{en otro caso} \end{cases} \qquad j = 1,\ldots,9\,,\ k = p,a,b$$

Estas últimas variables definidas sólo para los jugadores con capacidad para jugar en varias posiciones.

Maximización de la capacidad defensiva

$$\max 3x_1 + 2x_2 + 2x_3 + x_4 + 2x_5 + 3x_6 + x_7 + 2x_8 + 3x_9$$

Selección de cinco jugadores

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = 5$$

Selección de número mínimo de jugadores en cada posición

$$\begin{aligned} &x_1 + x_{3p} + x_{5p} + x_{7p} + x_8 \geq 2 \\ &x_{3a} + x_{4a} + x_{5a} + x_{6a} + x_{7a} + x_9 \geq 2 \\ &x_2 + x_{4b} + x_{6b} \geq 1 \end{aligned}$$

Niveles medios mínimos

$$\begin{aligned} &2x_1 + 3x_2 + 2x_3 + x_4 + x_5 + 3x_6 + 3x_7 + 2x_8 + 3x_9 \geq 10 \\ &x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 + 2x_7 + x_8 + 3x_9 \geq 10 \\ &3x_1 + x_2 + 2x_3 + 3x_4 + x_5 + 2x_6 + 2x_7 + 3x_8 + x_9 \geq 10 \end{aligned}$$

Incompatibilidad entre dos jugadores

$$x_3 = 1 \rightarrow x_6 = 0 \text{ equivale a } x_3 \leq 0 \text{ ó } x_6 \leq 0$$

que, como disyunción, se modela con las restricciones

$$\begin{cases} x_3 \le y_1 \\ x_6 \le 1 - y_1 \end{cases}$$

o bien, modelando directamente la implicación, con la restricción siguiente

$$x_3 + x_6 \le 1$$

La formulación con las dos ecuaciones $\begin{cases} x_3 \le y_1 \\ x_6 \le 1 - y_1 \end{cases}$ es más fuerte (mejor) que la de una única ecuación $x_3 + x_6 \le 1$, porque la región factible del problema relajado es menor para el primer caso que para el segundo. En el apartado I.6.5.2 se explican algunas técnicas de reformulación de problemas MIP para conseguir soluciones más fuertes.

Afinidad entre jugadores

$x_1 \ge 1 \to x_4 + x_5 = 1$ es equivalente a $x_1 \le 0$ ó $(x_4 + x_5 \le 1$ y $x_4 + x_5 \ge 1)$

que, como disyunción, se modela con las restricciones

$$\begin{cases} x_1 \le y_2 \\ x_4 + x_5 - 1 \le (1 - y_2) \\ x_4 + x_5 - 1 \ge -1(1 - y_2) \end{cases}$$

o bien, modelando directamente la implicación, con la restricción siguiente

$$\begin{cases} x_4 + x_5 \le 2 - x_1 \\ x_4 + x_5 \ge x_1 \end{cases}$$

Al menos un jugador de entre varios

$$x_8 + x_9 = 1$$

Relación de coherencia entre variables binarias

$$\begin{aligned} x_{3p} + x_{3a} - x_3 &= 0 \\ x_{4a} + x_{4b} - x_4 &= 0 \\ x_{5p} + x_{5a} - x_5 &= 0 \\ x_{6a} + x_{6b} - x_6 &= 0 \\ x_{7p} + x_{7a} - x_7 &= 0 \end{aligned}$$

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

La solución resulta ser $x_1 = x_4 = x_6 = x_7 = x_9 = 1$ y el resto 0 y la defensa total del equipo es 11.

I.3.3.4. Modelado de productos con variables binarias

Las variables binarias también se pueden utilizar para eliminar algunos productos de variables que convertirían el problema en no lineal pero que con esta transformación resulta un problema lineal entero mixto, más fácil de resolver.

En la siguiente tabla se muestran algunas conversiones posibles. La primera columna indica los productos de variables que se desea modelar. En la siguiente la equivalencia para obtener las restricciones que se introducen en el problema de programación lineal, expresadas en la tercera columna.

$\delta_1\delta_2 = 0$ $\delta_i \in \{0,1\}$	$\delta_1 = 0$ ó $\delta_2 = 0$	$\delta_1 + \delta_2 \le 1$ $\delta_i \in \{0,1\}$
$\delta_1\delta_2$ $\delta_i \in \{0,1\}$	Reemplazar $\delta_1\delta_2$ por δ_3 $\delta_3 = 1 \leftrightarrow \delta_1 = 1$ y $\delta_2 = 1$	$\delta_3 \le \delta_1$ $\delta_3 \le \delta_2$ $\delta_1 + \delta_2 \le 1 + \delta_3$ $\delta_i \in \{0,1\}$
$x\delta$ $x \ge 0$ $\delta \in \{0,1\}$	Reemplazar $x\delta$ por y $\delta = 0 \rightarrow y = 0$ $\delta = 1 \rightarrow y = x$	$y \ge 0$ $y \le M\delta$ $-x + y \le 0$ $x - y + M\delta \le M$ $x \le M$

I.3.4. Modelos característicos de programación no lineal

Aunque la programación lineal juega un papel fundamental en el campo de la optimización ya que en multitud de problemas prácticos se pueden considerar lineales todas las funciones que intervienen, esto no siempre es así. De hecho, muchos economistas han encontrado que cierto grado de no linealidad es la regla, y no la excepción, en la mayoría de los problemas de planificación económica, por lo que se plantea como necesario manejar problemas de programación no lineal.

De forma muy aproximada, se puede afirmar que el tiempo de resolución de problemas de programación no lineal es un orden de magnitud mayor que el de un problema lineal del mismo tamaño y, además, los algoritmos de NLP son menos fiables y robustos.

Veamos algunos ejemplos donde existen funciones objetivos o restricciones no lineales.

Problemas de producción con elasticidad en los precios y/o costes

En algunos problemas de producción se puede suponer que hay una ganancia unitaria fija asociada a cada producto, con lo que la función objetivo de beneficio que se obtiene es lineal. Sin embargo, en otros problemas ciertos factores introducen no linealidades en la función objetivo. Por ejemplo, un gran fabricante puede encontrar *precios elásticos* mediante los cuales la cantidad que se puede vender de un producto va en relación inversa con el precio que se cobra. La curva precio-demanda, $p(x)$, que representa el precio unitario que se necesita para poder vender x unidades, sería una función no lineal decreciente, nunca inferior al coste unitario de producción c.

Así el margen de contribución de la empresa (ingreso bruto menos coste de producción, beneficio neto, EBITDA) vendría determinado por

$$P(x) = xp(x) - cx$$

Si, además, la empresa tiene una función semejante para cada uno de los n productos que puede fabricar la función objetivo global sería una suma de funciones no lineales.

$$f(x) = \sum_{j=1}^{n} P_j(x) = \sum_{j=1}^{n} \left[x_j p_j(x_j) - c_j x_j \right]$$

Otra razón por la que pueden surgir no linealidades en la función objetivo es a causa de los costes de producción, ya que éstos pueden variar con el nivel de producción. Por ejemplo, el coste puede decrecer cuando aumenta el nivel de producción gracias al efecto de una *curva de aprendizaje* (mayor eficiencia con más experiencia) o aumentar por necesidad de tiempos extra o instalaciones más costosas.

Las restricciones también se pueden ver afectadas por estos tipos de no linealidades. Una que surge inmediatamente es la restricción de presupuesto, si existe, cuando los costes de producción varían como se ha descrito anteriormente. También serán funciones no lineales las asociadas a los recursos, siempre que el uso de un determinado recurso no sea proporcional a los niveles de los respectivos productos.

Problema de transporte con descuentos por volumen

El problema de transporte que se ha considerado hasta el momento supone que el coste por unidad enviada de un origen a un destino dados es fijo, independientemente de la cantidad mandada. Sin embargo, una situación muy habitual es que se disponga de *descuentos por cantidad* para volúmenes grandes, con lo que la función de coste unitaria sería una función no lineal con pendiente no creciente. Una alternativa es aproximar esta función no lineal por una poligonal.

Así pues, el coste de embarcar x unidades viene dado por una función *poligonal*, $C(x)$, continua, con pendiente en cada tramo igual al coste unitario de transporte. En consecuencia, si cada combinación de origen y destino tiene una función semejante, la función objetivo sería

$$f(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij}(x_{ij})$$

Al ser una función poligonal cóncava en un problema de minimización se modelará introduciendo variables binarias de selección del segmento de la poligonal.

Selección de una cartera de inversiones

Actualmente, cuando se plantea la selección de una cartera de inversiones, los inversores se preocupan tanto por el rendimiento esperado como por el riesgo asociado a su inversión y para obtener un modelo que permita determinar una cartera que, con ciertas suposiciones, combine de forma óptima estos factores se utiliza la programación no lineal.

Supongamos que se están considerando n tipos de acciones para incluirlas en la cartera; las variables de decisión x_j, $j = 1,\ldots,n$ representan el número de acciones j que se van a incluir. Sean μ_j y σ_{jj} la media y la varianza del rendimiento sobre cada acción de tipo j, en donde σ_{jj} es una medida del riesgo de estas acciones. Sea σ_{ij} la covarianza del rendimiento sobre una acción de cada tipo i y j. Entonces, el valor esperado $R(x)$ y la varianza $V(x)$ del rendimiento total de la cartera son

$$R(x) = \sum_{j=1}^{n} \mu_j x_j$$

$$V(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

con lo que la función objetivo del modelo resultante es

$$f(x) = R(x) - \beta V(x)$$

donde β se denomina factor de aversión al riesgo, ya que cuanto mayor sea mayor importancia (negativa) se le da en la función objetivo a la variabilidad (la volatilidad del rendimiento no es más que su desviación estándar) de la inversión final.

Como restricción se incluye la restricción del presupuesto y la no negatividad de las variables (P_j representa el coste de cada acción de tipo j y B es el presupuesto):

$$\sum_{j=1}^{n} P_j x_j \leq B$$
$$x_j \geq 0 \quad j = 1,...,n$$

Problemas de sistemas de energía eléctrica

Un problema no lineal de sistemas de energía eléctrica muy característico es el flujo de cargas óptimo en corriente alterna AC. Se trata de minimizar los costes variables de operación de los grupos de generación sujeto al conjunto de restricciones de la red y a las restricciones de seguridad preventiva y/o correctiva. En este caso, tanto la función objetivo como las restricciones son no lineales. La función objetivo porque los costes de generación se suelen considerar cuadráticos en función de la producción. Las restricciones porque tanto la potencia activa como la reactiva son funciones no lineales del módulo y argumento de las tensiones en los nudos.

I.3.5. Referencias

Hillier, F.S., Lieberman, G.J. (2002) *Investigación de Operaciones*. 7ª edición. McGraw Hill.

Nemhauser, G.L. and Wolsey, L.A. (1999) *Integer and Combinatorial Optimization*. John Wiley and Sons.

Williams, H.P. (1999) *Model Building in Mathematical Programming*. 4th Edition. John Wiley and Sons.

Wolsey, L.A. (1998) *Integer Programming*. John Wiley and Sons.

I.3.6. Biblioteca de problemas

PROBLEMA: AYUDA EN EMERGENCIAS

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. 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.

	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	

PROBLEMA: CONSTRUCCIÓN DE ALMACENES

Una compañía planea construir varios almacenes para guardar un cierto producto. Estos almacenes surtirán a dos grandes clientes con las unidades demandadas mensualmente apuntadas en la última fila de la tabla. Se pueden construir hasta tres almacenes, que se tienen como candidatos, con capacidades expresadas en la última columna. Usando el coste estimado de construcción de los almacenes, su vida útil y el valor del dinero en el tiempo, los costes de construcción por mes para los tres almacenes se han estimado en 8000, 12000 y 7000. A continuación se dan los costes de transporte por unidad desde los tres almacenes candidatos a los clientes.

	Cliente 1	Cliente 2	Capacidad
Almacén 1	1.50	2.00	4000
Almacén 2	2.00	1.50	5000
Almacén 3	2.50	2.25	6000
Demanda	3000	5000	

Determinar qué almacenes se deben construir y cómo se ha de satisfacer la demanda de los clientes.

PROBLEMA: TRANSPORTE DE ELECTRODOMÉSTICOS

Una compañía tiene dos fábricas, una en Alicante y otra en Huelva. Las dos fábricas producen frigoríficos y lavadoras. Las capacidades de producción de estos artículos en Alicante son de 5000 y 7000, respectivamente, y en Huelva de 8000 y 4000. La compañía entrega estos productos a tres grandes clientes en las ciudades de Barcelona, A Coruña y Valencia, siendo las demandas:

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Demanda/cliente	Barcelona	A Coruña	Valencia
Frigoríficos	4000	5000	4000
Lavadoras	3000	3000	4000

Los artículos se transportan por ferrocarril. En la tabla siguiente se muestran los costes unitarios de transporte y las limitaciones para enviar cualquiera de los dos productos de cada fábrica a cada cliente:

		Barcelona	A Coruña	Valencia
Alicante	Coste unitario	6	14	7
	Máximo unidades	6000	3000	7500
Huelva	Coste unitario	10	8	15
	Máximo unidades	3000	9000	3000

Se desea minimizar el coste total de transporte.

Problema: Logística

Una empresa tiene dos factorías, F1 y F2, con las que abastece a tres almacenes de distribución, D1, D2 y D3, de dos artículos, A1 y A2.

Los costes de transporte de una unidad de cualquiera de los dos artículos desde cada factoría a cada almacén se dan en la tabla izquierda, en tanto que los precios de venta unitarios de cada artículo en cada almacén se dan en la tabla derecha.

Coste Tr	D1	D2	D3
F1	4	7	5
F2	6	5	7

Precio	D1	D2	D3
A1	17	20	18
A2	19	17	21

El tiempo, expresado en minutos, que se tarda en fabricar una unidad de cada artículo en cada una de las factorías se refleja en la tabla izquierda, en tanto que los costes unitarios de fabricación de cada artículo en cada factoría aparecen en la tabla derecha.

Tiempo	A1	A2
F1	6	7.5
F2	10	5

Coste Fb	A1	A2
F1	8	6
F2	5	10

La capacidad de producción de la factoría 1 es de 260 horas y la de la factoría 2 de 240 horas.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Las demandas mínimas de cada uno de los artículos que en cada almacén deben ser satisfechas son expresadas en la tabla siguiente.

	D1	D2	D3
A1	600	800	500
A2	700	500	1200

Por último y por cuestiones de tipo técnico y de política de empresa, nunca se pueden producir en cualquiera de las factorías más de 500 unidades de un artículo que de otro.

Se trata de elaborar un modelo que proporcione el mejor programa de producción y distribución para maximizar el beneficio neto.

¿Se modifica la solución si el tiempo de ejecución de A2 en F1 se reduce en medio minuto?

PROBLEMA: GESTIÓN DE AUTOBUSES

En una ciudad se intenta disminuir la contaminación reduciendo la circulación interurbana. Un primer estudio busca determinar el mínimo número de autobuses que satisfagan las necesidades de transporte. Después de recoger la información se observa que este número varía según la hora del día, pero se puede considerar constante en intervalos sucesivos de cuatro horas:

00:00 a.m. – 4:00 a.m.	4
4:00 a.m. – 8:00 a.m.	8
8:00 a.m. – 12:00 m.	10

12:00 m. – 4:00 p.m.	7
4:00 p.m. – 8:00 p.m.	12
8:00 p.m. – 00:00 a.m.	4

Los turnos de autobuses funcionan durante ocho horas seguidas y pueden comenzar al principio de cualquiera de los seis periodos descritos anteriormente. Además, si en el turno que comienza a las 8:00 p.m. hay estrictamente más de 4 autobuses, en el siguiente ha de haber también estrictamente más de 4. Plantear un problema de programación lineal entera para determinar el mínimo número de autobuses diario que satisface las necesidades anteriores.

PROBLEMA: ADQUISICIÓN DE CAMIONES

Una compañía de transportes tiene 10 camiones con capacidad 40000 kg y 5 camiones de 30000 kg. Los camiones grandes tienen un coste variable de combustible de 0.30 €/km y los pequeños de 0.25 €/km.

En una semana la empresa debe transportar 400000 kg en un recorrido de 800 km. La posibilidad de otros compromisos recomienda que por cada dos

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

camiones pequeños mantenidos en reserva debe quedarse por lo menos uno de los grandes.

¿Cuál es el número óptimo de camiones de ambas clases que deben movilizarse para ese transporte y teniendo en cuenta las restricciones?

PROBLEMA: PLANIFICACIÓN DEL METRO

En una determinada ciudad se va a construir la red del metro. La empresa encargada ha de decidir qué líneas construir y para ello tiene varias opciones. Existen 10 puntos claves por los que la red ha de pasar y se ha visto que son 8 las posibles líneas a construir. Las líneas posibles, los puntos clave por los que pasaría cada una y su coste estimado de construcción en unidades apropiadas, son:

	Puntos clave	Coste
L1	P1 P2 P3 P4	4
L2	P1 P3 P5 P7	4
L3	P2 P3 P4 P6	4
L4	P5 P7 P9 P10	4
L5	P2 P7 P8	3
L6	P1 P4 P5 P10	4
L7	P3 P8 P9	3
L8	P2 P6 P10	3

Además, por el punto P2 han de pasar al menos dos líneas; y, si los puntos P3 y P7 no quedan conectados por una línea directa, entonces debe existir un transbordo en P8 de modo que pase una línea que una este punto con el P3 y otra con el P7. Plantear como un problema de programación lineal entera el problema de decidir qué líneas construir de la forma más económica con estas restricciones teniendo en cuenta que por cada punto clave debe pasar al menos una línea.

PROBLEMA: OFICINA DE CORREOS

Una oficina de correos necesita distinto número de empleados de jornada completa para cada día de la semana, tal como se da en la tabla adjunta. Las reglas sindicales señalan que cada empleado de jornada completa tiene que trabajar durante cinco días consecutivos y, a continuación, descansar dos días. Por ejemplo, un empleado que trabaje de lunes a viernes tiene que descansar sábado y domingo. La oficina de correos quiere cumplir con sus requerimientos diarios y utilizar sólo empleados de jornada completa. Formular mediante programación matemática un modelo que pueda utilizar la oficina de correos para minimizar el número de empleados de jornada completa a contratar.

	Empleados
Lunes	17
Martes	13
Miércoles	15
Jueves	19
Viernes	14
Sábado	16
Domingo	11

PROBLEMA: ABASTECIMIENTO

Una empresa abastecedora de agua tiene que llevar agua de un punto s a un punto t y para realizar la conexión entre ambos puntos ha de pasar por unos puntos intermedios. Cada conexión entre un par de puntos tiene un coste estimado de construcción y, una vez construida, un coste unitario de envío de cada litro y una capacidad por hora que se recogen en la siguiente tabla:

Conexión	Coste construcción	Coste envío litro/min	Capacidad litros/min
s–1	100000	40	100
s–2	200000	50	200
1–3	80000	60	50
1–t	100000	70	30
2–3	200000	40	20
2–t	200000	70	100
3–t	150000	60	60

Plantear un problema de programación matemática si se quieren enviar 180 litros por minuto de la forma más económica posible, teniendo en cuenta que si se construye la conexión de s a 2 ha de hacerse la de 2 a t.

PROBLEMA: ADQUISICIÓN DE MÁQUINAS TROQUELADORAS

Una compañía tiene tres tipos de máquinas troqueladoras de diferente velocidad y precisión:

	Velocidad (piezas/hora)	Precisión (%)	Coste (€/hora)

Tipo 1	20	99	2.00
Tipo 2	15	95	1.75
Tipo 3	10	99	1.50

Cada día (8 horas) se deben procesar por lo menos 3500 piezas y hay disponibles 8 máquinas del tipo 1, 10 del tipo 2 y 20 del tipo 3.

Si cada pieza errónea le cuesta a la compañía 1 céntimo. ¿Cuántas máquinas de cada tipo se deben utilizar para minimizar los costes?

PROBLEMA: SECUENCIACIÓN DE TRABAJOS EN UNA MÁQUINA

Dados unos trabajos que realizar, una duración de éstos y una fecha de entrega prevista, plantear un problema de programación lineal entera para encontrar la secuencia que minimiza el retraso o demora media con que los trabajos son entregados, con los siguientes datos:

Tarea	T1	T2	T3	T4
Tiempo de proceso	9	12	7	14
Fecha de entrega	15	19	23	31

PROBLEMA: PRODUCCIÓN II

En una empresa familiar se producen dos tipos de productos, 1 y 2, procesando materia prima. Se pueden comprar hasta 90 kg de materia prima a un coste de 10 €/kg. Se puede usar un kilo de materia prima para producir un kilo de producto 1 o para producir 1/2 kg del producto 2. Usar un kilo de materia prima para producir el producto 1 requiere 2 horas de mano de obra. Usar un kilo de materia prima para procesar el producto 2 requiere 3 horas de mano de obra. Se dispone de 300 horas de mano de obra a 3 €/hora. Se pueden vender a lo sumo 40 kg. del producto 2. El producto 1 se vende a 29 €/kg y el producto 2 a 69 €/kg.

Además existen una limitación inferior y superior en caso de que se produzca alguna cantidad de cada artículo. Es decir, si se produce algo del producto 1 ha de ser más de 15 y menos de 30 kg y si se produce algo del producto 2 ha de ser más de 10 y menos de 20 kg. Plantear el problema y obtener la solución óptima.

PROBLEMA: PRODUCCIÓN E INVENTARIO

Una empresa desea planear su política de producción/inventario para los meses de agosto, septiembre, octubre y noviembre. La demanda estimada del producto para esos meses es de 500, 600, 800 y 1000 unidades, respectivamente. En la actualidad, la capacidad de producción mensual es de 600 unidades con un coste de 2500 €. La administración ha decidido instalar un nuevo sistema de producción con capacidad mensual de 1100 unidades a un coste por unidad de 3000 €. Sin embargo, el nuevo sistema no puede ser instalado hasta noviembre. Supóngase que el inventario inicial es de 250 unidades y que, durante cualquier mes dado, se pueden almacenar a lo sumo 400 unidades. Si el coste mensual por unidad por mantener en inventario es de 300 €, minimizar el coste total de producción e inventario. Suponer que se debe satisfacer la demanda y que se requiere tener 100 unidades en inventario al final de noviembre.

PROBLEMA: MISIÓN PACÍFICA

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 éstos:

	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: MEZCLA DE CRUDO

La empresa Sunco Oil produce dos tipos de gasolina (1 y 2), cada una de ellas mezclando dos tipos de crudo (1 y 2). Los precios de venta de cada barril de gasolina son 7000 y 6000 €, respectivamente. Por su parte, los precios de compra de los dos tipos de crudo son de 4500 y 3500 € por barril, respectivamente. Se pueden comprar hasta 5000 barriles de cada crudo diarios. Los dos tipos de gasolina difieren en su índice de octano y en su contenido en azufre. La mezcla del petróleo crudo que se utiliza para obtener la gasolina 1 ha de tener un índice de octano promedio de al menos 10 y a lo sumo un 1 % de azufre. La mezcla que se obtiene para la gasolina 2 ha de tener un índice promedio de octano de por lo menos 8 y a lo sumo un 2 % de azufre. Los índices de octano y el contenido en azufre de los dos tipos de crudo son:

Crudo	Octano	Azufre
1	12	0.5
2	6	2.0

La transformación de un barril de petróleo en un barril de gasolina cuesta 400 € y la refinería de Sunco puede producir diariamente hasta 9000 barriles de gasolina.

Los clientes de Sunco actualmente demandan 3000 barriles de la gasolina 1 y 2000 de la gasolina 2. Sin embargo, Sunco tiene la posibilidad de estimular la demanda mediante la publicidad, de modo que cada euro invertido en la publicidad de cada tipo de gasolina, aumenta la demanda diaria de ese tipo de gasolina en 0.1 barriles (si por ejemplo gasta 1000 € en publicidad de la gasolina 1 aumenta la demanda de gasolina 1 en 100 barriles). Formular un problema de programación lineal que permita a Sunco maximizar sus ganancias diarias.

PROBLEMA: PRODUCCIÓN VI

Una planta de producción dispone de m máquinas para llevar a cabo su producción. La demanda semanal del producto es conocida para las siguientes n semanas, dem_j siendo j cada una de las semanas, y ha de ser satisfecha. Cada una de las máquinas i puede estar arrancada y produciendo durante cada semana o no, pero si lo está tiene un coste fijo por estar arrancada de cf_i €, siendo su producción máxima pm_i. Además, el coste unitario de producción con cada una de las máquinas es variable con las semanas, siendo cv_{ij} € por unidad de producto, y el coste de almacenamiento de una semana a otra está estimado en $calm$ € por unidad de producto. Por otra parte, arrancar una máquina para acoplarla una semana si no lo estaba la anterior tiene un coste de arranque $carr_i$. Se supone que todas las máquinas inicialmente están arrancadas (no hay coste de arranque para la primera semana).

a) Plantear un modelo lineal para optimizar la planificación de la producción siendo el horizonte de planificación las n semanas.

b) Sobre la formulación anterior supóngase ahora que cuando una máquina para, ha de hacerlo al menos dos semanas consecutivas por razones técnicas, ¿cómo se modelaría esta nueva condición?

PROBLEMA: TRANSPORTE POR FERROCARRIL

Una empresa de transporte opera en una línea ferroviaria con un determinado material rodante que puede transportar un volumen máximo V y un peso máximo P de mercancías, transportando distintas mercancías de otras empresas. Para un determinado día dispone de distintas solicitudes de mercancías, de modo que de cada mercancía i conoce la cantidad máxima a transportar mx_i, el volumen v_i y peso p_i unitarios, el pago que el propietario de la mercancía está dispuesto a pagar a la empresa por cada unidad transportada, b_i, y el coste que estima la empresa por cada unidad a transportar, c_i. La empresa de transporte desea tener un modelo que le permita elegir cada día las cantidades de las mercancías que le proporcionen mayor beneficio a partir de estos datos diarios.

A su vez, puede variar la capacidad del material rodante, tanto en volumen o peso, de modo que desea saber en cuánto puede valorar cada unidad extra de volumen y de peso de la que puede disponer, para saber cuánto puede estar dispuesto a pagar por un aumento de éstos.

Por otra parte, a los propietarios de las mercancías que no son transportadas, puede y debe informarles de la cantidad en la que deberían aumentar su oferta para que su mercancía fuera transportada (en detrimento de otras).

a) Presentar un modelo lineal para este problema, suponiendo que las cantidades se pueden fraccionar. ¿Qué elementos del modelo darías como resultado para responder a las distintas preguntas planteadas?
b) Supóngase ahora que la oferta conlleva un descuento por volumen de modo que la cantidad que un propietario paga por cada unidad es de la forma $a_i - g_i x_i$, donde x_i es la cantidad transportada. Plantear el nuevo modelo
c) Sobre el planteamiento del apartado a), supóngase que lo que hay es una capacidad de volumen y peso por vagón, $\overline{V}$ y $\overline{P}$, y que hay que decidir además de cuánto transportar, cuántos vagones hay que enganchar, existiendo un coste unitario por vagón utilizado, $cvag$. Plantear el nuevo modelo.

I.3.7. Resultados de la biblioteca de problemas

RESULTADO DEL PROBLEMA AYUDA EN EMERGENCIAS

$$\max_{x_{ij}} \sum_{i,j} a_{ij} x_{ij}$$
$$\sum_j x_{ij} \le a_i \quad \forall i$$
$$\sum_i x_{ij} \le v_j \quad \forall j$$
$$x_{ij} \ge 0$$

El avión A1 hace 50 viajes a la aldea V1, el avión A2 hace 70 a la aldea V3 y 20 a la V5 y el avión A3 hace 20 a la aldea V2 y 40 a la V4. La cantidad total de alimentos repartidos es 1678.

RESULTADO DEL PROBLEMA CONSTRUCCIÓN DE ALMACENES

$$\begin{aligned}
&\min_{x_{ij},y_i} \sum_{i,j} v_{ij}x_{ij} + \sum_i f_i y_i \\
&\sum_j x_{ij} \le c_i y_i \quad \forall i \\
&\sum_i x_{ij} \ge d_j \qquad \forall j \\
&x_{ij} \ge 0, y_i \in \{0,1\}
\end{aligned}$$

Construir los almacenes 1 y 3 y servir 3000 unidades del almacén 1 al cliente 1 y 1000 unidades al cliente 2 y del almacén 3 4000 unidades al cliente 2.

RESULTADO DEL PROBLEMA TRANSPORTE DE ELECTRODOMÉSTICOS

$$\begin{aligned}
&\min_{x_{ijk}} \sum_{i,j,k} c_{ij}x_{ijk} \\
&\sum_j x_{ijk} \le a_{ik} \quad \forall i,k \\
&\sum_i x_{ijk} \ge b_{jk} \quad \forall j,k \\
&\sum_k x_{ijk} \le t_{ij} \quad \forall i,j \\
&x_{ijk} \ge 0
\end{aligned}$$

Frigoríficos	Barcelona	A Coruña	Valencia
Alicante	1500	0	3500
Huelva	2500	5000	500

Lavadoras	Barcelona	A Coruña	Valencia
Alicante	3000	0	4000
Huelva	0	3000	0

El coste total de transporte es de 176000 u.m.

RESULTADO DEL PROBLEMA LOGÍSTICA

$$\max_{x_{ijk}} \sum_{i,j,k} v_{jk} x_{ijk} - \sum_{i,j,k} f_{ik} x_{ijk} - \sum_{i,j,k} c_{ij} x_{ijk}$$
$$\sum_{j,k} t_{ik} x_{ijk} \le h_i \quad \forall i$$
$$\sum_{i} x_{ijk} \ge b_{jk} \quad \forall j,k$$
$$-500 \le \sum_{j} x_{ij1} - \sum_{j} x_{ij2} \le 500 \quad \forall i$$
$$x_{ijk} \ge 0$$

A1	D1	D2	D3
F1	600	0	500
F2	0	840	0

A1	D1	D2	D3
F1	0	0	1200
F2	700	500	0

El beneficio neto es de 29000.

RESULTADO DEL PROBLEMA GESTIÓN DE AUTOBUSES

$$\min x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$
$$x_1 + x_6 \ge 4$$
$$x_1 + x_2 \ge 8$$
$$x_2 + x_3 \ge 10$$
$$x_3 + x_4 \ge 7$$
$$x_4 + x_5 \ge 12$$
$$x_5 + x_6 \ge 4$$
$$x_5 + x_6 \le 4 + 12\delta$$
$$x_1 + x_6 \ge 5\delta$$
$$x_i \in \mathbb{Z}^+, \delta \in \{0,1\}$$

El número mínimo de autobuses es de 26.

RESULTADO DEL PROBLEMA ADQUISICIÓN DE CAMIONES

$$\min (0.30x + 0.25y)800$$
$$x \leq 10$$
$$y \leq 5$$
$$40x + 30y \geq 400$$
$$2(10 - x) \geq 5 - y$$
$$x, y \in \mathbb{Z}^+$$

RESULTADO DEL PROBLEMA PLANIFICACIÓN DEL METRO

$$\min 4x_1 + 4x_2 + 4x_3 + 4x_4 + 3x_5 + 4x_6 + 3x_7 + 3x_8$$
$$x_1 + x_2 + x_6 \geq 1$$
$$x_1 + x_3 + x_5 + x_8 \geq 2$$
$$x_1 + x_2 + x_3 + x_7 \geq 1$$
$$x_1 + x_3 + x_6 \geq 1$$
$$x_2 + x_4 + x_6 \geq 1$$
$$x_3 + x_8 \geq 1$$
$$x_2 + x_4 + x_5 \geq 1$$
$$x_5 + x_7 \geq 1$$
$$x_4 + x_7 \geq 1$$
$$x_4 + x_6 + x_8 \geq 1$$
$$\left.\begin{array}{r} x_2 + \delta \geq 1 \\ x_5 + x_7 - 2\delta \geq 0 \end{array}\right\} \text{ o bien } \left\{ x_5 + x_7 \geq 2 - 2\,x_2 \right.$$
$$x_i, \delta \in \{0,1\}$$

RESULTADO DEL PROBLEMA OFICINA DE CORREOS

$$
\begin{aligned}
&\min \sum_{i=1}^{7} x_i \\
&x_1 + x_4 + x_5 + x_6 + x_7 \geq 17 \\
&x_1 + x_2 + x_5 + x_6 + x_7 \geq 13 \\
&x_1 + x_2 + x_3 + x_6 + x_7 \geq 15 \\
&x_1 + x_2 + x_3 + x_4 + x_7 \geq 19 \\
&x_1 + x_2 + x_3 + x_4 + x_5 \geq 14 \\
&x_2 + x_3 + x_4 + x_5 + x_6 \geq 16 \\
&x_3 + x_4 + x_5 + x_6 + x_7 \geq 13 \\
&x_i \in \mathbb{Z}^+
\end{aligned}
$$

Resultado del problema Abastecimiento

$$
\begin{aligned}
&\min 40x_{s1} + 50x_{s2} + 60x_{13} + 70x_{1t} + 40x_{23} + 70x_{2t} + 60x_{3t} + \\
&+100000y_{s1} + 200000y_{s2} + 80000y_{13} + \\
&+10000y_{1t} + 200000y_{23} + 200000y_{2t} + 150000y_{3t} \\
&x_{s1} + x_{s2} = 180 \\
&x_{s1} = x_{13} + x_{1t} \\
&x_{s2} = x_{23} + x_{2t} \\
&x_{13} + x_{23} = x_{3t} \\
&x_{1t} + x_{2t} + x_{3t} = 180 \\
&x_{s1} \leq 100y_{s1} \\
&x_{s2} \leq 200y_{s2} \\
&x_{13} \leq 50y_{13} \\
&x_{1t} \leq 30y_{1t} \\
&x_{23} \leq 20y_{23} \\
&x_{2t} \leq 100y_{2t} \\
&x_{3t} \leq 60y_{3t} \\
&y_{2t} \geq y_{s2} \\
&x_{ij} \geq 0, y_{ij} \in \{0,1\} \quad \forall i, j
\end{aligned}
$$

Se eligen las conexiones (s–1), (s–2), (1–3), (1–t), (2–t) y (3–t) y pasa un flujo de 80, 100, 50, 30, 100 y 50 respectivamente. El coste total es 853300.

Resultado del problema Adquisición de máquinas troqueladoras

$$\min(2x + 1.75y + 1.5z + 0.01 \cdot 20x + 0.01 \cdot 15y + 0.01 \cdot 10z)8$$
$$20 \cdot 8x + 15 \cdot 8z + 10 \cdot 8z \geq 3500$$
$$x \leq 8$$
$$y \leq 10$$
$$z \leq 20$$
$$x, y, z \in \mathbb{Z}^+$$

RESULTADO DEL PROBLEMA SECUENCIACIÓN DE TRABAJOS EN UNA MÁQUINA

Denominamos d_j al tiempo de proceso del trabajo j y r_j a la fecha de entrega del trabajo j.

Definimos las variables del problema como

$$x_{ij} = \begin{cases} 1 & \text{si el trabajo } j \text{ se hace en la posición } i \\ 0 & \text{en otro caso} \end{cases}$$

La función objetivo será la minimización de la demora media

$$\min \frac{1}{4} \sum_i p_i$$

sujeto a estas restricciones

cada trabajo se hace una vez

$$\sum_i x_{ij} = 1 \quad \forall j$$

en cada posición sólo un trabajo

$$\sum_j x_{ij} = 1 \quad \forall i$$

Para cada posición i se acaba un trabajo en ella y su fecha de entrega es $\sum_j r_j x_{ij}$.

Por otra parte, el trabajo j que acaba en esa posición acaba en el instante $\sum_j d_j \sum_{k \leq i} x_{kj}$. Las variables n_i y p_i, cuentan si acaba antes de tiempo (adelantado) o después (retrasado). Por eso p_i, que es la demora, es la que aparece en la función objetivo

$$\sum_j d_j \sum_{k \leq i} x_{kj} + n_i - p_i = \sum_j r_j x_{ij} \quad \forall i$$

$$n_i, p_i \geq 0 \quad x_{ij} \in \{0,1\}$$

Resultado del problema Producción II

$$
\begin{aligned}
&\max 2900y_1 + 6900y_2 - 600y_1 - 1800y_2 - 1000y_1 - 2000y_2 \\
&y_1 + 2y_2 \le 90 \\
&2y_1 + 6y_2 \le 300 \\
&y_2 \le 40 \\
&15u_1 \le y_1 \le 30u_1 \\
&10u_2 \le y_2 \le 20u_2 \\
&y_1, y_2 \ge 0, u_1, u_2 = \{0,1\}
\end{aligned}
$$

$$
\begin{aligned}
&\max \sum_k p_k y_k - e\sum_k t_k x_k - c\sum_k x_k \\
&\sum_k x_k \le \overline{x} \\
&\sum_k t_k x_k \le \overline{t} \\
&y_k = a_k x_k, \forall k \\
&u_k \underline{y}_k \le y_k \le u_k \overline{y}_k, \forall k \\
&x_k, y_k \ge 0, u_k = \{0,1\}, \forall k
\end{aligned}
$$

Resultado del problema Producción e inventario

$$
\begin{aligned}
&\min \sum_i \left(ci_i + c'p_i\right) \\
&i_i + p_i - d_i = i_{i+1} \\
&p_i \le \overline{p}_i \\
&i_i \le \overline{i}_i \\
&i_1 = 250, i_5 = 100 \\
&i_i, p_i \ge 0
\end{aligned}
$$

Resultado del problema Misión pacífica

$$
\begin{aligned}
&\max 2x_{ef} + 5x_{ei} + 4x_{eg} + 3x_{ep} + 4x_{fi} + 4x_{fg} + 2x_{fp} + 5x_{ig} + 4x_{ip} + 3x_{gp} \\
&x_{ef} + x_{ei} + x_{eg} + x_{ep} \le 1 \\
&x_{ef} + x_{fi} + x_{fg} + x_{fp} \le 1 \\
&x_{ei} + x_{fi} + x_{ig} + x_{ip} \le 1 \\
&x_{eg} + x_{fg} + x_{ig} + x_{gp} \le 1 \\
&x_{ep} + x_{fp} + x_{ip} + x_{gp} \le 1 \\
&x_{ef} + x_{ei} + x_{eg} + x_{ep} + x_{fi} + x_{fg} + x_{fp} + x_{ig} + x_{ip} + x_{gp} = 2 \\
&x_{ei} \le x_{fg} \\
&x_{ef}, x_{ei}, x_{eg}, x_{ep}, x_{fi}, x_{fg}, x_{fp}, x_{ig}, x_{ip}, x_{gp} \in \{0,1\}
\end{aligned}
$$

Resultado del problema Mezcla de crudo

$$
\begin{aligned}
&\max 7000(x_{11} + x_{21}) + 6000(x_{12} + x_{22}) - 4500(x_{11} + x_{12}) - 3500(x_{21} + x_{22}) \\
&-400(x_{11} + x_{12} + x_{21} + x_{22}) - y_1 - y_2 \\
&x_{11} + x_{12} \le 5000 \\
&x_{21} + x_{22} \le 5000 \\
&x_{11} + x_{12} + x_{21} + x_{22} \le 9000 \\
&12x_{11} + 6x_{21} \ge 10(x_{11} + x_{21}) \\
&12x_{12} + 6x_{22} \ge 8(x_{12} + x_{22}) \\
&0.5x_{11} + 2x_{21} \le x_{11} + x_{21} \\
&0.5x_{12} + 2x_{22} \le 2(x_{12} + x_{22}) \\
&x_{11} + x_{21} = 3000 + 0.1y_1 \\
&x_{12} + x_{22} = 2000 + 0.1y_2 \\
&x_{11}, x_{12}, x_{21}, x_{22}, y_1, y_2 \ge 0
\end{aligned}
$$

Resultado del problema Producción VI

a)

$$
\begin{aligned}
&\min \sum_j \left[calm \cdot Al_j + \sum_i \left(cv_{ij} P_{ij} + cf_i \lambda_{ij} + carr_i Arr_{ij} \right) \right] \\
&\sum_i P_{ij} Al_{j-1} = dem_j + Al_j \quad \forall j \quad (Al_0 = 0) \\
&P_{ij} \leq pm_i \lambda_{ij} \quad \forall i, j \\
&Arr_{ij} - Par_{ij} = \lambda_{ij} - \lambda_{ij-1} \quad \forall i, \forall j \geq 2 \\
&\lambda_{i0} = 1 \\
&P_{ij}, Al_j \geq 0, \lambda_{ij} \in \{0,1\} \\
&Arr_{ij} \in \{0,1\} \quad \text{ó} \quad Arr_{ij} \geq 0
\end{aligned}
$$

b)

$$
\begin{aligned}
&\lambda_{ij-1} + \lambda_{ij+1} - \lambda_{ij} \leq 1 \quad \forall i, 2 \leq j \leq n-1 \\
&\text{o bien} \\
&\lambda_{ij} + \lambda_{ij+1} \leq 2 - 2Par_{ij} \quad \forall i, 1 \leq j \leq n-1
\end{aligned}
$$

RESULTADO DEL PROBLEMA TRANSPORTE POR FERROCARRIL

a)

$$
\begin{aligned}
&\max z = \sum_i (b_i - c_i) x_i \\
&\sum_i v_i x_i \leq V \\
&\sum_i p_i x_i \leq P \\
&0 \leq x_i \leq mx_i \quad \forall i
\end{aligned}
$$

Se deben dar las cantidades de las mercancías a transportar, el valor de la variable dual de la primera restricción para dar el valor de cada unidad de volumen, el valor de la variable dual de la segunda restricción para dar el valor de cada unidad de peso y a cada propietario del que no se transporte nada el coste reducido de su variable correspondiente que sería en lo que tendría que aumentar su oferta para que se transporte algo de su mercancía.

b)

$$
\begin{aligned}
&\max z = \sum_i (a_i - g_i x_i - c_i) x_i \\
&\sum_i v_i x_i \leq V \\
&\sum_i p_i x_i \leq P \\
&0 \leq x_i \leq mx_i \quad \forall i
\end{aligned}
$$

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

c)

$$\max z = \sum_{i,j} (b_i - c_i) x_{ij} - cvag \sum_j y_j$$
$$\sum_i v_i x_{ij} \le \bar{V} y_j \quad \forall j$$
$$\sum_i p_i x_{ij} \le \bar{P} y_j \quad \forall j$$
$$\sum_j x_{ij} \le mx_i \quad \forall i$$
$$x_{ij} \ge 0, \quad y_j \in \{0,1\}$$

o bien

$$\max z = \sum_i (b_i - c_i) x_i - cvag \cdot y$$
$$\sum_i v_i x_i \le \bar{V} y$$
$$\sum_i p_i x_i \le \bar{P} y$$
$$0 \le x_i \le mx_i \quad \forall i$$
$$y \in \mathbb{Z}^+$$

I.4. Codificación de problemas de optimización

I.4.1. Lenguajes de modelado(OAE)

I.4.1.1. Lenguajes de modelado

Las principales alternativas actuales para el desarrollo de modelos de optimización suelen ser, Sharda (1995):

- *Lenguajes de programación de propósito general* (C, C++, Java, Visual Basic, FORTRAN 90) que llaman a una biblioteca de optimización
  Tienen sentido cuando el tiempo de solución es crítico o el modelo es ejecutado con mucha frecuencia o cuando se necesitan interfaces a medida para la entrada de datos o salida de resultados o cuando el modelo tiene que ser integrado en otra aplicación o se necesitan algoritmos de optimización específicos. Además permiten la implantación del modelo en un entorno software o hardware especial. Como contrapartida requiere un tiempo de desarrollo muy elevado y, sobre todo, presenta una gran dificultad y consumo de recursos para el mantenimiento del código.
  Actualmente ya existen bibliotecas de componentes orientados a objetos (clases C++) dedicadas exclusivamente a optimización, por ejemplo, Concert de ILOG, LINDO API de LINDO Systems, OptiMax 2000 de Maximal Software, FLOPC++ de Universidade de Aveiro.

- *Lenguajes o entornos de cálculo numérico o simbólico* (hojas de cálculo, lenguajes para cálculo numérico intensivo, como MATLAB, o para cálculo simbólico, como Maple o Mathematica, etc.)
  Los optimizadores de las hojas de cálculo, por ser aplicaciones muy comunes y conocidas, pueden ser un vehículo eficaz de difusión de un modelo entre cierto tipo de usuarios y facilitan el manejo de datos que se encuentren ya en dicho formato [Ragsdale, 1998]. Como ventajas específicas se pueden mencionar: su facilidad de uso, su integración total con la hoja de cálculo, la familiaridad con el entorno que facilita la explicación del modelo y de sus resultados, así como la facilidad de presentación de resultados en gráficos. Sin embargo, no inducen una buena práctica de programación, presentan la dificultad de su desarrollo, verificación, validación, actualización,

documentación y, en general, el mantenimiento del modelo y no permiten modelar problemas complejos o de gran tamaño [Gass, 1995].
Los lenguajes de cálculo numérico o simbólico no son específicos de problemas de optimización pero facilitan la manipulación numérica o simbólica de matrices y vectores. También disponen de funciones de optimización.
Todas estas alternativas pueden ser utilizadas para desarrollo rápido de un prototipo o una demostración ya que presentan capacidades de presentación gráfica que pueden ser aprovechadas. Son difícilmente utilizables cuando se plantean problemas de optimización de tamaño medio o superior.

- *Lenguajes algebraicos de modelado*
  Son las alternativas más complejas y potentes por su capacidad de indexación de las variables y ecuaciones, permiten cambiar sin dificultad las dimensiones del modelo, de forma natural separan datos de resultados. Desde el punto de vista del modelador permiten la detección de errores de consistencia en la definición y verificación del modelo. Desde el punto de vista del usuario simplifican drásticamente su mantenimiento. Entre los lenguajes de modelado más conocidos se pueden mencionar: GAMS (www.gams.com), AMPL (www.ampl.com) de origen estadounidense y MPL (www.maximalsoftware.com) y AIMMS (www.aimms.com) y XPRESS-MP (www.dash.co.uk) de origen europeo, por citar algunos. De algunos de ellos se pueden descargar versiones de estudiante desde sus páginas web. GAMS es el más antiguo, pero con el conjunto de usuarios más amplio, quizá por eso con algunas limitaciones en sus capacidades de modelado. AMPL es más nuevo, muy potente para el modelado pero con un conjunto reducido de usuarios. MPL es otro lenguaje de modelado robusto, cuya versión de estudiante acompaña al libro [Hillier y Lieberman, 2002].
  Existe una herramienta integrada denominada OPLStudio (www.ilog.com), en la que se dispone de un lenguaje de modelado (OPL) y varios optimizadores dependiendo del modelo propuesto. Está especialmente desarrollada para problemas de programación (*scheduling*) y planificación, aunque admite también cualquier modelo de optimización lineal y lineal entera mixta. Es una herramienta integrada ya que además del lenguaje de modelado, incluye sus propios optimizadores, Scheduler, Solver, CPLEX, estando los dos primeros basados en la programación de restricciones[4] y el último en programación matemática.

[4] Se denomina programación de restricciones a un tipo de programación lógica donde el dominio de las variables viene definido por relaciones lógicas y por restricciones.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

GAMS es el lenguaje más ampliamente difundido comercialmente con su propia lista de discusión de usuarios (gams-l@listserv.gmd.de) mientras que AMPL se está potenciando mucho en las universidades estadounidenses. Existe un proyecto denominado NEOS Server for Optimization (http://www-neos.mcs.anl.gov/neos/server-solver-types.html) para el cálculo distribuido que permite el envío de problemas de optimización escritos en AMPL o GAMS a través de internet y éstos son resueltos por optimizadores específicos para el tipo de problema enviado en servidores de la red devolviendo los resultados de la optimización.

Existen libros específicos que describen sus características y que sirven como guías de usuario tanto para el lenguaje GAMS [Brooke, 1998], [McCarl, 1998], para AMPL, [Fourer, 2000], o para OPL [Van Hentenryck, 1999]. Incluso en España se ha publicado un libro de optimización que se apoya en GAMS para la presentación de ejemplos [Mocholí, 1996]. Los campos de aplicación de estos lenguajes son tan amplios como los de la optimización propiamente dicha. Abarcan desde la micro y macroeconomía, a la economía de la energía, a la planificación energética o eléctrica, a la ingeniería química o forestal, a la planificación del desarrollo económico o del comercio internacional, a la cobertura de riesgos financieros, a problemas de transporte y comunicaciones, a organización de la producción o fabricación o a la planificación de grandes proyectos. En el caso de la programación de restricciones ésta aparece especialmente en problemas combinatorios para modelar restricciones lógicas.

I.4.1.2. Lenguajes algebraicos de modelado

Los lenguajes algebraicos son lenguajes de alto nivel que han sido diseñados específicamente para el desarrollo e implantación de modelos de optimización de forma más directa para los programadores y más inteligible para los usuarios. En consecuencia, el campo de actuación y utilidad de los modelos de optimización se ha ampliado tremendamente al utilizar estos lenguajes. Entre sus *características* y *ventajas* principales destacan las siguientes:

- Proporcionan una formulación sencilla de modelos grandes y complejos
- Facilitan sobremanera el desarrollo de prototipos
- Mejoran sustancialmente la productividad de los modeladores al permitir dedicar más tiempo al diseño, ejecución del modelo y análisis de los resultados y menos a la codificación del mismo
- Estructuran los buenos hábitos de modelado al exigir una representación concisa y exacta de los parámetros/variables y sus relaciones
- Recogen simultáneamente la estructura del modelo y su documentación

- Separan de manera natural los datos de la estructura del modelo y ésta de los algoritmos de solución
- La formulación del problema es independiente del tamaño. Permiten el uso de la estructura del modelo para diferentes casos[5]
- Los optimizadores pueden ser intercambiados sin dificultad, se pueden probar nuevos optimizadores, nuevos métodos o nuevas versiones
- Por ejemplo, en el lenguaje GAMS se encuentran entre otros disponibles los optimizadores CPLEX, OSL, XA y XPRESS para problemas LP y MIP, MINOS y CONOPT para problemas NLP, DICOPT para problemas MINLP y MILES y PATH para problemas MCP.
- Permiten la realización de cambios en el modelo de manera sencilla y segura, es decir, se puede afrontar un refinamiento continuo en la formulación del problema
- Cualquier tipo de problemas de programación lineal, no lineal, flujos en redes o mixta complementaria resulta muy fácil implantar su formulación
- Permiten la implantación de algoritmos avanzados, que incluyan varias llamadas al optimizador o procedimientos específicos para el problema (como por ejemplo los métodos de descomposición)
- Permiten la portabilidad de los modelos entre plataformas y sistemas operativos

Como *desventajas* principales se pueden mencionar las siguientes:

- No son adecuados para la resolución de problemas de pequeño tamaño por parte de usuarios esporádicos por la barrera de entrada que supone el aprendizaje de un nuevo lenguaje
- No pueden utilizarse para la resolución directa de problemas gigantescos cuya formulación completa incluso no se puede realizar (por ejemplo, a partir de 1 millón de restricciones y/o variables)
- En la ejecución se incluye un tiempo de creación del modelo y de interfaz con el optimizador que ralentiza la obtención de la solución, por lo tanto no es recomendable cuando el tiempo de ejecución es un factor crítico.

Las *tendencias* o características más actuales en el desarrollo de lenguajes algebraicos se mueven hacia:

[5] Una manera habitual de desarrollar es utilizar una maqueta (caso ejemplo) para la depuración y verificación del modelo y una vez comprobada su validez utilizar el caso real a ser resuelto.

- Interfaces de entrada y salida de datos más estrechamente relacionadas con bases de datos u hojas de cálculo
- El desarrollo de interfaces gráficas que faciliten al usuario la formulación visual y el entendimiento de problemas de optimización
- Interfaz con lenguajes de propósito general para la incorporación de funciones externas definidas por el usuario dentro de la optimización
- El avance en las capacidades de resolución directa de problemas estocásticos (con adición de características específicas en el propio lenguaje y uso de algoritmos de descomposición en el optimizador) o problemas no lineales complejos
- La posibilidad de ocultar el código fuente produciendo versiones ejecutables para usuarios finales
- La selección automática del método y optimizador

La experiencia personal por el uso de estos lenguajes de modelado ha sido tremendamente positiva. En el Instituto de Investigación Tecnológica (IIT) de la Escuela Técnica Superior de Ingeniería se pasó a partir del año 1991 de utilizar lenguajes de propósito general (como FORTRAN) para el desarrollo de modelos de optimización a utilizar exclusivamente lenguajes de modelado (como GAMS). Esto ha representado un salto importante en cuanto a la productividad de los modeladores. Aplicaciones que antes requerían decenas de miles de líneas de código (en FORTRAN) ahora se desarrollan con una décima parte de la longitud original y con un esfuerzo muy inferior en tiempo (menos de la cuarta parte). De hecho, el IIT ha desarrollado desde el año 1993 numerosos modelos de optimización para el sector eléctrico y ha sido el precursor en España del uso de lenguajes algebraicos en el campo de la planificación, operación y economía del sector eléctrico. En este sector, cuya regulación ha cambiado recientemente, ha sido de vital importancia la capacidad de mantener o modificar de manera muy sencilla los modelos de planificación debido al uso de estos lenguajes.

I.4.1.3. Referencias

Brooke, A., Kendrick, D. and Meeraus, A. (1998) *GAMS: A User's Guide.* GAMS Development Co.

Fourer, R., Gay, D.M. and Kernighan, B.W. (2000) *AMPL: A Modeling Language for Mathematical Programming.* The Scientific Press. 2nd ed.

Ragsdale, C. T. (1998) *Spreadsheet modeling and decision analysis: a practical introduction to management science.* South-Western College. 2nd ed.

Gass, S.I., Hirshfeld, D.S. and Wasil, E.A. (1995) "Model World: The Spreadsheeting of OR/MS" *Interfaces* pp. 72-81. September-October.

McCarl, B.A. and Spreen, Th.H. (1998) *Applied Mathematical Programming using Algebraic Systems.* Technical Report.

Mocholí, M. y Sala, R. (1996) *Decisiones de optimización* Tirant lo Blanch. Valencia.

Sharda, R. and Rampal, G. (1995) "Algebraic Modeling Languages on PCs" *OR/MS Today* pp. 58-63. June.

Van Hentenryck, P. (1999) *The OPL Optimization Programming Language.* The MIT Press.

I.4.2. Modelado en GAMS(OAE)

En este apartado se presentan varios ejemplos sencillos que permiten mostrar algunas de las características del lenguaje GAMS. Sin embargo, el manual de usuario contiene un capítulo tutorial y la referencia de todas las características del lenguaje.

I.4.2.1. Ejemplo de transporte

Veamos a continuación un caso típico de un problema de optimización lineal clásico y cómo este problema se codifica en el lenguaje GAMS. En el apartado de modelado en programación lineal entera mixta se presenta formalmente este problema y sus características. Sean i fábricas de envasado y j mercados de consumo. Cada fábrica tiene una capacidad máxima de producción de a_i cajas y cada mercado demanda una cantidad b_j de cajas (se supone que la capacidad de producción total de las fábricas es superior a la demanda total para que el problema sea factible). El coste de transporte entre cada fábrica i y cada mercado j por cada caja es c_{ij}. Se desea satisfacer la demanda de cada mercado al mínimo coste. Las variables de decisión del problema serán las cajas transportadas entre cada fábrica i y cada mercado j, x_{ij}. Las ecuaciones que deben satisfacerse son:

Límite de capacidad máxima de producción de cada fábrica

$$\sum_j x_{ij} \leq a_i \text{ para cada fábrica } i$$

Satisfacción de la demanda de cada mercado

$$\sum_i x_{ij} \geq b_j \text{ para cada mercado } j$$

La función objetivo será la minimización de los costes totales de transporte

$$\sum_i \sum_j c_{ij} x_{ij}$$

Ésta es la forma algebraica de representación de este problema de optimización. La codificación en lenguaje GAMS aparece a continuación.

```
$TITLE MODELO DE TRANSPORTE

SETS
   I fábricas de envasado / VIGO, ALGECIRAS /
   J mercados de consumo  / MADRID, BARCELONA, VALENCIA /

PARAMETERS
   A(i) capacidad de producción de la fábrica i [cajas]
       / VIGO      350
         ALGECIRAS 700 /

   B(j) demanda del mercado j [cajas]
       / MADRID    400
         BARCELONA 450
         VALENCIA  150 /

TABLE C(i,j) coste unitario transporte entre i y j [miles de euros por caja]
          MADRID BARCELONA VALENCIA
VIGO       0.06     0.12     0.09
ALGECIRAS  0.05     0.15     0.11

VARIABLES
   X(i,j) cajas transportadas entre fábrica i y mercado j [cajas]
   CT     coste de transporte [miles de euros]

POSITIVE VARIABLE X

EQUATIONS
   COSTE        coste total de transporte     [miles de euros]
   CAPACIDAD(i) capacidad máxima de cada fábrica i     [cajas]
   DEMANDA(j)   satisfacción demanda de cada mercado j [cajas] ;

COSTE ..        CT =E= SUM[(i,j), C(i,j) * X(i,j)] ;

CAPACIDAD(i) .. SUM[j, X(i,j)] =L= A(i) ;

DEMANDA(j) ..   SUM[i, X(i,j)] =G= B(j) ;

MODEL TRANSPORTE / COSTE, CAPACIDAD, DEMANDA /

SOLVE TRANSPORTE USING LP MINIMIZING CT
```

Se ha puesto especial cuidado en presentar de forma clara, concisa y limpia el código. El contenido de este código resulta prácticamente autoexplicativo. El resultado de la ejecución del modelo de transporte se presenta a continuación.

```
C o m p i l a t i o n

COMPILATION TIME     =        0.000 SECONDS    0.7 Mb      WIN-19-115

Equation Listing    SOLVE TRANSPORTE USING LP FROM LINE 39

---- COSTE  =E=  coste total de transporte   [miles de euros]

COSTE..  - 0.06*X(VIGO,MADRID) - 0.12*X(VIGO,BARCELONA) - 0.09*X(VIGO,VALENCIA)

      - 0.05*X(ALGECIRAS,MADRID) - 0.15*X(ALGECIRAS,BARCELONA)

      - 0.11*X(ALGECIRAS,VALENCIA) + CT =E= 0 ; (LHS = 0)

---- CAPACIDAD  =L=  capacidad máxima de cada fábrica i     [cajas]

CAPACIDAD(VIGO)..  X(VIGO,MADRID) + X(VIGO,BARCELONA) + X(VIGO,VALENCIA) =L= 350 ;

      (LHS = 0)

CAPACIDAD(ALGECIRAS)..  X(ALGECIRAS,MADRID) + X(ALGECIRAS,BARCELONA)
```


UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

```
      + X(ALGECIRAS,VALENCIA) =L= 700 ; (LHS = 0)

---- DEMANDA  =G=  satisfacción demanda de cada mercado j [cajas]

DEMANDA(MADRID)..  X(VIGO,MADRID) + X(ALGECIRAS,MADRID) =G= 400 ;

      (LHS = 0, INFES = 400 ***)

DEMANDA(BARCELONA)..  X(VIGO,BARCELONA) + X(ALGECIRAS,BARCELONA) =G= 450 ;

      (LHS = 0, INFES = 450 ***)

DEMANDA(VALENCIA)..  X(VIGO,VALENCIA) + X(ALGECIRAS,VALENCIA) =G= 150 ;

      (LHS = 0, INFES = 150 ***)

Column Listing      SOLVE TRANSPORTE USING LP FROM LINE 39

---- X  cajas transportadas entre fábrica i y mercado j [cajas]

X(VIGO,MADRID)
                (.LO, .L, .UP = 0, 0, +INF)
       -0.06    COSTE
        1       CAPACIDAD(VIGO)
        1       DEMANDA(MADRID)

X(VIGO,BARCELONA)
                (.LO, .L, .UP = 0, 0, +INF)
       -0.12    COSTE
        1       CAPACIDAD(VIGO)
        1       DEMANDA(BARCELONA)

X(VIGO,VALENCIA)
                (.LO, .L, .UP = 0, 0, +INF)
       -0.09    COSTE
        1       CAPACIDAD(VIGO)
        1       DEMANDA(VALENCIA)

REMAINING 3 ENTRIES SKIPPED

---- CT  coste de transporte [miles de euros]

CT
                (.LO, .L, .UP = -INF, 0, +INF)
        1       COSTE

Model Statistics    SOLVE TRANSPORTE USING LP FROM LINE 39

MODEL STATISTICS

BLOCKS OF EQUATIONS       3     SINGLE EQUATIONS        6
BLOCKS OF VARIABLES       2     SINGLE VARIABLES        7
NON ZERO ELEMENTS        19

GENERATION TIME      =        0.140 SECONDS    1.4 Mb      WIN-19-115

EXECUTION TIME       =        0.140 SECONDS    1.4 Mb      WIN-19-115

               S O L V E      S U M M A R Y

     MODEL   TRANSPORTE          OBJECTIVE  CT
     TYPE    LP                  DIRECTION  MINIMIZE
     SOLVER  CPLEX               FROM LINE  39

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE               93.5000

 RESOURCE USAGE, LIMIT          0.401     1000.000
 ITERATION COUNT, LIMIT         5        10000

GAMS/Cplex    May 18, 2000 WIN.CP.CP 19.3 016.014.038.WAT For Cplex 6.6
Cplex 6.6.1, GAMS Link 16, Using a GAMS/Cplex demo license installed at runtime.

Optimal solution found.

Objective :          93.500000

                       LOWER     LEVEL     UPPER    MARGINAL

---- EQU COSTE           .         .         .        1.000

  COSTE  coste total de transporte   [miles de euros]

---- EQU CAPACIDAD  capacidad máxima de cada fábrica i     [cajas]

             LOWER     LEVEL     UPPER    MARGINAL
```

```
VIGO          -INF    350.000   350.000    -0.030
ALGECIRAS     -INF    650.000   700.000      .

---- EQU DEMANDA  satisfacción demanda de cada mercado j [cajas]

             LOWER     LEVEL     UPPER    MARGINAL

MADRID      400.000   400.000     +INF      0.050
BARCELONA   450.000   450.000     +INF      0.150
VALENCIA    150.000   150.000     +INF      0.110

---- VAR X  cajas transportadas entre fábrica i y mercado j [cajas]

                       LOWER     LEVEL     UPPER    MARGINAL

VIGO     .MADRID         .         .        +INF      0.040
VIGO     .BARCELONA      .      350.000     +INF       .
VIGO     .VALENCIA       .         .        +INF      0.010
ALGECIRAS.MADRID         .      400.000     +INF       .
ALGECIRAS.BARCELONA      .      100.000     +INF       .
ALGECIRAS.VALENCIA       .      150.000     +INF       .

                       LOWER     LEVEL     UPPER    MARGINAL

---- VAR CT             -INF     93.500     +INF       .

  CT  coste de transporte [miles de euros]

**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED

EXECUTION TIME       =        0.030 SECONDS    0.7 Mb      WIN-19-115

**** FILE SUMMARY

INPUT      D:\TR.GMS
OUTPUT     D:\TR.LST
```

I.4.2.2. Ejemplo de planificación de la producción

A continuación se presenta un ejemplo de planificación de la producción de una fábrica de papel. Se dispone de varias máquinas para producir diferentes tipos de papel. Se trata de determinar cuáles son las cantidades óptimas a producir de cada tipo de papel en cada máquina para maximizar el beneficio. La demanda de cada tipo de papel se considera fija y conocida y existen limitaciones en el tiempo de producción disponible en cada máquina.

```
$TITLE Planificación de la producción de una papelera

* la papelera puede producir cuatro tipos diferentes de papel en tres máquinas
* distintas. Dada una demanda fija se trata de determinar la producción que
* maximiza los beneficios mensuales

SETS
   M máquinas       / maquina1 * maquina3 /
   P tipos de papel / prensa, folio, imprenta, reciclado /

TABLE TASAPROD(p,m) tasa de producción (tm por h)
                 maquina1  maquina2  maquina3
      prensa        53        52        49
      folio         51        49        44
      imprenta      52        45        47
      reciclado     42        44        40

TABLE COSTEPROD(p,m) coste de producción (€ por tm)
                 maquina1  maquina2  maquina3
      prensa        76        75        73
      folio         82        80        78
      imprenta      96        95        92
      reciclado     72        71        70

TABLE DATDEM(p,*) demanda y precio de venta
                  demanda   precio
*              (tm por mes)(€ por tm)
      prensa        30000      77
      folio         20000      81
```

```
      imprenta      12000      99
      reciclado      8000     105

PARAMETER TIEMPOMAQ(m) tiempo disponible al mes de cada máquina (h)
                   / maquina1   672
                     maquina2   600
                     maquina3   480 /

VARIABLES
   PRODUCC(p,m) producción de cada tipo papel en cada máquina (tm por mes)
   BENEFICIO    beneficio  (€ por mes)

POSITIVE VARIABLE PRODUCC

EQUATIONS
   CAPACMAQ(m) capacidad de cada máquina     (h  por mes)
   DEMANDAP(p) demanda de cada tipo de papel (tm por mes)
   BENEF       beneficio                     (€  por mes) ;

CAPACMAQ(m) .. SUM[p, PRODUCC(p,m)/TASAPROD(p,m)] =L= TIEMPOMAQ(m) ;

DEMANDAP(p) .. SUM[m, PRODUCC(p,m)] =E= DATDEM(p,'demanda') ;

BENEF       .. BENEFICIO =E= SUM[p, DATDEM(p,'demanda')*DATDEM(p,'precio')]
                           - SUM[(p,m), COSTEPROD(p,m)*PRODUCC(p,m)] ;

MODEL PAPELERA / ALL /

SOLVE PAPELERA USING LP MAXIMIZING BENEFICIO
```

I.4.2.3. Ejemplo de secuenciación de órdenes de trabajo

Dada una máquina y 5 trabajos que hay que realizar en ella, en cualquier orden, se dispone del tiempo de ejecución de cada trabajo

TR1	TR2	TR3	TR4	TR5
15	13	12	14	16

y del tiempo de ajuste de la máquina para pasar de ejecutar el trabajo i (fila) a ejecutar el trabajo j (columna)

	TR1	TR2	TR3	TR4	TR5
TR1		2	5	1	6
TR2	3		4	2	5
TR3	4	2		3	4
TR4	5	3	6		5
TR5	4	4	4	3	

Resolver el problema de determinar cuál es el menor tiempo posible para completar los 5 trabajos y cómo hacerlo. Se considera un ciclo de trabajo cerrado, que se repite y vuelve a comenzar.

```
$TITLE Secuenciación de órdenes de trabajo

* La segunda formulación evita subciclos de parejas de trabajos

SETS
   I trabajos que se van a ejecutar / TR1 * TR5 /

ALIAS (i,j)

TABLE C(i,j) tiempo de ajuste para pasar del trabajo i al trabajo j
```

```
        TR1    TR2    TR3    TR4    TR5
TR1             2      5      1      6
TR2      3             4      2      5
TR3      4      2             3      4
TR4      5      3      6             5
TR5      4      4      4      3

VARIABLES
   X(i,j) paso del trabajo i al trabajo j
   TT     tiempo total en completar los trabajos

BINARY VARIABLE X

EQUATIONS
   TIEMPO       tiempo total de trabajo
   ANTERIOR(i)  de cada trabajo se parte una vez
   POSTERIOR(j) a  cada trabajo se llega una vez
   PAREJAS(i,j) suma de los trabajos por parejas ;

TIEMPO       .. TT =E= SUM[(i,j) $(NOT SAMEAS(i,j)), C(i,j)*X(i,j)] ;
ANTERIOR(i)  .. SUM[j $(NOT SAMEAS(i,j)), X(i,j)] =E= 1 ;
POSTERIOR(j) .. SUM[i $(NOT SAMEAS(i,j)), X(i,j)] =E= 1 ;
PAREJAS(i,j) $(ORD(i) < ORD(j)) .. X(i,j) + X(j,i) =L= 1 ;

MODEL AJUSTE1 / TIEMPO, ANTERIOR, POSTERIOR          /
MODEL AJUSTE2 / TIEMPO, ANTERIOR, POSTERIOR, PAREJAS /

SOLVE AJUSTE1 USING MIP MINIMIZING TT

* número de restricciones de parejas son C(5,2)=10

SOLVE AJUSTE2 USING MIP MINIMIZING TT
```

I.4.2.4. Ejemplo del viajante de comercio

Veamos tres posibles formulaciones del problema del viajante de comercio escritas en GAMS. La primera formulación es la denominada de Miller, Tucker y Zemlin

$$\begin{aligned}
&\min_{x_{ij},u_i} \sum_{i,j} c_{ij} x_{ij} \\
&\sum_i x_{ij} = 1 \quad \forall j \\
&\sum_j x_{ij} = 1 \quad \forall i \\
&u_i - u_j + [\text{card}(i) - 1] x_{ij} \le \text{card}(i) - 2 \quad \forall i, j \\
&x_{ij} \in \{0,1\}, u_i \ge 0
\end{aligned} \tag{1.13}$$

y las dos últimas a las presentadas previamente en la sección I.3.2 de formulación de problemas de optimización lineal entera.

```
$TITLE Problema del viajante de comercio

SETS
I        Ciudades
K        Etapas

ALIAS(I,J,R,S)

PARAMETER
COSTE(i,j)       Coste de ir de la ciudad i a la ciudad j [€]

VARIABLE
FOBJ             Función objetivo [€]

BINARY VARIABLES
X1(i,j)          Indica si se viaja de la ciudad i a la ciudad j
X2(i,j,k)        Indica si se viaja de la ciudad i a la ciudad j en la etapa k

POSITIVE VARIABLE
```

```
U(i)             Etapa en que se visita una cierta ciudad

EQUATIONS
E_FOBJ1                Función objetivo
E_ORIGEN1(i)           Cada ciudad es origen  una sola vez
E_DESTINO1(j)          Cada ciudad es destino una sola vez
E_SUBCICLO1(i,j)       Restricciones para eliminar subciclos
E_SUBCICLO2(i,j)       Restricciones para eliminar subciclos de orden 2
E_SUBCICLO3(i,j,r)     Restricciones para eliminar subciclos de orden 3
E_SUBCICLO4(i,j,r,s)   Restricciones para eliminar subciclos de orden 4

E_FOBJ2          Función objetivo
E_ORIGEN2(i)     Cada ciudad es origen  una sola vez
E_DESTINO2(j)    Cada ciudad es destino una sola vez
E_ETAPA(k)       En cada etapa se hace sólo un recorrido
E_ORIG_DEST(j,k) La ciudad destino en k es origen en k+1 ;

E_FOBJ1       .. FOBJ =E= SUM[(i,j) $(NOT SAMEAS(i,j)), COSTE(i,j) * X1(i,j)];

E_ORIGEN1(i)  .. SUM[j $(NOT SAMEAS(i,j)), X1(i,j)] =E= 1;

E_DESTINO1(j) .. SUM[i $(NOT SAMEAS(i,j)), X1(i,j)] =E= 1;

E_SUBCICLO1(i,j) $(ORD(i) > 1 AND ORD(j) > 1 AND NOT SAMEAS(i,j)) ..
                   U(i) - U(j) + [CARD(i) - 1] * X1(i,j) =L= CARD(i) - 2;

E_SUBCICLO2(i,j) $(NOT SAMEAS(i,j)) .. X1(i,j) + X1(j,i) =L= 1;

E_SUBCICLO3(i,j,r)   $(NOT SAMEAS(i,j) AND NOT SAMEAS(j,r) AND NOT SAMEAS(r,i)) ..
                       X1(i,j) + X1(j,r) + X1(r,i) =L= 2;

E_SUBCICLO4(i,j,r,s) $(NOT SAMEAS(i,j) AND NOT SAMEAS(j,r) AND NOT SAMEAS(r,i) AND
                       NOT SAMEAS(r,s) AND NOT SAMEAS(s,i) AND NOT SAMEAS(s,j)) ..
                       X1(i,j) + X1(j,r) + X1(r,s) + X1(s,i) =L= 3;

E_FOBJ2          .. FOBJ =E= SUM[(i,j,k) $(NOT SAMEAS(i,j)), COSTE(i,j) * X2(i,j,k)];

E_ORIGEN2(i)     .. SUM[(j,k) $(NOT SAMEAS(i,j)), X2(i,j,k)] =E= 1;

E_DESTINO2(j)    .. SUM[(i,k) $(NOT SAMEAS(i,j)), X2(i,j,k)] =E= 1;

E_ETAPA(k)       .. SUM[(i,j) $(NOT SAMEAS(i,j)), X2(i,j,k)] =E= 1;

E_ORIG_DEST(j,k) .. SUM[i $(NOT SAMEAS(i,j)), X2(i,j,k)] =E=
                    SUM[r $(NOT SAMEAS(j,r)), X2(j,r,k ++ 1)];

MODEL TSP1  / E_FOBJ1, E_ORIGEN1, E_DESTINO1, E_SUBCICLO1 /

MODEL TSP1B / E_FOBJ1, E_ORIGEN1, E_DESTINO1, E_SUBCICLO2, E_SUBCICLO3, E_SUBCICLO4 /

MODEL TSP2  / E_FOBJ2, E_ORIGEN2, E_DESTINO2, E_ETAPA, E_ORIG_DEST /

********************************************************************************
* Caso ejemplo: Schrage (1997) p. 319
* Esta información se podría introducir mediante ficheros de texto
* utilizando la instrucción $INCLUDE nombrefichero

SETS
I        Ciudades
/ Atl, Chi, Cin, Hou, LA, Mon, NY, Phi, Pit, StL, SD, SF /

K        Etapas
/ 1 * 12 /

TABLE COSTE(i,j) Coste de ir de la ciudad i a la ciudad j [€]
      Atl   Chi   Cin   Hou   LA    Mon   NY    Phi   Pit   StL    SD    SF
Atl   0     702   454   842   2396  1196  864   772   714   554   2363  5679
Chi   702   0     324   1093  2136  764   845   764   459   294   2184  2187
Cin   454   324   0     1137  2180  798   664   572   284   338   2228  2463
Hou   842   1093  1137  0     1616  1857  1706  1614  1421  799   1521  2021
LA    2396  2136  2180  1616  0     2900  2844  2752  2464  1842  95    405
Mon   1196  764   798   1857  2900  0     396   424   514   1058  2948  2951
NY    864   845   664   1706  2844  396   0     92    386   1002  2892  3032
Phi   772   764   572   1614  2752  424   92    0     305   910   2800  2951
Pit   714   459   284   1421  2464  514   386   305   0     622   2512  2646
StL   554   294   338   799   1842  1058  1002  910   622   0     1890  2125
SD    2363  2184  2228  1521  95    2948  2892  2800  2512  1890  0     500
SF    2679  2187  2463  2021  405   2951  3032  2951  2646  2125  500   0    ;

COSTE(i,j) = COSTE(i,j) / 1e3 ;

U.UP(i) = CARD(i) ;

OPTION OPTCR = 0

* Orden de ejecución

SOLVE TSP1 USING MIP MINIMIZING FOBJ
```

```
DISPLAY X1.L, U.L

SOLVE TSP1B USING MIP MINIMIZING FOBJ
DISPLAY X1.L

SOLVE TSP2 USING MIP MINIMIZING FOBJ
DISPLAY X2.L
```

I.4.2.5. Ejemplo de asignación de grupos térmicos

El problema de la asignación de grupos térmicos de producción de electricidad consiste en la decisión de qué grupos térmicos hay que acoplar en cada hora del día (o semana) de manera que:

- Se minimicen los costes variables de generación (incluyendo costes de combustible y costes de arranque y parada)
- Se suministre la demanda en cada hora
- Se mantenga un cierto nivel de reserva rodante
- Se respeten los parámetros de funcionamiento de los grupos térmicos (mínimos técnicos, potencia nominal, rampas de subida y bajada)

Datos

D_h demanda térmica en la hora h [MW]

R coeficiente de reserva rodante con respecto a la demanda [p.u.]

a_t término lineal del coste de combustible del grupo térmico t [€/MWh]

b_t término fijo del coste de combustible del grupo térmico t [€/h]

ca_t coste de arranque del grupo térmico t [€]

cp_t coste de parada del grupo térmico t [€]

$\overline{P}_t$ potencia máxima del grupo térmico t [MW]

$\underline{P}_t$ potencia mínima del grupo térmico t [MW]

rs_t rampa de subida del grupo térmico t [MW/h]

rb_t rampa de bajada del grupo térmico t [MW/h]

Variables

P_{ht} potencia producida por el grupo térmico t en la hora h [MW]

A_{ht} acoplamiento del grupo térmico t en la hora h [0,1]

AR_{ht} arranque del grupo térmico t en la hora h [0,1]

PR_{ht} parada del grupo térmico t en la hora h [0,1]

$$\min \sum_{h=1}^{H} \sum_{t=1}^{T} \left(a_t P_{ht} + b_t A_{ht} + ca_t AR_{ht} + cp_t PR_{ht} \right)$$

$$\sum_{t=1}^{T} P_{ht} = D_h \qquad H$$

$$\sum_{t=1}^{T} (\overline{P}_t A_{ht} - P_{ht}) = RD_h \qquad H$$

$$\underline{P}_t A_{ht} \leq P_{ht} \leq \overline{P}_t A_{ht} \qquad 2HT$$

$$A_{ht} - A_{h-1t} = AR_{ht} - PR_{ht} \qquad (H-1)T$$

$$P_{ht} - P_{h-1t} \leq rs_t \qquad (H-1)T$$

$$P_{h-1t} - P_{ht} \leq rb_t \qquad (H-1)T$$

$$P_{ht} \geq 0 \qquad A_{ht}, AR_{ht}, PR_{ht} \in \{0,1\}$$

```
$TITLE ASIGNACIÓN HORARIA DE GRUPOS TÉRMICOS

SETS
T grupos térmicos /GALICIA, CATALUNA, MADRID, VALENCIA,
                   EXTREMAD, ANDALUCI, CASTLEON/
H hora h /h1 * h5/

SCALAR
r porcentaje de reserva rodante sobre la demanda [p.u.] /0.2/

PARAMETERS
d(h) demanda cada hora [MW]
/h1 1000 , h2 1400 , h3 2400 , h4 2000 , h5 1000/
pmax(t) pot máxima de cada térmico [MW]
/GALICIA 400, CATALUNA 500, MADRID 700, VALENCIA 400,
 EXTREMAD 300, ANDALUCI 800, CASTLEON 800/
pmin(t) pot mínima de cada térmico [MW]
/GALICIA 100, CATALUNA 150, MADRID 150, VALENCIA 50,
 EXTREMAD 50, ANDALUCI 400, CASTLEON 200 /
rs(t) rampa de subida [MW por hora]
/GALICIA 200, CATALUNA 300, MADRID 500, VALENCIA 300,
 EXTREMAD 100, ANDALUCI 500, CASTLEON 400/
rb(t) rampa de bajada [MW por hora]
/GALICIA 300, CATALUNA 300, MADRID 200, VALENCIA 100,
 EXTREMAD 100, ANDALUCI 500, CASTLEON 400/
c(t)  coste lineal de producción [€ por MWh]
/GALICIA 4, CATALUNA 4, MADRID 4, VALENCIA 4,
 EXTREMAD 3, ANDALUCI 2, CASTLEON 7/
b(t)  coste fijo de producción [€]
/GALICIA 50, CATALUNA 30, MADRID 30, VALENCIA 25,
 EXTREMAD 30, ANDALUCI 80, CASTLEON 70/
ca(t) coste de arranque
/GALICIA 10, CATALUNA 20, MADRID 10, VALENCIA 15,
 EXTREMAD 20, ANDALUCI 10, CASTLEON 15/
cp(t) coste de parada
/GALICIA 5, CATALUNA 10, MADRID 5, VALENCIA 10,
 EXTREMAD 5, ANDALUCI 15, CASTLEON 10/

VARIABLES
CT      coste variable total del sistema [M€]
A(t,h)  acoplamiento del grupo t a las h horas [0-1]
AR(t,h) arranque     del grupo t a las h horas [0-1]
PR(t,h) parada       del grupo t a las h horas [0-1]
P(t,h)  generación producida por el grupo t a las h horas [MW]

BINARY VARIABLE A,AR,PR
POSITIVE VARIABLE P

EQUATIONS
COSTE          costes variables de generación-función objetivo [€]
DEMANDA(h)     abastecimiento de la demanda [MW]
RESERVA(h)     reserva rodante del sistema [MW]
COTASUP(t,h)   cota superior de producción del grupo t [MW]
COTAINF(t,h)   cota inferior de producción del grupo t [MW]
RAMPASUB(t,h)  limitación de rampa de subida del grupo t [MW]
RAMPABAJ(t,h)  limitación de rampa de bajada del grupo t [MW]
LOGICA(t,h)    relación lógica entre variables de acoplamiento arranque y parada ;

COSTE .. CT =E= SUM[(T,H), c(t)*P(t,h)+b(t)*A(t,h)+ca(t)*AR(t,h)+cp(t)*PR(t,h)] ;
```

```
DEMANDA(h) .. SUM[T, P(t,h)] =E= d(h) ;
RESERVA(h) .. SUM[T, A(t,h)*pmax(t)-P(t,h)] =G= d(h)*r;
COTASUP(t,h) .. P(t,h) =L= pmax(t)*A(t,h);
COTAINF(t,h) .. P(t,h) =G= pmin(t)*A(t,h);
RAMPASUB(t,h) .. P(t,h)-P(t,h-1) =L= rs(t);
RAMPABAJ(t,h) .. P(t,h-1)-P(t,h) =L= rb(t);
LOGICA(t,h) .. A(t,h)-A(t,h-1) =E= AR(t,h)-PR(t,h);
MODEL ASIGNA /COSTE,DEMANDA,RESERVA,COTASUP,COTAINF,RAMPASUB,RAMPABAJ,LOGICA/ ;
P.UP(t,h) = pmax(t)
OPTION OPTCR = 0
SOLVE ASIGNA USING MIP MINIMIZING CT
```

I.4.2.6. Ejemplo de flujo de cargas óptimo

Veamos a continuación la formulación de un ejemplo de sistemas de energía eléctrica. En particular, se trata de un *flujo de cargas óptimo en corriente continua con y sin pérdidas óhmicas* que se formula como un problema de optimización lineal y no lineal dependiendo de la consideración o no de las pérdidas. Este ejemplo ilustra alguna característica muy interesante del GAMS como son los conjuntos dinámicos y su uso para restringir las variables y ecuaciones del problema. Aunque no se entienda el significado físico del problema por carecer de los conocimientos adecuados en sistemas de energía eléctrica su formulación en GAMS se puede seguir sin dificultad.

La *estructura* general de un modelo de optimización escrito en GAMS se presenta en la tabla 1.2. Esta estructura es la que se ha empleado en este caso ejemplo.

Índices y parámetros	Todos los índices y parámetros del modelo se declaran al comienzo del mismo. Se inicializarán a sus valores por omisión aquellos que sea necesario.
Variables	Definición de las variables según sean positivas, libres, binarias, etc.
Ecuaciones	Declaración y definición de las restricciones. Se controlará con cuidado las condiciones de validez u ocurrencia de las mismas.
Modelo	Declaración de las ecuaciones que componen el modelo.
Inclusión y manipulación de datos de entrada	Los datos de entrada se introducen desde ficheros independientes, después se realizan los cálculos de parámetros auxiliares dependientes de los datos de entrada.

Acotamiento e inicialización de variables	Acotamiento de las variables a sus cotas físicas e inicialización cuando tenga sentido.
Resolución del problema de optimización	
Presentación de resultados	Presentación de los resultados elaborados a partir de la solución del problema de optimización.

Tabla 1.2 Estructura de un modelo de optimización en GAMS.

Se trata de minimizar los costes variables de operación en un intervalo horario unitario compuestos por los costes variables de los grupos térmicos, los costes de oportunidad de los grupos hidráulicos cuando producen por encima de su potencia programada y el coste variable de la potencia no suministrada.

$$\sum_{t=1}^{T} v_t GTR_t + \sum_{h=1}^{H} v_h GHE_h + \sum_{n=1}^{N} v_n PNS_n$$

siendo datos v_t el coste variable unitario de generación del grupo térmico t, v_h el coste de oportunidad unitario de la hidráulica de emergencia h y v_n el coste variable unitario de la potencia no suministrada en el nudo n. Las variables son: GTR_t potencia producida por el grupo térmico t, GHE_h potencia hidráulica de emergencia del grupo hidráulico h y PNS_n potencia no suministrada en el nudo n. T, H y N son todos los grupos térmicos, hidráulicos y nudos del sistema respectivamente.

Las restricciones que condicionan este problema se muestran a continuación.

La primera ley de Kirchhoff que establece el balance entre generación y demanda en cada nudo:

$$\sum_{t \in n} GTR_t + \sum_{h \in n} (GHP_h + GHE_h) + PNS_n + \sum_{i=1}^{I} F_{i \to n} - \sum_{j=1}^{J} F_{n \to j} = D_n$$

siendo $t \in n$, $h \in n$ los grupos térmicos e hidráulicos localizados en el nudo n, GHP_h es la variable potencia hidráulica programada del grupo hidráulico h, $F_{i \to j}$ es la variable flujo de potencia que va del nudo i al nudo j y D_n corresponde al dato de la demanda de potencia en el nudo n.

La segunda ley de Kirchhoff nos dice que el flujo por una línea es proporcional a la diferencia de los ángulos de tensión de sus nudos extremos:

$$\frac{X_{i \to j}}{S_B} F_{i \to j} = \theta_i - \theta_j$$

donde se conocen $X_{i\to j}$ es la reactancia de la línea que une los nudos i y j y S_B es la potencia base del sistema y θ_i es la variable de ángulo de tensión del nudo i.

Algunas variables del problema están acotadas entre ciertos valores. La potencia térmica producida de cada grupo t se encuentra entre su valor mínimo $\underline{GTR_t}$ y máximo $\overline{GTR_t}$.

$$\underline{GTR_t} \leq GTR_t \leq \overline{GTR_t}$$

La potencia hidráulica programada de cada grupo h puede tomar como valor máximo $\overline{GHP_h}$, valor dado por un programa de coordinación hidrotérmica de jerarquía superior.

$$0 \leq GHP_h \leq \overline{GHP_h}$$

La potencia hidráulica de emergencia de cada grupo h puede alcanzar como máximo el valor $\left(\overline{GHM_h} - \overline{GHP_h}\right)$, es decir, la potencia máxima del grupo menos su potencia programada.

$$0 \leq GHE_h \leq \left(\overline{GHM_h} - \overline{GHP_h}\right)$$

La potencia no suministrada como mucho será la demanda del nudo.

$$0 \leq PNS_n \leq D_n$$

El flujo por la línea está acotado en valor absoluto por $\overline{F_{i\to j}}$.

$$-\overline{F_{i\to j}} \leq F_{i\to j} \leq \overline{F_{i\to j}}$$

Además de la formulación anterior, se presenta otra donde se eliminan las variables de flujo y se sustituyen por la expresión de la segunda ley de Kirchhoff en función de los ángulos. La ecuación de balance entre generación y demanda tiene ahora esta formulación

$$\begin{gathered}\sum_{t\in n} GTR_t + \sum_{h\in n}(GHP_h + GHE_h) + PNS_n + \\ \sum_{i=1}^{I}(\theta_i - \theta_n)S_B / X_{i\to n} - \sum_{j=1}^{J}(\theta_n - \theta_j)S_B / X_{n\to j} = D_n\end{gathered}$$

y las cotas de las variables de flujo se transforman ahora en restricciones

$$\theta_i - \theta_j \leq \overline{F_{i\to j}}\frac{X_{i\to j}}{S_B}$$

$$\theta_i - \theta_j \geq -\overline{F_{i\to j}}\frac{X_{i\to j}}{S_B}$$

En la primera formulación se tienen más variables pero menos restricciones[6] que en la segunda y el número total de elementos no nulos de la matriz de restricciones será menor.

En el código escrito en GAMS se añade, además, una formulación del *flujo de cargas óptimo en corriente continua con pérdidas óhmicas*. Las pérdidas óhmicas de una línea se modelan con una expresión *no lineal* en función del coseno de la diferencia angular. Esto convierte el problema de optimización en no lineal.

$$L_{i\to j} = 2S_B \frac{r_{i\to j}}{r_{i\to j}^2 + X_{i\to j}^2}\left[1 - \cos(\theta_i - \theta_j)\right]$$

siendo $r_{i\to j}$ la resistencia de la línea que une los nudos i y j.

Éstas se incluyen como dos cargas adicionales iguales en los extremos de la línea. La primera ley de Kirchhoff tiene ahora esta expresión

$$\sum_{t\in n} GTR_t + \sum_{h\in n} (GHP_h + GHE_h) + PNS_n + \sum_{i=1}^{I} F_{i\to n} - \sum_{j=1}^{J} F_{n\to j} = D_n + L_n$$

siendo L_n las pérdidas en el nudo n

$$L_n = \left(\sum_{i=1}^{I} L_{i\to n} + \sum_{j=1}^{J} L_{n\to j} \right) / 2$$

```
$TITLE Flujo de cargas en corriente continua con y sin pérdidas

SETS
   ND          nudos
   GR          generadores
   TR(gr)      generadores térmicos
   HD(gr)      generadores hidráulicos
   NDGR(nd,gr) localización de generadores en nudos
   LN(nd,nd)   líneas

   CN características nudos       / dem, cpns /
   CG características generadores / coste, pmin, pmax, cshd, hdrpro, hdrmax /
   CL características líneas      / r, x, flmax /

ALIAS (nd, ni, nf) ;

SCALARS
   SBASE  potencia base [GW]                           / 0.1 /
   OPCPRD opción de modelado de las pérdidas (no 0 si 1) / 0 /

* definición de la estructura de datos sin incluir explícitamente éstos

PARAMETERS
   DATNUD(nd,cn)    datos de los nudos
   DATGEN(gr,cg)    datos de los generadores
   DATLIN(nd,nd,cl) datos de las líneas

* planteamiento matemático del problema

VARIABLES
   COSTE     función objetivo                        [M€]
   TT(nd)    ángulo de tensión en el nudo            [rad]
   FL(ni,nf) flujo de potencia                       [GW]

POSITIVE VARIABLES
   GTR(gr)   generación térmica                      [GW]
```

[6] Las cotas en las variables no cuentan como restricciones desde el punto de vista del tiempo de cálculo, ya que los algoritmos de optimización las tratan de forma específica.

```
   GHP(gr)   generación hidráulica programada         [GW]
   GHE(gr)   generación hidráulica de emergencia      [GW]
   PNS(nd)   potencia no suministrada                 [GW]
   PRDAS(nd) pérdidas de las líneas conectadas al nudo [GW]

EQUATIONS
   FO          costes de generación y de indisponibilidad [M€]
   KR1F(nd)    primera ley de Kirchhoff para cada nudo en función de flujos
   KR1A(nd)    primera ley de Kirchhoff para cada nudo en función de ángulos
   FLJ(ni,nf)  flujo en función de ángulos de tensión
   FLJP(ni,nf) diferencia angular máxima en cada línea en un   sentido
   FLJN(ni,nf) diferencia angular máxima en cada línea en otro sentido
   EPRDAS(nd)  pérdidas de las líneas conectadas al nudo ;

FO      .. COSTE =E= SUM[tr, DATGEN(tr,'coste') * GTR(tr)]
                  +  SUM[hd, DATGEN(hd,'cshd')  * GHE(hd)]
                  +  SUM[nd, DATNUD(nd,'cpns')  * PNS(nd)] ;

KR1F(nd) ..
   SUM[NDGR(nd,tr), GTR(tr)] + SUM[NDGR(nd,hd), GHP(hd) + GHE(hd)]
 + SUM[LN(ni,nd), FL(ni,nd)] - SUM[LN(nd,nf), FL(nd,nf)]
 + PNS(nd) =E= DATNUD(nd,'dem') + PRDAS(nd) $OPCPRD ;

KR1A(nd) ..
   SUM[NDGR(nd,tr), GTR(tr)]
 + SUM[NDGR(nd,hd), GHP(hd) + GHE(hd)]
 + SUM[LN(ni,nd), (TT(ni) - TT(nd)) / DATLIN(ni,nd,'x')] * SBASE
 - SUM[LN(nd,nf), (TT(nd) - TT(nf)) / DATLIN(nd,nf,'x')] * SBASE
 + PNS(nd) =E= DATNUD(nd,'dem') + PRDAS(nd) $OPCPRD ;

FLJ(LN(ni,nf)) .. FL(ni,nf) * DATLIN(ni,nf,'x') / SBASE =E= TT(ni) - TT(nf) ;

FLJP(LN(ni,nf)) ..
   TT(ni) - TT(nf) =L=   DATLIN(ni,nf,'flmax') * DATLIN(ni,nf,'x') / SBASE ;

FLJN(LN(ni,nf)) ..
   TT(ni) - TT(nf) =G= - DATLIN(ni,nf,'flmax') * DATLIN(ni,nf,'x') / SBASE ;

EPRDAS(nd) .. PRDAS(nd) =E=
          SBASE * SUM[LN(ni,nd), (1-cos(TT(ni) - TT(nd)))*
          DATLIN(ni,nd,'r')/(DATLIN(ni,nd,'r')**2+DATLIN(ni,nd,'x')**2)]
        + SBASE * SUM[LN(nd,nf), (1-cos(TT(nd) - TT(nf)))*
          DATLIN(nd,nf,'r')/(DATLIN(nd,nf,'r')**2+DATLIN(nd,nf,'x')**2)] ;

MODEL FC  / FO, KR1F, FLJ         / ;
MODEL FCA / FO, KR1A, FLJP, FLJN  / ;
MODEL FCP / FO, KR1F, FLJ, EPRDAS / ;

* caso de estudio
*** esta parte iría en ficheros independientes y se introduciría con $include

SETS
   ND     nudos                    / nudo-1 * nudo-9 /
   GR     generadores              / genr-1 * genr-9, genh-1 * genh-4 /
   NDGR(nd,gr) localización de generadores en nudos
  /
   nudo-1 . genr-1
   nudo-1 . genr-2
   nudo-1 . genr-3
   nudo-2 . genr-4
   nudo-2 . genr-5
   nudo-2 . genr-6
   nudo-3 . genr-7
   nudo-3 . genr-8
   nudo-3 . genr-9
   nudo-1 . genh-1
   nudo-3 . genh-2
   nudo-6 . genh-3
   nudo-8 . genh-4
  / ;

TABLE DATNUD(nd,cn) datos de los nudos
             dem   cpns
*             MW  €/kWh
   nudo-1      1   1500
   nudo-2    240   1500
   nudo-3     40   1500
   nudo-4    160   1500
   nudo-5    240   1500
   nudo-6     80   1500
   nudo-7    100   1500
   nudo-8     15   1500
   nudo-9    100   1500

TABLE DATGEN(gr,cg) datos de los generadores
            coste   pmin  pmax   cshd  hdrpro hdrmax
*           €/MWh    MW    MW    €/kWh   MW     MW
   genr-1    65       0    75
```


UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

```
   genr-2    70       30   125
   genr-3    75       10   100
   genr-4    59       10   100
   genr-5    67        0    50
   genr-6    74        0    50
   genr-7    61       10   100
   genr-8    76        0    50
   genr-9    80        0    50
   genh-1                            10   300    300
   genh-2                            10   150    160
   genh-3                            10   120    150
   genh-4                            10    90    100

TABLE DATLIN(ni,nf,cl) datos de las líneas
                    r       x      flmax
*                   p.u.    p.u.     MW
   nudo-1 . nudo-2  0.0777  0.2913  500
   nudo-1 . nudo-4  0.0544  0.2041  500
   nudo-2 . nudo-3  0.0424  0.1695  500
   nudo-2 . nudo-4  0.1     0.4     500
   nudo-2 . nudo-5  0.05    0.2     500
   nudo-2 . nudo-6  0.1     0.4     500
   nudo-3 . nudo-5  0.0248  0.099   500
   nudo-3 . nudo-8  0.1     0.4     500
   nudo-4 . nudo-6  0.15    0.6     500
   nudo-5 . nudo-6  0.05    0.2     500
   nudo-5 . nudo-8  0.1     0.4     500
   nudo-6 . nudo-7  0.15    0.6     500
   nudo-6 . nudo-9  0.05    0.2     500
   nudo-7 . nudo-9  0.05    0.2     500 ;

*** hasta aquí son ficheros independientes

* activación de generadores térmicos hidráulicos y líneas

TR(gr)    $DATGEN(gr,'pmax')   = YES ;
HD(gr)    $DATGEN(gr,'hdrpro') = YES ;
LN(ni,nf) $DATLIN(ni,nf,'x')   = YES ;

* escalación de datos de potencia a GW

DATNUD(nd,'dem')    = DATNUD(nd,'dem')    / 1e3 ;
DATGEN(tr,'pmin')   = DATGEN(tr,'pmin')   / 1e3 ;
DATGEN(tr,'pmax')   = DATGEN(tr,'pmax')   / 1e3 ;
DATGEN(hd,'hdrpro') = DATGEN(hd,'hdrpro') / 1e3 ;
DATGEN(hd,'hdrmax') = DATGEN(hd,'hdrmax') / 1e3 ;
DATLIN(ln,'flmax')  = DATLIN(ln,'flmax')  / 1e3 ;

* acotamiento de las variables (cotas físicas)

GTR.LO(tr) = DATGEN(tr,'pmin') ;
GTR.UP(tr) = DATGEN(tr,'pmax') ;

GHP.UP(hd) = DATGEN(hd,'hdrpro') ;
GHE.UP(hd) = DATGEN(hd,'hdrmax') - DATGEN(hd,'hdrpro') ;

PNS.UP(nd) = DATNUD(nd,'dem') ;

FL.LO(ln)  = - DATLIN(ln,'flmax') ;
FL.UP(ln)  =   DATLIN(ln,'flmax') ;

* cotas algorítmicas de los ángulos

TT.LO(nd) = - 1.5 ;
TT.UP(nd) =   1.5 ;

* nudo de referencia

TT.FX(nd) $(ORD(nd) EQ 1) = 0 ;

* opción sin pérdidas

OPCPRD = 0 ;

* flujo de cargas con variables de flujo

SOLVE FC  USING  LP MINIMIZING COSTE ;

* control sobre aprovechamiento de base previa

OPTION BRATIO = 1 ;

* flujo de cargas con variables de ángulos de tensión

SOLVE FCA USING  LP MINIMIZING COSTE ;

* opción con pérdidas
```

```
OPCPRD = 1 ;
* flujo de cargas con variables de flujo
SOLVE FCP USING NLP MINIMIZING COSTE ;
```

I.4.3. Elementos de estilo de programación$^{(DOCT)}$

> "En los últimos años se ha reconocido la programación de computadores como una disciplina cuyo dominio es básico y crucial para el éxito de muchos proyectos de ingeniería" Niklaus Wirth (1976).

I.4.3.1. Generales

La programación no es un castigo divino para los humanos, es *ciencia* y *arte*. Es ciencia en la medida que se pueden implantar modelos matemáticos complejos y que el pensamiento, la disciplina, la rigurosidad y la experimentación acompañan este desarrollo. El resultado es arte por la belleza, elegancia, sensación que puede transmitir un modelo y la profesionalidad de su creador.

Una forma de aprender a escribir con estilo y estructura ordenada es mediante la *lectura* de ejemplos ilustrativos o de código ajeno. Una manera de programar es por *refinamiento gradual* de los detalles. Es importante recordar que en el desarrollo de una aplicación el diablo se esconde en los detalles.

La potencia y concisión de los modelos escritos en un lenguaje de modelado hacen que el propio código forme parte de la documentación. De hecho la reutilización de modelos fue una de las causas que dieron origen a los lenguajes de modelado. La etapa de diseño del modelo cobra gran importancia para permitir posteriores ampliaciones. Por esta razón es importante el estilo en la programación, que incide en la *calidad* y *mantenibilidad*[7] del código desarrollado. Piénsese que el tiempo dedicado a mantenimiento y ampliación de un modelo es muy superior al inicial de desarrollo. El desarrollo y la depuración del modelo se debe hacer con una maqueta (caso ejemplo sencillo) para finalmente utilizar un problema real.

He aquí algunas *recomendaciones* para la escritura de un modelo que inciden en la *calidad* del desarrollo:

MODULARIDAD

Estructurar el modelo en diversos módulos con diferentes propósitos. Por ejemplo, la inclusión de los datos[8] y la escritura de resultados deben separarse en diferentes ficheros que son convenientemente insertados mediante la instrucción

[7] Se entiende por mantenibilidad la reutilización, reparación o modificación de un modelo.

[8] Se recomienda la introducción de los datos tal como son recogidos y entendidos por el usuario y se hacen en el modelo los cálculos auxiliares que sean necesarios.

`$include` en el módulo principal, que contiene la formulación del problema de optimización.

Utilización de las entidades (parámetros, escalares, etc.) con el mismo propósito y significado en las diferentes partes del código. Es decir, mantener la definición y uso de cada parámetro y escalar en todo el código para evitar la confusión del lector.

Comprobación de la pertenencia de un subconjunto a un conjunto de forma explícita en su definición, es decir, evitar el uso de índices comodín en vectores y matrices. Esta es una manera de validar y evitar errores en la introducción de los datos.

Escribir código para facilitar su lectura

Estas otras *recomendaciones* están orientadas al cuidado exquisito de la estética. Es el primer paso en el desarrollo profesional de un modelo. Son fundamentales, aunque aparentemente carecen de importancia para el desarrollador, pero se hacen imprescindibles para su *mantenimiento* y ampliación. El código debe ser limpio y claro para que pueda ser mantenido.

Mantener una coherencia en las reglas de escritura, de manera que se observe una norma sistemática en todo el código. Por ejemplo, endentación en las instrucciones repetitivas, sangría de tres espacios cada vez que se realiza una instrucción tipo `LOOP`, `IF`. Las palabras reservadas del lenguaje van en mayúsculas (`LOOP`, `IF`, `THEN`, `ELSE`, `SET`, `SCALAR`, `PARAMETER`, `TABLE`, etc.). La coma del final de instrucción va separada por un blanco. El signo de igualdad en las asignaciones se separa por espacios en blanco a ambos lados.

Establecer paralelismos o réplicas entre instrucciones consecutivas semejantes.

Las líneas de código deben tener una longitud aproximada de 100 columnas, no sobrepasando nunca las 110. Romper la instrucción en cuantas líneas sea necesario para cumplir esta recomendación.

Los comentarios deben ser suficientemente ilustrativos del contenido y estar bien localizados. Deben ayudar a documentar la naturaleza y origen de los datos

Se deben utilizar nombres largos y descriptivos para las entidades del modelo.

Los nombres y los índices de los parámetros, variables y ecuaciones han de ser acrónimos que representen su significado. Se recomienda una longitud de hasta 10 caracteres para los primeros y de hasta 2 para los segundos. Los comentarios explicativos pueden hacerse de hasta 80 caracteres.

Las definiciones de las entidades del modelo deben llevar las dimensiones físicas del problema.

Universidad Pontificia ICAI ICADE Comillas Madrid
Escuela Técnica Superior de Ingeniería
Departamento de Organización Industrial

Hacer un uso sistemático de mayúsculas y minúsculas con algún criterio predefinido, que debe ser coherente y mantenerse a lo largo de todo el programa. Por ejemplo, los nombres de los parámetros, variables y ecuaciones van en mayúsculas. Los nombres de sus índices van en minúsculas.

REFORMULACIÓN MANUAL DEL PROBLEMA

Un primer estadio en la formulación de un problema está en la elección de la propia formulación. A veces se pueden utilizar formulaciones semejantes con coste computacional muy diferente. Por ejemplo, para la representación de las pérdidas en un circuito eléctrico se puede utilizar una función no lineal o una poligonal aproximada. Habitualmente la formulación poligonal convexa requiere mucho menos tiempo.

Diferentes formulaciones matemáticamente equivalentes de un mismo problema de optimización pueden requerir tiempos de optimización muy distintos. Esta afirmación es especialmente relevante en problemas de programación lineal entera mixta y programación no lineal. Por esta razón, siempre es conveniente un ejercicio continuo de experimentación y reformulación de los problemas.

Veamos estas tres formulaciones de un problema NLP

$$
\begin{aligned}
&\min \sum_{i=1}^{n} \sum_{j=i+1}^{n} q_{ij} x_i x_j \\
&\sum_{j=1}^{n} x_j = 1 \\
&\sum_{j=1}^{n} r_j x_j = r_0
\end{aligned}
$$

$$
\begin{aligned}
&\min \sum_{i=1}^{n} x_i \sum_{j=i+1}^{n} q_{ij} x_j \\
&\sum_{j=1}^{n} x_j = 1 \\
&\sum_{j=1}^{n} r_j x_j = r_0
\end{aligned}
$$

$$
\begin{aligned}
&\min \sum_{i=1}^{n} x_i w_i \\
&w_i = \sum_{j=i+1}^{n} q_{ij} x_j \\
&\sum_{j=1}^{n} x_j = 1 \\
&\sum_{j=1}^{n} r_j x_j = r_0
\end{aligned}
$$

La ventaja de la formulación segunda con respecto a la primera es inmediata. La formulación 1 requiere para evaluar la función objetivo aproximadamente $2n\,{}^{n}\!/_{2}$ multiplicaciones. En la formulación 2 se necesitan $n + n\,{}^{n}\!/_{2}$ aproximadamente. La tercera formulación tiene esencialmente las mismas multiplicaciones pero aparecen en restricciones lineales. El número de restricciones aumenta sustancialmente pero todas son lineales y los métodos de manipulación de restricciones lineales son extremadamente eficientes. La formulación 3 resulta ser la más eficiente.

De hecho, algunos optimizadores realizan una etapa previa de preproceso[9] del problema antes de su resolución (parámetro de control `presolve` en el CPLEX). Como ejemplo, el impacto en el tamaño de dos problemas LP debido al preproceso realizado por el optimizador CPLEX 6.0 se muestra en la tabla 1.3.

	Caso 1			**Caso 2**		
	Restricc.	Variables	Elementos	Restricc.	Variables	Elementos
Sin preproceso	19047	27262	81215	48971	63935	187059
Con preproceso	15744	21982	51079	40794	56133	135361
Decremento	17%	19%	37%	17%	12%	28%

Tabla 1.3 Reducción de tamaños con la opción de preproceso.

En la formulación del problema la opción `profile` muestra el tiempo y memoria consumidos y el número de asignaciones realizadas o restricciones creadas.

Entre las principales consideraciones para mejorar la formulación de un problema de optimización se pueden citar:

[9] El desarrollo de las técnicas de preproceso y reformulación han originado avances muy importantes en la resolución de problemas MIP.

- Cálculo analítico del número de restricciones y variables
  Éste es una ayuda para ser consciente del tamaño esperable del problema y ver su dependencia en función de los elementos básicos que lo componen. El número real de restricciones para un caso concreto se muestra con la opción `profile`. Puede ser utilizado para detectar errores en la formulación. Por ejemplo, por excesivo número de ecuaciones al haber puesto dimensiones superfluas no controladas convenientemente con conjuntos dinámicos.
  Es conveniente también conocer la estructura de la matriz de restricciones, es decir, los bloques que la componen. Existe alguna utilidad, citada en el siguiente apartado, que lo permite hacer.

- No crear variables ni ecuaciones superfluas.
  Hay que tener cuidado con lo que se entiende por superfluas porque algunas condiciones redundantes pueden realmente llevar a obtener un modelo más fuerte en el contexto de programación entera. Sin embargo, el conocimiento de la naturaleza del problema permite introducir condiciones lógicas (mediante el uso del operador `$`) que eliminan algunas de ellas en la escritura de las ecuaciones o de las variables. Por ejemplo, en el caso de una red se suprimen variables o ecuaciones asociadas a líneas entre nudos no conectados entre sí. Aunque los optimizadores pueden detectar algunas de estas ecuaciones/variables superfluas, es más eficiente evitarlo mediante condiciones expresas.
  La opción `solprint=on` o la utilidad `gamschk` puede ayudar en la detección de las variables o ecuaciones superfluas (porque toman valor 0 o conocido bajo toda circunstancia en la solución).

- Reducción del número de restricciones y/o elementos de la matriz aun a costa de aumentar el número de variables.
  Una manera de reducir el número de ecuaciones o de variables es introduciendo expresamente el conocimiento que se tiene del problema real (casos particulares que pueden aparecer y sus implicaciones). Es más conveniente hacerlo manualmente a dejar que lo intente el preproceso del optimizador.
  Se puede hacer mediante sustitución, definición de nuevas variables, reformulación en general se debe intentar reducir el número de restricciones y/o de elementos de la matriz.
  Como norma general para la formulación de problemas lineales es conveniente saber que el tiempo necesario para su solución por el método simplex depende aproximadamente del cubo del número de restricciones, no siendo demasiado influyente el número de variables. En el método de punto

interior el tiempo de ejecución depende principalmente del número de elementos (densidad) de la matriz de restricciones.

- Escalación tanto de variables como de coeficientes y valores de restricciones a números alrededor de 1
  Esto mejora el comportamiento numérico en la resolución del problema y reduce el tiempo de ejecución. La escalación resulta muy conveniente en problemas LP de gran tamaño pero es imprescindible en problemas NLP. Implícitamente los valores por omisión de los parámetros de control de los optimizadores están fijados suponiendo que el problema está bien escalado alrededor de 1. Con una escalación razonable puede haber como mucho 6 órdenes de magnitud de diferencia (por ejemplo, coeficientes de las variables entre 0.001 y 1000). La utilidad `gamschk` es una herramienta muy útil para observar los intervalos de variación de los coeficientes de las variables en las restricciones y de las costas de éstas para detectar potenciales problemas de escalado.
  La escalación se puede hacer *manualmente* –expresando las variables, parámetros y ecuaciones en unidades naturales con sentido físico para el problema– o *automáticamente* mediante las opciones disponibles en el lenguaje (`nombre_modelo.scaleopt=1`) o en el optimizador (`scale`). La escalación manual requiere más cuidado y control pero es preferible porque conserva la naturaleza física del problema dentro del código y es igual de efectiva que la automática.
  En cualquier caso hay que tener cuidado al realizar el escalado, especialmente en problemas no lineales, donde pueden existir efectos no lineales que invaliden la nueva formulación.

- Acotamiento de las variables.
  Las cotas en las variables no cuentan como restricciones desde el punto de vista del tiempo de cálculo, ya que los algoritmos de optimización las tratan de forma específica. Las cotas pueden tener sentido *físico* (y, por tanto, forman parte de la naturaleza del problema) o ser *algorítmicas* (es decir, cotas superfluas que nunca deben ser activas en la solución óptima pero que reducen el tiempo de optimización).
  El preproceso generalmente incluye procedimientos para el fortalecimiento de las cotas de las variables (reducción de las cotas superiores y aumento de las inferiores).

TRATAMIENTO EXPLÍCITO DE CONJUNTOS ORDENADOS SOSn

Los conjuntos ordenados (*Special Ordered Sets SOS*) son conjuntos de variables que cumplen las siguientes condiciones:

Como mucho n elementos del conjunto toman valores diferentes de 0. El resto de elementos ha de ser 0

Si hay n elementos que son diferentes de 0 deben ser contiguos

Los conjuntos ordenados tienen un tratamiento especial en la optimización, por lo que su definición puede mejorar mucho el tiempo requerido para la resolución.

Selección del optimizador y tipo algoritmo de optimización

Un lenguaje de modelado permite utilizar diferentes optimizadores para la resolución de un mismo problema de optimización. Esta característica representa una gran ventaja por la flexibilidad que aporta en la selección del optimizador más adecuado a las características del problema.

En Internet (www-c.mcs.anl.gov/otc/guide/faq/linear-programming.html) y en la revista *OR/MS Today*, Fourer (1999), se pueden encontrar opiniones y revisiones del software disponible para la resolución de problemas de optimización de todo tipo. Entre los optimizadores a los que se ha tenido acceso destacan CPLEX y OSL para LP, MINOS y CONOPT para NLP y MILES y PATH para MCP.

El mejor método para un problema concreto depende de las características del problema, de los detalles de implantación del método simplex o del punto interior y del ordenador utilizado. Por esta razón los paquetes comerciales de LP importantes incluyen métodos de punto interior (habitualmente primal-dual predictivo-correctivo), métodos simplex (en su versión primal y dual) y de resolución de flujos de redes (simplex de red). Se debe utilizar el método de optimización (punto interior o barrera, simplex primal o simplex dual) más adecuado al tipo o tamaño del problema.

Como recomendación general, para problemas de tamaño medio (hasta aproximadamente de 10000 x 10000) el método más adecuado es el simplex y para problemas de gran tamaño (desde 10000 x 10000 hasta 100000 x 100000) el mejor método es el de punto interior (especialmente en problemas degenerados). Para problemas de tamaño superior se requiere el uso de técnicas de optimización específicas (como, por ejemplo, las de descomposición entre otras [Ramos, 1996]). La selección de un método u otro se debe realizar principalmente en función del tamaño del problema. [Bixby, 2000] es un artículo práctico reciente donde se presentan algunas comparaciones entre métodos de solución del optimizador CPLEX tanto para problemas lineales como enteros mixtos.

El método simplex también resulta adecuado en la realización de análisis de sensibilidad, es decir, cuando se trata de resolver problemas similares disponiendo de una solución próxima y una base previa, como sucede en el método de ramificación y acotamiento para resolver problemas lineales enteros.

A continuación se presenta la tabla 1.4 de comparación entre varios optimizadores y métodos de optimización. En la tabla 1.5 se muestra la diferencia de funcionamiento entre las opciones de preproceso de dos optimizadores.

		Caso 1			**Caso 2**		
		Tiempo	Índice	Iter.	Tiempo	Índice	Iter.
CPLEX 6.0	Punto interior	41.8	1.0	32	237.3	1.0	35
	Simplex dual	99.8	1.4	12692	1812.6	6.6	48695
	Simplex primal	156.2	3.7	21622	1217.5	5.1	50280
MINOS 5.3	Simplex primal	1863.6	44.6	23927	–	–	–
OSL 2.1	Punto interior	163.9	3.9	10798	774.4	3.3	19524
	Simplex primal	530.9	12.7	12685	7426.6	31.3	62019

Tabla 1.4 Comparación entre diferentes optimizadores en problemas LP.

	Caso 1			**Caso 2**		
	Restricc.	Variables	Elementos	Restricc.	Variables	Elementos
Sin prep	19047	27847	82295	49715	64679	189477
Prep CPLEX	–14,8%	–19,3%	–36,2%	–17,9%	–13,2%	–28,6%
Prep OSL	–4,9%	0,0%	–2,4%	–15,6%	0,0%	–9,1%

Tabla 1.5 Comparación entre diferentes preprocesos.

Las diferencias en tiempo de resolución que pueden encontrarse entre métodos de optimización o entre implantaciones de un mismo método llegan a ser significativas (de hasta 45 veces para una comparación entre CPLEX 6.0 utilizando un método de punto interior y MINOS 5.3 utilizando el método simplex para un problema de 19000 restricciones, 28000 variables y 82000 elementos no nulos). Para un mismo método de optimización se han encontrado diferencias de hasta 3 veces entre implantaciones.

Utilización de últimas versiones

En general, las últimas versiones aportan mejoras de tiempo o funcionalidad con respecto a versiones previas.

En particular, una característica muy atractiva de los lenguajes de modelado es la posibilidad de actualizar la versión del optimizador o cambiar de optimizador sin necesidad de realizar modificaciones en el código del modelo. Ser consciente de ello y aprovecharlo forma parte de un uso avanzado del lenguaje.

Ajuste de parámetros de control del optimizador

Habitualmente los parámetros de control de un optimizador toman unos valores por omisión generalmente adecuados para un problema estándar de optimización. Sin embargo, cuando se trata de problemas difíciles, como pueden ser los LP de muy gran tamaño o los NLP o MIP, son convenientes pruebas específicas de ajuste con algunos parámetros. En particular, algunos relacionados con la eficiencia y estabilidad numérica del algoritmo.

Los parámetros son propios de cada optimizador y también pueden serlo de cada método de optimización. Por mencionar algunos que pueden ser importantes en MINOS (`linesearch tolerance`, `penalty`, `major iterations`, `minor iterations, factorization frequency`) y en CPLEX (`epopt, eprhs, epmrk`).

Como consejo para evitar errores o confusiones es conveniente la creación de los ficheros de parámetros de control del optimizador dentro del código en lugar de editarlos manualmente.

Uso de soluciones iniciales y/o bases previas

El uso de puntos iniciales es particularmente importante en el caso de problemas no lineales, donde se debe ejecutar un problema lineal cuya solución resulte cercana a la previsible solución del problema no lineal.

Cuando se trata de ejecuciones sucesivas es conveniente, desde el punto de vista de cálculo, aprovechar en el algoritmo del simplex las bases de soluciones previas del mismo problema u otro similar que haya sido resuelto previamente. La base contiene la información relativa a las variables primales y duales del problema. El aprovechamiento se controla con la opción `bratio` que marca un criterio de aceptación o rechazo de la misma.

Como ejemplo del impacto en el tiempo de optimización del aprovechamiento de la base, un problema LP de 8000 restricciones, 10000 variables y 30000 elementos requiere 10.3, 4.4, 4.7 y 2.6 segundos en sucesivas resoluciones.

A pesar de ello esta ventaja puede no ser suficiente para ciertos tamaños como para superar al método de punto interior. Por ejemplo, aproximadamente a partir de 20000 restricciones por 20000 variables el método de punto interior resulta más competitivo que el simplex aun comenzando éste con una base previa de un problema anterior.

Universidad Pontificia ICAI ICADE Comillas Madrid
Escuela Técnica Superior de Ingeniería
Departamento de Organización Industrial

DETECCIÓN DE INFACTIBILIDADES

Un método muy sencillo aunque laborioso y que puede producir problemas de consistencia y de dimensiones, es introducir variables de holgura en cada restricción y penalizarlas en la función objetivo. Este procedimiento hace las *restricciones elásticas.*

Alternativamente algunos optimizadores tienen un parámetro que detecta el núcleo menor de restricciones infactibles de un problema (parámetro *Irreducible Infeasible Subsets* `iis`) y, por consiguiente, ayudan a localizar su posible causa. Una vez conocidas el desarrollador debe modificar o eliminar alguna del conjunto para que el problema se haga factible. Un artículo reciente de John W. Chinneck sobre algoritmos para encontrar este conjunto mínimo es [Guieu, 1999]

ANÁLISIS DE SENSIBILIDAD

Proporciona información adicional sobre la solución de un problema de optimización lineal.

Algunos optimizadores permiten realizar directamente un análisis de sensibilidad a cambios en los coeficientes de la función objetivo que no producen una alteración de la base óptima o a cotas de las restricciones que no producen pérdida de factibilidad (parámetros `objrng, rhsrng`).

I.4.3.2. Específicos de GAMS

Existe un informe técnico que incluye un conjunto general de recomendaciones para el desarrollo y la reparación de modelos escritos en GAMS, [McCarl, 1998].

USO AVANZADO PARA OPTIMIZACIÓN

En este apartado se desarrollan algunas consideraciones que permiten la implantación avanzada de modelos escritos en GAMS. Estas recomendaciones recogen la experiencia práctica adquirida en años de uso del lenguaje GAMS, principalmente en modelos de optimización lineal y estocástica, de ahí el valor que tienen a la hora de implantar problemas de optimización de gran tamaño.

Una implantación ingenua (de novato, "no profesional") de un problema de optimización puede llevar aparejado un consumo excesivo de recursos computacionales. Los más relevantes son el *tiempo de ejecución* y/o la *memoria.* El tiempo de ejecución de un modelo es especialmente crítico en aplicaciones de muy gran tamaño (e.g., a partir de 100000 restricciones por 100000 variables en el caso lineal) o, sobre todo, en el caso de resolución iterativa de numerosos (e.g., más de 100) problemas de optimización de mediano tamaño (como sucede en los métodos de descomposición o en la simulación de Monte Carlo). Los requisitos de memoria pueden ser limitativos en el caso de problemas de muy gran tamaño. Algunas de las acciones que permiten reducir tiempo también disminuyen los requerimientos de memoria.

Los métodos de descomposición no son más que técnicas matemáticas que permiten resolver problemas gigantescos (por ejemplo, de más de 1 millón de restricciones y variables) con una estructura especial, que ni siquiera se pueden formular explícitamente, mediante la solución iterativa de problemas de menor tamaño. Como ejemplo se puede mencionar el caso de un problema de coordinación hidrotérmica en un sistema eléctrico cuya resolución se efectúa mediante descomposición anidada estocástica, ver Jacobs (1995).

Los valores numéricos que se aportan para contrastar el impacto de algunas recomendaciones deben tomarse como indicaciones relativas de las mejoras esperables nunca como seguros. Piénsese que cualquier mejora está asociada a un tipo de problemas, no necesariamente es generalizable para todos.

El tiempo de ejecución de un modelo escrito en GAMS se puede descomponer en estos tres tipos principales[10]:

- tiempo de *creación*
  formulación del problema de optimización específico, es decir, creación de las variables y de las restricciones.

- tiempo de *interfaz*
  escritura del problema de optimización en disco para su lectura por el optimizador y viceversa.

- tiempo de optimización
  resolución del problema de optimización por parte del optimizador.

Además de éstos hay que añadir el tiempo de compilación del modelo. Sin embargo, este tiempo se da únicamente una vez al comienzo y habitualmente es despreciable frente al resto.

El valor e importancia de cada uno de estos tiempos se puede conocer con las opciones `stepsum`, que resume el consumo de tiempo entre llamadas al optimizador, y `profile`, que informa sobre el consumo de tiempo y memoria en cada instrucción del código. Antes de iniciar las acciones de mejora es necesario

[10] Esta clasificación del tiempo de ejecución de un modelo en tres componentes es relevante para los modelos escritos en GAMS. Quizá con otros lenguajes de modelado alguno de estos tiempos puede ser despreciable.

realizar un análisis de los consumos de tiempo del modelo y de cómo se reparten.

La relación entre ellos depende de las diversas características del problema: tamaño y estructura de la matriz de restricciones, número de optimizaciones, variación de los parámetros en sucesivas optimizaciones, como más importantes. Las direcciones de mejora que se presentan a continuación tienen una orientación o bien informática o bien matemática, aunque indudablemente en el tiempo de ejecución resultante influyen ambas. Las primeras están basadas en el uso del lenguaje GAMS. Las segundas modifican el problema o su resolución. La efectividad de cada mejora dependerá de las características del problema de optimización. Se sugieren algunos criterios heurísticos que permiten utilizarlas adaptándose al caso concreto tal como se menciona posteriormente. Éstos han de tomarse con cautela. En ningún momento se les quiere dar a estos criterios más que un valor indicativo, ajeno a cualquier tipo de generalización.

USO DE UN DISCO VIRTUAL

El tiempo de interfaz se debe a la escritura de los ficheros[11] de comunicación con el optimizador. El tiempo del proceso de escritura depende del hardware del equipo utilizado, en particular, del manejo y tamaño de la memoria caché[12] y del tiempo de escritura en el disco. A su vez, el disco puede ser local (localizado en la máquina que ejecuta el modelo) o remoto (conectado a través de una red de área local).

La minimización del tiempo de interfaz exige un conocimiento detallado de los recursos de hardware utilizables. Por ejemplo, uso de discos locales en lugar de remotos, tamaños elevados memorias caché, etc. En cualquier caso, una solución sencilla en un PC es la creación y uso de discos virtuales localizados en memoria RAM (utilidad RAMDISK), siempre más rápida que los discos magnéticos. Un tamaño de 16 MB de disco RAM es suficiente para estos casos de estudio.

La reducción en tiempo esperable depende de la relación entre el tiempo de acceso al disco frente al acceso a la RAM. Por ejemplo, para el anterior PC se ha obtenido una reducción en tiempo de un 20 % del tiempo de interfaz.

CAMBIOS EN INSTRUCCIONES DE ASIGNACIÓN

[11] En GAMS la comunicación entre el lenguaje y los optimizadores se hace mediante ficheros. En otros lenguajes esta relación se establece a través de variables localizadas en la memoria principal.

[12] La memoria caché mantiene una copia de la última información leída o escrita en disco, de manera que pueden evitarse accesos a disco cuyo tiempo de acceso es superior.

GAMS es un lenguaje "peligroso" desde el punto de vista de consumo de tiempo por su naturaleza intrínseca, ser muy compacto y de alto nivel. Es relativamente fácil escribir instrucciones sencillas que involucren entidades con múltiples dimensiones que consuman un tiempo y/o memoria elevada. La opción `profile` permite conocer el consumo de tiempo y memoria y el número de asignaciones realizadas en cada instrucción.

Como recomendaciones específicas para no desperdiciar tiempo:

- El orden de colocación de los índices/dimensiones debe ser consistente para todos los parámetros, ecuaciones y variables.
- Se debe pensar desde el punto de vista de una ordenación natural de todos los índices para el conjunto del problema. Esta ordenación influye también en la formulación de las ecuaciones.
- El orden de colocación de los índices en las instrucciones reiterativas (sumatorios, productorios, bucles) debe ser el mismo que en los parámetros, variables o ecuaciones que se estén manipulando.
- Se debe hacer un uso extensivo de la exclusión mediante condiciones en asignaciones (uso del operador `$`) controladas preferentemente mediante conjuntos dinámicos (mejor que con valores de parámetros), es decir, activar sólo las variables y restricciones necesarias. En el ejemplo previo del flujo de cargas se observa que se consideran sólo las líneas eléctricas que existen y no cualquier posible conexión entre dos nudos.

TAMAÑOS MÁXIMOS ALCANZADOS

Como referencia final una indicación sobre el tamaño de los problemas que se están resolviendo y a los que se ha llegado aplicando estas recomendaciones. Se han podido resolver sin dificultad problemas de 150000 restricciones por 227000 variables con 566000 elementos no nulos en la matriz de restricciones en 300 segundos en un PC con procesador Pentium III Mobile a 1 GHz.

ALGUNAS INSTRUCCIONES ADICIONALES

A continuación se presenta una miscelánea de instrucciones de GAMS que no están suficientemente documentadas en el manual de usuario o se han utilizado con otra perspectiva.

Máxima supresión de información de salida

La supresión de la información de salida en el `nombre_fichero.lst` se consigue con las siguientes opciones.

```
$OFFSYMLIST, OFFSYMXREF, OFFUELLIST, OFFUELXREF
OPTION LIMROW=0, LIMCOL=0, SOLPRINT=OFF, SYSOUT=OFF
nombre_modelo.SOLPRINT=2 ;
```

y escribiendo en la invocación de GAMS

```
gams nombre_modelo.gms suppress 1
```

Además, también se puede suprimir la información en pantalla que produce el optimizador con los consiguientes parámetros (por ejemplo, para CPLEX `simdisplay 0` `bardisplay 0` y `mipdisplay 0`).

```
gams nombre_modelo.gms ll 0 lo 0
```

Presentación de información por pantalla

Las siguientes instrucciones permiten la definición de la pantalla para posteriormente escribir información en ella. En el uso real hay que tener en cuenta posibles buffers que hacen que esta información no se muestre inmediatamente.

```
$SET CONSOLA
$IF %system.filesys% == UNIX $SET CONSOLA /dev/tty
$IF %system.filesys% == MS95 $SET CONSOLA CON
$IF %system.filesys% == MSNT $SET CONSOLA CON
$IF "%consola%." == "." ABORT "Fichero no reconocido" ;
FILE PANTALLA / '%consola%' / ;
```

Opciones SAVE y RESTART.

Permiten la segregación de una parte de código para su depuración evitando su ejecución completa.

También permiten la creación y distribución de una versión ejecutable, es decir, aquélla donde el usuario final no tiene acceso a la definición del problema de optimización. Para ello el código se separa en dos partes. La primera contiene las declaraciones y definiciones de variables y ecuaciones y la segunda la inclusión de ficheros de datos y resolución del problema.

Esta opción también puede utilizarse para paralelizar bucles[13]. La parte común se genera con la instrucción `SAVE` en el procesador principal. Después, la parte paralelizada (cada ciclo del bucle) se ejecuta con un `RESTART` en cada procesador independiente y asíncronamente. Una vez terminadas todas las ejecuciones se integran los resultados obtenidos.

Recorrido inverso de un índice

La siguiente instrucción permite recorrer el parámetro `PP` en sentido inverso

```
PP(i+[card(i)-2*ord(i)+1])
```

empezando por `card(i), card(i)-1, .... 2, 1`.

Eliminación de las variables fijas

[13] El IIT ha desarrollado una utilidad que permite la ejecución asíncrona de scripts de UNIX que pueden ser utilizados para la paralelización de aplicaciones en GAMS.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Se trata de aquellas variables cuyas cotas inferior y superior coinciden y el mismo lenguaje las convierte en parámetros, de manera que no son consideradas como tal por el optimizador. Por consiguiente, no se puede obtener información dual sobre ellas.

```
nombre_modelo.HOLDFIXED = 1 ;
```

Otras características no documentadas en el manual

```
SAMEAS(elemento_de_set1.elemento_de_set2)
```

Función que devuelve verdadero si las cadenas de caracteres de los nombres de los elementos de set son iguales o falso en caso contrario.

```
DIAG(elemento_de_set1.elemento_de_set2)
```

Devuelve 1 en el caso de igualdad y 0 en caso contrario.

```
$CALL
```

Llamada externa que se ejecuta en el momento de la compilación.

```
$EXECUTE
```

Llamada externa a una aplicación que devuelve el control a GAMS cuando ésta finaliza.

```
option SOLSLACK = 1
```

Presenta el valor de las variables de holgura de las restricciones en lugar del valor de la restricción como tal.

Utilidades complementarias

Existen algunas aplicaciones conectadas con GAMS que añaden funcionalidad, facilitan la interfaz con el lenguaje o la presentación de resultados. Las nueve primeras se apoyan en la instrucción `$libinclude`. Entre ellas cabe citar:

Aplicaciones de análisis, depuración y mejora de modelos

```
gams-f
```

Permite definir funciones que posteriormente son sustituidas mediante un preproceso en las definiciones de parámetros o ecuaciones.

```
gamschk
```

Aplicación que permite examinar empíricamente modelos escritos en GAMS para detectar posibles errores.

```
gamsbas
```

Aplicación que permite guardar la información relativa a la base que posteriormente puede ser utilizada para modelos subsiguientes.

Exportación del modelo a otros sistemas

GAMS permite la exportación del problema de optimización en formato MPS o LP que pueden ser leídos por numerosos optimizadores. En el fichero MPS se define la matriz de restricciones del problema, vista por columnas, y las cotas de las variables. Los nombres de las restricciones y de las variables están limitados a 8 caracteres y el formato de los datos es fijo por columnas. En el fichero LP se define el problema de forma más natural al poner directamente las expresiones de las ecuaciones pero utilizando nombres de las variables no indexados.

Interfaz con una hoja de cálculo

`ssimport.gms`
Lee datos de una hoja de cálculo durante la compilación.
`ssdump.gms`
Escribe datos y etiquetas en una hoja de cálculo. Admite tamaños dinámicos.
`ssexport.gms`
Escribe datos en una hoja de cálculo. Los intervalos de escritura son fijos.

Interfaz de presentación de resultados

Algunas de estas utilidades se pueden usar para realizar interfaces más sencillas con bases de datos.
`gams2tbl.gms`
Facilita la escritura automática de informes en forma de tablas.
`gams2txt.gms`
Escribe en un fichero los valores de parámetros, variables, ecuaciones o sets.
`gams2prm.gms`
Escribe en un fichero la declaración y valores de parámetros, variables. ecuaciones o sets.
`gams2zip.gms`
Escribe en un fichero comprimido la declaración y valores de parámetros, variables, ecuaciones o sets.
`zip2gams.gms`
Recupera de un fichero comprimido la declaración y valores de parámetros, variables, ecuaciones o sets.
`gnuplot.gms`
Permite la creación de gráficos con GNUPLOT que representen valores de parámetros.

Interfaz con MATLAB

`matout.gms` y `gams.dll`
Permite el uso desde MATLAB de las capacidades de optimización que proporciona GAMS y la facilidad de visualización de MATLAB de los resultados de una optimización.

I.4.3.3. Referencias

Bixby, R.E., Fenelon, M., Gu, Z., Rothberg, E. and Wunderling, R. (2000) *MIP: Theory and Practice - Closing the Gap*. Technical Report.

Guieu, O. and Chinneck, J.W. (1999) "Analyzing Infeasible Mixed-Integer and Integer Linear Programs", INFORMS Journal on Computing, vol. 11, no. 1, pp. 63-77.

Fourer. R. (1999) "Linear Programming" *OR/MS Today* pp. 64-71. August.

McCarl. B. A. (1998) *So Your GAMS Model Didn't Work Right. A Guide to Model Repair*. Technical Report.

Jacobs. J., Freeman. G., Grygier. J., Morton. D., Schultz. G., Staschus. K. and Stedinger. J. (1995) "SOCRATES: A system for scheduling hydroelectric generation under uncertainty" *Annals of Operations Research* 59. pp. 99-133.

Ramos. A. et al. (1996) "Computational Experience with Optimization for a Bulk Production Cost Model" *12th PSCC*. Dresden, Germany.

I.5. Optimización lineal

I.5.1. Introducción

La programación lineal (LP) es la aplicación clásica por excelencia y la más desarrollada de la optimización. En cada momento se están ejecutando miles de aplicaciones basadas en LP. Los modelos de LP son más utilizados que todos los otros tipos de optimización juntos. Abarcan cualquier tipo de actividad humana como economía, finanzas, marketing, organización de la producción, planificación de la operación, selección de procesos, asignación de tareas, etc. Su importancia se debe a la existencia de técnicas potentes, estables y robustas para encontrar el óptimo que han permitido su uso en multitud de aplicaciones.

Sea el siguiente problema de programación lineal genérico en *forma estándar*:

$$\begin{aligned}
&\min_{x_j} z = \sum_j c_j x_j \\
&\sum_j a_{ij} x_j = b_i \\
&x_j \geq 0
\end{aligned} \tag{1.14}$$

Las variables reciben también el nombre de *actividades*, *decisiones* o *columnas*. Las restricciones se denominan también *recursos* o *filas*.

La programación lineal se sustenta en las siguientes hipótesis sobre las ecuaciones y variables que conforman el problema matemático:

- *Proporcionalidad*
  La contribución de cada actividad (variable) x_j al valor de la función objetivo z es proporcional al nivel de la actividad, $c_j x_j$. La contribución de cada actividad x_j al valor de la parte izquierda de cada restricción es proporcional al nivel de la actividad, $a_{ij} x_j$.

- *Aditividad*
  Cada ecuación en un problema LP es la suma de las contribuciones individuales de las respectivas actividades.

- *Divisibilidad*
  Cualquier variable puede tomar cualquier valor, no necesariamente entero, que satisfaga las restricciones incluyendo las de no negatividad.

- *Certidumbre*
  Los parámetros (constantes) de un problema LP se suponen conocidos con certidumbre (pueden ser estimaciones, pero éstas se tratan como valores conocidos).

I.5.2. Solución gráfica

Supongamos el siguiente problema de programación lineal en un espacio bidimensional, que gráficamente queda representado en la figura adjunta.

$$
\begin{array}{llll}
\max z = 3x_1 + 5x_2 & & & \\
x_1 & & \leq 4 & \\
 & 2x_2 & \leq 12 & \\
3x_1 & +2x_2 & \leq 18 & \\
x_1, & x_2 & \geq 0 &
\end{array}
$$

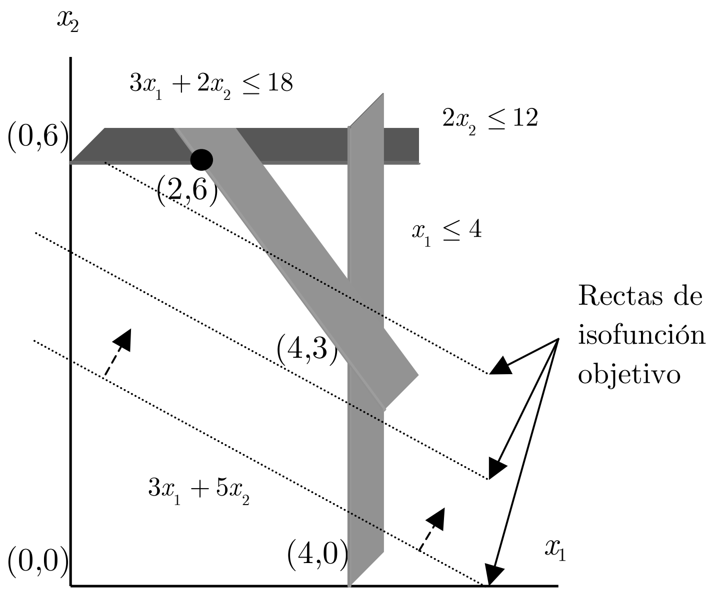


Figura 2.1 Solución gráfica de un problema LP.

La *región factible* está encerrada por las tres rectas que constituyen las restricciones más los dos ejes, que son las rectas correspondientes a la no negatividad de las variables. La solución óptima se halla gráficamente desplazando paralela a sí misma la recta de isofunción objetivo en el sentido de maximización (dirección del gradiente de la función objetivo, c) hasta que se alcanza el último punto de la región factible. Ésta corresponde al punto $(x_1, x_2) = (2, 6)$ con un valor de la función objetivo $\hat{z} = 36$.

Otras posibilidades que pueden aparecer al resolver un problema de programación lineal son las siguientes: solución con múltiples óptimos y solución no acotada. Por otra parte, si la región factible es el conjunto vacío, el problema es infactible.

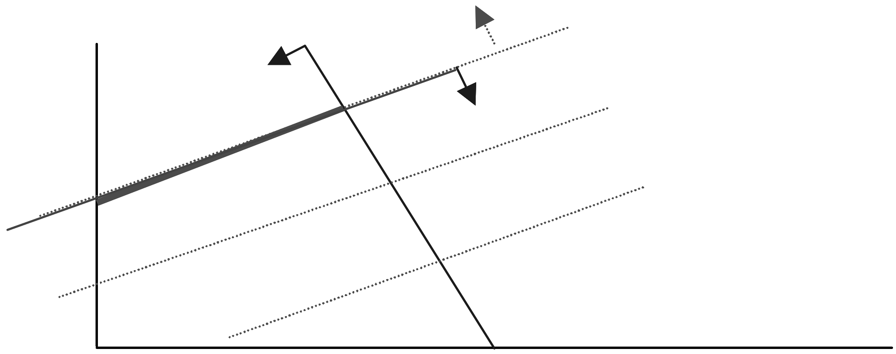


Figura 2.2 Solución gráfica de un problema LP con múltiples óptimos.

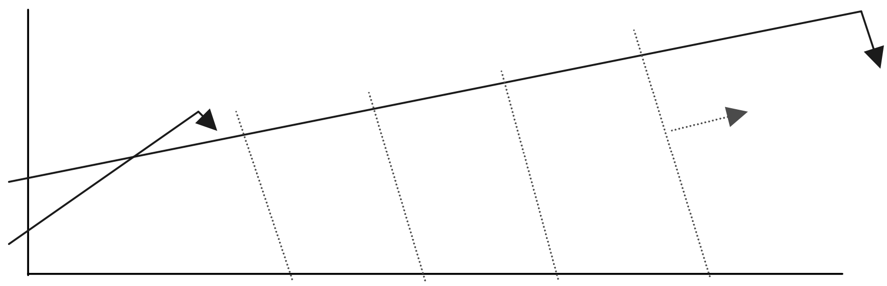


Figura 2.3 Solución gráfica de un problema LP con solución no acotada.

I.5.3. Geometría de la programación lineal

Hay una serie de conceptos geométricos que hay que definir antes de analizar la geometría de un problema de programación lineal.

Un *hiperplano* está definido por el conjunto de puntos x_j que cumplen una ecuación cualquiera del problema LP, $\sum_j a_{ij}x_j = b_i$. Un hiperplano en dos dimensiones es una recta. El hiperplano divide el espacio en dos semiespacios definidos por $\sum_j a_{ij}x_j \le b_i$ y $\sum_j a_{ij}x_j \ge b_i$.

Un *poliedro* es la región definida por la intersección de un conjunto finito de semiespacios. La región factible de un problema de programación lineal es un poliedro. Un *politopo* es un poliedro no vacío y acotado (por ejemplo, las regiones factibles de las figuras 2.1 y 2.2).

Un *vértice* o *punto extremo* de un poliedro se define como un punto que no puede ponerse como combinación lineal convexa de dos puntos diferentes del conjunto. Se determina por la intersección de n hiperplanos. En dos dimensiones, son dos rectas.

El *contorno* de una región factible contiene las soluciones factibles que están en uno o más hiperplanos. *Arista* es el segmento obtenido por solución de $n-1$ hiperplanos, siendo sus extremos soluciones factibles vértices.

Teorema de existencia de puntos extremos

Sea $S = \{x \,/\, Ax = b, x \geq 0\} \neq \varnothing$, $A \in \mathbb{R}^{m\times n}$, $r(A) = m$. Entonces S tiene al menos un punto extremo.

Teorema de representación de un politopo

Sea $S = \{x \,/\, Ax = b, x \geq 0\} \neq \varnothing$ y acotado, $A \in \mathbb{R}^{m\times n}$, $r(A) = m$. Sean $x_1, \ldots, x_k$ los puntos extremos de S. Entonces $x \in S$ si y sólo si x se puede expresar como una combinación lineal convexa de los puntos extremos del conjunto, es decir, $\exists \lambda_1, \ldots, \lambda_k \geq 0$, tales que $\sum_i \lambda_i = 1$ y $x = \sum_i \lambda_i x_i$. Cualquier punto de un politopo se puede expresar como combinación lineal convexa de los vértices del mismo.

Una aplicación inmediata del resultado anterior es su aplicación al problema de programación lineal, ya que es fácilmente demostrable que buscar en una región factible acotada el punto que optimiza una determinada función objetivo es equivalente a buscar el punto extremo que lo hace. Es decir, entonces la búsqueda del óptimo se limita a optimizar en un número finito de puntos: los vértices o puntos extremos.

Sin embargo, este resultado no es de una aplicación inmediata ya que, por un lado, el número total de vértices o puntos extremos posibles del politopo (no todos ellos son necesariamente factibles) puede ser muy grande (es de $C_{n,m} = \begin{pmatrix} n \\ m \end{pmatrix} = \dfrac{n!}{m!(n-m)!}$ y crece exponencialmente con el tamaño del problema) y, por otro lado, hay que calcular cuáles son los puntos extremos, que si bien están caracterizados y existe el procedimiento para hacerlo puede ser largo y costoso. Además en el caso de que no haya ningún punto factible, no sería detectado hasta la prueba de todos los puntos candidatos a ser extremos ($C_{n,m}$ puntos) y, en el caso de que la solución sea no acotada, con tal procedimiento no se detectaría esta situación.

Por lo tanto, se impone la búsqueda del óptimo de una forma dirigida, es decir, con ciertos criterios que favorezcan esta búsqueda y que además permitan la detección de situaciones como las descritas en el apartado anterior. Estos criterios se plasman en un método o algoritmo de tipo algebraico que se describe en la siguiente sección.

UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

I.5.4. Método simplex

El método simplex, que debe su origen a George B. Dantzig en 1947, es un procedimiento algebraico iterativo para resolver un problema de programación lineal que tiene conceptos geométricos subyacentes. Recorre exclusivamente los vértices (o puntos extremos) de la región factible. La idea geométrica subyacente se describe a continuación. Partiendo de un vértice inicial (luego se describirá el procedimiento para obtenerlo), se va moviendo de un vértice a otro vértice adyacente mejorando la función objetivo. Se selecciona la arista con mayor tasa de mejora de la función objetivo para alcanzar el vértice adyacente, hasta alcanzar un punto en el que no existe ninguna arista incidente por la que mejorar. En el ejemplo anterior, partiendo del vértice (0,0) con función objetivo $\hat{z} = 0$, el método simplex llegaría al óptimo (2,6) con función objetivo $\hat{z} = 36$ después de pasar por el vértice (0,6) con función objetivo $\hat{z} = 30$, tal como se aprecia en la figura. Se comprueba la condición de óptimo porque en dicho vértice las tasas de mejora de la función objetivo hacia vértices adyacentes son negativas.

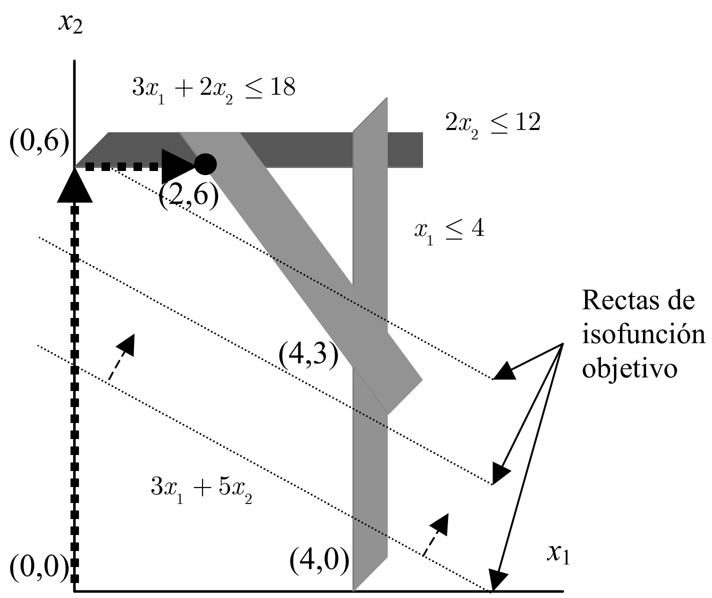


Figura 2.4 Representación gráfica del método simplex.

Un problema de optimización lineal tiene la siguiente *forma estándar*[14]:

[14] Por convención en la formulación los vectores son columna, su transposición se representa por un superíndice T, las variables se ubican a la izquierda de las ecuaciones y los coeficientes de las variables preceden a éstas.

$$\begin{aligned} &\min z = c^T x \\ &Ax = b \\ &x \geq 0 \end{aligned} \tag{1.15}$$

donde $x \in \mathbb{R}^n$ es el vector de las variables del problema, $A \in \mathbb{R}^{m\times n}$ es la matriz de restricciones del problema o matriz del lado izquierdo (*left hand side* LHS), $c \in \mathbb{R}^n$ es el vector de los coeficientes de las variables en la función objetivo o vector de costes, $b \in \mathbb{R}^m$ es el vector del lado derecho (*right hand side* RHS) o vector de cotas de las restricciones, siendo $b \geq 0$, y la variable $z \in \mathbb{R}$ es el valor de la función objetivo.

Por consiguiente, $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$, $c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$ y $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$.

El número de restricciones m, para el problema expresado en su forma estándar, ha de ser estrictamente menor que el número de variables n, $m < n$, y el rango de la matriz de restricciones máximo, $r(A) = m$.

La forma estándar se utiliza como base para introducir y desarrollar el método simplex, no es necesario que el problema se exprese de esta forma para ser resuelto por un optimizador. Los códigos comerciales no necesitan que los problemas sean escritos de esta manera.

El ejemplo anterior expresado en la forma estándar aparece como:

$$\begin{array}{llllll} \min\ z = -3x_1 - 5x_2 & & & & & \\ x_1 & & +x_3 & & & = 4 \\ & 2x_2 & & +x_4 & & = 12 \\ 3x_1 & +2x_2 & & & +x_5 & = 18 \\ x_1, & x_2, & x_3, & x_4, & x_5, & \geq 0 \end{array}$$

siendo $A = \begin{bmatrix} 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & 2 & \cdot & 1 & \cdot \\ 3 & 2 & \cdot & \cdot & 1 \end{bmatrix}$, $b = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$, $c = \begin{bmatrix} -3 \\ -5 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$ y $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$.

Para convertir un problema de programación lineal cualquiera a su forma estándar se efectúan algunas transformaciones:

Función objetivo

La dirección de maximización se trata como una minimización del valor negativo de la función objetivo.

$$\max z \Rightarrow \min -z \qquad (\max z = -\min -z)$$

Restricciones

En las restricciones de tipo ≤ se introduce una *variable de holgura* $u_i \geq 0$.

$$\sum_j a_{ij} x_j \leq b_i \Rightarrow \sum_j a_{ij} x_j + u_i = b_i$$

En las restricciones de tipo ≥ se introduce una *variable de exceso o de holgura* $v_i \geq 0$.

$$\sum_j a_{ij} x_j \geq b_i \Rightarrow \sum_j a_{ij} x_j - v_i = b_i$$

Variables

Si una variable es negativa y no acotada inferiormente $-\infty \leq x_j \leq 0$, simplemente se sustituye por una variable que tome su valor opuesto $x_j = -y_j$ siendo $0 \leq y_j \leq \infty$.
Si la variable negativa está acotada inferiormente por un valor $L_j < 0$, $L_j \leq x_j \leq 0$, se desplaza en dicha cantidad $x_j = y_j + L_j$ siendo $0 \leq y_j \leq -L_j$. Posteriormente se ve cómo se tratan las variables acotadas.
Si la variable es libre (no acotada inferior ni superiormente, $-\infty \leq x_j \leq \infty$) se sustituye por dos variables que corresponden a su componente positiva y negativa $x_j = x_j^+ - x_j^-$ siendo $0 \leq x_j^+ \leq \infty$ y $0 \leq x_j^- \leq \infty$.

Examinemos los tipos de soluciones del problema de programación lineal:

Solución factible

Aquélla que satisface todas las restricciones.
Cualquier punto del interior o del contorno de la región factible.

Solución infactible

Aquélla que viola al menos una restricción.
Cualquier punto del exterior de la región factible.

Solución básica factible

Una *base* de la matriz de restricciones o matriz del problema es una matriz cuadrada no singular (es decir, tiene inversa) de rango máximo en A,

$r(A) = m$. Las variables asociadas a las columnas que forman la base se denominan *variables básicas* y el resto *variables secundarias* (o *no básicas*).

Solución básica factible es aquélla con m variables básicas que pueden tomar valor diferente de 0 y $(n-m)$ variables no básicas que toman valor 0, está asociada a una base de la matriz. Los valores de las variables básicas, si la base es B, se calculan por resolución de un sistema de $m \times m$ ecuaciones, es decir, se obtienen calculando $B^{-1}b$.

Las variables no básicas en un problema de programación lineal con variables acotadas superior e inferiormente se hallan en uno de sus límites mientras que las variables básicas se encuentran estrictamente dentro de los límites. Este caso corresponde a una *solución básica factible no degenerada.* Si alguna de las variables básicas se encuentra en uno de sus límites se denomina *solución básica factible degenerada.*

Toda solución básica factible está asociada a un único punto extremo o vértice y todo punto extremo está asociado a alguna solución básica factible. Sin embargo, puede existir un punto extremo que corresponda a dos bases diferentes en presencia de degeneración.

Dos soluciones factibles vértices son *adyacentes* si la línea que los conecta es una arista, es decir, si todas excepto una de sus variables son iguales. Cada solución factible vértice tiene n soluciones factibles vértices adyacentes. Moverse de una solución básica factible a otra adyacente es cambiar de una variable no básica a básica y al contrario para otra variable.

Solución óptima

Aquella solución básica factible (vértice) con mejor valor de la función objetivo. Pueden existir múltiples soluciones óptimas. Esta situación, que por inspección del problema de dos dimensiones parecería infrecuente, es una situación habitual en la realidad por razón de que el criterio de optimalidad se comprueba numéricamente, es decir, cuando la tasa de mejora de la función objetivo está por debajo de cierta tolerancia muy pequeña pero no cero.

Existe un resultado fundamental en el que se fundamenta la validez del algoritmo del simplex,

Teorema fundamental de la programación lineal

Dado un problema de programación lineal en forma estándar, se tiene que:

- Si admite una solución factible, admite al menos una solución básica factible.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

- Si admite una solución óptima finita, admite al menos una solución básica óptima.

El primer apartado es equivalente al teorema geométrico de existencia de puntos extremos y el segundo a la aplicación del problema de representación que asegura que la solución óptima se alcanza en un punto extremo (solución básica). Si hay múltiples óptimos, en una región factible acotada, al menos dos deben ser soluciones factibles vértices adyacentes.

El comportamiento del método simplex en el peor caso es muy pobre. Sin embargo, en la práctica recorre un número muy inferior al número total de vértices posibles. Para el problema anterior el método simplex recorre 3 vértices (incluyendo el inicial) de un número total posible (factibles e infactibles) de $C_{5,2} = \begin{pmatrix} 5 \\ 2 \end{pmatrix} = 10$.

Como criterio aproximado se puede estimar que en media el número de iteraciones necesarias para resolver un problema lineal por el método simplex es proporcional al número de restricciones m y el tiempo de solución es proporcional a m^3.

Veamos a continuación cómo se explica y desarrolla el método simplex. Supongamos que se dispone de una solución básica factible $x^T = \begin{bmatrix} x_B & x_N \end{bmatrix}^T$. Sea B la base, $A = \begin{bmatrix} B & N \end{bmatrix}$ y $c^T = \begin{bmatrix} c_B & c_N \end{bmatrix}^T$, siendo

$x_B \in \mathbb{R}^m$	vector de *variables básicas*
$x_N \in \mathbb{R}^{n-m}$	vector de *variables no básicas*
$B \in \mathbb{R}^{m\times m}$	matriz *base*
$N \in \mathbb{R}^{m\times(n-m)}$	matriz no básica
$c_B \in \mathbb{R}^m$	coeficientes de las variables básicas en la función objetivo
$c_N \in \mathbb{R}^{n-m}$	coeficientes de las variables no básicas en la función objetivo

Reformulando el sistema de ecuaciones $Ax = b$ y siempre que B sea no singular

$$Bx_B + Nx_N = b \Leftrightarrow x_B = B^{-1}(b - Nx_N) = B^{-1}b - B^{-1}Nx_N \tag{1.16}$$

La función objetivo se puede expresar entonces como

$$z = c_B^T x_B + c_N^T x_N = c_B^T B^{-1} b - c_B^T B^{-1} N x_N + c_N^T x_N = c_B^T B^{-1} b + \left[c_N^T - c_B^T B^{-1} N\right] x_N \tag{1.17}$$

Lo que se ha hecho es simplemente un cambio de base, es decir, expresar los elementos del problema en función de la base B. En este caso, la solución básica asociada a la base B, viene determinada por los valores de x_B ya que las variables no básicas x_N toman valor 0 (es decir, se resuelve un sistema de ecuaciones lineales para determinar el valor de las variables básicas)

$$x_B = \hat{b} = B^{-1}b \tag{1.18}$$

$$\hat{z} = c_B^T B^{-1} b = c_B^T x_B \tag{1.19}$$

Definamos

$$w^T \equiv c_B^T B^{-1} \tag{1.20}$$

$$Y \equiv B^{-1}N \tag{1.21}$$

$$\hat{c}_N^T \equiv c_N^T - c_B^T B^{-1} N = c_N^T - w^T N = c_N^T - c_B^T Y \tag{1.22}$$

y para cada variable no básica x_j su *coste reducido* $\hat{c}_j$ (o *derivada direccional*) en particular sería

$$\hat{c}_j \equiv c_j - c_B^T B^{-1} a_j = c_j - w^T a_j = c_j - c_B^T y_j = c_j - z_j \tag{1.23}$$

siendo a_j la columna correspondiente a la variable x_j en la matriz A.

Conviene resaltar el significado del vector de costes reducidos $\hat{c}_N$, tal como se desprende de la expresión de la función objetivo. Para ello reformulando la expresión de la función objetivo en función de la base resulta

$$z = c_B^T x_B + \sum_{j \in I_N} \hat{c}_j x_j = \hat{z} + \sum_{j \in I_N} (c_j - z_j) x_j$$

Así, el c*oste reducido* de una variable $\hat{c}_j$ es el cambio en la función objetivo debido a un incremento unitario de una variable no básica. En el óptimo de un problema de minimización todos los costes reducidos de las variables no básicas son positivos o nulos, ya que en tal caso incrementar una variable desde su valor 0 a otro mayor irá en contra de la minimización. Los costes reducidos de las variables básicas son cero. Si el coste reducido de una variable no básica es cero en la solución óptima entonces pueden existir múltiples soluciones óptimas, siendo al menos dos de ellas vértices. En un apartado posterior se menciona analiza expresamente esta posibilidad.

Veamos a continuación cómo se utilizan las expresiones derivadas anteriormente. Si elegimos como variables básicas $x_B = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix}$ y como no básicas $x_N = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ entonces $c_B = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$, $c_N = \begin{bmatrix} -3 \\ -5 \end{bmatrix}$, $B = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & 1 \end{bmatrix} = B^{-1}$ y $N = \begin{bmatrix} 1 & \cdot \\ \cdot & 2 \\ 3 & 2 \end{bmatrix}$.

Los valores que toman dichas variables serán $x_B = \hat{b} = B^{-1}b = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$ y $x_N = \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$, $w^T = c_B^T B^{-1} = \begin{bmatrix} \cdot & \cdot & \cdot \end{bmatrix}$, el valor de la función objetivo será $\hat{z} = c_B^T B^{-1} b = 0$ y los costes reducidos de las variables no básicas $\hat{c}_N^T = c_N^T - w^T N = \begin{bmatrix} -3 & -5 \end{bmatrix}$.

Si alguna variable no básica incrementara su valor desde 0, $x_N > 0$, el valor que tomarían las variables básicas sería

$$x_B = B^{-1}b - B^{-1}Nx_N = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} - \begin{bmatrix} 1 & \cdot \\ \cdot & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

y el de la función objetivo

$$z = c_B^T B^{-1} b + \begin{bmatrix} c_N^T - c_B^T B^{-1} N \end{bmatrix} x_N = 0 + \begin{bmatrix} -3 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

De acuerdo con esta ecuación la función objetivo disminuye al incrementar desde 0 cualquiera de las variables no básicas x_1 o x_2 por ser negativos sus costes reducidos $\hat{c}_N^T = \begin{bmatrix} -3 & -5 \end{bmatrix}$, es decir, la base actual no es óptima.

La *prueba de optimalidad* se basa en comprobar si existe algún coste reducido negativo (en un problema de minimización) en la función objetivo. En caso de que no exista, la solución básica actual es óptima. En otro caso, se puede seleccionar una de las variables no básicas con coste reducido negativo para entrar en la base. La variable seleccionada x_t, denominada *variable básica entrante*, será la que tenga coste reducido negativo de mayor valor absoluto.

Para determinar hasta dónde se puede incrementar dicha variable sin violar ninguna restricción de no negatividad de las variables básicas se observan las ecuaciones de las variables básicas

$$x_B = B^{-1}b - B^{-1}Nx_N = \hat{b} - Yx_N = \hat{b} - y_t x_t \tag{1.24}$$

siendo a_t la columna de A correspondiente a x_t e $y_t = B^{-1}a_t$ la *columna pivote.*

Podemos escribir esta ecuación componente a componente

$$(x_B)_i = \hat{b}_i - y_{it}x_t \tag{1.25}$$

Si $y_{it} > 0$ entonces $(x_B)_i$ disminuye cuando la variable básica entrante x_t aumenta, hasta alcanzar el valor 0 para $x_t = \hat{b}_i / y_{it}$, $\hat{b}_i > 0$ por ser el valor de la variable básica.

Para valores $y_{it} \leq 0$ la variable $(x_B)_i$ aumenta o permanece igual. Si todos los valores fueran $y_{it} \leq 0$ el problema sería *no acotado*, puesto que la variable básica entrante se puede incrementar indefinidamente sin que una variable básica alcance el valor 0. La dirección de no acotamiento para dicha variable no básica x_t es un vector formado por el vector $d_B = -y_t = -B^{-1}a_t$ para las variables básicas, $d_t = 1$ para la variable básica entrante x_t y $d_i = 0$ para el resto de variables no básicas. Esta dirección satisface la condición $Ad = 0$ y $d \geq 0$, es un rayo extremo y pertenece al cono de recesión, que se define como el conjunto de direcciones en las que el problema resulta no acotado. Obsérvese que la detección de problema no acotado se puede hacer con cualquier variable no básica no necesariamente con la de coste reducido negativo de mayor valor absoluto.

La variable x_t puede aumentar su valor hasta $\overline{x}_t$ mientras todas las variables básicas sigan siendo no negativas. Así el máximo valor que puede tomar y en el que además una variable básica se hace 0 y puede salir de la base es

$$\overline{x}_t = \min_{1 \leq i \leq m} \left\{ \frac{\hat{b}_i}{y_{it}} : y_{it} > 0 \right\} \tag{1.26}$$

El valor mínimo[15] de la relación previa identifica la *variable básica saliente*, es decir, la variable básica que se hace no básica (toma valor 0).

Después del intercambio de variables la nueva solución básica factible es un nuevo vértice adyacente al anterior con mejor valor de la función objetivo. Para la nueva base se calcula el nuevo valor de la función objetivo y de las variables básicas. El resto de variables, las no básicas, siguen siendo cero.

$$\hat{z} = \hat{z} + \hat{c}_t \overline{x}_t \tag{1.27}$$

$$x_B = \hat{b} - y_t \overline{x}_t \tag{1.28}$$

[15] Si existe más de una relación que alcance el valor mínimo al mismo tiempo usualmente se toma aquél con mayor coeficiente en la columna pivote.

El *método simplex* consiste en el siguiente algoritmo, que realiza iterativamente este cambio de base mejorando la función objetivo:

- *Inicialización*
  Se supone que se dispone de una matriz base B correspondiente a una *solución básica factible inicial* $x_B = \hat{b} = B^{-1}b \geq 0$ y una función objetivo $\hat{z} = c_B^T B^{-1} b = c_B^T x_B$.

- *Prueba de optimalidad*
  Se calcula el vector de *costes reducidos* $\hat{c}_N^T = c_N^T - c_B^T B^{-1} N = c_N^T - c_B^T Y$. Si todos los costes reducidos son no negativos $\hat{c}_N^T \geq 0$ la base actual es óptima. En caso contrario, se selecciona[16] como *variable básica entrante* x_t una variable con coste reducido estrictamente negativo $\hat{c}_t < 0$.
  La prueba de optimalidad es local pero como los problemas de programación lineal son convexos, la solución óptima será global.

- *Iteración*
  Sea $y_t = B^{-1} a_t$, la *columna pivote* t correspondiente a la variable básica entrante. Se busca la *fila pivote* s que verifica

$$\frac{\hat{b}_s}{y_{st}} = \min_{1 \leq i \leq m} \left\{ \frac{\hat{b}_i}{y_{it}} : y_{it} > 0 \right\}$$

  y determina la *variable básica saliente* x_s y el elemento pivote y_{st}.
  Si $y_{it} \leq 0$ para toda fila i entonces el problema es no acotado.

- *Pivotamiento*
  Actualización de la matriz B^{-1} y del vector de variables básicas y volver al paso 2. Más adelante al presentar la forma tabular se muestra cómo se actualiza la matriz B^{-1} en un número reducido de operaciones.

Sigamos aplicando el método simplex al caso ejemplo anterior. Elegimos x_2 como la variable básica entrante por tener el coste reducido negativo con mayor valor absoluto. Calculamos los elementos de la columna pivote

[16] La variable básica entrante seleccionada es la de coste reducido negativo de mayor valor absoluto. Sin embargo, esta opción no considera el cálculo de la relación mínima (es decir, el incremento que se le va a dar a dicha variable) luego la mejora en la función objetivo puede ser pequeña. Tampoco tiene en cuenta el efecto del escalado de las variables en el problema. En general, no existe manera práctica de predecir qué elección de variable básica entrante puede dar lugar al menor número de iteraciones.

$$y_t = y_2 = B^{-1}a_2 = \begin{bmatrix} \cdot \\ 2 \\ 2 \end{bmatrix}$$

Las relaciones correspondientes a los dos últimos elementos de y_t, los únicos estrictamente positivos, son

$$\hat{b}_2 / y_{22} = 12/2 = 6 \text{ y } \hat{b}_3 / y_{32} = 18/2 = 9$$

Como la primera relación es la menor, x_4 será la variable básica saliente. Se reemplaza en la base la variable básica x_4 por la variable no básica x_2 y tendremos

$$x_B = \begin{bmatrix} x_3 \\ x_2 \\ x_5 \end{bmatrix} \text{ y } x_N = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}$$

Entonces $B = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 2 & \cdot \\ \cdot & 2 & 1 \end{bmatrix}$, $B^{-1} = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1/2 & \cdot \\ \cdot & -1 & 1 \end{bmatrix}$ y $N = \begin{bmatrix} 1 & \cdot \\ \cdot & 1 \\ 3 & \cdot \end{bmatrix}$. Los costes de la variables básicas son $c_B^T = \begin{bmatrix} \cdot & -5 & \cdot \end{bmatrix}$ y los de las no básicas $c_N^T = \begin{bmatrix} -3 & \cdot \end{bmatrix}$.

El valor de las variables básicas será ahora

$$x_B = \hat{b} = B^{-1}b = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1/2 & \cdot \\ \cdot & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 6 \end{bmatrix}$$

y el valor de la función objetivo será $\hat{z} = c_B^T B^{-1} b = -30$.

Los coeficientes

$$w^T = c_B^T B^{-1} = \begin{bmatrix} \cdot & -5 & \cdot \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1/2 & \cdot \\ \cdot & -1 & 1 \end{bmatrix} = \begin{bmatrix} \cdot & -5/2 & \cdot \end{bmatrix}$$

y los costes reducidos

$$\hat{c}_N^T = c_N^T - w^T N = \begin{bmatrix} -3 & \cdot \end{bmatrix} - \begin{bmatrix} \cdot & -5/2 & \cdot \end{bmatrix} \begin{bmatrix} 1 & \cdot \\ \cdot & 1 \\ 3 & \cdot \end{bmatrix} = \begin{bmatrix} -3 & 5/2 \end{bmatrix}.$$

Elegimos x_1 como la variable básica entrante por ser la única con un coste reducido negativo. Calculamos los elementos de la columna pivote

$$y_t = y_1 = B^{-1}a_1 = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1/2 & \cdot \\ \cdot & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \cdot \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ \cdot \\ 3 \end{bmatrix}$$

Las relaciones correspondientes a los elementos de y_t estrictamente positivos son

$$\hat{b}_1 / y_{11} = 4/1 = 4 \text{ y } \hat{b}_3 / y_{31} = 6/3 = 2$$

Como la segunda relación es la menor x_5 será la variable básica saliente. Se reemplaza en la base la variable básica x_5 por la variable x_1 y tendremos

$$x_B = \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix} \text{ y } x_N = \begin{bmatrix} x_5 \\ x_4 \end{bmatrix}$$

Entonces $B = \begin{bmatrix} 1 & \cdot & 1 \\ \cdot & 2 & \cdot \\ \cdot & 2 & 3 \end{bmatrix}$, $B^{-1} = \begin{bmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{bmatrix}$ y $N = \begin{bmatrix} \cdot & \cdot \\ \cdot & 1 \\ 1 & \cdot \end{bmatrix}$. Los costes de la variables básicas son $c_B^T = \begin{bmatrix} \cdot & -5 & -3 \end{bmatrix}$ y los de las no básicas $c_N^T = \begin{bmatrix} \cdot & \cdot \end{bmatrix}$.

El valor de las variables básicas será ahora

$$x_B = \hat{b} = B^{-1}b = \begin{bmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}$$

y el valor de la función objetivo será $\hat{z} = c_B^T B^{-1} b = -36$.

Los coeficientes

$$w^T = c_B^T B^{-1} = \begin{bmatrix} \cdot & -5 & -3 \end{bmatrix} \begin{bmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{bmatrix} = \begin{bmatrix} \cdot & -3/2 & -1 \end{bmatrix}$$

y los costes reducidos

$$\hat{c}_N^T = c_N^T - w^T N = \begin{bmatrix} \cdot & \cdot \end{bmatrix} - \begin{bmatrix} \cdot & -3/2 & -1 \end{bmatrix} \begin{bmatrix} \cdot & \cdot \\ \cdot & 1 \\ 1 & \cdot \end{bmatrix} = \begin{bmatrix} 1 & 3/2 \end{bmatrix}$$

Como todos los costes reducidos son positivos no se puede mejorar la función objetivo, es decir, se ha alcanzado la solución óptima.

I.5.4.1. Problema de maximización

Si bien se ha dicho que un problema de maximización puede ser transformado en uno de minimización para ser resuelto, también es cierto que el algoritmo anterior es fácilmente adaptable al caso de maximización, ya que sólo afecta a la dirección de mejora y, por lo tanto, sólo hay que modificar el criterio de optimalidad y el de entrada en la base.

En tal caso, una solución será óptima si todos los costes reducidos son menores o iguales que 0 y, en caso de que no sea así, la variable básica entrante es la que tenga coste reducido estrictamente positivo de mayor valor.

I.5.4.2. Múltiples óptimos

En algunos problemas no existe un único óptimo, sino que puede haber varios. Esta condición es fácilmente detectable pues para que existan múltiples óptimos al finalizar el algoritmo del simplex debe haber al menos una variable no básica cuyo coste reducido sea igual a 0, de modo que si la variable aumenta su valor la función objetivo no se ve afectada, con lo que puede ser introducida en la base.

Sin embargo, no es una condición suficiente que el coste reducido sea 0, hay que asegurar también que cuando esta variable entra en la base lo hace con valor distinto de 0, pues en otro caso, si sustituye a una variable con valor 0, se produce un cambio de base pero no de punto.

Una vez identificadas todas las soluciones básicas factibles (puntos extremos) óptimas, serán soluciones óptimas todas las combinaciones lineales convexas de éstas. Por ejemplo, si existen dos soluciones básicas óptimas, todo el segmento que une los puntos correspondientes son soluciones óptimas del problema.

En algunos problemas, identificar más de una solución óptima puede ser de vital importancia, especialmente si existe más de un criterio de optimización y éstos han sido jerarquizados (optimización multiobjetivo).

I.5.4.3. Convergencia del algoritmo

La convergencia del algoritmo está asegurada si no existe degeneración. Sin embargo, en caso de degeneración de varias variables básicas el algoritmo puede ciclar, moviéndose por una secuencia de bases que correspondan al mismo punto extremo sin mejorar la función objetivo. Existen procedimientos, que aquí no se presentan, para evitar esta situación, siendo el más popular de ellos, que no el único, el método lexicográfico.

I.5.4.4. Variables acotadas superiormente

Las cotas superiores no se deben tratar como restricciones, esto tiene ventajas computacionales. En el método simplex las cotas superiores no se deben sobrepasar cuando la variable básica entrante se incrementa hasta alcanzar un nuevo vértice.

Supongamos una variable acotada superiormente $0 \le x_j \le u_j$, entonces su variable complementaria definida como $y_j = u_j - x_j$ también está acotada por el mismo valor $0 \le y_j \le u_j$. Si $x_j = 0$, x_j es no básica. Si $x_j = u_j$, entonces $y_j = 0$ y, por tanto, y_j será no básica.

En el método simplex, la variable básica saliente es la primera que alcanza su cota 0 al incrementar la variable básica entrante. Ahora esta comprobación se extiende para la variable básica entrante a su cota u_j. Cuando se alcanza la cota superior se cambia la variable x_j a y_j como nueva variable no básica.

I.5.4.5. Forma tabular

Una forma sencilla de presentar este método y resolver ejemplos pequeños es mediante la *tabla* del simplex. En esta tabla se refleja el problema en función de la base actual en cada iteración. La aportación de esta tabla es que la actualización de los valores para pasar de una base a otra (actualización de la matriz B o B^{-1}) se puede realizar por simple eliminación gaussiana[17], lo que también se denomina *pivotamiento.*

Para el problema original la tabla tiene este aspecto

Variables básicas	z	x_N	x_B	**Cotas**
$-z$	-1	c_N^T	c_B^T	0
x_B	0	N	B	b

[17] La eliminación gaussiana actualiza la tabla para que en ella aparezca un 1 en el elemento pivote y 0 en el resto de la columna. Para aplicarla seguir los siguientes pasos:

1. Dividir la fila pivote por el elemento pivote y_{st}
2. Restar a las demás filas de la tabla la nueva fila pivote multiplicada por el elemento que se desea pase a ser cero, es decir, el elemento correspondiente de la columna pivote.

Además, se han de hacer 0 los coeficientes de la función objetivo correspondientes a las variables básicas.

Tabla 2.1 Tabla (A) del simplex en la primera iteración.

y en una iteración cualquiera en función de la base B (siendo B las columnas bajo las variables básicas actuales de la matriz original A y N las columnas de las variables no básicas actuales de la matriz original A) se tiene

Variables básicas	z	x_N	x_B	**Cotas**
$-z$	-1	$c_N^T - c_B^T B^{-1} N$	0	$-c_B^T B^{-1} b$
x_B	0	$B^{-1}N$	I	$B^{-1}b$

Tabla 2.2 Tabla (A) del simplex en una iteración cualquiera.

c_B^T y c_N^T son los coeficientes de las variables básicas y no básicas actuales en la función objetivo. La ecuación de la función objetivo de las nuevas tablas resulta de multiplicar las ecuaciones de las restricciones de las tablas originales respectivas por $w^T = c_B^T B^{-1}$ y restarla de la ecuación de la función objetivo de las tablas originales respectivas. Las ecuaciones de las restricciones de las nuevas tablas resultan de multiplicar las de las tablas originales respectivas por B^{-1}.

Si en la solución inicial la base fuera una matriz identidad y los costes de las variables básicas fueran 0, la tabla original inicial sería la siguiente (por notación se designa con un apóstrofo a lo que es inicial)

Variables básicas	z	x'_N	x'_B	**Cotas**
$-z$	-1	$c_N'^T$	0	0
x_B	0	N'	I	b

Tabla 2.3 Tabla (B) del simplex en la primera iteración partiendo de una matriz identidad.

y, manteniendo la asignación (ordenación) original por columnas de las variables, la tabla de simplex tiene este aspecto para una iteración cualquiera (siendo N' las columnas de las variables no básicas iniciales en la matriz original A)

Variables básicas	z	x'_N	x'_B	**Cotas**
$-z$	-1	$c_N'^T - c_B^T B^{-1} N'$	$-c_B^T B^{-1}$	$-c_B^T B^{-1} b$
x_B	0	$B^{-1}N'$	B^{-1}	$B^{-1}b$

Tabla 2.4 Tabla (B) del simplex en una iteración cualquiera.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Bajo las columnas de las variables básicas originales inicialmente estaba la matriz identidad I y ahora está la matriz inversa de la base B^{-1}. Luego esta matriz contiene las manipulaciones hechas a las restricciones hasta una iteración cualquiera. $c_N'^T$ son los coeficientes de las variables no básicas iniciales y c_B^T son los coeficientes de las variables básicas de cada iteración.

Observando las tablas 2.1 y 2.2 se puede decir que el método simplex persigue hallar la partición adecuada entre variables básicas y no básicas que cumplen la condición de optimalidad $\hat{c}_N^T = c_N^T - c_B^T B^{-1} N \geq 0$. Observando las tablas 2.3 y 2.4 consistiría en hallar los valores que cumplen la condición de optimalidad del problema, es decir, coeficientes de la función objetivo no negativos en el caso de minimización $\hat{c}_N^T = c_N'^T - c_B^T B^{-1} N' \geq 0$ y $-w^T = -c_B^T B^{-1} \geq 0$.

Veamos con un ejemplo las iteraciones del método simplex en forma tabular. La tabla inicial del problema anterior es:

Variables básicas	z	x_1	x_2	x_3	x_4	x_5	**Cotas**	**Relación**
$-z$	–1	–3	–5				0	
x_3		1		1			4	
x_4			2		1		12	12/2=6
x_5		3	2			1	18	18/2=9

La solución básica factible inicial es $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 4, 12, 18)$ y $z = 0$. El recuadro dentro de la tabla identifica la matriz inversa de la base B^{-1}.

Se toma x_2 como variable básica entrante por tener el coste reducido negativo de mayor valor absoluto. Luego su columna es la columna pivote.

Se toma x_4 como variable básica saliente por tener la menor relación. Luego su fila es la fila pivote.

Se actualiza la tabla para anular los elementos de la columna pivote excepto el elemento pivote que se hace 1 (es decir, se anula el coeficiente de la variable básica entrante en la función objetivo y se crea una columna de la matriz identidad en dicha variable) para resolver el sistema de ecuaciones por eliminación gaussiana. Se hacen 0 los coeficientes de la función objetivo correspondientes a las variables básicas.

Variables básicas	z	x_1	x_2	x_3	x_4	x_5	**Cotas**	**Relación**
$-z$	–1	–3			5/2		30	
x_3		1		1			4	4/1=4
x_2			1		1/2		6	
x_5		3			–1	1	6	6/3=2

La nueva solución básica factible es $(x_1, x_2, x_3, x_4, x_5) = (0, 6, 4, 0, 6)$ y $z = -30$.

Se toma x_1 como variable básica entrante por tener el único coste reducido negativo. Luego su columna es la columna pivote.

Se toma x_5 como variable básica saliente por tener la menor relación. Luego su fila es la fila pivote.

Se vuelve a actualizar la tabla del simplex.

Variables básicas	z	x_1	x_2	x_3	x_4	x_5	**Cotas**
$-z$	–1				3/2	1	36
x_3				1	1/3	–1/3	2
x_2			1		1/2		6
x_1		1			–1/3	1/3	2

La nueva solución básica factible es $(x_1, x_2, x_3, x_4, x_5) = (2, 6, 2, 0, 0)$ y $\hat{z} = -36$. Ésta es ya la solución óptima por ser todos los costes reducidos, correspondientes a las variables no básicas (x_5, x_4), no negativos $\hat{c}_N^T = \begin{bmatrix} 1 & 3/2 \end{bmatrix}$.

I.5.4.6. Solución básica factible inicial

El método simplex parte de una solución básica factible y se mueve en cada iteración a otra hasta llegar a la solución óptima. Resuelve este problema de optimización lineal, presentado en su forma estándar

$$\begin{aligned} &\min c^T x \\ &Ax = b \\ &x \geq 0 \end{aligned} \qquad (P)$$

Sin embargo, no siempre es fácil proporcionar una solución básica factible inicial. Como se ha visto, la forma sencilla para hacerlo es partiendo de una matriz identidad en los coeficientes de las variables básicas iniciales.

Para ello en las restricciones de tipo $\leq$ se aprovechan las *variables de holgura*. En las restricciones de tipo $\geq$ se introduce una *variable de exceso* (a veces a ésta también se la denomina de holgura) con signo negativo $v_i \geq 0$ y una *variable artificial* $w_i \geq 0$. Y en las restricciones de tipo $=$ se introduce una *variable artificial* $w_i \geq 0$.

$$\sum_j a_{ij} x_j \leq b_i \Rightarrow \sum_j a_{ij} x_j + u_i = b_i$$

$$\sum_j a_{ij} x_j \geq b_i \Rightarrow \sum_j a_{ij} x_j - v_i + w_i = b_i$$

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

$$\sum_j a_{ij} x_j = b_i \Rightarrow \sum_j a_{ij} x_j + w_i = b_i$$

Con las variables de holgura y artificiales se parte de una matriz identidad para las variables básicas y el problema de optimización lineal a resolver es

$$\begin{aligned} &\min c^T x \\ &Ax + Ix_a = b \\ &x, x_a \geq 0 \end{aligned} \qquad (P(a))$$

siendo x_a las variables artificiales.

Pero si la solución incluye variables artificiales no es una solución factible del problema original. Por consiguiente, hay que eliminar esas variables aprovechando el propio método simplex, de modo que en su desarrollo estas variables dejen de ser variables básicas. Si al final no se consiguen anular las variables artificiales el problema original será infactible.

Obsérvese que en las hipótesis del método simplex se ha supuesto que la matriz era de rango máximo, sin embargo, esta hipótesis no se ha comprobado nunca. La razón es que al partir de una matriz identidad de rango máximo la hipótesis necesariamente se verifica. Sin embargo, si no se parte de una matriz identidad habría que comprobar que se cumple esta condición. Obsérvese también que aunque el problema original no la cumpliera, al añadir las variables artificiales sí lo hará.

Para lograr anular las variables artificiales se emplean habitualmente dos métodos:

- el *método de la M mayúscula*
- el *método de las dos fases*

El método de la M mayúscula introduce las variables artificiales en la función objetivo original pero penalizadas con un coeficiente M de valor muy elevado. Este coeficiente de penalización puede introducir problemas numéricos por lo que, en general, es preferible el método siguiente.

En el método de las dos fases, la fase I del método simplex tiene como función objetivo la suma de las variables artificiales. Si esta fase I acaba con valor de la función objetivo igual a cero entonces el problema original es factible y se dispone de una solución básica factible siendo no básicas las variables artificiales. La fase II restablece la función objetivo original y aplica de nuevo el método simplex hasta alcanzar la solución óptima a partir de la solución básica factible del final de la fase I.

Supóngase el problema original (P) y el problema una vez que se han introducido las variables artificiales $(P(a))$. El método de las dos fases consta de los siguientes pasos:

Fase I: Resolver el problema

$$\begin{aligned}
&\min z' = \sum_a x_a \\
&Ax + Ix_a = b \\
&x, x_a \geq 0
\end{aligned}$$

Existen dos posibles finales de esta fase

- $z' > 0$, entonces el problema (P) no es factible.
- $z' = 0$, entonces el problema (P) es factible y tenemos una solución factible inicial.

Sin embargo, todavía es posible distinguir dos casos en esta situación, antes de pasar a la fase II:

1. No hay variables artificiales en la base se puede pasar a la fase II.
2. Hay variables artificiales en la base con valor 0 (degeneradas). En este caso, la solución obtenida no es una solución básica del problema original ya que incluye variables artificiales. Para eliminarlas de la base, se sustituyen mediante pivotamiento por cualquier variable original no básica (incluyendo de holgura y exceso) cuyo coste reducido sea 0 y cuyo elemento en la fila de la variable artificial sea distinto de 0 (incluso puede ser negativo). Si ninguna variable original tuviera un valor distinto de cero en esa fila (la fila de la variable artificial tiene todo ceros en las variables originales), quiere decir que esa fila es redundante y puede ser eliminada directamente.

Fase II: Eliminar las variables artificiales del problema[18] y retomar la función objetivo original, recalculando los costes reducidos para la solución básica factible obtenida en la fase I.

Veamos con un ejemplo el método de las dos fases.

$$\begin{aligned}
&\min z = 0.4x_1 + 0.5x_2 \\
&0.3x_1 \quad +0.1x_2 \quad \leq 2.7 \\
&0.5x_1 \quad +0.5x_2 \quad = 6 \\
&0.6x_1 \quad +0.4x_2 \quad \geq 6 \\
&\ x_1, \qquad x_2 \qquad \geq 0
\end{aligned}$$

[18] En muchos casos no se eliminan estas variables, sino que se mantienen aunque no se cuente con ellas para entrar en la base, ya que aportan información al final (B^{-1}, ...).

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Se convierte el problema a la forma estándar y se añaden las variables artificiales pertinentes

$$\begin{array}{llllllll}
\min z = 0.4x_1 + 0.5x_2 \\
0.3x_1 & +0.1x_2 & +x_3 & & & & = 2.7 \\
0.5x_1 & +0.5x_2 & & +\overline{x}_4 & & & = 6 \\
0.6x_1 & +0.4x_2 & & & -x_5 & +\overline{x}_6 & = 6 \\
x_1, & x_2, & x_3, & \overline{x}_4, & x_5, & \overline{x}_6 & \geq 0
\end{array}$$

siendo x_3 una variable de holgura, x_5 una variable de exceso y $\overline{x}_4$ y $\overline{x}_6$ variables artificiales.

La fase I del método simplex minimiza la suma de las variables artificiales

$$\begin{array}{llllllll}
\min z' = \overline{x}_4 + \overline{x}_6 \\
0.3x_1 & +0.1x_2 & +x_3 & & & & = 2.7 \\
0.5x_1 & +0.5x_2 & & +\overline{x}_4 & & & = 6 \\
0.6x_1 & +0.4x_2 & & & -x_5 & +\overline{x}_6 & = 6 \\
x_1, & x_2, & x_3, & \overline{x}_4, & x_5, & \overline{x}_6 & \geq 0
\end{array}$$

Si esta fase acaba con la función objetivo con valor 0 significa que se ha encontrado una solución básica factible inicial del problema original y se puede seguir con la fase II. En caso contrario el problema es infactible.

Variables básicas	z'	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Cotas**
$-z'$	–1					1	1	
x_3		0.3	0.1		1			2.7
$\overline{x}_4$		0.5	0.5			1		6
$\overline{x}_6$		0.6	0.4	–1			1	6

Se transforma la tabla para hacer 0 los coeficientes de las variables básicas

Variables Básicas	z'	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**	**Relación**
$-z'$	–1	–1.1	–0.9	1				–12	
x_3		0.3	0.1		1			2.7	9
$\overline{x}_4$		0.5	0.5			1		6	12
$\overline{x}_6$		0.6	0.4	–1			1	6	10

Se elige x_1 como variable básica entrante y entonces x_3 es la variable básica saliente. Se hacen 0 los elementos de la columna pivote excepto el elemento pivote que se hace 1.

Variables Básicas	z'	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**	**Relación**
$-z'$	–1		–0.53	1	3.67			–2.1	
x_1		1	0.33		3.33			9	27
$\overline{x}_4$			0.33		–1.67	1		1.5	4.5
$\overline{x}_6$			0.2	–1	–2		1	0.6	3

Se elige x_2 como variable básica entrante y entonces $\overline{x}_6$ es la variable básica saliente. Se hacen 0 los elementos de la columna pivote excepto el elemento pivote que se hace 1.

Variables Básicas	z'	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**	**Relación**
$-z'$	–1			–1.67	–1.67		2.67	–0.5	
x_1		1		1.67	6.67		–1.67	8	1.2
$\overline{x}_4$				1.67	1.67	1	–1.67	0.5	0.3
x_2			1	–5	–10		5	3	

Existen dos variables no básicas x_3 y x_5 con igual coeficiente negativo de mayor valor absoluto en la función objetivo. Se puede elegir cualquiera de ellas como variable básica entrante. Arbitrariamente, se elige x_3 y, por tanto, $\overline{x}_4$ es la variable básica saliente. Se hacen 0 los elementos de la columna pivote excepto el elemento pivote que se hace 1.

Variables Básicas	z'	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**
$-z'$	–1					1	1	
x_1		1		–5		–4	5	6
x_3				1	1	0.6	–1	0.3
x_2			1	5		6	–5	6

La solución $(x_1, x_2, x_3, \overline{x}_4, x_5, \overline{x}_6) = (6, 6, 0.3, 0, 0, 0)$ es óptima y como $\hat{z}' = 0$ ésta es una solución básica factible inicial para el problema original.

Se plantea entonces la fase II donde se introduce la función objetivo original.

Variables Básicas	z	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**
$-z$	–1	0.4	0.5					
x_1		1		–5		–4	5	6

x_3				1	1	0.6	–1	0.3
x_2			1	5		6	–5	6

Se deben hacer 0 los coeficientes de las variables básicas x_1, x_2 y x_3 en la función objetivo

Variables Básicas	z	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**	**Relación**
$-z$	–1			–0.5		–1.4	0.5	–5.4	
x_1		1		–5		–4	5	6	
x_3				1	1	0.6	–1	0.3	0.3
x_2			1	5		6	–5	6	1.2

A partir de ahora se ignoran los coeficientes de las variables artificiales para elegir la variable básica entrante para evitar que éstas tomen valor positivo. Por tanto, se elige x_5 como variable básica entrante y entonces x_3 es la variable básica saliente. Se hacen 0 los elementos de la columna pivote excepto el elemento pivote que se hace 1.

Variables Básicas	z	x_1	x_2	x_5	x_3	$\overline{x}_4$	$\overline{x}_6$	**Valores**
$-z$	–1				0.5	–1.1		–5.25
x_1		1			5	–1		7.5
x_5				1	1	0.6	–1	0.3
x_2			1		–5	3		4.5

Como no aparecen costes reducidos negativos (excepto en una variable artificial) la solución $(x_1, x_2, x_3, \overline{x}_4, x_5, \overline{x}_6) = (7.5, 4.5, 0, 0, 0.3, 0)$ es óptima y el valor de la función objetivo es $\hat{z} = 5.25$.

I.5.4.7. Método simplex revisado(DOCT)

El método simplex en formato tabular efectúa más cálculos de los estrictamente necesarios en cada iteración. De hecho en una iteración cualquiera sólo se necesitan los coeficientes de la variable básica entrante (la columna pivote) no los del resto de variables no básicas. El *método simplex revisado* calcula en cada iteración sólo la información que necesita en dicha iteración. Por consiguiente, requiere menos cálculo y menos almacenamiento.

En forma tabular el método simplex revisado se desarrolla sobre una tabla de dimensiones $(m+1)\times(m+1)$, siendo m el número de restricciones. La tabla del método simplex revisado es la siguiente:

x'_B	**Valores**
$w^T = c_B^T B^{-1}$	$\hat{z} = c_B^T B^{-1} b$
B^{-1}	$\hat{b} = B^{-1} b$

Tabla 2.5 Tabla (B) del simplex revisado en una iteración cualquiera.

En cada iteración se calculan los costes reducidos de las variables no básicas ($\hat{c}_j = c_j - z_j = c_j - w^T a_j$) fuera de la tabla para aplicar el criterio de optimalidad y, en su caso, determinar la variable entrante, x_t. Igualmente, se calcula la columna pivote fuera de la tabla ($y_t = B^{-1} a_t$) para aplicar el criterio de salida y determinar la variable básica saliente ($\frac{\hat{b}_s}{y_{st}} = \min_{1 \le i \le m} \left\{ \frac{\hat{b}_i}{y_{it}} : y_{it} > 0 \right\}$). El pivotamiento se hace sobre la tabla actual, suponiendo que existe una columna adicional (realmente no es añadida) que es la columna pivote.

Obsérvese que todos los elementos necesarios de la iteración anterior están recogidos en esta tabla del simplex revisado y que de una iteración a otra los elementos que se obtienen mediante pivotamiento son los mismos, pero actualizados con la nueva base (consecuencia de la eliminación gaussiana).

I.5.4.8. Forma producto de la inversa(DOCT)

El mayor esfuerzo computacional del método simplex se lo llevan los cálculos asociados a la inversa de la base B^{-1}. La razón estriba en que, mientras la matriz base B para problemas grandes suele ser cuasivacía, la inversa tiende a ser densa. En primer lugar, este hecho disuade del cálculo explícito de B^{-1} a partir de B en cada iteración. En segundo lugar, B cambia sólo ligeramente en cada iteración luego será posible actualizar tanto B como B^{-1} sin necesidad de recalcularlas en cada paso tal como se hace en el método simplex explicado hasta ahora. Para ello se utiliza el método denominado forma producto de la inversa, que es un avance del método simplex revisado.

Como el cambio en la matriz base B se produce en una columna en cada iteración es fácil actualizar dicha matriz y también la inversa. Si en una iteración cualquiera la columna pivote corresponde a x_t y la fila pivote es la s, la nueva matriz base $\overline{B}$ se puede calcular a partir de la matriz B anterior mediante transformaciones elementales simplemente reemplazando la columna s por la columna pivote y_t. Como $y_t = B^{-1} a_t$ o $B y_t = a_t$ entonces

$$\overline{B} = BF \tag{1.29}$$

siendo

$$F = \begin{pmatrix} 1 & & \Bigg[& \Bigg] & & \\ & \ddots & & & & \\ & & & y_t & & \\ & & & & \ddots & \\ & & \Bigg[& \Bigg] & & 1 \end{pmatrix}$$

obtenida a partir de la matriz identidad reemplazando la columna s por y_t.

Si denominamos $E = F^{-1}$ la nueva matriz inversa $\bar{B}^{-1}$ se obtendrá a partir de la anterior premultiplicando por la matriz E.

$$\bar{B}^{-1} = EB^{-1} \tag{1.30}$$

donde

$$E = \begin{pmatrix} 1 & & \Bigg[& \Bigg] & & \\ & \ddots & & & & \\ & & & \eta & & \\ & & & & \ddots & \\ & & \Bigg[& \Bigg] & & 1 \end{pmatrix} \text{ siendo } \eta = \begin{bmatrix} -y_{1t}/y_{st} \\ \vdots \\ 1/y_{st} \\ \vdots \\ -y_{mt}/y_{st} \end{bmatrix}.$$

La matriz E se llama *matriz elemental* y esta forma de actualizar la matriz inversa se denomina *forma producto*, porque representa la inversa de la base en cualquier iteración k como un producto de matrices elementales E_k de las iteraciones $(k-1, k-2, \ldots, 2, 1)$, suponiendo que la matriz base inicial es la matriz identidad.

$$B_k^{-1} = E_{k-1}E_{k-2}\cdots E_2E_1 \tag{1.31}$$

Veamos la actualización de la inversa de la base del ejemplo anterior con la forma producto de la inversa. En la primera iteración la columna pivote es $y_t = \begin{pmatrix} \cdot & 2 & 2 \end{pmatrix}^T$ y el elemento pivote es el segundo, $s = 2$. Luego

$$\eta_1 = \begin{pmatrix} \cdot \\ 1/2 \\ -1 \end{pmatrix}, \; E_1 = \begin{pmatrix} 1 & \cdot & \cdot \\ \cdot & 1/2 & \cdot \\ \cdot & -1 & 1 \end{pmatrix} \text{ y } B_2^{-1} = E_1B_1^{-1} = E_1 = \begin{pmatrix} 1 & \cdot & \cdot \\ \cdot & 1/2 & \cdot \\ \cdot & -1 & 1 \end{pmatrix}$$

En la segunda iteración la columna pivote es $y_t = \begin{pmatrix} 1 & \cdot & 3 \end{pmatrix}^T$ y el elemento pivote es el tercero, $s = 3$. Luego

$$\eta_2 = \begin{pmatrix} -1/3 \\ \cdot \\ 1/3 \end{pmatrix},\ E_2 = \begin{pmatrix} 1 & \cdot & -1/3 \\ \cdot & 1 & \cdot \\ \cdot & \cdot & 1/3 \end{pmatrix} \text{ y } B_3^{-1} = E_2 E_1 = \begin{pmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{pmatrix}$$

En la implantación informática más que actualizar en cada iteración la matriz se almacenan las transformaciones elementales de manera compacta. Cuando el almacenamiento ocupado es excesivo se elimina y se vuelve a refactorizar la matriz B^{-1} desde el principio.

De hecho, ni siquiera es necesario disponer de la inversa de la matriz base B^{-1} sino utilizarla en productos de dicha matriz por vectores. Puede ser postmultiplicación por un vector como para el cálculo del valor de las variables básicas $\hat{b} = B^{-1}b$ o de la columna pivote $y_t = B^{-1}a_t$ o premultiplicación por un vector como en el caso del cálculo de $w^T = c_B^T B^{-1}$.

La postmultiplicación por un vector se realiza secuencialmente de esta forma

$$B_k^{-1}a = E_{k-1}(E_{k-2}(E_{k-3}(\cdots(E_2(E_1 a))\cdots))) \tag{1.32}$$

premultiplicando en primer lugar el vector a por la matriz elemental E_1, el vector resultante premultiplicándolo por E_2 y así sucesivamente. Esta operación se denomina transformación hacia delante (FTRAN) porque recorre hacia delante las matrices elementales.

La premultiplicación por un vector se realiza secuencialmente de esta forma

$$c^T B_k^{-1} = ((\cdots(((c^T E_{k-1})E_{k-2})E_{k-3})\cdots)E_2)E_1 \tag{1.33}$$

postmultiplicando en primer lugar el vector c^T por la matriz elemental E_{k-1}, el vector resultante postmultiplicándolo por E_{k-2} y así sucesivamente. Esta operación se denomina transformación hacia atrás (BTRAN) porque recorre hacia atrás las matrices elementales.

Cada una de estas operaciones de producto matriz-vector es muy rápida y simple. Sea E una matriz elemental con un vector η en su columna s. El producto genérico Ea resulta ser

$$Ea = \begin{pmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & \\ & & \eta_s & & \\ & & \vdots & \ddots & \\ & & \eta_m & & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_s \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ 0 \\ \vdots \\ a_m \end{pmatrix} + a_s \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_s \\ \vdots \\ \eta_m \end{pmatrix} \tag{1.34}$$

de manera que ni siquiera la matriz elemental E necesita ser formada explícitamente. Se reemplaza el término s del vector a por 0 y a dicho vector se le añade el vector η multiplicado por el escalar a_s.

El producto genérico $c^T E$ resulta ser

$$c^T E = \begin{pmatrix} c_1 & c_2 & \cdots & c_s & \cdots & c_m \end{pmatrix} \begin{pmatrix} 1 & & & \eta_1 & & \\ & 1 & & \vdots & & \\ & & \ddots & \vdots & & \\ & & & \eta_s & & \\ & & & \vdots & \ddots & \\ & & & \eta_m & & 1 \end{pmatrix} \tag{1.35}$$
$$= \begin{pmatrix} c_1 & c_2 & \cdots & c_{s-1} & c^T\eta & c_{s+1} & \cdots & c_m \end{pmatrix}$$

dejando el vector c sin modificación excepto en el término s que es reemplazado por el escalar $c^T\eta$.

I.5.4.9. Factorización de la matriz base(DOCT)

En cada iteración del método simplex se realiza el cálculo de los costes reducidos de las variables no básicas, del valor de las variables básicas y de la columna pivote en función de la inversa de la base B^{-1}

$$\begin{aligned} w^T &= c_B^T B^{-1} \\ \hat{b} &= B^{-1} b \\ y_t &= B^{-1} a_t \end{aligned} \tag{1.36}$$

Estos cálculos se pueden expresar también como sistemas de ecuaciones en función de la matriz base B

$$\begin{aligned} w^T B &= c_B^T \\ B\hat{b} &= b \\ By_t &= a_t \end{aligned} \tag{1.37}$$

La factorización LU de la matriz base expresa ésta como producto de dos matrices

$$B = LU \tag{1.38}$$

donde L y U son matrices triangular inferior y superior respectivamente. Estas matrices mantienen la condición de ser cuasivacías como la matriz A original.

Por consiguiente, se pueden aplicar técnicas específicas de manejo de dichas matrices.

Con la factorización la resolución de un sistema de ecuaciones se realiza en dos etapas sin necesidad de invertir la matriz. Por ejemplo, para el cálculo del valor de las variables básicas $\hat{b}$

$$\begin{aligned} B\hat{b} &= b \\ LU\hat{b} &= b \end{aligned} \tag{1.39}$$

Si denominamos $\hat{b}_1 = U\hat{b}$ se resuelve primeramente este sistema triangular *hacia delante* para calcular $\hat{b}_1$

$$L\hat{b}_1 = b \tag{1.40}$$

y después se resuelve este otro sistema triangular *hacia atrás* para calcular $\hat{b}$

$$U\hat{b} = \hat{b}_1 \tag{1.41}$$

En cada iteración del método simplex se actualiza la matriz base B mediante una matriz elemental F

$$\overline{B} = BF \tag{1.42}$$

siendo

$$F = \begin{pmatrix} 1 & & & & \\ & \ddots & \left[\begin{matrix} \\ \\ y_t \\ \\ \\ \end{matrix}\right] & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

obtenida a partir de la matriz identidad reemplazando la columna s por y_t.

También se pueden calcular las nuevas matrices $\overline{L}$ y $\overline{U}$.

$$\overline{B} = \overline{L}\,\overline{U} \tag{1.43}$$

Luego, en una iteración cualquiera, éstas se pueden obtener como producto de matrices elementales y de las matrices originales de la factorización de manera análoga a como se hace en la forma producto de la inversa.

Alternativamente a la factorización LU se puede utilizar la descomposición de Cholesky

$$B = R^T R \tag{1.44}$$

donde R es una matriz triangular superior.

I.5.4.10. Estrategias de cálculo de costes reducidos(DOCT)

Para aplicar el criterio de optimalidad en cada iteración es necesario calcular los costes reducidos de las variables no básicas.

$$\hat{c}_N^T \equiv c_N^T - c_B^T B^{-1} N$$

Hasta ahora se suponía que se calculaban los costes reducidos de todas las variables no básicas y se seleccionaba como variable básica entrante a aquélla con coste reducido de menor valor negativo. Existen otras estrategias de cálculo de costes reducidos y de selección de la variable básica entrante (denominadas *pricing strategies* en inglés).

La primera gran división es calcular los costes reducidos de todas las variables no básicas (*full pricing*) o sólo de algunas (*partial pricing*). En el primer caso, se hace mayor esfuerzo computacional en cada iteración y, en principio, menor número de iteraciones del simplex. En el segundo, lo contrario.

Entre las alternativas para la selección de la variable básica entrante t están:

- La de coste reducido de menor valor negativo $\hat{c}_j$
- La primera con coste reducido negativo $\hat{c}_j$
- La de mayor decremento de la función objetivo calculado como $\hat{c}_j x_j$, siendo $\hat{c}_j$ el coste reducido de la variable no básica y $x_j = \min_{1 \le i \le m} \left\{ \frac{\hat{b}_i}{y_{ij}} : y_{ij} > 0 \right\}$
- La de mayor decremento de la función objetivo por incremento unitario de la variable no básica calculado como $\frac{\hat{c}_j y_{ij}}{\left\| y_{ij} \right\| + 1}$. Esta estrategia es la más utilizada habitualmente, en inglés se denomina *steepest-edge*.

I.5.5. Dualidad

Supongamos un problema de programación lineal, denominado *problema primal* (P), expresado en forma estándar como

$$\begin{aligned} &\min z = c^T x \\ &Ax = b \qquad (P) \\ &x \ge 0 \end{aligned} \tag{1.45}$$

Supóngase que se quiere calcular una cota inferior de la función objetivo de este problema antes de resolverlo. Una forma de hacerlo sería operar con las filas de la matriz, multiplicándolas por algún valor $y^T A x = y^T b$ y sumándolas entre ellas de modo que no se superen los coeficientes de la función objetivo

$y^T A \leq c^T$, en tal caso el lado derecho resultante de tales operaciones sería una cota inferior para el valor de la función objetivo $y^T b \leq c^T x$. Desde luego desearemos que esa cota sea lo más alta posible, ya que se podría asegurar que la función objetivo no puede ser menor que ese valor.

Por lo tanto, el problema de encontrar esa cota inferior, sería el problema de encontrar coeficientes y_1 a y_m para las filas de la matriz tales que al multiplicarlos por las filas y sumarlos no superen los coeficientes de la función objetivo, es decir, tales que

$$A^T y \leq c$$

y tal que el resultado obtenido en el lado derecho al operar sea máximo. Este nuevo problema planteado a raíz del anterior se denomina *problema dual* (D)

$$\begin{aligned} \max y_0 &= b^T y \\ A^T y &\leq c \end{aligned} \quad (D) \tag{1.46}$$

siendo $y \in \mathbb{R}^m$ el vector de variables del problema dual o *variable duales*.

Cualquier problema de programación lineal denominado *primal* tiene asociado a él otro problema también de programación lineal llamado *dual*. Tal como se observa en la tabla siguiente, si el problema primal es de maximización con m restricciones y n variables, el problema dual es de minimización con n restricciones y m variables. Se elige este caso particular de optimización para facilitar la presentación de la dualidad pero son generalizables a cualquier problema de optimización lineal.

Primal $(m \times n)$	Dual $(n \times m)$
$\max_x z = c^T x$ $Ax \leq b$ $x \geq 0$	$\min_y y_0 = b^T y$ $A^T y \geq c$ $y \geq 0$
$\max_{x_j} z = \sum_{j=1}^{n} c_j x_j$ $\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, \ldots, m$ $x_j \geq 0 \quad j = 1, \ldots, n$	$\min_{y_i} y_0 = \sum_{i=1}^{m} b_i y_i$ $\sum_{i=1}^{m} a_{ij} y_i \geq c_j \quad j = 1, \ldots, n$ $y_i \geq 0 \quad i = 1, \ldots, m$

Para el ejemplo habitual el problema primal y dual son los siguientes

Primal	Dual

$\begin{array}{llcc} \multicolumn{4}{l}{\max z = 3x_1 + 5x_2} \\ x_1 & & \le & 4 \\ & 2x_2 & \le & 12 \\ 3x_1 & +2x_2 & \le & 18 \\ x_1, & x_2 & \ge & 0 \end{array}$	$\begin{array}{lllcc} \multicolumn{5}{l}{\min y_0 = 4y_1 + 12y_2 + 18y_3} \\ y_1 & & +3y_3 & \ge & 3 \\ & 2y_2 & +2y_3 & \ge & 5 \\ y_1, & y_2, & y_3 & \ge & 0 \end{array}$
$\max z = \begin{bmatrix} 3 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\begin{bmatrix} 1 & \cdot \\ \cdot & 2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$ $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \ge \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\min y_0 = \begin{bmatrix} 4 & 12 & 18 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ $\begin{bmatrix} 1 & \cdot & 3 \\ \cdot & 2 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \ge \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ $\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \ge \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

A continuación se muestra la tabla completa (vista de izquierda a derecha o viceversa según sea el problema primal) que permite la transformación de un problema primal cualesquiera en su dual y viceversa.

min		max
Variable		**Restricción**
≥ 0	$\Leftrightarrow$	$\le$
≤ 0	$\Leftrightarrow$	$\ge$
no restringida	$\Leftrightarrow$	$=$
Restricción		**Variable**
$\ge$	$\Leftrightarrow$	≥ 0
$\le$	$\Leftrightarrow$	≤ 0
$=$	$\Leftrightarrow$	no restringida

Tabla 2.6 Tabla de conversión entre primal y dual.

Como caso ejemplo general para observar la conversión entre primal y dual se presenta el siguiente problema.

$$
\begin{array}{ll}
\min\limits_{x_1,x_2,x_3} c_1x_1 + c_2x_2 + c_3x_3 & \max\limits_{y_1,y_2,y_3} y_1b_1 + y_2b_2 + y_3b_3 \\
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leq b_1 & y_1a_{11} + y_2a_{21} + y_3a_{31} \leq c_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \geq b_2 & y_1a_{12} + y_2a_{22} + y_3a_{32} \geq c_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 & y_1a_{13} + y_2a_{23} + y_3a_{33} = c_3 \\
x_1 \geq 0, x_2 \leq 0, x_3 \text{ libre} & y_1 \leq 0, y_2 \geq 0, y_3 \text{ libre}
\end{array}
$$

Existe un conjunto de propiedades básicas de dualidad que ligan las soluciones del primal y dual.

Teorema: El dual del problema dual es el problema primal.

Propiedad de la simetría

Para cualquier problema primal y su dual, todas las relaciones entre ellos son simétricas porque el dual de su dual es el primal.

Implícitamente las propiedades que siguen se muestran suponiendo que el *primal* es de *maximización* y el *dual* de *minimización*, pero son completamente generalizables.

Propiedad débil de la dualidad

Si x es una solución factible del primal e y es una solución factible del dual, se verifica que $c^T x \leq b^T y$.

Propiedad fuerte de la dualidad

Si $\hat{x}$ es una solución óptima del primal e $\hat{y}$ es una solución óptima del dual, se verifica que $c^T \hat{x} = b^T \hat{y}$.

Propiedad de soluciones básicas complementarias

En cada iteración del método simplex se encuentra una solución básica (vértice) factible del primal y una solución básica complementaria infactible del dual tal que $c^T x = b^T y$ o $z = y_0$. Si x no es óptima para el primal y no es factible para el dual.

Si la solución básica del primal está asociada a la base B, la *solución dual complementaria* tiene la expresión $y^T \equiv c_B^T B^{-1}$. Nótese que son los coeficientes que aparecen en la expresión de los costes reducidos del problema primal (denotados entonces por w^T) y aparecen, cambiados de signo, en la tabla del método simplex bajo las variables básicas iniciales x'_B. Luego, las variables duales son los coeficientes de las variables básicas iniciales en la función objetivo cambiados de signo.

Variables básicas	z	x'_N	x'_B	Cotas
$-z$	-1	$c'^T_N - c^T_B B^{-1} N'$	$-c^T_B B^{-1}$	$-c^T_B B^{-1} b$
x_B	0	$B^{-1} N'$	B^{-1}	$B^{-1} b$

Tabla 2.7 Tabla (B) del simplex en una iteración cualquiera.

A partir de esta expresión la comprobación de la propiedad enunciada es inmediata y se explican las relaciones de transformación de un problema a otro.

Propiedad de solución básica óptima complementaria

En la iteración final se encuentra simultáneamente la solución básica óptima $\hat{x}$ del primal y la básica óptima complementaria $\hat{y}$ del dual y se verifica $c^T \hat{x} = b^T \hat{y}$ o $\hat{z} = \hat{y}_0$. Es decir, en el óptimo del problema primal la solución dual complementaria es también óptima para el problema dual.

Además, esta solución dual puede ser fácilmente identificada en la forma tabular del simplex, corresponde a los coeficientes de las variables básicas iniciales (variables de holgura y artificiales) en la función objetivo. Para las variables de holgura, la solución dual complementaria puede ser vista en los costes reducidos de estas variables, simplemente cambiando el signo. Si la variable fuera de exceso es todavía más directo, ya que no habría que cambiar el signo. Sólo en el caso de las restricciones donde no se hayan introducido variables de holgura o exceso, es decir, restricciones de igualdad (o si se hubieran eliminado las variables artificiales tras la fase I) no es posible ver el valor de las variables duales asociadas a esas restricciones.

Variables básicas	z	x'_N	x'_B	Cotas
$-z$	-1	–variables duales de exceso y de holgura	–variables duales originales	$-c^T_B B^{-1} b$
x_B	0	$B^{-1} N'$	B^{-1}	$B^{-1} b$

Propiedad de la complementariedad de holguras

Si una solución óptima del primal tiene una variable de holgura o exceso > 0 en una restricción (es decir, es variable básica > 0), la variable dual asociada a esa restricción tiene valor 0 y, de la misma forma, si una variable original del primal (no las de holgura, exceso o artificiales) en el óptimo tiene valor > 0, su correspondiente restricción del dual no tiene holgura (es decir, su variable asociada del dual es 0). Por el contrario, si una variable de holgura o exceso del primal es 0 en el óptimo, la variable dual asociada a dicha restricción tiene valor distinto de 0 (positivo o negativo según corresponda al tipo de restricción y de función objetivo –maximización o minimización–) en el óptimo del problema dual.

Supongamos, los problemas primal y dual ya mencionados

$$\begin{aligned} &\min z = c^T x \\ &Ax = b \qquad (P) \\ &x \geq 0 \end{aligned}$$

$$\left.\begin{aligned} &\max y_0 = b^T y \\ &A^T y \leq c \end{aligned}\right. \quad (D)$$

Introducimos variables de holgura en las restricciones del problema dual

$$\begin{aligned} &\max y_0 = b^T y \\ &A^T y + s = c \\ &s \geq 0 \end{aligned}$$

Alternativamente, la propiedad de la complementariedad de holguras para las soluciones óptimas se puede expresar como

$$x_j s_j = 0 \quad j = 1,\ldots,n \tag{1.47}$$

Esta propiedad impide que los valores de x_j y de s_j en las soluciones óptimas de primal y dual sean simultáneamente diferentes de 0. Por otra parte, permite que ambos sean 0.

Sean dos puntos factibles del problema primal y dual, es decir, que cumplen

$$\begin{aligned} Ax = b \qquad & x \geq 0 \\ A^T y + s = c \quad & s \geq 0 \end{aligned} \tag{1.48}$$

Se define el intervalo de dualidad (*duality gap*) como la diferencia

$$c^T x - b^T y = x^T s = \sum_{j=1}^{n} x_j s_j \tag{1.49}$$

Para las respectivas soluciones óptimas del problema primal y dual este intervalo de dualidad es 0 según la propiedad de la complementariedad de holguras.

Teorema de dualidad

Las únicas relaciones posibles entre primal y dual son:

- Si un problema tiene soluciones óptimas factibles entonces también las tiene el otro. Se pueden aplicar las propiedades débil y fuerte de dualidad.
- Si un problema tiene soluciones factibles y función objetivo no acotada, el otro problema no tiene soluciones factibles.
- Si un problema no tiene soluciones factibles, el otro o no tiene soluciones factibles o no tiene función objetivo acotada.

Otra forma de expresar estas mismas relaciones entre primal y dual es el lema de Farkas. Sea el problema en su forma estándar

$$\begin{aligned} &\min c^T x \\ &Ax = b \\ &x \geq 0 \end{aligned} \tag{1.50}$$

Entonces exactamente uno de los dos sistemas de ecuaciones siguientes tiene solución.

- Sistema 1: $A^T x = b$ y $x \geq 0$ para algún $x \in \mathbb{R}^n$
- Sistema 2: $A^T y \leq 0$ y $b^T y > 0$ para algún $y \in \mathbb{R}^m$

Veamos estas propiedades en el caso ejemplo.

$$\begin{array}{lllll} \max\ z = 3x_1 + 5x_2 & & & & \\ x_1 & & \leq 4 & & \\ & 2x_2 & \leq 12 & (P) & \\ 3x_1 & +2x_2 & \leq 18 & & \\ x_1, & x_2 & \geq 0 & & \end{array}$$

El problema dual de éste es

$$\begin{array}{lllll} \min y_0 = 4y_1 + 12y_2 + 18y_3 & & & & \\ y_1 & & +3y_3 & \geq 3 & \\ & 2y_2 & +2y_3 & \geq 5 & (D) \\ y_1, & y_2, & y_3 & \geq 0 & \end{array}$$

Sea el problema primal anterior en su forma estándar y la tabla inicial correspondiente

$$\begin{array}{llllll}
\min\ z = -3x_1 - 5x_2 & & & & & \\
x_1 & & +x_3 & & & = 4 \\
 & 2x_2 & & +x_4 & & = 12 \\
3x_1 & +2x_2 & & & +x_5 & = 18 \\
x_1, & x_2, & x_3, & x_4, & x_5, & \geq 0
\end{array}$$

Variables básicas	z	x_1	x_2	x_3	x_4	x_5	**Cotas**
$-z$	–1	–3	–5				0
x_3		1		1			4
x_4			2		1		12
x_5		3	2			1	18

El problema dual en forma estándar, aunque sin añadir variables artificiales porque no lo queremos resolver por el método simplex tal como se ha explicado, y la tabla inicial correspondiente son

$$\begin{array}{llllll}
\min y_0 = 4y_1 + 12y_2 + 18y_3 & & & & & \\
y_1 & & +3y_3 & -y_4 & & = 3 \\
 & 2y_2 & +2y_3 & & -y_5 & = 5 \\
y_1, & y_2, & y_3, & y_4, & y_5 & \geq 0
\end{array}$$

Variables básicas	y_0	y_4	y_5	y_1	y_2	y_3	**Cotas**
$-y_0$	–1			4	12	18	0
$-y_4$		–1		1		3	3
$-y_5$			–1		2	2	5

La *solución básica factible* inicial para el problema primal es $(x_1, x_2, x_3, x_4, x_5) = (0,0,4,12,18)$, siendo $x_B^T = \begin{bmatrix} x_3 & x_4 & x_5 \end{bmatrix}$ y $x_N^T = \begin{bmatrix} x_1 & x_2 \end{bmatrix}$, y la función objetivo $\hat{z} = 0$. Para el problema dual la *solución básica infactible* correspondiente es $(y_1, y_2, y_3, y_4, y_5) = (0,0,0,-3,-5)$, siendo $y_B^T = \begin{bmatrix} y_4 & y_5 \end{bmatrix}$ y $y_N^T = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}$, con función objetivo $\hat{y}_0 = 0$.

Los valores de las variables del problema dual (variables duales) corresponden a los coeficientes de las variables del problema primal en la función objetivo. Las variables originales del problema dual en las columnas correspondientes a las variables básicas iniciales del problema primal y las variables de exceso del problema dual en las columnas de las variables no básicas iniciales del primal. En cualquier iteración, excepto la última, del método simplex aplicado al problema primal algún coste reducido de las variables no básicas toma valor negativo luego existe una variable dual negativa, luego el problema dual para esa solución básica es infactible.

En el óptimo del problema primal se tienen la siguiente tabla

Variables básicas	z	x_1	x_2	x_3	x_4	x_5	**Cotas**
$-z$	–1				3/2	1	36
x_3				1	1/3	–1/3	2
x_2			1		1/2		6
x_1		1			–1/3	1/3	2

La solución básica óptima del primal es $(x_1, x_2, x_3, x_4, x_5) = (2, 6, 2, 0, 0)$ y $\hat{z} = -36$ para el de minimización, $\hat{z} = 36$ para el original de maximización. Ésta es la solución óptima porque los costes reducidos de las variables no básicas $\hat{c}_N^T = \begin{bmatrix} 1 & 3/2 \end{bmatrix}$ son no negativos. Las variables duales de este problema de minimización son $(y_1, y_2, y_3) = (0, -3/2, -1)$, los costes reducidos de las variables básicas originales cambiados de signo. Las variables duales del problema original de maximización son las opuestas $(y_1, y_2, y_3) = (0, 3/2, 1)$.

Para el problema dual la tabla final es la siguiente

Variables básicas	y_0	y_4	y_5	y_1	y_2	y_3	**Cotas**
$-y_0$	–1	2	6	2			–36
y_3		–1/3		1/3		1	1
y_2		1/3	–1/2	–1/3	1		3/2

La solución básica óptima del dual es $(y_1, y_2, y_3, y_4, y_5) = (0, 3/2, 1, 0, 0)$ e $\hat{y}_0 = 36$. Ésta es la solución óptima porque los costes reducidos de las variables no básicas $\hat{c}_N^T = \begin{bmatrix} 2 & 6 & 2 \end{bmatrix}$ son no negativos. Las variables duales de este problema son $(x_1, x_2) = (2, 6)$, corresponden a los costes reducidos de las variables de exceso y_4 e y_5 sin cambiar de signo.

En el óptimo del problema primal la primera restricción no es activa, por ser su variable de holgura x_3 estrictamente positiva, mientras que las dos últimas restricciones lo son por ser nulas las variables de holgura correspondientes, x_4 y x_5. Efectivamente, en el problema dual la variable y_1 asociada a la primera restricción toma valor 0, mientras que las variables y_2 y y_3 correspondientes a la segunda y tercera toman valor no nulo.

En cuanto a la interpretación de las variables duales, éstas tienen un claro sentido *económico* en el propio problema primal. Obsérvese que dadas las relaciones existentes entre ambos problemas se ha dicho que los valores óptimos de ambos problemas son iguales, es decir, suponiendo un problema primal de maximización se tiene que

$$\max c^T x = \min b^T y$$

o lo que es lo mismo,

$$\max \sum_j c_j x_j = \min \sum_i b_i y_i$$

Por lo tanto, si se incrementa una unidad la disponibilidad de un recurso b_i el óptimo de la función objetivo se verá incrementado en y_i unidades, es decir, las variables duales representan el valor marginal de cada recurso, también llamado *precio sombra* o *precio justo*, ya que representa la cantidad máxima que se podría llegar a pagar por una unidad más de recurso.

Las variables básicas son estrictamente positivas. Si alguna de las variables de holgura o exceso es básica entonces la variable dual de dicha restricción es 0 (propiedad de la complementariedad de holguras). Económicamente es evidente, ya que si hay holgura de un recurso su precio marginal es cero, no se está dispuesto a pagar por disponer de una unidad más.

Hay que tener precaución con esta interpretación pues todo tiene un límite, es decir, esto será cierto mientras no cambie el valor de las variables duales, lo que se puede asegurar mientras la base actual sea óptima (ya que como hemos visto las variables duales tienen la expresión $y^T = c_B^T B^{-1}$, luego no dependen de b). De ahí que una interpretación completa de una solución requiera también el análisis de los límites en que puede darse esta interpretación. El cálculo de estos límites corresponden a lo que se denomina análisis de sensibilidad que se presenta en la siguiente sección.

Veamos a continuación un ejemplo muy sencillo de planificación de la operación de sistemas de energía eléctrica escrito en GAMS. El problema primal es la minimización de los costes variables de explotación de un sistema para una hora, denominado problema de despacho económico. El problema dual corresponde a la maximización de la función de utilidad de consumidores y generadores.

Se trata de un sistema eléctrico con tres generadores cuya potencia máxima es de 200, 200 y 100 MW y con costes variables de 30, 50 y 70 €/MWh. La demanda a cubrir es de 300 MW.

$$\begin{array}{llllll} \min\limits_{P_1,P_2,P_3} & c_1P_1 + c_2P_2 + c_3P_3 \\ P_1 & +P_2 & +P_3 & \geq d & :\lambda \\ P_1 & & & \leq \overline{P}_1 & :\mu_1 \\ & P_2 & & \leq \overline{P}_2 & :\mu_2 \\ & & P_3 & \leq \overline{P}_3 & :\mu_3 \\ P_1, & P_2, & P_3 & \geq 0 \end{array}$$

$$\begin{array}{llllll} \max\limits_{\lambda,\mu_1,\mu_2,\mu_3} & d\lambda + \overline{P}_1\mu_1 + \overline{P}_2\mu_2 + \overline{P}_3\mu_3 \\ \lambda & +\mu_1 & & & \leq c_1 & :P_1 \\ \lambda & & +\mu_2 & & \leq c_2 & :P_2 \\ \lambda & & & +\mu_3 & \leq c_3 & :P_3 \\ \lambda \geq 0 & \mu_1, & \mu_2, & \mu_3 & \leq 0 \end{array}$$

```
set i grupos de generación / g1 * g3 /

parameter
   cv(i)   coste variable  / g1  30, g2  50, g3  70 /
   ptmx(i) potencia máxima / g1 200, g2 200, g3 100 /

scalar dem demanda         / 300 /

positive variables P(i) potencia producida
variable coste

equations
fo         coste variable total
demanda    cobertura de la demanda
prodmax(i) limitación de la potencia producida ;

fo         .. coste =E= sum[i, cv(i) * P(i)] ;
demanda    .. sum[i, P(i)] =G= dem ;
prodmax(i) .. P(i) =L= ptmx(i) ;

model opera / all /

solve opera using lp minimizing coste
```

```
Solution Report     SOLVE opera Using LP From line 23


               S O L V E     S U M M A R Y

     MODEL   opera               OBJECTIVE  coste
     TYPE    LP                  DIRECTION  MINIMIZE
     SOLVER  CPLEX               FROM LINE  23

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE            11000.0000

 RESOURCE USAGE, LIMIT          0.530      1000.000
 ITERATION COUNT, LIMIT         0         10000

GAMS/Cplex    May 15, 2003 WIN.CP.CP 21.0 023.025.041.VIS For Cplex 8.1
Cplex 8.1.0, GAMS Link 23

Optimal solution found.
Objective :       11000.000000


                           LOWER          LEVEL          UPPER         MARGINAL

---- EQU fo                  .              .              .             1.0000
---- EQU demanda           300.0000       300.0000        +INF           50.0000

  fo  coste variable total
  demanda  cobertura de la demanda

---- EQU prodmax  limitación de la potencia producida

          LOWER          LEVEL          UPPER         MARGINAL

g1        -INF          200.0000       200.0000       -20.0000
g2        -INF          100.0000       200.0000         .
g3        -INF            .            100.0000         .

---- VAR P  potencia producida

          LOWER          LEVEL          UPPER         MARGINAL

g1          .           200.0000       +INF             .
```

```
g2          :           100.0000        +INF          .
g3          :              .            +INF        20.0000

                           LOWER          LEVEL          UPPER         MARGINAL

---- VAR coste             -INF       11000.0000        +INF             .
```

```
set i grupos de generación / g1 * g3 /

parameter
   cv(i)    coste variable  / g1  30, g2  50, g3  70 /
   ptmx(i) potencia máxima / g1 200, g2 200, g3 100 /

scalar dem demanda          / 300 /

positive variable  LA    precio marginal
negative variables MU(i) precio marginal
variable benef

equations
fo          remuneración total
prgrupo(i) limitación del pago al grupo ;

fo          .. benef =E= dem * LA + sum[i, ptmx(i) * MU(i)] ;
prgrupo(i) .. LA + MU(i) =L= cv(i) ;

model opera / all /

solve opera using lp maximizing benef
```

```
Solution Report     SOLVE opera Using LP From line 22


               S O L V E     S U M M A R Y

     MODEL   opera               OBJECTIVE  benef
     TYPE    LP                  DIRECTION  MAXIMIZE
     SOLVER  CPLEX               FROM LINE  22

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE            11000.0000

 RESOURCE USAGE, LIMIT          0.030      1000.000
 ITERATION COUNT, LIMIT         2         10000

GAMS/Cplex    May 15, 2003 WIN.CP.CP 21.0 023.025.041.VIS For Cplex 8.1
Cplex 8.1.0, GAMS Link 23

Optimal solution found.
Objective :       11000.000000


                           LOWER          LEVEL          UPPER         MARGINAL

---- EQU fo                  .              .              .             1.0000

  fo  remuneración total

---- EQU prgrupo  limitación del pago al grupo

          LOWER          LEVEL          UPPER         MARGINAL

g1        -INF          30.0000        30.0000       200.0000
g2        -INF          50.0000        50.0000       100.0000
g3        -INF          50.0000        70.0000         .

                           LOWER          LEVEL          UPPER         MARGINAL

---- VAR LA                  .            50.0000        +INF             .

  LA  precio marginal

---- VAR MU  precio marginal

          LOWER          LEVEL          UPPER         MARGINAL

g1        -INF         -20.0000          .             .
g2        -INF            .              .           100.0000
g3        -INF            .              .           100.0000

                           LOWER          LEVEL          UPPER         MARGINAL

---- VAR benef             -INF       11000.0000        +INF             .
```

El coste variable total resultante es de 11000 €. El grupo 1 produce sus 200 MW y el grupo 2 produce 100 MW y es el marginal. Para este problema, la variable dual de la ecuación de la demanda indica el incremento en costes variables del sistema por incremento marginal de la demanda. Tiene un valor de 50 €/MWh, lo que quiere decir que si la demanda se incrementara en 1 MW el coste variable total se incrementaría en 50 €. Corresponde al coste variable del grupo marginal.

Una aplicación directa de la dualidad se plantea a la hora de resolver un problema ya que, como se ha visto, resolviendo uno u otro se obtiene la misma solución óptima. Entonces, dado un problema primal cualquiera y sabiendo que se puede obtener su dual y resolverlo, ¿cuál de los dos se debe resolver? Sabiendo que el tiempo de resolución crece proporcionalmente a m^3, si $m \ll n$ entonces se resolverá el problema primal y si $m \gg n$ se resolverá el problema dual en estas etapas: (i) se pasa del primal al dual, (ii) se resuelve el dual y (iii) se pasa de nuevo del dual al primal.

I.5.6. Análisis de sensibilidad

El análisis de sensibilidad o de *postoptimalidad* estudia los efectos sobre la solución óptima debidos a cambios en cualquier parámetro del problema de optimización lineal A, b o c partiendo de la solución óptima ya alcanzada.

Observando la tabla 2.8, tabla del simplex en cualquier iteración, se advierte que las columnas de la tabla bajo las variables básicas iniciales x'_B no dependen de los parámetros del problema original. Entonces, si éstos cambian ($b \to \overline{b}$, $A \to \overline{A}$ o $c'_N \to \overline{c}'_N$) se deberán recalcular para los nuevos valores solamente los bloques de la tabla que se alteran y continuar a partir de ahí con el método simplex.

Variables Básicas	z	x'_N	x'_B	**Valores**
$-z$	-1	$\overline{c}_N'^T - c_B^T B^{-1}\overline{N}'$	$-c_B^T B^{-1}$	$-c_B^T B^{-1}\overline{b}$
x_B	0	$B^{-1}\overline{N}'$	B^{-1}	$B^{-1}\overline{b}$

Tabla 2.8 Tabla del simplex (B) en una iteración cualquiera con cambios en parámetros.

También se pueden determinar los cambios de manera incremental. Si ($b \to b + \Delta b$, $A \to A + \Delta A$ y $c'_N \to c'_N + \Delta c'_N$) la tabla presenta ahora este aspecto

Variables Básicas	z	x'_N	x'_B	**Valores**
$-z$		$\Delta c'^T_N - c^T_B B^{-1}\Delta N'$		$-c^T_B B^{-1}\Delta b$
x_B		$B^{-1}\Delta N'$		$B^{-1}\Delta b$

Tabla 2.9 Cambios incrementales en la tabla del simplex (B) en una iteración cualquiera con cambios en parámetros.

Luego las modificaciones de los bloques correspondientes se pueden escribir como $\Delta\hat{c}^T_N = \Delta c'^T_N - c^T_B B^{-1}\Delta N'$, $\Delta\hat{z} = w^T\Delta b$, $\Delta\hat{b} = B^{-1}\Delta b$

El procedimiento genérico completo de análisis de sensibilidad tiene estas etapas:

1. introducción de los cambios en los bloques correspondientes de la tabla del simplex
2. preparación para adecuarla a una iteración del simplex (función objetivo con coeficientes nulos en variables básicas y matriz identidad en las columnas de dichas variables)
3. prueba de factibilidad (todas las variables básicas no negativas)
4. prueba de optimalidad (costes reducidos de las variables no básicas no negativos en un problema de minimización)
5. nueva iteración del método simplex o del método simplex dual (éste se ve en un siguiente apartado)

En el ejemplo habitual se ha hecho mediante GAMS análisis de sensibilidad con respecto a las cotas de las restricciones y a los coeficientes de la función objetivo y da estos resultados.

$$
\begin{array}{lll}
\max\ z = 3x_1 + 5x_2 & & \\
x_1 & & \le 4 \\
 & 2x_2 & \le 12 \\
3x_1 & +2x_2 & \le 18 \\
x_1, & x_2 & \ge 0
\end{array}
$$

```
positive variables x1, x2
variable z

equations fo, r1, r2, r3 ;

fo .. z =e= 3 * x1 + 5 * x2 ;
r1 .. x1 =l= 4 ;
r2 .. 2 * x2 =l= 12 ;
r3 .. 3 * x1 + 2 * x2 =l= 18 ;

model caso / all / ;

caso.optfile= 1

solve caso maximazing z using lp
```

```
Solution Report     SOLVE caso Using LP From line 15

               S O L V E     S U M M A R Y

     MODEL   caso                OBJECTIVE  z
     TYPE    LP                  DIRECTION  MAXIMIZE
     SOLVER  CPLEX               FROM LINE  15

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE               36.0000

 RESOURCE USAGE, LIMIT          0.010      1000.000
 ITERATION COUNT, LIMIT         0         10000

GAMS/Cplex    May 15, 2003 WIN.CP.CP 21.0 023.025.041.VIS For Cplex 8.1
Cplex 8.1.0, GAMS Link 23


User supplied options:
objrng all
rhsrng all

Optimal solution found.
Objective :          36.000000


EQUATION NAME                               LOWER      CURRENT        UPPER
-------------                               -----      -------        -----
fo                                           -INF            0         +INF
r1                                              2            4         +INF
r2                                              6           12           18
r3                                             12           18           24


VARIABLE NAME                               LOWER      CURRENT        UPPER
-------------                               -----      -------        -----
x1                                             -3            0          4.5
x2                                             -3            0         +INF
z                                               0            1         +INF



                       LOWER     LEVEL     UPPER    MARGINAL

---- EQU fo              .         .         .       1.000
---- EQU r1            -INF      2.000     4.000      .
---- EQU r2            -INF     12.000    12.000     1.500
---- EQU r3            -INF     18.000    18.000     1.000

                       LOWER     LEVEL     UPPER    MARGINAL

---- VAR x1              .       2.000     +INF       .
---- VAR x2              .       6.000     +INF       .
---- VAR z             -INF     36.000     +INF       .
```

A continuación se analizan específicamente algunos cambios: en las cotas de las restricciones, en un coeficiente de una variable en las restricciones o en la función objetivo, introducción de una nueva variable o restricción.

I.5.6.1. Cambios en cotas de restricciones

Observando la tabla 2.8 se ve que los cambios en las cotas de las restricciones sólo afectan al valor de la función objetivo y a los valores de las variables básicas. Como éstos se pueden hacer negativos se puede perder factibilidad. En este caso, se aplica el método simplex dual que se verá más adelante.

Geométricamente, los cambios en las cotas de las restricciones suponen desplazar paralelas a sí mismas las restricciones. Dado que el punto óptimo

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

viene dado por las restricciones que son activas, al realizar este desplazamiento también se desplaza el punto óptimo. El nuevo punto puede seguir dentro de la región factible, en cuyo caso sigue siendo el óptimo o quedar fuera, con lo que hay que aplicar el método denominado simplex dual para volver a la región factible, es decir, recuperar la factibilidad.

Veamos un ejemplo de cambio de la cota en la segunda restricción del problema. Las cotas pasan de ser $b = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$ a tomar el valor $\overline{b} = \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix}$. La inversa de la matriz base en la iteración final era $B^{-1} = \begin{bmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{bmatrix}$ y las variables duales $y^T = c_B^T B^{-1} = \begin{bmatrix} \cdot & -3/2 & -1 \end{bmatrix}$.

El nuevo valor de la función objetivo será $\hat{z} = c_B^T B^{-1}\overline{b} = y^T\overline{b} = \begin{bmatrix} \cdot & -3/2 & -1 \end{bmatrix} \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix} = -54$ y el de las variables básicas $x_B = \hat{b} = B^{-1}\overline{b} = \begin{bmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{bmatrix} \begin{bmatrix} 4 \\ 24 \\ 18 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \\ -2 \end{bmatrix}$, que resulta ser infactible por tomar un valor negativo.

Estos cambios se pueden realizar también de forma incremental

$$\hat{z} = \hat{z} + \Delta\hat{z} = \hat{z} + y^T \Delta b = -36 + \begin{bmatrix} \cdot & -3/2 & -1 \end{bmatrix} \begin{bmatrix} \cdot \\ 12 \\ \cdot \end{bmatrix} = -36 - 18 = -54$$

$$x_B = x_B + \Delta x_B = x_B + B^{-1}\Delta b = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & 1/3 & -1/3 \\ \cdot & 1/2 & \cdot \\ \cdot & -1/3 & 1/3 \end{bmatrix} \begin{bmatrix} \cdot \\ 12 \\ \cdot \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix} + \begin{bmatrix} 4 \\ 6 \\ -4 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \\ -2 \end{bmatrix}$$

Es interesante determinar también el intervalo de valores para los que la solución básica óptima se mantiene factible cambiando únicamente el valor de b_i en el problema.

Continuando con el ejemplo de incremento de la cota de la segunda restricción $\hat{b} = \hat{b} + B^{-1}\Delta b = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1/2 \\ -1/3 \end{bmatrix} \Delta b_2 \geq 0$. Por tanto, $\begin{Bmatrix} 2 + \Delta b_2 / 3 \geq 0 \\ 6 + \Delta b_2 / 2 \geq 0 \\ 2 - \Delta b_2 / 3 \geq 0 \end{Bmatrix}$. Es decir, $\Delta b_2 \geq -6$, $\Delta b_2 \geq -12$ y $\Delta b_2 \leq 6$. Luego, $-6 \leq \Delta b_2 \leq 6$ da el intervalo de modificación de la cota de la segunda restricción sin que se pierda factibilidad con la misma solución básica factible.

I.5.6.2. Cambio en un coeficiente de una variable no básica

Se analiza el cambio en los coeficientes de una variable no básica en la iteración final o bien en la función objetivo $\hat{c}_N^T = \overline{c}_N'^T - c_B^T B^{-1} \overline{N}'$ o bien en las restricciones $B^{-1}\overline{N}'$. Estos cambios no afectan a la factibilidad de la solución pero pueden afectar a la optimalidad. Geométricamente, el cambio de un coeficiente de una variable en la función objetivo, sea básica o no, supone un cambio de dirección de optimización, pero no afecta a la región factible. Así pues, lo único que puede ocurrir es que el punto óptimo actual deje de serlo y haya que iterar para encontrar el nuevo óptimo.

Si $\hat{c}_N^T = \overline{c}_N'^T - c_B^T B^{-1} \overline{N}' \geq 0$ la solución sigue siendo óptima. Se puede determinar el intervalo de valores del coeficiente de la variable no básica en la función objetivo (cambiando solamente éste) para los que se mantiene la optimalidad. En caso contrario la solución no es óptima, ésta será la nueva variable básica entrante y se debe continuar con el método simplex.

De acuerdo con esta expresión el *coste reducido* se puede interpretar como la cantidad mínima en que tiene que reducirse el coeficiente de la variable en la función objetivo para que interese realizar esa actividad (incrementar desde 0), o bien como el máximo decremento permisible de dicho coeficiente para mantener la actual solución básica factible como óptima.

I.5.6.3. Introducción de una nueva variable

Es un caso particular de la situación anterior. Una variable que no existe es equivalente a una variable no básica en la solución final (igual a 0) a la que se le cambian los coeficientes de la función objetivo y de las restricciones.

Para introducir una nueva variable se calcula el coste reducido de la variable y si éste es menor que 0 será una variable cuyo valor interese aumentar su valor y por lo tanto hay que introducirla en la tabla. Para ello se calcula $B^{-1}a_j$ y se añade como columna de esa variable en la tabla y su coste reducido en la fila de la función objetivo. A continuación se sigue con las iteraciones del simplex.

I.5.6.4. Cambio en un coeficiente de una variable básica

El cambio en un coeficiente de una variable básica en la iteración final (que era una variable no básica en la iteración inicial) en la función objetivo supone un cambio en todos los costes reducidos de las variables no básicas que han de ser actualizados. Una vez hecho esto, la solución actual puede seguir siendo óptima (con diferente valor de la función objetivo) o no, en cuyo caso hay que seguir iterando con el método simplex (pérdida de optimalidad).

El cambio de algún coeficiente de una variable básica en la matriz de restricciones $\overline{B}^{-1}$ altera la tabla del simplex de manera que se requiere ponerla de nuevo en la forma adecuada para eliminación gaussiana. Esto puede causar pérdida de factibilidad (resuelto mediante método simplex dual) y/o de optimalidad (resuelto mediante método simplex), e incluso las variables básicas actuales pueden dejar de formar una base, requiriendo la introducción de una variable artificial que la sustituya en la base.

I.5.6.5. Introducción de nueva restricción

Al introducir una nueva restricción se comprueba si ésta se satisface para la solución óptima. Si lo hace, la solución sigue siendo óptima. Si no, se introduce la restricción en la tabla final del simplex con la variable de holgura, por exceso o artificial, que se toma como variable básica. Se prepara la tabla para eliminación gaussiana y se sigue con el método simplex dual ya que se ha perdido factibilidad.

Es importante apuntar, que dado que la solución óptima actual no verifica la restricción añadida, al introducirla en la tabla y adaptar ésta para la eliminación gaussiana, necesariamente el valor de la variable básica en esa restricción ha de ser negativo, pues en caso contrario, es que la solución actual sí verificaba la restricción.

I.5.7. Método simplex dual

El método simplex dual es conveniente utilizarlo cuando:

- Se necesitan introducir muchas variables artificiales para construir una solución básica factible inicial
- Se hace infactible la solución óptima tras una perturbación en análisis de sensibilidad

Opera en el problema primal como si el método simplex se aplicara simultáneamente al problema dual. Maneja soluciones básicas que satisfacen la condición de optimalidad del primal (la solución dual complementaria es factible), aunque no sean factibles para el primal, y se mueve hacia la solución óptima para conseguir factibilidad del primal, sin perder nunca la factibilidad dual (optimalidad primal). Es decir, mantiene los coeficientes de la ecuación de la función objetivo mayores o iguales que cero y los de las cotas de las restricciones con alguno estrictamente negativo, excepto en el óptimo en que logra la factibilidad primal.

Este método, por lo tanto, se mueve por fuera de la región factible, moviéndose por vértices exteriores (infactibles) con mejor valor de la función objetivo que el óptimo que luego alcanza y entra en la región factible sólo al alcanzar el óptimo.

A continuación se presenta el procedimiento del método simplex dual

- *Inicialización*
  Se parte de una solución básica que cumpla el criterio de optimalidad (dual factible), es decir, cuyos costes reducidos sean todos mayores o iguales que 0. Si la solución es factible (todas las variables básicas tienen valor ≥ 0), la solución es óptima. En caso contrario, iterar.

- *Iteración*
  - Determinar la *variable básica saliente* (variable básica con valor negativo) que determina la *fila pivote* s.
    Se suele elegir dentro de las negativas la mayor en valor absoluto, aunque esto no garantiza una convergencia más rápida.
  - Determinar la *variable básica entrante*
    Se ha de utilizar un criterio de relación semejante al del método simplex primal para asegurar que se mantiene la factibilidad dual. De entre las variables que tienen coeficientes < 0 en la fila pivote se selecciona aquélla que minimice en valor absoluto la relación entre el coste reducido y su coeficiente en la fila pivote s:

$$\min_{1\leq j\leq n}\left\{\left|\frac{c_j - z_j}{y_{sj}}\right| : y_{sj} < 0\right\}$$

    En el caso de que no haya ningún elemento estrictamente menor que cero en la fila pivote, quiere decir que el problema dual no está acotado y, por lo tanto, el problema primal es infactible.
  - Determinar la nueva solución básica por eliminación gaussiana.

Veamos con un ejemplo el método simplex dual. Sea el siguiente problema de programación lineal

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

$$\begin{aligned} \min 4x_1 + x_2 \\ 3x_1 + x_2 \geq 3 \\ 4x_1 + 3x_2 \geq 6 \\ x_1 + 2x_2 \geq 4 \\ x_1, x_2 \geq 0 \end{aligned}$$

Como las restricciones son de tipo ≥ habría que introducir muchas variables artificiales y emplear el método de las dos fases para solucionarlo. En su lugar, se transforman las restricciones en tipo ≤ y se aplica el método simplex dual.

$$\begin{aligned} \min 4x_1 + x_2 \\ -3x_1 - x_2 \leq -3 \\ -4x_1 - 3x_2 \leq -6 \\ -x_1 - 2x_2 \leq -4 \\ x_1, x_2 \geq 0 \end{aligned}$$

Una vez transformado en la forma estándar se obtiene el siguiente problema y la siguiente tabla del simplex

$$\begin{aligned} \min 4x_1 + x_2 \\ -3x_1 - x_2 + x_3 = -3 \\ -4x_1 - 3x_2 + x_4 = -6 \\ -x_1 - 2x_2 + x_5 = -4 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Variables Básicas	z	x_1	x_2	x_3	x_4	x_5	**Valores**
Relación		–1	–1/3				
$-z$	–1	4	1				
x_3		–3	–1	1			–3
x_4		–4	–3		1		–6
x_5		–1	–2			1	–4

Obsérvese cómo la tabla tiene las características necesarias para utilizar el simplex dual: coeficientes de la función objetivo mayores o iguales que 0 (factibilidad del dual, optimalidad del primal) y alguna cota de las restricciones negativa (no optimalidad del dual, infactibilidad del primal).

Se elige x_4 como variable básica saliente por tener el mayor coeficiente negativo en la función objetivo del dual (mayor valor negativo de cota). Ésta define la fila pivote. Se calcula la relación entre los coeficientes de la función objetivo y los estrictamente negativos de la fila pivote. Se elige como variable básica entrante x_2, aquélla con menor relación en valor absoluto, será la primera que alcance valor 0 al incrementar x_4. Se realizan ahora las transformaciones sobre la tabla para prepararla para la eliminación gaussiana.

Variables Básicas	z	x_1	x_2	x_3	x_4	x_5	**Valores**
Relación		–8/5			–1		
$-z$	–1	8/3			1/3		–2
x_3		–5/3		1	–1/3		–1
x_2		4/3	1		–1/3		2
x_5		5/3			–2/3	1	

Se elige x_3 como variable básica saliente y x_4 como variable básica entrante y se hacen las transformaciones pertinentes.

Variables Básicas		x_1	x_2	x_3	x_4	x_5	**Valores**
$-z$	–1	1		1			–3
x_4		5		–3	1		3
x_2		3	1	–1			3
x_5		5		–2		1	2

Ésta es la solución óptima $(x_1, x_2, x_3, x_4, x_5) = (0, 3, 0, 3, 2)$ y la función objetivo $\hat{z} = 3$. Las variables duales son $(y_1, y_2, y_3) = (-1, 0, 0)$ para las restricciones de $\leq$, para las restricciones originales $\geq$ serán $(y_1, y_2, y_3) = (1, 0, 0)$.

Veamos el problema dual del problema original y la tabla inicial del simplex para este problema

$$\begin{aligned} \max y_0 &= 3y_1 + 6y_2 + 4y_3 \\ 3y_1 + 4y_2 + y_3 &\leq 4 \\ y_1 + 3y_2 + 2y_3 &\leq 1 \\ y_1, y_2, y_3 &\geq 0 \end{aligned}$$

Variables Básicas	y_0	y_1	y_2	y_3	y_4	y_5	**Valores**	**Relación**
$-y_0$	–1	–3	–6	–4				
y_4		3	4	1	1		4	1
y_5		1	3	2		1	1	1/3

Entra la variable y_2 y sale la variable y_5. Reorganizando la matriz de restricciones queda la siguiente tabla.

Variables Básicas	y_0	y_1	y_2	y_3	y_4	y_5	**Valores**	**Relación**

$-y_0$	–1	–1				2	2	
y_4		5/3		–5/3	1	–4/3	8/3	8/5
y_2		1/3	1	2/3		1/3	1/3	1

Entra la variable y_1 y sale la variable y_2. Reorganizando la matriz de restricciones queda la siguiente tabla, que corresponde a la solución óptima.

Variables Básicas	y_0	y_1	y_2	y_3	y_4	y_5	**Valores**
$-y_0$	–1		3	2		3	3
y_4			–5	–5	1	–3	1
y_1		1	3	2		1	1

La solución óptima es $(y_1, y_2, y_3, y_4, y_5) = (1, 0, 0, 1, 0)$ y función objetivo $\hat{y}_0 = 3$ para el problema de maximización. Las variables duales son $(x_1, x_2) = (0, 3)$. Es decir, el método simplex dual es equivalente a aplicar el método simplex al problema dual.

I.5.8. Programación lineal paramétrica

Así como el análisis de sensibilidad determinaba los cambios en la solución óptima del problema original por cambios discretos en los valores de algún parámetro, el análisis paramétrico determina el cambio en la función objetivo en función de cambios continuos en los parámetros. Se presenta a continuación el análisis paramétrico frente a cambios en los coeficientes de la función objetivo y cotas de las restricciones.

I.5.8.1. Cambios simultáneos en coeficientes de la función objetivo

Suponemos la función objetivo $z = \sum_{j=1}^{n} c_j x_j$. Se desea conocer el cambio en la función objetivo óptima $\hat{z}(\theta) = \sum_{j=1}^{n} \left(c_j + \alpha_j \theta\right) x_j$ al cambiar el parámetro θ desde 0, supuestos conocidos los pesos α_j para cada variable x_j.

1. Resolver hasta optimalidad el problema con $\theta = 0$ por el método simplex.
2. Mediante análisis de sensibilidad introducir $\Delta c_j = \alpha_j \theta$.
3. Modificar la ecuación de la función objetivo para hacer 0 los coeficientes de las variables básicas en la función objetivo. Aplicar la condición de optimalidad (coeficientes de variables no básicas ≥ 0) para obtener el valor de θ.
4. Incrementar θ hasta anular un coeficiente de variable no básica. Ésta es la nueva variable básica entrante.

5. Encontrar la nueva solución óptima mediante el algoritmo simplex primal. Ir al paso 3.

Obsérvese que en este tipo de análisis la región factible no cambia, sólo se va modificando la dirección de optimización y para los distintos valores del parámetro θ se va detectando un nuevo punto óptimo. El proceso de análisis lleva a definir zonas del espacio paramétrico, siendo el óptimo en cada una de ellas un punto extremo diferente.

I.5.8.2. Cambios simultáneos en cotas de las restricciones

Se desea conocer el cambio en la función objetivo óptima $\hat{z}$ al cambiar θ desde 0 en las cotas de las restricciones $b_i + \alpha_i \theta$, supuestos conocidos los pesos α_i. Los cambios en las cotas b_i son equivalentes a cambios en coeficientes de la función objetivo del problema dual.

1. Resolver hasta optimalidad el problema con $\theta = 0$ por el método simplex.
2. Mediante análisis de sensibilidad introducir $\Delta b_i = \alpha_i \theta$.
3. Expresar la condición de optimalidad de las variables del dual (cotas de las restricciones ≥ 0).
4. Incrementar θ hasta anular las cotas de las restricciones (la solución se haría infactible). Ésta es la nueva variable básica saliente en el simplex dual.
5. Encontrar la nueva solución óptima mediante el algoritmo simplex dual. Ir al paso 2.

Obsérvese en este caso, que los diferentes valores del parámetro θ determinan diferentes regiones factibles, así el espacio paramétrico es dividido en intervalos de modo que en cada uno de ellos hay una base que es óptima (unas restricciones activas que determinan el punto extremo óptimo). En este caso, el proceso de análisis acaba con un intervalo donde para cualquier valor del parámetro hay una base que es óptima o con un intervalo del parámetro donde el problema resulta no factible.

I.5.9. Método de punto interior primal-dual$^{(\mathrm{DOCT})}$

La investigación en métodos de punto interior aplicados a programación lineal nació a consecuencia del teórico mal comportamiento del método simplex en el peor caso (cuando ha de recorrer todos los vértices del contorno de la región factible).

El método simplex requiere un número de iteraciones proporcional al número de ecuaciones. Sin embargo, el número de iteraciones puede aumentar sustancialmente en problemas degenerados o disminuir drásticamente si se dispone de una base inicial. Cada iteración del simplex es poco costosa. Esencialmente es el cálculo de una actualización de la inversa de la matriz base.

Los métodos de punto interior requieren pocas iteraciones para resolver un problema lineal. Sin embargo, estas iteraciones son muy costosas. Esencialmente cada iteración conlleva la resolución de un sistema de ecuaciones lineales, como se verá más adelante.

En 1979 Khachiyan presentó el *método del elipsoide* y en 1984 Karmarkar describió un *método proyectivo*, reclamando ambos un requerimiento de tiempo polinómico. De hecho, los métodos barrera fueron desarrollados originalmente para programación no lineal en los años 60 por Fiacco y McCormick.

El *método primal-dual*, que se presenta a continuación, es un caso particular de ellos y ha dado muy buenos resultados en la práctica. Se denomina así porque resuelve simultáneamente el problema primal y el dual.

Supongamos un problema lineal en su forma estándar

$$\begin{aligned} \min z &= c^T x \\ Ax &= b \quad (P) \\ x &\geq 0 \end{aligned} \tag{1.51}$$

y su correspondiente problema dual

$$\left.\begin{aligned} \max y_0 &= b^T y \\ A^T y &\leq c \end{aligned}\right. \quad (D) \tag{1.52}$$

siendo $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^m$.

Añadiendo variables de holgura en las restricciones del dual tenemos

$$\begin{aligned} \max y_0 &= b^T y \\ A^T y + s &= c \quad (D) \\ s &\geq 0 \end{aligned} \tag{1.53}$$

Sea $\overline{x}$ una solución factible del problema primal (P) y $(\overline{y}, \overline{s})$ una solución factible del problema dual (D), no necesariamente básicas. Serán óptimos en sus respectivos problemas si cumplen la propiedad de la complementariedad de holguras

$$x_j s_j = 0 \quad j = 1,\ldots,n \tag{1.54}$$

La idea principal del método primal-dual es moverse por una secuencia de soluciones estrictamente factibles[19] en ambos problemas tratando de que se verifiquen las condiciones de complementariedad de holguras. Específicamente, se trata de encontrar $x(\mu)$, $y(\mu)$ y $s(\mu)$ para $\mu > 0$ que satisfagan el siguiente sistema de $(m+n)$ ecuaciones lineales (denominadas condiciones de factibilidad) y n ecuaciones no lineales (denominadas condiciones de complementariedad)

$$\begin{aligned} Ax &= b \\ x &\geq 0 \\ A^T y + s &= c \quad (PD) \\ s &\geq 0 \\ x_j s_j &= \mu \quad \forall j \end{aligned} \tag{1.55}$$

El valor del parámetro μ se va reduciendo hasta lograr la convergencia. Mientras $\mu > 0$ la condición $x_j s_j = \mu$ implica que $x > 0$ y $s > 0$, es decir, que ambos son puntos estrictamente factibles. Este mismo avance se puede ver como una reducción del intervalo de dualidad (*duality gap*) hasta alcanzar el valor cero en la solución óptima

$$c^T x - b^T y = (A^T y + s)^T x - b^T y = y^T A x + s^T x - b^T y = x^T s = n\mu \tag{1.56}$$

El método primal-dual se puede interpretar como una variante del método de Newton de programación no lineal que resuelve iterativamente un sistema de ecuaciones no lineales donde las direcciones de movimiento corresponden a Δx^k, Δy^k y Δs^k. Para ello se pasa de una solución estrictamente factible $x^k > 0$ y $s^k > 0$ en la iteración k a otra en la iteración $k+1$ que mantenga estricta factibilidad (de ahí el nombre de método de punto interior) y disminuya el intervalo de dualidad $x^{k+1} \leftarrow x^k + \Delta x^k$, $y^{k+1} \leftarrow y^k + \Delta y^k$ y $s^{k+1} \leftarrow s^k + \Delta s^k$

$$\begin{aligned} A\left(x^k + \Delta x^k\right) = b \quad &\rightarrow \quad A\Delta x^k = 0 \\ A^T\left(y^k + \Delta y^k\right) + s^k + \Delta s^k = c \quad &\rightarrow \quad A^T \Delta y^k + \Delta s^k = 0 \end{aligned} \tag{1.57}$$

que es un sistema de $(m+n)$ ecuaciones lineales en función de Δx^k, Δy^k e Δs^k.

Además se deben satisfacer las condiciones de complementariedad de holguras para cada dimensión j

[19] Se define una solución como estrictamente factible en el poliedro $Ax = b$, $x \geq 0$ a aquélla que verifica $Ax = b$ y $x > 0$.

$$\begin{aligned} x_j^{k+1}s_j^{k+1} &= (x_j^k + \Delta x_j^k)(s_j^k + \Delta s_j^k) = \mu^{k+1} \\ s_j^k\Delta x_j^k + x_j^k\Delta s_j^k + \Delta x_j^k\Delta s_j^k &= \mu^{k+1} - x_j^k s_j^k \end{aligned} \tag{1.58}$$

que es un sistema de n ecuaciones no lineales en función de Δx_j^k y Δs_j^k. Para obtener un sistema de ecuaciones lineales despreciamos el término $\Delta x_j^k\Delta s_j^k$ ya que se supone que los incrementos de las variables en cada iteración serán pequeños

$$s_j^k\Delta x_j^k + x_j^k\Delta s_j^k = \mu^{k+1} - x_j^k s_j^k \tag{1.59}$$

Representemos la condición de complementariedad de holguras en forma matricial en función de $X = \operatorname{diag}(x)$, matriz diagonal donde x_j ocupa la posición j, $S = \operatorname{diag}(s)$ y $e = \begin{pmatrix}1 & \cdots & 1\end{pmatrix}^T$. Luego $x = Xe$, $s = Se$ y, por consiguiente,

$$XSe = \mu e \tag{1.60}$$

De forma completa, una iteración cualquiera del método primal-dual resuelve este sistema de ecuaciones lineales

$$\begin{aligned} S\Delta x + X\Delta s &= \mu e - XSe \\ A\Delta x &= 0 \\ A^T\Delta y + \Delta s &= 0 \end{aligned} \tag{1.61}$$

Obteniendo Δs en función de Δy de la tercera ecuación y sustituyendo en la primera

$$\begin{aligned} S\Delta x - XA^T\Delta y &= \mu e - XSe \\ A\Delta x &= 0 \end{aligned} \tag{1.62}$$

Premultiplicando por AS^{-1} y aprovechando la segunda ecuación obtenemos este sistema de ecuaciones lineales para determinar Δy

$$-AS^{-1}XA^T\Delta y = AS^{-1}(\mu e - XSe) \tag{1.63}$$

Si definimos la matriz diagonal $D \equiv S^{-1}X$ y $v(\mu) = \mu e - XSe$, los otros vectores solución se pueden calcular secuencialmente de esta manera

$$\begin{aligned} \Delta s &= -A^T\Delta y \\ \Delta x &= S^{-1}v(\mu) - D\Delta s \end{aligned} \tag{1.64}$$

El coste computacional más importante de este método está en la resolución del sistema de ecuaciones lineales para el cálculo de la dirección de movimiento

Δy. En el caso de columnas densas de A la matriz ADA^T tiene muchos más elementos no nulos que la misma matriz A.

Si en el método simplex tanto el manejo de matrices cuasivacías como la factorización de la inversa son fundamentales para una buena implantación en problemas de gran tamaño estas técnicas resultan críticas en el método de punto interior.

El algoritmo completo parte de una solución estrictamente factible en la iteración k para $x^k > 0$, y^k y $s^k > 0$. Se calculan las direcciones de movimiento Δx^k, Δy^k e Δs^k que definen las nuevas soluciones $x^k + \Delta x^k$, $y^k + \Delta y^k$ y $s + \Delta s^k$. Entonces se reduce el valor del parámetro μ^{k+1} de la siguiente iteración $k+1$ y se repite el proceso. El parámetro μ se puede disminuir rápidamente (hasta un orden de magnitud en cada iteración) siempre que se mantenga estricta factibilidad. Para ello se modifica la manera de actualizar las soluciones

$$\begin{aligned} x(\alpha_P, \mu) &= x + \alpha_P \Delta x \\ y(\alpha_D, \mu) &= y + \alpha_D \Delta y \\ s(\alpha_D, \mu) &= s + \alpha_D \Delta s \end{aligned} \tag{1.65}$$

donde $\alpha_P = \min\limits_{\Delta x_j < 0} \left(-x_j / \Delta x_j\right)$ y $\alpha_D = \min\limits_{\Delta s_j < 0} \left(-s_j / \Delta s_j\right)$ son las longitudes del paso que aseguran $x > 0$ y $s > 0$. Una estrategia habitual es tomar una longitud de paso única $\alpha = \min(\alpha_P, \alpha_D)$

$$\begin{aligned} x_j + \alpha \Delta x_j &\geq 0 \\ y_j + \alpha \Delta y_j &\geq 0 \\ s_j + \alpha \Delta s_j &\geq 0 \end{aligned} \tag{1.66}$$

Si además de asegurar estricta factibilidad en la selección de α_P y α_D se satisface la condición de centralidad

$$(x^k, s^k) \in C^k \tag{1.67}$$

siendo $C^k = \left\{(x,s) : x_j s_j \geq \gamma^k \mu^k, j = 1,\ldots,n\right\}$ para $\gamma \in (0,1)$ estamos en la estrategia *path following* del método primal-dual.

La selección de valores adecuados de la longitud de paso α, parámetro de centrado γ e intervalo de dualidad μ en cada iteración son imprescindibles para obtener en la práctica un comportamiento robusto del algoritmo.

La obtención de un punto factible inicial puede ser difícil. Sin embargo, las transformaciones anteriores se pueden modificar para incluir puntos iniciales no factibles.

$$
\begin{aligned}
S\Delta x + X\Delta s &= \mu e - XSe \equiv v(\mu) \\
A\Delta x &= b - Ax \equiv r_P \\
A^T\Delta y + \Delta s &= c - A^T y - s \equiv r_D
\end{aligned}
\tag{1.68}
$$

siendo r_P y r_D precisamente los residuos de las ecuaciones del primal y dual respectivamente. Entonces, el cálculo de los vectores de actualización en el método primal-dual para puntos iniciales infactibles queda de esta manera

$$
\begin{aligned}
\Delta y &= -(ADA^T)^{-1}\left[AS^{-1}v(\mu) - ADr_D - r_P\right] \\
\Delta s &= -A^T\Delta y + r_D \\
\Delta x &= S^{-1}v(\mu) - D\Delta s
\end{aligned}
\tag{1.69}
$$

El método primal-dual *predictivo-correctivo* de Mehrotra está diseñado para no ignorar los términos de segundo orden $\Delta x_j \Delta s_j$ que anteriormente se habían despreciado. Esto se realiza en dos etapas. En la *etapa predictiva* o *afín* se predicen (calculan) los valores de Δx y de Δs junto con un valor del parámetro μ relacionado con los valores de x y de s según el sistema de ecuaciones anterior. En la *etapa correctiva* con paso de Newton (i.e., segunda derivada) se utilizan estos valores para resolver el siguiente sistema de ecuaciones lineales

$$
\begin{aligned}
S\Delta x + X\Delta s &= \mu e - XSe - \Delta X\Delta Se \equiv v(\mu) \\
A\Delta x &= b - Ax \equiv r_P \\
A^T\Delta y + \Delta s &= c - A^T y - s \equiv r_D
\end{aligned}
\tag{1.70}
$$

donde $\Delta X = \text{diag}(\Delta x_j)$ e $\Delta S = \text{diag}(\Delta s_j)$ son los valores obtenidos en la etapa predictiva.

Los métodos de punto interior no producen soluciones básicas óptimas. Por ello cuando llegan al óptimo se suele cambiar (*crossover*) al método simplex para determinar la solución básica óptima.

Veamos a continuación la resolución de este problema por el método primal-dual.

$$
\begin{aligned}
\max\ z = 3x_1 + 5x_2 & \\
x_1 \qquad\qquad & \le 4 \\
2x_2 \quad & \le 12 \\
3x_1 \quad +2x_2 \quad & \le 18 \\
x_1, x_2 & \ge 0
\end{aligned}
$$

Lo convertimos en un problema de minimización con restricciones de igualdad

$$
\begin{array}{lllllll}
 & & & & & & \min -3x_1 - 5x_2 \\
x_1 & & +x_3 & & & & = 4 \\
 & 2x_2 & & +x_4 & & & = 12 \\
3x_1 & +2x_2 & & & +x_5 & & = 18 \\
 & & & & & & x_1, x_2, x_3, x_4, x_5 \geq 0
\end{array}
$$

Su problema dual será el siguiente

$$
\begin{array}{llll}
\max 4y_1 + 12y_2 + 18y_3 \\
y_1 & & +3y_3 & \leq -3 \\
 & 2y_2 & +2y_3 & \leq -5 \\
y_1 & & & \leq 0 \\
 & y_2 & & \leq 0 \\
 & & y_3 & \leq 0
\end{array}
$$

y pasándolo a la forma estándar

$$
\begin{array}{lllllllll}
 & & & & & & & & \max 4y_1 + 12y_2 + 18y_3 \\
y_1 & & +3y_3 & +s_1 & & & & & = -3 \\
 & 2y_2 & +2y_3 & & +s_2 & & & & = -5 \\
y_1 & & & & & +s_3 & & & = 0 \\
 & y_2 & & & & & +s_4 & & = 0 \\
 & & y_3 & & & & & +s_5 & = 0 \\
 & & & & & & & & s_1, s_2, s_3, s_4, s_5 \geq 0
\end{array}
$$

I.5.10. Referencias

Bazaraa, M.S. and Jarvis, J.J. (1998) *Programación Lineal y Flujo en Redes.* Limusa.

Bertsimas. D. and Tsitsiklis. J.N. (1997) *Introduction to Linear Optimization.* Athena Scientific.

Chvátal. V. (1983) *Linear Programming.* W.H. Freeman and Co.

Dantzig, G. and Thapa, M. (1997) *Linear Programming. Introduction.* Springer.

Gill, P.E., Murray, W. and Wright, M.H. (1991) *Numerical Linear Algebra and Optimization.* Addison Wesley.

Nash. S.G. and Sofer. A. (1996) *Linear and Nonlinear Programming.* McGraw-Hill.

Ríos Insúa, S. (1996) *Investigación operativa. Programación lineal y aplicaciones* Ed. Centro de estudios Ramón Areces.

Ríos Insúa, S., Ríos Insúa, D., Mateos, A. y Martín, J. (1997) *Programación lineal y aplicaciones. Ejercicios resueltos* Ed. Ra-Ma.

Rivier, M. (1998) *Modelo probabilista de explotación de un sistema eléctrico: contribución a la teoría marginalista.* Tesis doctoral. Universidad Pontificia Comillas.

Vanderbei. R.J. (1996) *Linear Programming: Foundations and Extensions.* Kluwer Academic Pusblishers.

I.5.11. Biblioteca de problemas

PROBLEMA 1

Dado el problema de programación lineal:

$$\begin{aligned}
&\min x_1 + x_2 - 3x_3 \\
&3x_1 - x_3 = 5 \\
&x_2 - x_3 = 1 \\
&x_1, x_2, x_3 \geq 0
\end{aligned}$$

Se pide:

a) Dar todas las soluciones básicas factibles.

b) Comparar la función objetivo para todas las soluciones básicas factibles y decir cuál es la mejor de todas ellas.

c) ¿Es el vector $x = (1335, 4001, 4000)$ una solución para el problema anterior? ¿Es mejor solución que la obtenida en el apartado anterior?

d) Comprobar que la base $B = \begin{pmatrix} 3 & \cdot \\ \cdot & 1 \end{pmatrix}$ está asociada a una solución no acotada.

PROBLEMA 2

Resolver los siguientes problemas de programación lineal, utilizando el método de penalizaciones y el método de las dos fases, cuando sea necesario.

1. Discutir el problema geométricamente

$$\begin{aligned}&\max 3x + 2y\\&4x + y \leq 16\\&x + 4y \leq 16\\&5x + 6y \leq 30\\&x, y \geq 0\end{aligned}$$

2. Discutir el problema geométricamente

$$\begin{aligned}&\max 3y + 8z\\&y + 2z = 4\\&z \geq 3\\&y, z \geq 0\end{aligned}$$

3. Discutir el problema geométricamente

$$\begin{aligned}&\max 3x + 2y\\&5x + y \geq 0\\&y \geq x \geq 0\end{aligned}$$

4.

$$\begin{aligned}&\min 3x + 5y - 4z + 6t\\&x + 2z - t = 6\\&y + 4z + t = 9\\&2z - 2t = 3\\&x, y, z, t \geq 0\end{aligned}$$

5.

$$\begin{aligned}&\min - x_1 + 2x_2 - 3x_3\\&x_1 + x_2 + x_3 = 6\\&-x_1 + x_2 + 2x_3 = 4\\&2x_2 + 3x_3 = 10\\&x_3 \leq 2\\&x_1, x_2, x_3 \geq 0\end{aligned}$$

PROBLEMA 3

Dados los siguientes problemas de programación lineal, plantear los correspondientes problemas duales:

1.

$$\min 3x_1 + 2x_2 - x_3$$
$$x_1 + x_2 - x_3 \geq 5$$
$$-x_1 - x_2 + 7x_3 \geq 4$$
$$x_1, x_2 \geq 0$$
$$x_3 \text{ libre}$$

2.

$$\max 3x_1 + 4x_2$$
$$x_1 - x_2 \geq 7$$
$$x_1 + x_2 \geq 3$$
$$x_1, x_2 \text{ libres}$$

3.

$$\max x_1 + x_2 + x_3$$
$$x_1 - x_2 \geq 5$$
$$x_1 + x_3 = 8$$
$$x_1, x_2 \geq 0$$

Problema 4

Dados los siguientes problemas

$$\text{(i)} \quad \begin{aligned} &\min x_1 + x_2 + x_3 + x_4 \\ &2x_1 - x_2 + x_3 = 4 \\ &x_1 + 2x_3 - x_4 = 3 \\ &x_1, x_2, x_3 \geq 0 \end{aligned}$$

$$\text{(ii)} \quad \begin{aligned} &\max x_1 + 2x_2 + 3x_3 \\ &x_1 - x_2 + 2x_3 \geq 5 \\ &x_1 - x_3 = 3 \\ &x_1, x_2, x_3 \geq 0 \end{aligned}$$

Se pide:

a) Resolverlos mediante el algoritmo primal.

b) Plantear el dual y resolverlo.

c) Comprobar la relación existente entre ambas soluciones.

Problema 5

Dado el problema de programación lineal

$$\begin{aligned}&\min x_1 + x_2 + x_3 + x_4\\&2x_1 - x_2 + x_3 = 4\\&x_1 + 2x_3 - x_4 = 3\\&x_1, x_2, x_3 \geq 0\end{aligned}$$

Se pide: (cada apartado es independiente de los demás)

a) La solución óptima del problema si se añade la restricción $x_1 + x_2 - x_3 = 2$

b) Idem si se añade la restricción $x_1 + x_2 - x_3 = 3$

c) Idem al cambiar el vector $b = \begin{pmatrix}4\\3\end{pmatrix}$ por $\overline{b} = \begin{pmatrix}3\\1\end{pmatrix}$

d) Añadir x_5 con coste $c_5 = -3$ y $a_5 = \begin{pmatrix}1\\1\end{pmatrix}$

PROBLEMA 6

Dado el problema

$$\begin{aligned}&\min c_1x_1 + c_2x_2 + c_3x_3\\&x_1 + 2x_2 - x_3 \leq 5\\&x_1 + 2x_3 \leq 8\\&x_1, x_2, x_3 \geq 0\end{aligned}$$

donde

$$\begin{pmatrix}c_1\\c_2\\c_3\end{pmatrix} = \begin{pmatrix}3\\-2\\-1\end{pmatrix} + \begin{pmatrix}-3\\2\\1\end{pmatrix}\theta \qquad \theta \in [0, \infty]$$

PROBLEMA 7

Dado el problema de programación lineal

$$\begin{aligned}&\min 3x_1 + 2x_2 + 4x_3\\&3x_1 + 2x_2 - x_3 \leq 5\\&x_1 + x_3 = 3\\&x_1, x_2, x_3 \geq 0\end{aligned}$$

Se pide:

a) La solución óptima.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

b) Resolver el problema paramétrico asociado si el término independiente es $b^T(\theta) = (5,3)^T + (4,1)^T \theta$, $\theta \in [0, \infty)$

PROBLEMA 8

Un ama de casa, típico ejemplo de la economía sumergida, hace en sus ratos domésticos libres dos clases de salsa de tomate que vende en jarras de barro al supermercado de al lado. La primera requiere utilizar 3 kg de tomates y 4 tazas de vinagre por jarra de salsa. La segunda requiere 5 kg de tomates y 2 tazas de vinagre. La primera salsa le produce un beneficio neto de 0.4 € por jarra y la segunda 0.5.

El supermercado que evacúa su producción casera hacia los circuitos comerciales (no sabemos con qué beneficio relativo) le impone a nuestra amiga las siguientes condiciones:

- Que produzca como mínimo 3 jarras a la semana.
- Que le compre como máximo 24 kg de tomate y 3 botellas de vinagre a la semana.

Sabiendo que una botella de vinagre equivale a 16 tazas y que el supermercado monopoliza la venta de tomate y vinagre en la región, analizar los precios a los que estaría dispuesta a pagar el tomate y el vinagre nuestra ama de casa a un primo suyo contrabandista, que se los puede proporcionar recurriendo a otro de los variados mecanismos de la economía subterránea.

PROBLEMA 9

Dado el siguiente problema de programación lineal

$$\begin{aligned}
&\max 2x_1 - x_2 - 17x_3 \\
&5x_1 - 2x_2 + 31x_3 \leq -1 \\
&3x_1 - x_2 + 19x_3 \leq 0 \\
&x_1, x_2, x_3 \geq 0
\end{aligned}$$

Se pide:

a) Resolver por el algoritmo dual.
b) Resolver por el algoritmo primal.
c) Obtener, usando postoptimización, la solución para $\overline{b}_1 = 1$.

PROBLEMA 10

Dado el siguiente problema de programación lineal:

$$\max 2x_1 + x_2 + 0.5x_3$$
$$2x_1 + x_2 + 2x_3 \leq 2$$
$$x_1 - x_2 - x_3 \geq 1$$
$$x_1, x_2, x_3 \geq 0$$

a) Resolver por el algoritmo primal.
b) Identificar razonadamente, en la tabla óptima anterior, el valor de las variables duales.
c) Verificar las condiciones de holgura complementaria.
d) Hacer una iteración más del algoritmo primal, ¿qué ocurre?

PROBLEMA 11

Dado el siguiente problema y su correspondiente tabla óptima

$$\max 4x + 2y$$
$$x + y \leq 5$$
$$2x + y \leq 7$$
$$x \leq 3$$
$$x, y \geq 0$$

	x	y	s_1	s_2	s_3	
z	0	0	0	–2	0	–14
s_1	0	0	1	–1	1	1
y	0	1	0	1	–2	1
x	1	0	0	0	1	3

Se pide:

a) Resolverlo gráficamente.
b) ¿La solución es única? Si no es así, dar todas las posibles soluciones.
c) Plantear el problema dual.
d) Dar los precios sombra de los recursos.
e) Comprobar las relaciones existentes entre las soluciones primal y dual.
f) Obtener la nueva solución óptima, sin resolver todo el problema de nuevo, si se modifica la primera componente del vector b, cambiando de $b_1 = 5$ a $\overline{b}_1 = 4$.
g) Obtener la solución óptima, sin resolver todo el problema de nuevo, si se añade la restricción $5x + 3y \leq 15$.
h) Si se parametriza el vector de coeficientes de la función objetivo en la dirección (1,2), es decir, $c = (4,2) + (1,2)\theta$, $\theta \geq 0$, ¿para qué valores del parámetro la solución actual sigue siendo óptima?

PROBLEMA 12

Dado el siguiente problema lineal

$$\max z = 2x_1 + 2x_2$$
$$x_1 + x_2 \leq 3$$
$$x_1 + 2x_2 \leq 4$$
$$x_1, x_2 \geq 0$$

Se pide:

a) Resolverlo gráficamente.

b) Resolverlo aplicando el método simplex. Razonar la aplicación y el resultado.

c) ¿Qué valores del coeficiente de x_2 en la función objetivo mantienen la base óptima?

d) ¿Cuál es la solución si el lado derecho de la segunda restricción pasa a ser 10? Resolverlo mediante postoptimización.

e) Obtener los precios duales de los dos recursos. ¿Cuál es su significado? ¿cuál es el sentido de su signo?

f) Determinar, mediante postoptimización, la solución óptima si añadimos al problema original la restricción $2x_1 + x_2 \leq 2$.

PROBLEMA 13

En una fábrica se producen dos tipos de artículos. La semana siguiente se estima que han de producirse al menos 50 artículos en total. Para producirlos se usan dos tipos de materia prima, cuyo coste unitario es de 100 y 200 €, respectivamente. El problema que se plantea para maximizar los beneficios de la empresa y su tabla óptima son los siguientes:

$$\max 200x + 400y$$
$$x + y \geq 50$$
$$x + 2y \leq 80$$
$$x + y \leq 60$$
$$x, y \geq 0$$

	x	y	s_1	s_2	s_3	
z	0	0	0	–200	0	–16000
x	1	0	–2	–1	0	20
y	0	1	1	1	0	30
s_3	0	0	1	0	1	10

a) Resolver el problema gráficamente.
b) Hallar todas las posibles soluciones analíticamente.
c) Interpretar todos los elementos del problema y las soluciones.
d) Dar los precios sombra e interpretarlos.
e) Hallar mediante postoptimización la nueva solución si se dispone de 120 unidades de la primera materia en lugar de 80.
f) Estudiar mediante postoptimización qué ocurre si la producción del segundo producto se limita a 10 unidades.

PROBLEMA 14

Dado el siguiente problema de programación lineal:

$$\max 2x + y$$
$$x + y \geq 2$$
$$x + 4y \leq 16$$
$$3x + y \leq 15$$
$$x, y \geq 0$$

a) Resolver el problema mediante el algoritmo del simplex.
b) Obtener la nueva solución mediante postoptimización al añadir la restricción $x + y \leq 4$.

PROBLEMA 15

En una fábrica se producen dos productos, P1 y P2, cuyos beneficios unitarios son 20 y 70, respectivamente. En la producción se utilizan dos materias primas, disponiendo de 30 y 60 unidades cada una. La demanda total es de al menos 20 productos y ha de ser satisfecha. El planteamiento del problema es:

$$\max 20x + 70y$$
$$x + 3y \leq 30$$
$$2x + 4y \leq 60$$
$$x + y \geq 20$$
$$x, y \geq 0$$

a) Resolver mediante el algoritmo del simplex.
b) Dar los valores e interpretación de las variables originales, las de holgura, las artificiales (si existen), las duales, de los costes reducidos y de la función objetivo, si la tabla óptima fuera

	x	y	s_1	s_2	s_3	
z	0	0	–25	0	–5	–650
y	0	1	1/2	0	1/2	5
s_2	0	0	–1	1	1	10
x	1	0	–1/2	0	3/2	15

c) Surge ahora en el mercado la posibilidad de fabricar un tercer producto, P3. Este producto requiere 1 unidad de la primera materia prima (de la que se dispone de 30) y 1 unidad de la segunda, siendo su beneficio unitario de 60. La demanda total sería de 20 unidades igualmente. Sin embargo, este producto requiere del alquiler de un tipo de maquinaria cuyo coste es de 100 unidades monetarias, independientemente del tiempo que sea usada y de la cantidad que se produzca. Además, si se produce alguna cantidad de este producto P3 han de producirse al menos 8 unidades del producto P2. Plantear un problema de programación lineal entera para maximizar el beneficio de la empresa.

PROBLEMA 16

Dado el siguiente problema de programación lineal

$$\min 3x_1 + 5x_2 + 7x_3$$
$$x_1 + x_2 + x_3 \geq 300$$
$$x_1 \leq 200$$
$$x_2 \leq 200$$
$$x_3 \leq 100$$
$$x_1, x_2, x_3 \geq 0$$

Se pide:

a) Resolver por el algoritmo primal.

b) Resolver el problema dual por el algoritmo primal.

PROBLEMA 17

Dado el siguiente problema de programación lineal

$$\min 4x_1 + x_2$$
$$3x_1 + x_2 \geq 3$$
$$4x_1 + 3x_2 \geq 6$$
$$x_1 + 2x_2 \geq 4$$
$$x_1, x_2 \geq 0$$

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Se pide:
a) Resolver por el algoritmo dual.
b) Resolver el problema dual por el algoritmo primal.

PROBLEMA 18

Convertir el siguiente problema a uno de programación lineal en forma estándar

$$\min |x| + |y| + |z|$$
$$x + y \leq 1$$
$$2x + z = 3$$

PROBLEMA 19

Un cierto tipo de funciones poligonales se puede representar como

$$f(x) = \max\left\{c_1^T x + d_1, c_2^T x + d_2, \ldots, c_p^T x + d_p\right\}$$

donde $x, c_1, c_2, \ldots, c_p \in \mathbb{R}^n$ y $d_1, d_2, \ldots, d_p \in \mathbb{R}$

Para tal función se considera el problema

$$\min f(x)$$
$$Ax = b$$
$$x = 0$$

Mostrar cómo se puede convertir este problema en uno de programación lineal.

PROBLEMA 20

Un fabricante desea producir 90 toneladas de una aleación de modo que el porcentaje del metal A esté entre el 60 y el 70%. En el mercado hay cinco aleaciones, cuya composición y precio se indican a continuación, a partir de las cuales el metalúrgico desea conseguir al precio más barato la aleación deseada.

Aleación	Al1	Al2	Al3	Al4	Al5
%A	10	25	50	75	90
%otros	90	75	50	25	10
Precio/tm (€)	25	40	65	55	30

A tal efecto, el metalúrgico modela la situación con el siguiente problema de programación lineal

$$\min 25x_1 + 40x_2 + 65x_3 + 55x_4 + 30x_5$$
$$x_1 + x_2 + x_3 + x_4 + x_5 = 90$$
$$0.1x_1 + 0.25x_2 + 0.5x_3 + 0.75x_4 + 0.9x_5 \leq 0.7(x_1 + x_2 + x_3 + x_4 + x_5)$$
$$0.1x_1 + 0.25x_2 + 0.5x_3 + 0.75x_4 + 0.9x_5 \geq 0.6(x_1 + x_2 + x_3 + x_4 + x_5)$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

que tras agrupar términos en cada restricción, añadir dos variables de holgura (s_2 y s_3 en la segunda y tercera restricción, respectivamente) y resolver (añadiendo una variable artificial en la primera restricción, a_1), proporciona la siguiente tabla óptima

	x_1	x_2	x_3	x_4	x_5	s_2	s_3	a_1	Cotas
$c_j - z_j$	0	14.0625	37.5	25.9375	0	0	6.25	–28.125	–2531.25
x_5	0	0.1875	0.5	0.8125	1	0	–1.25	0.625	56.25
s_2	0	0	0	0	0	1	1	0.1	9
x_1	1	0.8125	0.5	0.1875	0	0	1.25	0.375	33.75

A la vista de los resultados, se pide (cada apartado es independiente de los demás):

1. Determinar mediante postoptimización las cantidades óptimas y el nuevo valor de la función objetivo si se tuvieran que producir 100 toneladas. A la vista de lo obtenido ¿qué conclusión puedes sacar?
2. Si el metalúrgico pudiera comprar una tonelada de aleación en el mercado con los porcentajes deseados del metal A para satisfacer su demanda sin tener que producirlo él, ¿a qué precio máximo la pagaría? ¿por qué? ¿hasta qué cantidad?
3. ¿Qué precios máximos tendrían que tener las aleaciones no utilizadas para que interesara incluirlas en la aleación final?
4. ¿Qué ocurriría si en el mercado aparece una nueva aleación con un 40 % del metal A y a un precio por tonelada de 30 €?
5. ¿Qué ocurriría si en el mercado sólo hay disponible 30 toneladas de la aleación Al1?

PROBLEMA 21

Un artesano fabrica trenes y camiones de juguete muy sencillos a base de tornillos, bloques de plástico y ruedas. Para la próxima semana dispone de 8000, 6000 y 6300 unidades de las citadas componentes. La adquisición de estas componentes y los gastos semanales de operación y amortización del local y utillaje ascienden a 40 €.

La estructura de producto de trenes y camiones se refleja en la siguiente tabla y los beneficios unitarios que reportan son de 1.6 y 1.4 € respectivamente.

	Tornillos	Bloques	Ruedas
Tren	10	15	18
Camión	20	10	6

El objetivo del artesano es el de determinar qué gama de producción le generará la mayor cantidad de ingresos esta semana. Para ello modela la situación con el siguiente modelo lineal

$$\begin{aligned}
&\max z = 1.6x_1 + 1.4x_2 + 0x_3 + 0x_4 + 0x_5 \\
&10x_1 + 20x_2 + x_3 = 8000 \\
&15x_1 + 10x_2 + x_4 = 6000 \\
&18x_1 + 6x_2 + x_5 = 6300 \\
&x_1, x_2, x_3, x_4, x_5 \geq 0
\end{aligned}$$

del que se adjunta la última tabla del método simplex de la que se deduce la solución óptima:

	x_1	x_2	x_3	x_4	x_5	Cotas
$c_j - z_j$	0	0	–0.025	–0.09	0	–740
x_5	0	0	0.45	–1.5	1	900
x_2	0	1	0.075	–0.05	0	300
x_1	1	0	–0.05	0.1	0	200

Situémonos sucesivamente e independientemente en los siguientes supuestos y tomemos las decisiones pertinentes utilizando, cuando corresponda, el análisis de postoptimalidad (o sensibilidad):

1. El beneficio unitario de los trenes baja a 1 euro y el de camiones aumenta a 2.2 €.
2. Por otro lado existe la posibilidad de adquirir 200 tornillos con un precio unitario incrementado sobre el precio normal de adquisición 0.05 €. ¿Merece la pena adquirirlos? ¿Se adquirirían los 200 tornillos si el citado incremento fuera tan sólo de 0.02 €? ¿Por qué?
3. Parece que una de las razones por las que los juguetes han perdido atractivo es que son un poco pobres de ornamentación. Cabe la posibilidad de añadirles algunas ventanillas como adornos: los trenes necesitan 7 ventanillas por unidad y los camiones 4. El artesano puede adquirir para la próxima

semana un máximo de 2100 ventanillas. ¿Cómo afecta esta incidencia a la producción óptima de juguetes?

4. Otro enterado en estos asuntos le sugiere la posibilidad de ampliar la gama de juguetes con la fabricación de aviones. Un avión necesita 8 tornillos, 12 bloques y 3 ruedas y proporciona un beneficio unitario de 1.1 €. ¿Interesa fabricar el nuevo juguete?

I.5.12. Resultados de la biblioteca de problemas

RESULTADO DEL PROBLEMA 1

a) Sólo una: $B = (a_1, a_2)$, solución (5/3,1,0)
b) Valor de la función objetivo 8/3
c) Sí, es solución factible (verifica las restricciones) con función objetivo 1335+4001–12000=–6664, luego es mejor solución.
d) $B^{-1} = \begin{pmatrix} 1/3 & \cdot \\ \cdot & 1 \end{pmatrix}$, $y_3 = B^{-1}a_3 = \begin{pmatrix} -1/3 \\ -1 \end{pmatrix}$ es un vector con todos sus componentes < 0 y el coste reducido de x_3 es menor que 0, $\hat{c}_3 = -5/3$.

RESULTADO DEL PROBLEMA 2

1. $(x, y) = (66/19, 40/19)$
2. No hay solución factible.
3. Solución no acotada, dirección (1,1) siendo $x = y$.
4. $(x, y, z, t) = (12/5, 0, 21/10, 3/5)$
5. $(x_1, x_2, x_3) = (2, 2, 2)$

RESULTADO DEL PROBLEMA 3

1.

$$\begin{aligned} &\max 5w_1 + 4w_2 \\ &w_1 \text{ - } w_2 \le 3 \\ &w_1 \text{ - } w_2 \le 2 \\ &\text{-}w_1 + 7w_2 = \text{-}1 \\ &w_1, w_2 \ge 0 \end{aligned}$$

2.

$$\begin{aligned} &\min 7w_1 + 3w_2 \\ &w_1 + w_2 = 3 \\ &\text{-}w_1 + w_2 = 4 \\ &w_1, w_2 \le 0 \end{aligned}$$

3.

$$\min 5w_1 + 8w_2$$
$$w_1 + w_2 \geq 1$$
$$-w_1 \geq 1$$
$$w_2 = 1$$
$$w_1 \leq 0$$

RESULTADO DEL PROBLEMA 4

i.a) $(x_1, x_2, x_3, x_4) = (2, 0, 0, -1)$, $z = 1$

i.b) $(w_1, w_2) = (1, -1)$, $z = 1$

i.c) Se cumplen las condiciones de holgura complementaria.

ii.a) Solución no acotada, dirección (1/3,1,1/3).

ii.b) Dual no factible (observar geométricamente).

ii.c) Obvias.

RESULTADO DEL PROBLEMA 5

a) La solución verifica la restricción, sigue siendo óptima.

b) $(x_1, x_2, x_3, x_4) = (7/3, 2/3, 0, -2/3)$

c) $(x_1, x_2, x_3, x_4) = (3/2, 0, 0, 1/2)$

d) Solución no acotada.

RESULTADO DEL PROBLEMA 6

$0 \leq \theta \leq 1$ $(x_1, x_2, x_3) = (0, 9/2, 4)$ $z = -13 + 13\theta$

$\theta \geq 1$ $(x_1, x_2, x_3) = (6, 0, 1)$ $z = 17 - 17\theta$

RESULTADO DEL PROBLEMA 7

a) $(x_1, x_2, x_3) = (2, 0, 1)$

b) $0 \leq \theta \leq 4$ $(x_1, x_2, x_3) = (2 + 5/4\theta, 0, 1 - 1/4\theta)$ $\theta \geq 4$ $(x_1, x_2, x_3) = (3 + \theta, 0, 0)$

RESULTADO DEL PROBLEMA 8

$$\max 0.4x_1 + 0.5x_2$$
$$x_1 + x_2 \geq 3 \quad :y_1$$
$$3x_1 + 5x_2 \leq 24 \quad :y_2$$
$$4x_1 + 2x_2 \leq 3 \cdot 16 \quad :y_3$$
$$x_1, x_2 \geq 0$$

Por kg de tomate $y_2 = 0.4/3$ €/kg e $y_3 = 0$ €/taza por el vinagre.

Resultado del problema 9

a) $x = (0, 0.5, 0)$

b) $x = (0, 0.5, 0)$

c) $x = (0, 0, 0)$

Resultado del problema 10

a) $x_1 = 1$, $x_2 = 0$, $x_3 = 0$

b) $w_1 = 1$, $w_2 = 0$

c) Comprobar que se verifican, evidente con estas soluciones.

d) Al introducir la segunda de holgura, cambia la base pero no la solución.

Resultado del problema 11

a) No. Las soluciones son $\lambda(2{,}3)+(1\text{-}\lambda)(3{,}1)$.

b)

c)

$$\begin{aligned}&\min 5w_1 + 7w_2 + 3w_3\\&w_1 + 2w_2 + w_3 \geq 4\\&w_1 + w_2 \geq 2\\&w_1, w_2, w_3 \geq 0\end{aligned}$$

d) $(w_1, w_2, w_3) = (0, 2, 0)$

e) Mismo valor de la función objetivo y condiciones de holgura complementaria.

f) $(x, y) = (3{,}1)$, sólo cambia la holgura que ahora es 0.

g) $(x, y) = (3, 0)$

h) Tan sólo para $\theta = 0$.

Resultado del problema 12

b) Múltiples soluciones, $\lambda(2{,}1) + (1 - \lambda)(3, 0) \quad \lambda \in [0{,}1] \quad z = 6$.

c) La solución (2,1) sigue siendo óptima para valores entre 2 y 4, y la solución (3,0) sigue siendo óptima para valores menores que 2.

d) Múltiples soluciones, $\lambda(0, 3) + (1 - \lambda)(3, 0) \quad \lambda \in [0{,}1] \quad z = 6$.

e) $w_1 = 2$, $w_2 = 0$

f) $(x_1, x_2) = (0, 2)$

Resultado del problema 13

b) Múltiples soluciones $(x, y) = \lambda(20, 30) + (1 - \lambda)(40, 20) \quad \lambda \in [0{,}1] \quad z = 16000$

d) $(w_1, w_2, w_3) = (0, 200, 0)$

e) $(x, y) = (0, 60)$, $z = 24000$

f) $(x, y) = (50{,}10)$, $z = 14000$

RESULTADO DEL PROBLEMA 14

a) $(x,y)=(4,3)\,,\ z=11$

b) $(x,y)=(4,0)\,,\ z=8$

RESULTADO DEL PROBLEMA 20

1.

$$B^{-1}\begin{pmatrix}100\\0\\0\end{pmatrix}=\begin{pmatrix}0.625&0&-1.25\\0.1&1&1\\0.375&0&1.25\end{pmatrix}\begin{pmatrix}100\\0\\0\end{pmatrix}=\begin{pmatrix}62.5\\10\\37.5\end{pmatrix}$$

Se puede concluir que son los porcentajes de la Al1 (37.5) y de la Al5 (62.5) sea cuál sea la cantidad a producir.

2. La pagaría a lo sumo a 28.125 que es el valor de la primera variable dual, y para cualquier cantidad hasta la demanda (es decir, hasta las 90 toneladas) ya que nunca se hará negativa ninguna variable básica.
3. Son sus costes menos los costes reducidos, es decir,
   Al2: 40–14.0625=25.9375, Al3: 65–37.5=27.5, Al4:55–25.9375=29.0625
4.

$$B^{-1}\begin{pmatrix}1\\0.4-0.7\\0.6-0.4\end{pmatrix}=\begin{pmatrix}0.375\\0\\0.625\end{pmatrix},\quad 30-\begin{pmatrix}30&0&25\end{pmatrix}\begin{pmatrix}0.375\\0\\0.625\end{pmatrix}=3.125$$

Nada, no interesa producir con la nueva.

5. Habría que añadir la restricción: $x_1 \le 30$, quedando (quitando la artificial):

	x_1	x_2	x_3	x_4	x_5	s_2	s_3	s_4	Cotas
$c_j - z_j$	0	14.0625	37.5	25.9375	0	0	6.25	0	– 2531.25
x_5	0	0.1875	0.5	0.8125	1	0	–1.25	0	56.25
s_2	0	0	0	0	0	1	1	0	9
x_1	1	0.8125	0.5	0.1875	0	0	1.25	0	33.75
s_4	1	0	0	0	0	0	0	1	30

Reorganizando (a la última restar la penúltima solamente):

	x_1	x_2	x_3	x_4	x_5	s_2	s_3	s_4	Cotas
$c_j - z_j$	0	14.0625	37.5	25.9375	0	0	6.25	0	–2531.25

x_5	0	0.1875	0.5	0.8125	1	0	–1.25	0	56.25
s_2	0	0	0	0	0	1	1	0	9
x_1	1	0.8125	0.5	0.1875	0	0	1.25	0	33.75
s_4	0	–0.8125	–0.5	–0.1875	0	0	–1.25	1	–3.75

Aplicando el dual entra la tercera de holgura y sale la cuarta

	x_1	x_2	x_3	x_4	x_5	s_2	s_3	s_4	Cotas
$c_j - z_j$	0	10	35	25	0	0	0	5	–2550
x_5	0	1	1	1	1	0	0	–1	60
s_2	0	–0.65	–0.4	–0.15	0	1	0	0.8	6
x_1	1	0	0	0	0	0	0	1	30
s_4	0	0.65	0.4	0.15	0	0	1	–0.8	3

RESULTADO DEL PROBLEMA 21

1.

	x_1	x_2	x_3	x_4	x_5	Cotas
$c_j - z_j$	0	0	–0.115	0.01	0	–860
x_5	0	0	0.45	–1.5	1	900
x_2	0	1	0.075	–0.05	0	300
x_1	1	0	–0.05	0.1	0	200

Cambia la solución actual:

	x_1	x_2	x_3	x_4	x_5	Cotas
$c_j - z_j$	–0.1	0	–0.11	0	0	–880
x_5	15	0	–0.3	0	1	3900
x_2	0.5	1	0.05	0	0	400
x_4	10	0	–0.5	1	0	2000

2. Incrementando en 0.05 no interesa, la variable dual es 0.025. Incrementando 0.02 sí interesa y la nueva solución es

$$B^{-1}\overline{b} = \begin{pmatrix} 0.45 & -1.5 & 1 \\ 0.075 & -0.05 & 0 \\ -0.05 & 0.1 & 0 \end{pmatrix} \begin{pmatrix} 8200 \\ 6000 \\ 6300 \end{pmatrix} = \begin{pmatrix} 990 \\ 315 \\ 190 \end{pmatrix}.$$

El nuevo beneficio es 740+200*(0.025–0.020)=741.

3. Hay que añadir la restricción $7x_1 + 4x_2 \leq 2100$ que no es verificada por la solución actual. La tabla que se tendría sería:
4.

	x_1	x_2	x_3	x_4	x_5	x_6	Cotas
$c_j - z_j$	0	0	–0.025	–0.09	0	0	–740
x_5	0	0	0.45	–1.5	1	0	900
x_2	0	1	0.075	–0.05	0	0	300
x_1	1	0	–0.05	0.1	0	0	200
x_6	0	0	0.05	–0.5	0	1	–500

Resultando la siguiente tabla óptima:

	x_1	x_2	x_3	x_4	x_5	x_6	Cotas
$c_j - z_j$	0	0	–0.034	0	0	–0.18	–650
x_5	0	0	0.3	0	1	–3	2400
x_2	0	1	0.07	0	0	–0.1	350
x_1	1	0	–0.04	0	0	0.2	100
x_4	0	0	–0.1	1	0	–2	1000

5. El coste reducido sería: $1.1 - \begin{pmatrix} 0.025 & 0.09 & 0 \end{pmatrix} \begin{pmatrix} 8 \\ 12 \\ 3 \end{pmatrix} = -0.18$. No interesa.

UNIVERSIDAD PONTIFICIA ICAI ICADE COMILLAS MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL