PROGRAMACIÓN MATEMÁTICA:

MODELOS DE OPTIMIZACIÓN

Begoña Vitoriano

bvitoriano@mat.ucm.es

www.mat.ucm.es/~bvitoria

Andrés Ramos

aramos@doi.icai.upcomillas.es

ÍNDICE

I. Optimización

I.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 (*operations research* en inglés de USA u *operational research* en inglés de UK) 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] o como la ciencia de aplicar los recursos disponibles para conseguir la satisfacción óptima de un objetivo específico deseado. 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. En este sentido también se pueden utilizar como sinónimos *management science* o análisis de las decisiones.

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

[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.

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.

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) (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, en los que se plantean modelos que 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 para fabricarn 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, $0 \le x \le u$, y estando u comprendida en el intervalo $2^N \le u \le 2^{N+1}$.

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

I.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*. Universidad Pontificia Comillas.

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

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

II. Modelos de Programación Matemática

II.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].

II.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	300	300
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.

II.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.

III. Formulación de problemas de optimización

III.1. Modelos característicos de programación matemática

A continuación se presentan algunos problemas característicos de programación lineal, entera y no lineal. 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.

III.1.1. 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.

- 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.

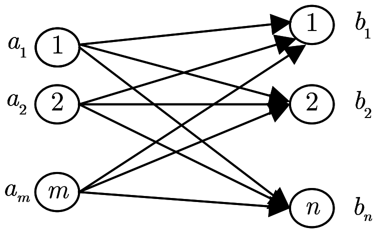
  Los resultados indican que la decisión óptima es comprar 10.833 kg de pienso y 8.333 kg de forraje cada día. Con estas decisiones el coste diario de los alimentos es de 6.1667 €.

  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. La solución óptima para este nuevo problema es comprar 14.933 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.

III.1.2. 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. 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 ha solido decir que se añade un sumidero universal con coste nulo, y 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. Esta opción se lleva a cabo para aplicar métodos específicos de solución para el problema de transporte.

Industrialmente, este modelo se utiliza muy a menudo, pero hay que tener en cuenta algunos detalles de formulación. En primer lugar, respecto al estilo, es importante que las variables reflejen su significado, así como los datos. Por ejemplo, es habitual para cantidades usar *Q*, para demandas *d*, etc. También es importante, distinguir de alguna forma datos de variables (por ejemplo, unos en mayúsculas y otros en minúsculas). Respecto a las condiciones de los datos, el modelo se formula sin hacer hipótesis sobre los valores iniciales ni la red (pueden no existir conexiones entre algunos orígenes y destinos, de modo que *A* es el conjunto de arcos de la red), siendo la siguiente formulación la que contempla que pueda haber más oferta que demanda (equivalente a poner un sumidero universal):

$$\begin{aligned} &\min \sum_{(i,j)\in A} c_{ij} Q_{ij} \\ &\sum_{j/(i,j)\in A} Q_{ij} \le o_i \quad i = 1,\dots,m \\ &\sum_{i/(i,j)\in A} Q_{ij} = d_j \quad j = 1,\dots,n \\ &Q_{ij} \ge 0 \end{aligned} \tag{1.5}$$

Por otra parte, para evitar infactibilidades que harían que el modelo no dé solución alguna, se introducen unas variables que recogen la demanda no suministrada en cada nodo y que son penalizadas en la función objetivo, por ejemplo con un valor *p:*

$$\begin{aligned} &\min \sum_{(i,j)\in A} c_{ij} Q_{ij} + \sum_j p N_j \\ &\sum_{j/(i,j)\in A} Q_{ij} \le o_i \qquad i = 1,\dots,m \\ &\sum_{i/(i,j)\in A} Q_{ij} = d_j + N_j \quad j = 1,\dots,n \\ &Q_{ij}, N_j \ge 0 \end{aligned} \tag{1.6}$$

Éste sería un modelo formulado para poder ser implementado de forma industrial.

En cualquier caso, la estructura que presenta la matriz de restricciones del problema tiene el siguiente aspecto.

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

Si tanto las ofertas como las demandas de los productos son números enteros, entonces el valor óptimo de las variables va a resultar 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.

III.1.3. 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}.

[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*).

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.

$$\begin{aligned}
&\min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\
&\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \quad i = 1, \ldots, n \\
&x_{ij} \geq 0
\end{aligned} \tag{1.7}$$

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.

También en este caso, el modelo académico del industrial difiere sensiblemente, ya que no es normal hacer la hipótesis de que la suma en la red sea 0 (no es normal que la demanda de un producto en una red sea exactamente la oferta, eso suele darse en casos límite). Por otra parte, también ha de protegerse respecto a posibles infactibilidades. Y por último, hablar un lenguaje propio del sector, que en general no va a considerar que producción y demanda sea lo mismo pero con signos opuestos. En este caso la formulación se podría plantear como sigue.

El problema es transportar a través de una red un determinado producto. La red está formada por n nodos y A arcos. Cada nodo puede ser de tres tipos: nodo de oferta donde hay disponible o se produce una cantidad del producto (s_i), nodo de demanda donde hay una demanda del producto (d_i), o nodo de transbordo donde ni se genera ni se consume ($s_i = 0, d_i = 0$).

Las variables del modelo son las siguientes:

Q_{ij}: cantidad transportada de nodo i a nodo j
N_j: demanda no suministrada en nodo j
P_j: cantidad suministrada en nodo j

Con estas variables se admite que la producción o cantidad suministrada en un nodo pueda ser inferior a la oferta existente, así como no suministrar la demanda completa de un nodo si es que no hay existencias o forma de enviar suficiente cantidad al nodo, aunque ha de ser altamente penalizada esta opción (con un valor p alto). El modelo entonces resulta el siguiente:

$$\begin{aligned}
&\min \sum_{(i,j)\in A} c_{ij}Q_{ij} + \sum_i p\, N_i \\
&\sum_{\substack{j/\\(i,j)\in A}} Q_{ij} - \sum_{\substack{j/\\(j,i)\in A}} Q_{ji} = P_i - d_i + N_i \quad \forall i \\
&P_i \le s_i \quad \forall i \\
&N_i \le d_i \quad \forall i \\
&Q_{ij} \ge 0 \quad \forall (i,j)\in A \quad P_i, N_i \ge 0 \quad \forall i
\end{aligned} \tag{1.8}$$

III.1.4. 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

$$x_{ij} = \begin{cases} 1 & \text{si se asigna la tarea } i \text{ a la persona } j \\ 0 & \text{en cualquier otro caso} \end{cases}, \ \forall i, j .$$

$$\begin{aligned}
&\min_{x_{ij}} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\
&\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1,\ldots,n \\
&\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1,\ldots,n \\
&x_{ij} \ge 0
\end{aligned} \tag{1.9}$$

III.1.5. 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

$$\begin{aligned} &\max_{x_j} \sum_{j=1}^{n} v_j x_j \\ &\sum_{j=1}^{n} c_j x_j \le b \\ &x_j \in \{0,1\} \end{aligned} \tag{1.10}$$

III.1.6. 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

$$
\begin{aligned}
&\min_{x_j} \sum_{j=1}^{n} c_j x_j \\
&\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1,\ldots,m \\
&x_j \in \{0,1\}
\end{aligned}
\tag{1.11}
$$

En la siguiente figura se representa gráficamente el problema de recubrimiento así como los de empaquetado y partición que se explican a continuación.

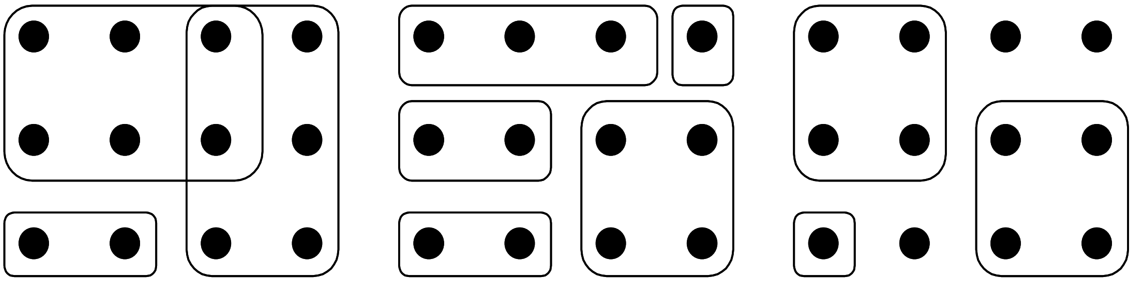


Figura 1.1 Representación gráfica de un recubrimiento, una partición y un empaquetado, respectivamente.

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

$$x_j = \begin{cases} 1 & \text{si se asigna la secuencia } j \\ 0 & \text{en cualquier otro caso} \end{cases}, \qquad x_j \in \{0,1\} \quad j = 1,\ldots,12$$

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

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

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.

III.1.7. 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

$$
\begin{aligned}
&\max_{x_j} \sum_{j=1}^{n} c_j x_j \\
&\sum_{j=1}^{n} a_{ij} x_j \le 1 \quad i = 1,\ldots,m \\
&x_j \in \{0,1\}
\end{aligned}
\tag{1.12}
$$

III.1.8. 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.

$$
\begin{aligned}
&\max_{x_j} \sum_{j=1}^{n} c_j x_j \\
&\sum_{j=1}^{n} a_{ij} x_j = 1 \quad i = 1,\ldots,m \\
&x_j \in \{0,1\}
\end{aligned}
\tag{1.13}
$$

III.1.9. 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 *OR/MS Today* (publicada por el *Institute of Operations Research and the Management Sciences* (*INFORMS*)) de junio de 2004, mencionaba que se había conseguido resolver un problema del viajante con 24978 ciudades. Los problemas de enrutamiento de vehículos (expedición o recogida de mercancías) pueden ser formulados basándose en este modelo. Una de las características más interesantes de este problema es que existen muchas formulaciones conocidas para el mismo, ver [Williams, 1999] y [Nemhauser, 1999]. Una de ellas es la siguiente. Sea c_{ij} la distancia entre las ciudades i y j.

Se definen las variables

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

$$T_j: \text{ instante de llegada a ciudad } j$$

La formulación del problema es:

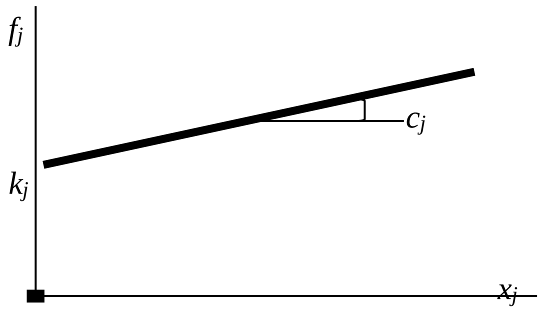
$$\begin{aligned} & \min_{x_{ij}} \sum_{i,j} c_{ij} x_{ij} \\ & \sum_i x_{ij} = 1 \quad \forall j \\ & \sum_j x_{ij} = 1 \quad \forall i \\ & T_j \geq T_i + c_{ij} - m(1 - x_{ij}) \qquad \forall i, j \\ & T_j \geq c_{1j} - m(1 - x_{1j}) \qquad \forall j \neq 1 \\ & x_{ij} \in \{0,1\}, T_j \geq 0 \end{aligned} \tag{1.14}$$

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. Sólo con estas variables no es suficiente para formular el problema, ya que se pueden formar subciclos. La forma de evitarlos es añadiendo las variables continuas.

III.1.10. 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 $x_j \le M y_j$, $j = 1,\ldots,n$, 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

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

III.1.11. Modelado de restricciones con variables binarias

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 $x_j \geq 1$. Entonces, la implicación es

$$x_A \geq 1 \rightarrow x_B \geq 1$$

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 $x_B \geq 1$) depende de que se cumpla otra ($x_A \geq 1$) 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.

III.1.11.1. Modelado de disyunciones

Disyunciones

Las disyunciones implican una pareja de restricciones donde 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.

$$f(x) \leq 0 \text{ ó } g(x) \leq 0$$

Supongamos el ejemplo de esta disyunción

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

Veamos cómo estas restricciones se pueden incorporar en un problema de programación matemática.

Añadir una constante de valor elevado M a una restricción es equivalente a eliminar (relajar) dicha restricción.

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

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

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

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

Algunas implicaciones son un caso semejante a las restricciones disyuntivas

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

es equivalente a

$$f(x) \le 0 \text{ ó } g(x) \le 0$$

ya que $P \rightarrow Q$ 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

$$\begin{array}{l} f_1(x_1, \ldots, x_n) \le 0 \\ f_2(x_1, \ldots, x_n) \le 0 \\ f_N(x_1, \ldots, x_n) \le 0 \end{array}$$

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

$$
\begin{aligned}
f_1(x_1,\ldots,x_n) &\le My_1 \\
f_2(x_1,\ldots,x_n) &\le My_2 \\
f_N(x_1,\ldots,x_n) &\le My_N
\end{aligned}
$$

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

$$\sum_{i=1}^{N} y_i = N-k \qquad\qquad y_i \in \{0,1\} \;\; i=1,\ldots,N$$

Seleccionar entre *N* valores

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

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

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

$$
\begin{aligned}
&f(x_1,\ldots,x_n)=\sum_{i=1}^{N} d_i y_i \\
&\sum_{i=1}^{N} y_i = 1 \\
&y_i \in \{0,1\} \quad i=1,\ldots,N
\end{aligned}
$$

III.1.11.2. 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

$$x \le M\delta$$

siendo M una cota superior positiva de x (por ejemplo, 10^6), $m \le x \le M$ 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 $x \le M$ se cumple por definición. Si $\delta = 0$ entonces $x \le 0$. Luego esta restricción permite modelar la implicación

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

Por otra parte, si $x > 0$ entonces $\delta = 1$. Si $x \le 0$ 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 $x \le M\delta$ nos permite representar dichas implicaciones en un problema lineal.

De forma análoga veamos la restricción $x \ge m\delta$

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

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

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

Por otra parte, si $x < 0$ entonces $\delta = 1$. Si $x \ge 0$ 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

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

$$\left.\begin{array}{l}\delta = 0 \rightarrow x \geq 0 \\ x < 0 \rightarrow \delta = 1\end{array}\right\} x \geq m\delta$$

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

Implicaciones de una restricción $\leq$

La implicación

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

es equivalente a

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

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

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

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

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

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

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

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

Claramente si $\delta = 1$ se cumplen ambas restricciones y si $\delta = 0$ ambas se relajan.

La implicación

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

es una combinación de los casos anteriores simultáneamente

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

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

Las implicaciones dobles se desdoblan en implicaciones unidireccionales

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

La siguiente tabla presenta todas las implicaciones lógicas y su formulación lineal:

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

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.

III.1.11.3. 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 $\rightarrow$ Q	no P o Q
P $\rightarrow$ (Q y R)	(P $\rightarrow$ Q) y (P $\rightarrow$ R)
P $\rightarrow$ (Q o R)	(P $\rightarrow$ Q) o (P $\rightarrow$ R)
(P y Q) $\rightarrow$ R	(P$\rightarrow$ R) o (Q $\rightarrow$ R)
(P o Q) $\rightarrow$ R	(P$\rightarrow$ R) y (Q $\rightarrow$ 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	$\delta_1 = 0$
$X_1 \to X_2$	$\delta_1 - \delta_2 \leq 0$
$X_1 \leftrightarrow X_2$	$\delta_1 - \delta_2 = 0$

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 $\delta_i = 1$, $x_i > 0 \to \delta_i = 1$. 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

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

se puede representar como

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

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

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

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

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

Aplicando este procedimiento al ejemplo anterior, la implicación

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

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

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

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

se podía haber transformado en $\left[X_A \to (X_C \text{ o } X_D \text{ o } X_E)\right]$ y $\left[X_B \to (X_C \text{ o } X_D \text{ o } X_E)\right]$

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

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

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

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

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

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

$$x_3 + x_6 \leq 1$$

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

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

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.

III.1.11.4. 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 $\delta = 0 \to y = 0$ $\delta = 1 \to y = x$	$y \ge 0$ $y \le M\delta$ $-x + y \le 0$ $x - y + M\delta \le M$ $x \le M$

III.1.12. Una aplicación: Modelo 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]

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

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

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

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

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

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

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

$$P_{ht} \ge 0 \quad A_{ht}, AR_{ht}, PR_{ht} \in \{0,1\}$$

```
$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
```

III.1.13. 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.

III.1.14. 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.

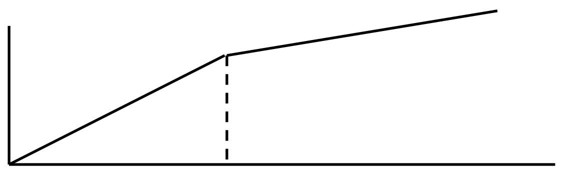
Así pues, el coste de embarcar x unidades viene dado por una función *poligonal*, $C(x)$, continua, con pendiente en cada tramo igual al coste unitario de transporte. En consecuencia, si cada combinación de origen y destino tiene una función semejante, la función objetivo sería

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

Al ser una función poligonal cóncava en un problema de minimización se modelará introduciendo variables binarias de selección del segmento de la poligonal, y formulándolo como un problema lineal entero.

Sin embargo, hay que distinguir dos casos al hacer la formulación, según el descuento se aplique a todas las unidades si la cantidad supera un valor, o sólo a las que superan ese valor.

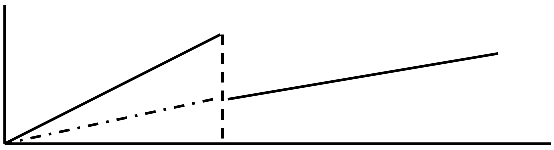
En el caso de que el descuento sólo se aplique a las cantidades que superan un valor, por ejemplo *k*, de modo que las *k* primeras siempre tienen un coste unitario *c* y las que sobrepasen a *k*, tengan un descuento siendo su coste unitario *c-a*, obsérvese que la función es continua:



El modelo para representar esta función, siendo *X* la variable pasa por dividir esta variable en dos, una hasta el valor *k* y otra para el exceso, obligando a que la segunda no sea distinta de cero mientras la otra no llegue al valor *k*:

$$\begin{aligned} \min \quad & cX_1 + (c-a)X_2 \\ & X = X_1 + X_2 \\ & \delta k \le X_1 \le k \\ & 0 \le X_2 \le \text{cot}\, a\ \delta \\ & \delta \in \{0,1\} \end{aligned}$$

En el caso, de aplicarse el descuento a todas las unidades la función es discontinua:



Y el modelo, aún siendo también una descomposición en dos variables, es distinto, pues serán variables que no puedan ser distintas de cero a la vez:

$$\begin{aligned} \min \quad & cX_1 + (c-a)X_2 \\ & X = X_1 + X_2 \\ & 0 \le X_1 \le k\delta \\ & k(1-\delta) \le X_2 \le \text{cot}\, a\ (1-\delta) \\ & \delta \in \{0,1\} \end{aligned}$$

III.1.15. 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):

$$\sum_{j=1}^{n} P_j x_j \le B$$

$$x_j \ge 0 \quad j = 1, \dots, n$$

III.1.16. 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.

III.2. 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.

III.3. Biblioteca de problemas

PROBLEMA: DISTRIBUCIÓN DE GASÓLEO

Una empresa de distribución de gasóleo dispone de tres depósitos D1, D2 y D3 donde guarda el gasóleo y desde los que abastece a cuatro estaciones de servicio, E1, E2, E3 y E4. En la tabla se muestra la capacidad máxima de almacenamiento de cada depósito, la demanda máxima de cada punto de venta y las capacidades máximas de transporte en las posibles rutas entre depósitos y estaciones de servicio.

	E1	E2	E3	E4	Capacidad
D1	80	–	70	–	150
D2	–	60	90	85	300
D3	40	60	–	50	250
Demanda	130	200	150	250	

Además, cada unidad que se envíe a las gasolineras le supone un beneficio de 7 u.m. y los envíos se llevan a cabo en camiones cuya capacidad es de 20 unidades cada uno, con un coste estimado por camión en cada una de las rutas de:

Costes (u.m.)	E1	E2	E3	E4
D1	100	–	125	–
D2	–	100	125	100

D3	75	100	–	125

Formular y resolver un problema para determinar el plan de rutas óptimo que maximice el beneficio de la empresa.

PROBLEMA: CONDUCTORES DE METRO

La Compañía Metropolitana de Transporte (CMT) explota el metro de la ciudad y planifica la asignación de turnos de su personal de operación. El convenio laboral exige que cada conductor disponga de dos días libres consecutivos por semana. La demanda de conductores para operar los trenes varía dependiendo del día, siendo, de lunes a domingo, de 18, 16, 15, 16, 19, 14 y 12 conductores respectivamente. El coste de personal varía a lo largo de la semana. Para un día cualquiera de lunes a viernes el coste es de 50 € por día, los sábados se pagan a 75 y los domingos a 90. Además se puede recurrir a la contratación parcial de hasta 3 personas que trabajarían los viernes, sábados y domingos por un coste de 200 €. Determinar el esquema óptimo de turnos de trabajo de dichos conductores.

PROBLEMA: PRODUCCIÓN

Una empresa puede fabricar 4 productos diferentes utilizando en su elaboración 5 tipos de materias primas. Las unidades requeridas de cada materia prima por cada unidad de cada producto se recogen en la tabla siguiente, así como el precio de venta de cada producto y la disponibilidad de cada materia prima:

	M1	M2	M3	M4	M5	Precio venta
P1	1	1		1		15
P2		1		1	1	20
P3	1	1	1			15
P4	1		1			10
Disponible	200	150	150	200	100	

El precio unitario de cada materia prima es de 2, 3, 4, 5 y 5, respectivamente, pero hay un coste adicional por hacer un pedido de cada materia prima, independiente de la cantidad que se pida, de 20, 25, 20, 25, 25 unidades, respectivamente. Determinar el plan de producción óptimo para maximizar los beneficios de la empresa.

PROBLEMA: 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:

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.

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 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 ha de pasar la red 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 l/min	Capacidad l/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 1 kg de materia prima para producir 1 kg de producto 1 o para producir 1/2 kg del producto 2. Usar 1 kg de materia prima para producir el producto 1 requiere 2 horas de mano de obra. Usar 1 kg 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 existe 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.

PROBLEMA: EDICIÓN DE CDS

Una compañía discográfica está pensando en editar una colección de grandes éxitos de uno de sus cantantes más famosos. Todas sus canciones han sido agrupadas en doce lotes. Cada lote ocupa tiene un cierto tamaño expresado en MB y tiene asociado un índice de marketing (relacionado con su demanda esperada).

Lote	Tamaño	Índice marketing
1	350	15
2	300	32
3	160	20
4	310	25
5	260	35
6	250	10
7	400	40
8	100	34
9	520	24
10	170	16
11	300	36
12	360	26

Los lotes van a ser grabados en una serie de CDs, cada uno con una capacidad máxima de 700 MB. Además, para que el CD funcione en el mercado, se estima que debe tener un índice de marketing superior a 45. Por otro lado, y dado el contenido de sus canciones, los lotes 1, 2 y 3 se han de grabar en CDs distintos. El objetivo de la compañía es maximizar el número de CDs editados, utilizando todos los lotes de canciones y sin editar más de una vez el mismo lote.

PROBLEMA: VUELOS CHARTER

Una compañía aérea tiene una flota de 15 aeronaves: 5 de cada uno de tres tipos A, B y C, cuyas respectivas capacidades para el transporte de viajeros son de 80, 68 y 55 personas. Una agencia de viajes le solicita presupuesto para trasladar a 372 personas. La compañía analiza sus costes, que dependen del número de aviones de cada tipo que quiera utilizar para transportar a esas personas, datos que se dan en la siguiente tabla en miles de euros

Tipo	1	2	3	4	5
A	11	20	30	40	50
B	9	17	24	34	45
C	8	15	21	26	31

Además la compañía aérea incurre en un coste fijo adicional de 6 k€ por cada tipo de aviones que utilice.

Proponer un modelo de programación lineal entera cuyo objetivo sea determinar la composición óptima de la flotilla de aviones que va a realizar el transporte para minimizar los costes de la operación.

PROBLEMA: PROVEEDORES

Tres almacenes deben abastecer a cuatro mercados de cierto número de unidades de un producto. Los costes unitarios de transporte (en €), las existencias en los almacenes y las necesidades de los mercados se dan en la tabla siguiente.

	M1	M2	M3	M4	Existencias en almacenes
A1	6	7	9	5	220
A2	7	10	9	6	350
A3	5	8	7	5	270
Demandas en mercados	80	90	100	135	

Cuando un almacén abastece a un mercado ha de firmar un contrato para establecerse como proveedor del mercado, de modo que ha de pagar una cantidad por el único hecho de ser proveedor, establecida en 100 €. La demanda de los mercados ha de ser satisfecha.

1. Establecer un programa lineal entero que permita obtener el plan de abastecimiento más económico.
2. Sobre el planteamiento anterior, si de un almacén cualquiera se envía a todos los mercados un total de menos de 100 unidades, sus costes unitarios de la tabla

anterior se incrementan en 2 €, es decir, el coste unitario de la tabla es para todas las unidades enviadas desde un almacén a un mercado si la cantidad total que el almacén envía es mayor o igual que 100. Establecer un programa lineal entero que permita obtener el plan de abastecimiento más económico.

PROBLEMA: COMPOSICIÓN DE PIENSO

Un pequeño ganadero alimenta sus reses con una mezcla de dos piensos compuestos, P1 y P2, que él mismo elabora, y en los que es posible encontrar tres nutrientes N1, N2 y N3, de acuerdo con lo reflejado en la tabla, donde se da el contenido, en gramos, de cada nutriente por kilo de pienso compuesto.

	N1	N2	N3	Aditivos
P1	100	300	400	200
P2	200	250	300	250

Los costes de fabricación de un kilo de cada pienso son de 0.3 € para P1 y de 0.36 € para P2. Las necesidades diarias de una res respecto a los nutrientes considerados son:

- N1: entre 250 y 300 gramos, con una cantidad óptima de 250 gramos
- N2: entre 325 y 460 gramos, con una cantidad óptima de 400 gramos
- N3: entre 450 y 600 gramos, con un óptimo de 500 gramos.

La desviación, en más o en menos, de la cantidad de nutrientes proporcionados a una res respecto al valor óptimo antes indicado requiere un tratamiento compensatorio, con costes de 0.01 € por gramo, en el caso de N1, 0.005 € por gramo en el caso de N2 y de 0.008 € por gramo en el caso de N3.

1. Plantear un modelo de programación lineal para determinar qué cantidad de cada pienso hay que proporcionar a cada res para minimizar el coste de alimentación
2. Además, si la cantidad utilizada de P1 fuera estrictamente mayor que el doble de la de P2 el coste de P1 se incrementaría en 0.05 €/kg. Modificar el modelo anterior añadiendo este supuesto y sin que deje de ser lineal.

PROBLEMA: EXPLOTACIONES GANADERAS

En dos explotaciones ganaderas G1 y G2 se crían vacas y cerdos, con los cuales se intenta satisfacer las demandas de tres mataderos industriales M1, M2 y M3.

Las disponibilidades y demandas actuales de vacas y cerdos en explotaciones y mercados, así como los costes de transporte unitarios entre ellos, en €, se resumen en las siguientes tablas.

	Vacas				Cerdos			
	M1	M2	M3	Disponibilidad	M1	M2	M3	Disponibilidad
G1	25	21	20	**25**	6	7	8	**130**
G2	27	19	25	**32**	8	6	7	**100**
Demanda	**17**	**6**	**22**		**45**	**85**	**76**	

En cada explotación hay 6 vehículos especialmente adaptados para el transporte de vacas y 8 para el de cerdos. Un vehículo para vacas tiene una capacidad en volumen de 100 unidades y uno para cerdos tiene una capacidad de 80 unidades de volumen. Con independencia de la explotación ganadera y del matadero entre los que puedan hacer un viaje, el uso de un camión para vacas tiene un coste fijo de 200 €, además del coste directamente asociado a las vacas transportadas. Por lo que hace a los camiones para cerdos, dicho coste fijo es de 165 €.

Una vaca ocupa 12 unidades de volumen y un cerdo 3.

1. Elaborar un modelo de programación lineal entera que satisfaga las demandas de los mataderos con un coste de transporte mínimo.

2. Supóngase ahora que también es posible, si conviene, transportar cerdos en un camión para vacas, aunque ello supone ciertas adaptaciones que incrementan el coste unitario de transporte de los cerdos en 2 €. De la misma forma, es posible transportar vacas en los vehículos para cerdos, con un coste unitario adicional de 10 €. Suponinedo que en un camión pueden ir vacas o cerdos simultáneamente, elaborar un modelo de programación lineal entera que satisfaga las demandas de los mataderos con un coste de transporte mínimo.

PROBLEMA: CORTE DE BOBINAS

La empresa EIVISSA se dedica a la fabricación de bobinas madre de plástico con un ancho de 6000 mm. Sus clientes son empresas de envasado cuyos pedidos son bobinas hija de anchos inferiores, tal como aparecen en la siguiente tabla:

Pedido	Anchura [mm]	Fecha de entrega [horas]
1	1500	6
2	1200	10
3	800	7
4	750	8
5	915	8
6	315	10
7	1450	5
8	650	4
9	725	7
10	1800	12

Se trata de plantear un modelo de optimización para determinar los cortes a realizar en las bobinas madre para satisfacer los pedidos de bobinas hija con estas posibles diferentes funciones objetivo alternativas:

1. Plantear el modelo de optimización completo para minimizar el número total de bobinas madre necesitadas
2. Minimizar la suma del desperdicio lateral sobrante de todas las bobinas
3. Minimizar la suma de los retrasos de cada pedido con respecto a las fechas de entrega teniendo en cuenta que el corte de una bobina madre en bobinas hija dura 6 horas y que sólo se dispone de una máquina de corte de bobinas madre por lo que el corte de cada bobina se hace consecutivamente

PROBLEMA EMPRESA DISTRIBUIDORA

Una empresa distribuidora tiene dos almacenes centrales en las ciudades C1 y C2 desde los que abastece a tres centros de distribución situados en D1, D2 y D3. Desde estos centros de distribución se aprovisionan a cuatro mercados M1, M2, M3 y M4.

Sean e_1 y e_2 las existencias en los almacenes centrales y d_1, d_2, d_3 y d_4 las demandas de los mercados. El coste de transporte unitario desde el almacén central i al centro de distribución j es c_{ij} y el de transportar una unidad desde el centro de distribución j al mercado k es w_{jk}. Cada unidad del artículo tiene un peso p.

1. Elaborar un modelo de programación lineal que permita satisfacer las demandas de los mercados al menor coste.
2. ¿Cómo se modifica dicho modelo si M1 no puede ser aprovisionado simultáneamente desde D1 y D2?
3. ¿Qué modificación adicional hay que introducir cuando el coste unitario de transporte entre un centro de distribución y un mercado cualesquiera de todas las unidades que excedan de un mínimo m, y sólo de ellas, se reduce en una cantidad h?
4. ¿Qué nueva modificación experimenta el modelo si las unidades se transportan en camiones que pueden transportar un peso máximo P, tal que P no es múltiplo de p ? El coste de cada camión es independiente del trayecto realizado y asciende a la cantidad C. Este coste se añade al coste unitario de transporte de las unidades.

III.4. Resultados de la biblioteca de problemas

RESULTADO DEL PROBLEMA DISTRIBUCIÓN DE GASÓLEO

X_{ij} : cantidad enviada de depósito i a estación j

Y_{ij} : camiones usados de depósito i a estación j

$$
\begin{aligned}
&\max \sum_{i,j}\left(7X_{ij}-c_{ij}Y_{ij}\right)\\
&s.a.\quad \sum_{j} X_{ij} \le cap_i \quad \forall i\\
&\qquad \sum_{i} X_{ij} \le dem_j \quad \forall j\\
&\qquad X_{ij} \le capac_{ij} \quad \forall i,j\\
&\qquad X_{ij} \le 20Y_{ij} \quad \forall i,j \qquad X_{ij}, Y_{ij} \ge 0, Y_{ij} \in \mathbb{Z}
\end{aligned}
$$

Objetivo: 825

Cantidades	E1	E2	E3	E4		Camiones	E1	E2	E3	E4
D1	80	0	60	0		D1	4	0	3	0
D2	0	60	80	80		D2	0	3	4	4
D3	40	60	0	40		D3	2	3	0	2

RESULTADO DEL PROBLEMA CONDUCTORES DE METRO

X_i : número de conductores que empiezan su turno el día i

T_i : conductores que trabajan el día i

Y : número de personas contratadas a tiempo parcial

$$
\begin{aligned}
&\min \sum_{i} c_i T_i + 200Y\\
&s.a.\quad T_i = \sum_{j\ne i+1,i+2} X_j \ge nec_i \quad \forall i \ne 5,6,7\\
&\qquad T_i = \sum_{j\ne i+1,i+2} X_j + Y \ge nec_i \quad \forall i = 5,6,7\\
&\qquad Y \le 3\\
&\qquad X_i, T_i, Y \ge 0
\end{aligned}
$$

L	M	X	J	V	S	D	Extra
8	2	2	4	3	3	0	0

COSTE: 6730

RESULTADO DEL PROBLEMA PRODUCCIÓN

X_i : unidades producidas de producto Pi

$$M_j : \begin{cases} 1 & \text{si pide materia prima j} \\ 0 & \text{si no pide materia prima j} \end{cases}$$

$$\max \sum_i pv_i X_i - \sum_{i,j} a_{ij} c_j X_i - \sum_j f_j M_j$$

$$s.a. \quad \sum_i a_{ij} X_i \le disp_j M_j \quad \forall j$$

$$X_i \ge 0, M_j \in \{0,1\}$$

Beneficio	1435
P1	50
P2	100
P3	0
P4	150

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 40 a la aldea V3 y 20 a la V5, y el avión A3 hace 50 a la aldea V2 y 40 a la V4. La cantidad total de alimentos repartidos es 1870.

RESULTADO DEL PROBLEMA CONSTRUCCIÓN DE ALMACENES

$$\min_{x_{ij}, y_i} \sum_{i,j} v_{ij} x_{ij} + \sum_i f_i y_i$$

$$\sum_j x_{ij} \le c_i y_i \quad \forall i$$

$$\sum_i x_{ij} \ge d_j \quad \forall j$$

$$x_{ij} \ge 0, y_i \in \{0,1\}$$

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

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

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

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

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

$$\begin{aligned}
&\min x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
&x_1 + x_6 \geq 4 \\
&x_1 + x_2 \geq 8 \\
&x_2 + x_3 \geq 10 \\
&x_3 + x_4 \geq 7 \\
&x_4 + x_5 \geq 12 \\
&x_5 + x_6 \geq 4 \\
&x_5 + x_6 \leq 4 + 12\delta \\
&x_1 + x_6 \geq 5\delta \\
&x_i \in \mathbb{Z}^+, \delta \in \{0,1\}
\end{aligned}$$

El número mínimo de autobuses es de 26.

RESULTADO DEL PROBLEMA ADQUISICIÓN DE CAMIONES

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

RESULTADO DEL PROBLEMA PLANIFICACIÓN DEL METRO

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

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

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

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

$$
\begin{aligned}
&\min(2x+1.75y+1.5z+0.01\cdot 20x+0.01\cdot 15y+0.01\cdot 10z)8\\
&20\cdot 8x+15\cdot 8z+10\cdot 8z\ge 3500\\
&x\le 8\\
&y\le 10\\
&z\le 20\\
&x,y,z\in\mathbb{Z}^+
\end{aligned}
$$

RESULTADO DEL PROBLEMA SECUENCIACIÓN DE TRABAJOS EN UNA MÁQUINA

Sea 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:

- 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 con fecha de entrega $\sum_j r_j x_{ij}$.

- Por otra parte, el trabajo j que acaba en esa posición acaba en el instante $\sum_j d_j \sum_{k \le i} x_{kj}$. 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 $\sum_j d_j \sum_{k \le i} x_{kj} + n_i - p_i = \sum_j r_j x_{ij} \quad \forall i$

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

RESULTADO DEL PROBLEMA PRODUCCIÓN II

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

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

RESULTADO DEL PROBLEMA PRODUCCIÓN E INVENTARIO

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

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)

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

b)

$$\lambda_{ij-1} + \lambda_{ij+1} - \lambda_{ij} \le 1 \quad \forall i, 2 \le j \le n-1$$

o bien

$$\lambda_{ij} + \lambda_{ij+1} \le 2 - 2Par_{ij} \quad \forall i, 1 \le j \le n-1$$

RESULTADO DEL PROBLEMA TRANSPORTE POR FERROCARRIL a)

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

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 ser incluida.

b)

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

c)

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

o bien

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

RESULTADO DEL PROBLEMA EDICIÓN DE CDS

$$
\begin{aligned}
&\max \sum_j y_j \\
&\sum_j x_{ij} = 1 \quad \forall i \\
&\sum_i tam_i x_{ij} \le 700 y_j \quad \forall j \\
&\sum_i indi_i x_{ij} \ge 45 y_j \quad \forall j \\
&x_{1j} + x_{2j} + x_{3j} \le 1 \quad \forall j \\
&x_{ij}, y_j \in \{0,1\}
\end{aligned}
$$

RESULTADO DEL PROBLEMA VUELOS CHARTER

$$
\begin{aligned}
&\min \sum_{i,j} C_{ij} x_{ij} + \sum_i 6 y_i \\
&\sum_i CAP_i \sum_j j \cdot x_{ij} \ge D \\
&\sum_j x_{ij} \le y_i \quad \forall i \\
&x_{ij}, y_i \in \{0,1\}
\end{aligned}
\qquad \text{o bien} \qquad
\begin{aligned}
&\min \sum_{i,j} C_{ij} x_{ij} + 6 \sum_{i,j} x_{ij} \\
&\sum_i CAP_i \sum_j j \cdot x_{ij} \ge D \\
&\sum_j x_{ij} \le 1 \quad \forall i \\
&x_{ij} \in \{0,1\}
\end{aligned}
$$

RESULTADO DEL PROBLEMA PROVEEDORES

1.
$$
\begin{aligned}
&\min \sum_{i,j} \left(c_{ij} X_{ij} + ccont_{ij} Y_{ij} \right) \\
&\sum_j X_{ij} \le a_i \quad \forall i \\
&\sum_i X_{ij} = d_j \quad \forall j \\
&X_{ij} \le M Y_{ij} \quad \forall i, j \ (M \text{ cota suf. grande, por ejemplo, } M = \max(d_j) = 135) \\
&X_{ij} \ge 0, Y_{ij} \in \{0,1\}
\end{aligned}
$$

$$\min \sum_{i,j} \left(c_{ij} X_{ij}^{+} + (c_{ij} + penal_i) X_{ij}^{-} + ccont_{ij} Y_{ij} \right)$$
$$\sum_j X_{ij} \le a_i \quad \forall i$$
$$\sum_i X_{ij} = d_j \quad \forall j$$
$$X_{ij} \le M Y_{ij} \quad \forall i, j \; (M \text{ cota suf. grande, por ejemplo, } M = \max(d_j) = 135)$$

2. $X_{ij} = X_{ij}^{+} + X_{ij}^{-} \quad \forall i, j$

$$\sum_j X_{ij} \ge k_i (1 - Z_i) \quad \forall i$$
$$X_{ij}^{-} \le m_{ij} Z_i \quad \forall i, j$$
$$X_{ij}^{+} \le m_{ij} (1 - Z_i) \quad \forall i, j \; (m_{ij} \text{ cota suf. grande, por ejemplo, } d_j)$$
$$X_{ij}, X_{ij}^{-}, X_{ij}^{+} \ge 0, Y_{ij}, Z_{ij} \in \{0,1\}$$

RESULTADO DEL PROBLEMA COMPOSICIÓN DE PIENSO

a)

$$\min \sum_i c_i X_i + \sum_j p_j (D_j + E_j)$$
$$\sum_i a_{ij} X_i \ge \min{}_j \quad \forall j$$
$$\sum_i a_{ij} X_i \le \max{}_j \quad \forall j$$
$$\sum_i a_{ij} X_i + D_j - E_j = opt_j \quad \forall j$$
$$X_i, D_j, E_j \ge 0$$

b) Dos modelos alternativos:

$$\min \sum_i c_i X_i + Extra + \sum_j p_j (D_j + E_j)$$
$$\sum_i a_{ij} X_i \ge \min{}_j \quad \forall j$$
$$\sum_i a_{ij} X_i \le \max{}_j \quad \forall j$$
$$\sum_i a_{ij} X_i + D_j - E_j = opt_j \quad \forall j$$
$$X_1 - 2X_2 \le M\delta$$
$$Extra \ge 0.05 X_1 - M(1 - \delta)$$
$$X_i, D_j, E_j, Extra \ge 0 \quad \delta \in \{0,1\}$$

$$
\begin{aligned}
&\min\ c_1 X_1^- + c_2 X_2 + (c_1 + 0.05) X_1^+ + \sum_j p_j (D_j + E_j) \\
&\sum_i a_{ij} X_i \geq \min_j \quad \forall j \\
&\sum_i a_{ij} X_i \leq \max_j \quad \forall j \\
&\sum_i a_{ij} X_i + D_j - E_j = opt_j \quad \forall j \\
&X_1 = X_1^- + X_1^+ \\
&X_1 - 2X_2 \leq M\delta \\
&X_1^- \leq M(1-\delta) \\
&X_1^+ \leq M\delta \\
&X_i, X_1^-, X_1^+, D_j, E_j \geq 0 \quad \delta \in \{0,1\}
\end{aligned}
$$

RESULTADO DEL PROBLEMA EXPLOTACIONES GANADERAS

a) i explotaciones ganaderas, granjas (G1 y G2)

j mataderos (M1, M2 y M3)

k tipo de animal (vaca y cerdo)

N_{ik} número de vehículos para transportar animales tipo k en explotación i

$DISP_{ik}$ disponibilidad de animales k en las explotaciones ganaderas i

DEM_{jk} demanda en cada matadero j de cada animal k

$COSTE_{ijk}$ coste unitario de transporte de explotaciones ganaderas i a mataderos j para cada animal k

CAP_k capacidad de cada tipo de vehículo k

$FIJO_k$ coste fijo de uso del camión para transportar animales tipo k

VOL_k volumen unitario de cada tipo de animal k

Transformar un dato: $ANIM_k = \lfloor CAP_k / VOL_k \rfloor$: número animales que caben en un camión de tipo k

Podrían ser dos problemas separados de transporte (nada los liga)

Variables:

x_{ijk} animales tipo k transportados entre la explotación i y el matadero j

z_{ijk} vehículos para animales tipo k usados entre explotación i y matadero j

$$\min \quad \sum_k \left(\sum_{i,j} \left(COSTE_{ijk} x_{ijk} + FIJO_k z_{ijk} \right) \right)$$

$$\sum_j x_{ijk} \le DISP_{ik} \qquad \forall i,k$$

$$\sum_i x_{ijk} \ge DEM_{jk} \qquad \forall j,k$$

$$\sum_j z_{ijk} \le N_{ik} \qquad \forall i,k$$

$$x_{ijk} \le ANIM_k z_{ijk} \qquad \forall i,j,k \quad Cuasicorrecto: VOL_k x_{ijk} \le CAP_k z_{ijk} \quad \forall i,j,k$$

$$x_{ijk}, z_{ijk} \in \mathbb{Z}^+$$

b)

i explotaciones ganaderas, granjas (G1 y G2)

j mataderos (M1, M2 y M3)

k tipo de animal (vaca y cerdo)

n vehículo

N_{ik} número de vehículos para transportar animales tipo k en explotación i

$DISP_{ik}$ disponibilidad de animales k en las explotaciones ganaderas i

DEM_{jk} demanda en cada matadero j de cada animal k

$COSTE_{ijk}$ coste unitario de transporte de explotación ganadera i a matadero j para cada animal k

CAP_k capacidad de cada tipo de vehículo k

$FIJO_k$ coste fijo de uso del camión para transportar animales tipo k

VOL_k volumen unitario de cada tipo de animal k

$INCCOSTE_k$ incremento de coste unitario de transporte para cada animal k

Variables

x_{ijk}^{kn} cantidad de animales de tipo k transportados entre la explotación ganadera i y el matadero j transportados en el vehículo número n del tipo k

$y_{ijk}^{k'n}$ cantidad de animales de tipo k transportados entre la explotación ganadera i y el matadero j transportados en el vehículo número n del tipo k' (en este caso k y k' son diferentes)

v_{ij}^{nk} uso o no del vehículo n diseñado para el transporte del animal k para transportar animales entre la explotación ganadera i y el matadero j

$$\min \sum_{knij} \left(COSTE_{ijk} x_{ijk}^{kn} + (COSTE_{ijk} + INCCOSTE_k) y_{ijk}^{k'n} + FIJO_k v_{ij}^{kn} \right)$$

$$VOL_k x_{ijk}^{kn} + VOL_{k'} y_{ijk'}^{kn} \le CAP_k v_{ij}^{kn} \quad \forall knij$$

$$\sum_{ni} (x_{ijk}^{kn} + y_{ijk}^{k'n}) \ge DEM_{jk} \quad \forall jk$$

$$\sum_{nj} (x_{ijk}^{kn} + y_{ijk}^{k'n}) \le DISP_{ik} \quad \forall ik$$

$$\sum_{j} v_{ij}^{kn} \le 1 \quad \forall ikn$$

$$x_{ijk}^{kn}, y_{ijk}^{kn} \in \mathbb{Z}^+, v_{ij}^{kn} \in \{0,1\}$$

La solución óptima con una tolerancia relativa del 3 % tiene un coste de 4943 €.

		Vehículos de vacas				Vehículos de cerdos	
	M1	M2	M3		M1	M2	M3
G1			(8,0) (8,0) (6,0)		(0,23) (**1**,22)	(0,26)	(0,24)
G2	(8,0) (8,0)	(6,**7**)				(0,26) (0,26)	(0,26) (0,26)

RESULTADO DEL PROBLEMA CORTE DE BOBINAS

Definimos i como el índice de cada pedido o bobina hija, I es el número total de pedidos, j como el índice de las bobinas madre, J es el número máximo de bobinas madre posibles (una cota superior es el número de pedidos). A_i es la anchura de cada pedido, F_i es la fecha de entrega de cada pedido y $\bar{A}$ es la anchura de la bobina madre.

Definimos las variables

$$X_{ij} = \begin{cases} 1 & \text{si el pedido } i \text{ se obtiene de la bobina madre } j \\ 0 & \text{en otro caso} \end{cases},$$

$Y_j = \begin{cases} 1 & \text{si se usa la bobina madre } j \\ 0 & \text{en otro caso} \end{cases}$, D_j el desperdicio de la bobina j.

La función objetivo de minimización del número total de bobinas se expresa como

$$\min \sum_{j=1}^{J} Y_j$$

- Satisfacción de cada pedido i: $\sum_{j=1}^{J} X_{ij} = 1 \quad \forall i$
- Los pedidos de una bobina no pueden exceder su ancho: $\sum_{i=1}^{I} A_i X_{ij} \leq \bar{A} Y_j \quad \forall j$

La función objetivo de minimización del desperdicio de las bobinas madre se expresa como

$$\min \sum_{j=1}^{J} D_j$$

- El desperdicio se calcula cambiando la restricción anterior

$$\sum_{i=1}^{I} A_i X_{ij} + D_j = \bar{A} Y_j \quad \forall j$$

La función objetivo de minimización del retraso en la fecha de entrega se expresa como $\min \sum_{i=1}^{I} RT_i$

siendo AD_i y RT_i el adelanto o retraso en la fecha de entrega del pedido i

- El retraso en la fecha de entrega de cada pedido se calcula en esta ecuación

$$\sum_{j=1}^{J} 6jX_{ij} + AD_i - RT_i = F_i \quad \forall i \text{ o bien } \sum_{j=1}^{J} 6jX_{ij} - RT_i \le F_i \quad \forall i$$

RESULTADO DEL PROBLEMA EMPRESA DISTRIBUIDORA

1. Variables

X_{ij} : número de unidades enviadas de i a j

Y_{ij} : número de unidades enviadas de j a k

$$\begin{aligned}
&\min \sum_{ij} c_{ij} X_{ij} + \sum_{jk} w_{jk} Y_{jk} \\
&\sum_{j} X_{ij} \le e_i \quad \forall i \\
&\sum_{j} Y_{jk} = d_k \quad \forall k \\
&\sum_{i} X_{ij} = \sum_{k} Y_{jk} \quad \forall j \\
&X_{ij}, Y_{jk} \ge 0 \quad X_{ij}, Y_{jk} \in \mathbb{Z}??
\end{aligned}$$

2. Nueva variable: $Z = \begin{cases} 1 & \text{le provee distribuidor 1} \\ 0 & \text{le provee distribuidor 2} \end{cases}$

   Añadir al modelo las restricciones:

$$\begin{aligned}
&Y_{11} \le MZ \\
&Y_{21} \le M(1-Z) \\
&Z \in \{0,1\}
\end{aligned}$$

   siendo, por ejemplo, $M = d_1$

3. Entiendo por el enunciado que las m primeras siempre cuestan lo mismo y las que se pasen son las que tienen descuento (no todas una vez se pasan).

   Nuevas variables

   N_{jk}, P_{jk} : exceso y defecto sobre el valor m

   $$\delta_{jk} = \begin{cases} 1 & \text{si } N_{jk} > 0 \\ 0 & \text{si } P_{jk} > 0 \end{cases}$$

   Modificar función objetivo añadiendo el término $-\sum_{jk} hN_{jk}$

   Añadir las restricciones:

   $$\begin{aligned} &Y_{jk} = m + N_{jk} - P_{jk} \quad \forall j,k \\ &N_{jk} \le M\delta_{jk} \quad \forall j,k \\ &P_{jk} \le M(1-\delta_{jk}) \quad \forall j,k \end{aligned}$$

   (Estas dos últimas son necesarias pues si no pueden ser distintas de 0 ambas a la vez, para un mismo valor de Y y lograr menor coste).

4. Sea $cap = \left\lfloor \frac{P}{p} \right\rfloor$ número de artículos que caben en un camión. Sean las variables:

   T_{ij} : número de camiones enviados de i a j

   S_{jk} T_{ij} : número de camiones enviados de j a k

   Modificar la función objetivo añadiendo el término $+C\left(\sum_{ij} T_{ij} + \sum_{jk} S_{jk}\right)$

   Añadir las restricciones:

   $$\begin{aligned} &X_{ij} \le cap\, T_{ij} \quad \forall i,j \\ &Y_{jk} \le cap\, S_{jk} \quad \forall j,k \end{aligned}$$

IV. Codificación de problemas de optimización

IV.1. Lenguajes de modelado

IV.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 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. Cabe también mencionar la iniciativa de desarrollo de software abierto para investigación operativa denominada *Computational Infrastructure for Operations Research* (COIN-OR) (www.coin-or.org).

- *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]. Frontline Systems (www.solver.com) ha desarrollado optimizadotes para Microsoft Excel.

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.dashoptimization.com) 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[4], estando los dos primeros basados en la programación de restricciones[5] y el último en programación matemática.

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 (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

[4] CPLEX es problablemente el mejor optimizador existente para problemas LP y MIP.

[5] 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.

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.

IV.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[6]
- 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 lo 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.

[6] 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.

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

- 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

IV.1.3. Referencias

Brooke, A., Kendrick, D., Meeraus, A. and Raman, R. (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.

IV.2. Casos de estudio con Excel

El Excel es una herramienta muy utilizada en gestión, aunque probablemente no sea la idónea para problemas de optimización. A pesar de todo, y dado lo extendido que está su uso, se muestran ejemplos para desarrollar con Excel, con el Complemento SOLVER. En Internet, se pueden encontrar complementos que mejoran fundamentalmente la escritura del modelo, pero no se distribuyen de forma gratuita con Excel, por lo que no se van a utilizar.

La implementación de un modelo en Excel se va a mostrar con un ejemplo.

IV.2.1. Caso Ejemplo

Un empresario ha de producir un compuesto basado en un determinado componente básico. Este componente puede ir en una cantidad variable entre el 50 y el 100% del total de la composición. Por otra parte del componente básico hay dos calidades, una calidad superior que cuesta 200€/unidad y otra que cuesta 100€. El mercado le obliga a que al menos un 20% ha de ser del componente de calidad superior. El empresario se plantea maximizar la calidad y minimizar el coste. Plantear un modelo para este problema (con los dos objetivos, aunque se traten por separado) e implementarlo en Excel.

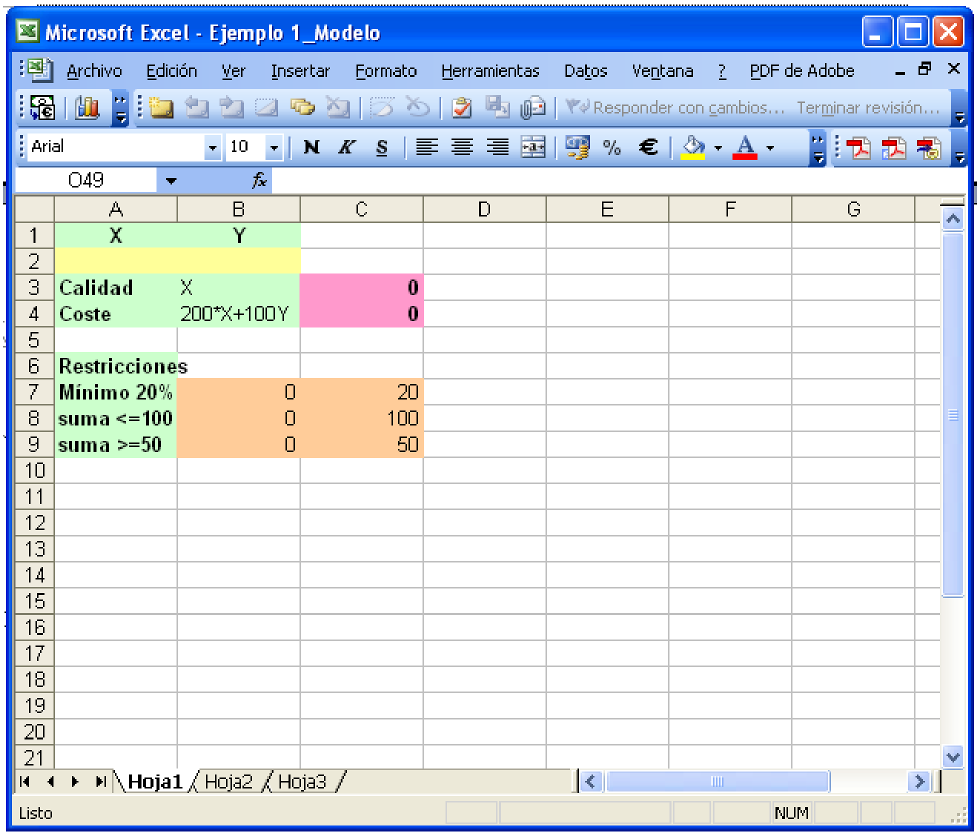
El siguiente modelo de optimización sirve para determinar el espacio de soluciones, donde *X* representa la calidad superior del componente e *Y* la calidad inferior:

$$
\begin{aligned}
& X \geq 20 \\
& X + Y \leq 100 \\
& X + Y \geq 50 \\
& X, Y \geq 0
\end{aligned}
$$

Por otra parte, los dos atributos planteados y su dirección de optimización, serían:

$$\text{Calidad:} \quad \max \quad X$$
$$\text{Coste:} \quad \min \quad 200X + 100Y$$

A continuación se muestra cómo formular este modelo de programación lineal en Excel. El archivo original se adjunta con el documento (Ejemplo 1_Modelo.xls)



El proceso para crearlo es el siguiente. En un archivo vacío de Excel, marcar dos celdas contiguas que contendrán los valores de las dos variables. En el ejemplo de la imagen, estas celdas se han marcado en amarillo para resaltarlas, y son las celda A2 y B2, para X e Y, respectivamente. Inicialmente están vacías pero contendrán la solución del modelo a resolver. Todas las celdas de fondo verde son meramente informativas.

A continuación, para cada objetivo, marcar una celda que contendrá la fórmula de ese objetivo como función de las variables. Por ejemplo, para el criterio Calidad, cuya expresión es la cantidad del componente 1, en el ejemplo se ha seleccionado la celda C3 donde se ha escrito $\boxed{\text{=A2}}$, ya que es la celda donde está el valor de X. Para el criterio

Coste, se ha introducido la expresión en la celda C4, = 200*A2+100*B2. Las celdas con las expresiones de los objetivos vienen resaltadas en rosa.

Para introducir cada una de las restricciones, se selecciona una celda con la expresión del lado izquierdo de la restricción, y otra con el lado derecho de la restricción. Por ejemplo, para indicar que del componente 1 ha de haber al menos un 20%, en la celda B7 se ha introducido =A2; y en la celda C7, se ha introducido 20. Para indicar que la suma de ambos componentes ha de ser a lo sumo el 100%, en la celda B8 se ha introducido =A2+B2, y en la celda C8 100 . Por último, para que la suma sea de al menos el 50%, se ha introducido en la celda B9 =A2+B2, y la celda C9 100 . Todas estas celdas están con fondo naranja.

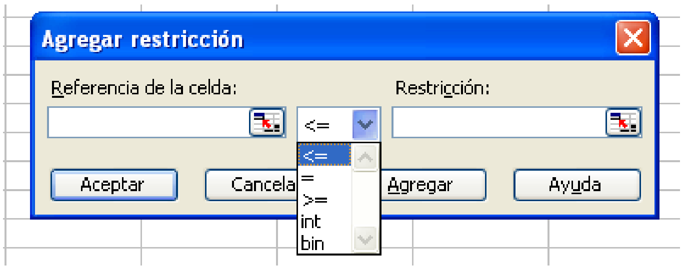
Con esto se han introducido los datos del modelo que se montará y resolverá con el complemento Solver. Este complemento viene por defecto en Excel, pero no siempre está activado. Si al ir al menú de Herramientas, no apareciera, haga click sobre Complementos, y en la pantalla que se abrirá, active la línea que pone Solver. Para montar el modelo, vaya al menú Herramientas y haga click sobre Solver. Aparecerá una pantalla como la siguiente, titulada Parámetros de Solver, pero vacía.

Donde pone Celda Objetivo, marque la celda donde está la expresión del objetivo que desea optimizar. En la imagen está C3 (por defecto Solver lo pone con dólares entre medias), que es que se va a optimizar la Calidad, cuya expresión pusimos en esa celda.

Donde pone Valor de la celda objetivo, puede marcar Máximo si desea maximizar ese atributo, Mínimo si desea minimizar, y otra opción no relevante para nuestro propósito. En este caso, siendo Maximizar la Calidad el criterio, marcamos Máximo.

Después pregunta acerca de las celdas que contienen el valor de las variables. En ese cuadro debe marcar las celdas donde están las variables, en nuestro caso A2 y B2.

En cuanto a las restricciones, pulse Agregar para ir añadiéndolas. Observará que se abre una ventana como la que sigue:



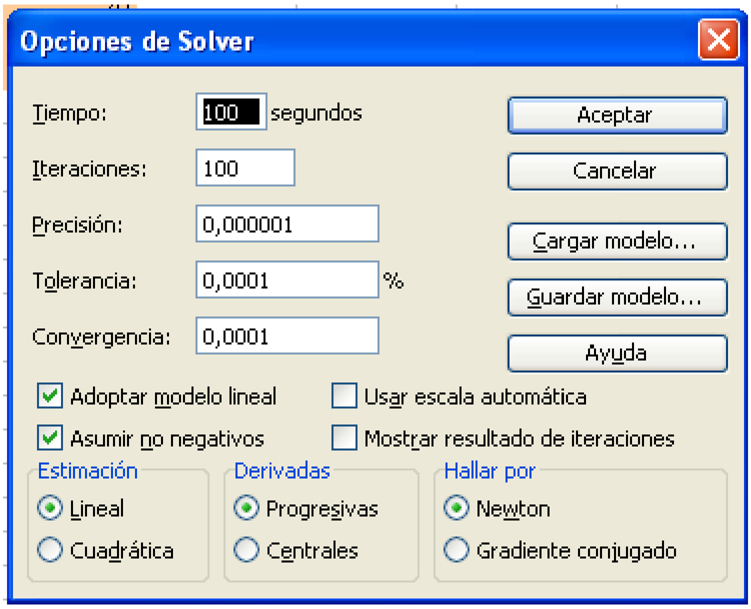
Donde pone Referencia de la celda, deber marcar la celda con la expresión del lado izquierdo de la restricción. Donde pone Restricción, la celda que recoge el lado derecho de la restricción. Y en medio, da a elegir entre 5 opciones: <= para restricciones de menor o igual, = para restricciones de igualdad, >= para restricciones de mayor o igual, int para que las variables marcadas en el lado izquierdo tengan que ser variables que sólo toman valores enteros, y bin para indicar que las variables marcadas en el lado izquierdo sólo puedan tomar valores 0 o 1.

En el ejemplo, la primera restricción (de la componente 1 ha de haber al menos un 20%) la hemos contemplado en las celdas B7 y C7, y es una restricción de mayor o igual con lo que debemos escribir:

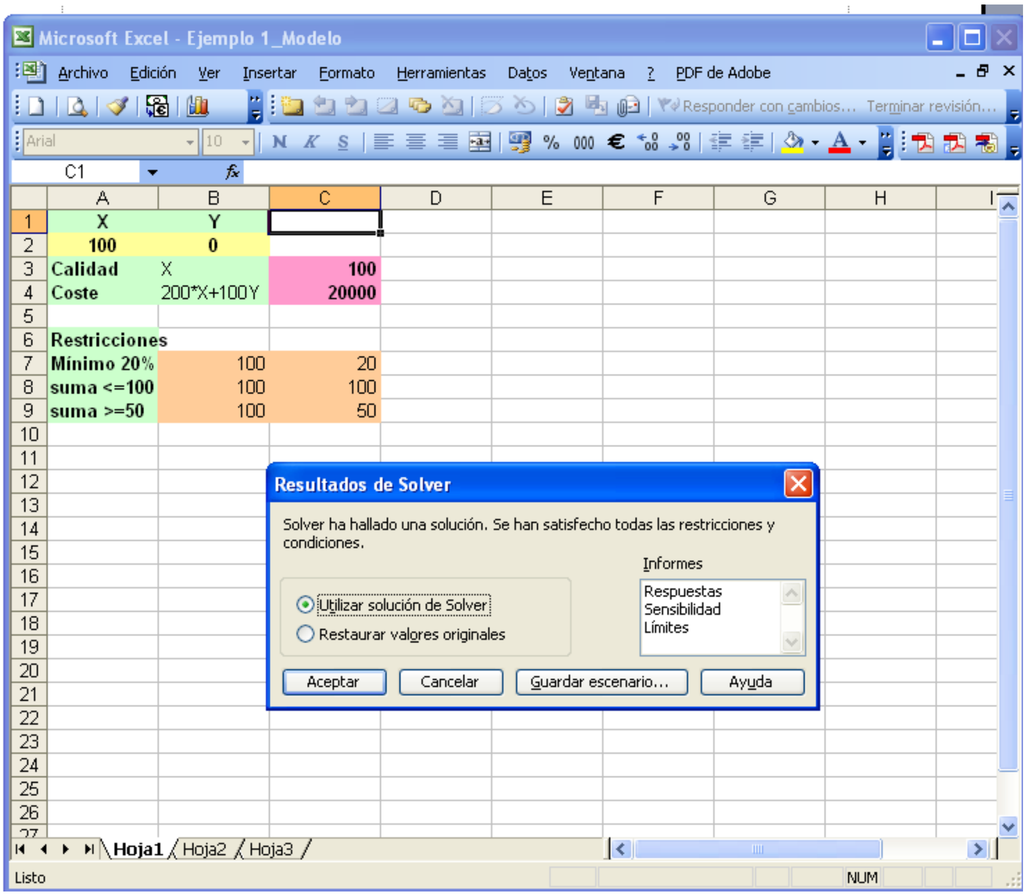
Y así sucesivamente, pulsando el botón de Agregar para ir añadiendo restricciones, con B8 <= C8, y B9 >= C9. Pulsando el botón Aceptar volvemos a la pantalla de Parámetros de Solver donde veremos las restricciones añadidas. Siempre se pueden

Agregar nuevas restricciones, Eliminar alguna ya creada, o Modificar alguna de ellas. Para nuestro ejemplo, tendríamos la siguiente pantalla de Parámetros.

Todavía, para terminar de montar el modelo, debemos pulsar en el botón Opciones, en el que se abrirá la siguiente pantalla:

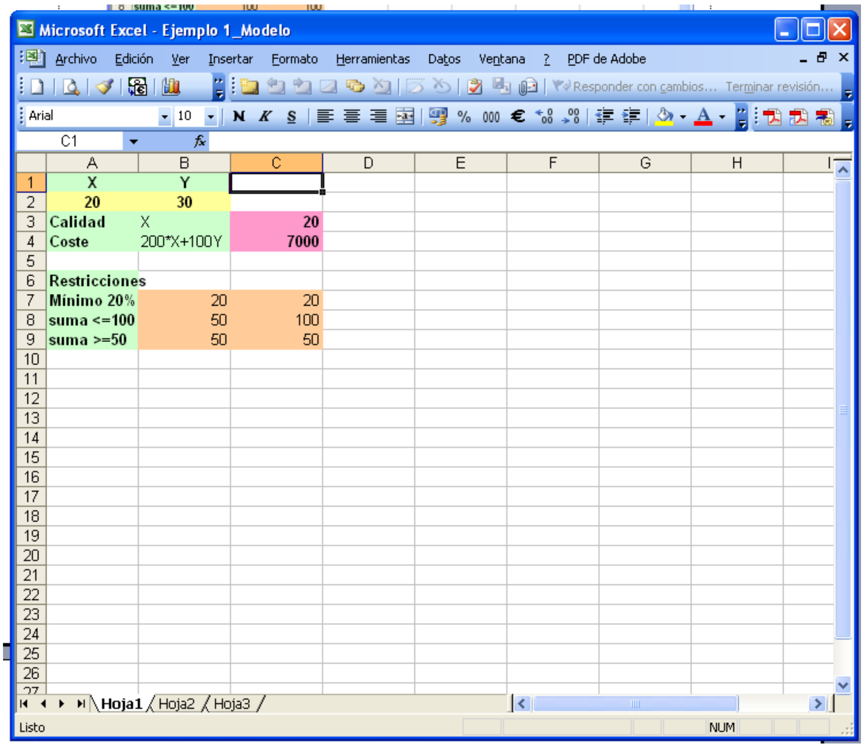


Como norma general, son parámetros bastante técnicos, pero hay dos que es necesario tocar, y alguno que es conveniente. Los que son necesarios es marcar Adoptar modelo lineal, y Asumir no negativos (especialmente éste último, ya que es donde indicamos que las variables sólo pueden tomar valores positivos). También es recomendable reducir algunos de los números que aparecen por defecto (los de Precisión, Tolerancia y Convergencia), pero no se puede llegar a poner 0 que sería lo ideal. Pulsar Aceptar para tener todo el modelo y volver a la pantalla de Parámetros de Solver, donde pulsando Resolver, resolverá el modelo dando los resultados en las celdas indicadas, después de emitir un mensaje en que dice: Solver ha hallado una solución. Se han satisfecho todas las restricciones y condiciones. Y pregunta si se quiere utilizar la solución de Solver, a lo que se debe decir Aceptar.



La cantidad de componente 1 (variable X), es 100 (celda A2), y del componente 2 (variable Y), es 0 (celda B2). La calidad obtenida es 100 (celda C3), que es la máxima posible, y el coste para esa solución es 20000 (celda C4).

Este modelo creado aparecerá siempre que se pulse Solver, y se puede modificar el objetivo simplemente marcando la otra celda objetivo (C4) en el primer parámetro de Solver, cambiando a Mínimo para minimizar el coste, y pulsando Resolver.



En este caso, la solución es 20% del componente 1, y 30 del componente 2. Obteniendo una calidad de 20, y un coste de 7000.

El archivo Excel que incluye las soluciones de los casos de estudio es Ejemplos MO_MIM.xls

IV.2.2. Caso 1: Distribución de gasóleo

Una empresa de distribución de gasóleo dispone de tres depósitos D1, D2 y D3 donde guarda el gasóleo y desde los que abastece a cuatro estaciones de servicio, E1, E2, E3 y E4. En la tabla se muestra la capacidad máxima de almacenamiento de cada depósito, la demanda máxima de cada punto de venta y las capacidades máximas de transporte en las posibles rutas entre depósitos y estaciones de servicio.

	E1	E2	E3	E4	Capacidad
D1	80	–	70	–	150
D2	–	60	90	85	300
D3	40	60	–	50	250
Demanda	130	200	150	250	

Además, cada unidad que se envíe a las gasolineras le supone un beneficio de 7 u.m. y los envíos se llevan a cabo en camiones cuya capacidad es de 20 unidades cada uno, con un coste estimado por camión en cada una de las rutas de:

Costes (u.m.)	E1	E2	E3	E4
D1	100	–	125	–
D2	–	100	125	100
D3	75	100	–	125

Formular y resolver un problema para determinar el plan de rutas óptimo que maximice el beneficio de la empresa.

SOLUCIÓN:

X_{ij} : cantidad enviada de depósito i a estación j

Y_{ij} : camiones usados de depósito i a estación j

$$\max \sum_{i,j} \left(7X_{ij} - c_{ij}Y_{ij}\right)$$

$$s.a. \quad \sum_{j} X_{ij} \le cap_i \quad \forall i$$

$$\sum_{i} X_{ij} \le dem_j \quad \forall j$$

$$X_{ij} \le capac_{ij} \quad \forall i, j$$

$$X_{ij} \le 20Y_{ij} \quad \forall i, j \qquad X_{ij}, Y_{ij} \ge 0, Y_{ij} \in \mathbb{Z}$$

Objetivo: 825

Cantidades	E1	E2	E3	E4
D1	80	0	60	0
D2	0	60	80	80
D3	40	60	0	40

Camiones	E1	E2	E3	E4
D1	4	0	3	0
D2	0	3	4	4
D3	2	3	0	2

IV.2.3. Caso 2: Conductores de metro

La Compañía Metropolitana de Transporte (CMT) explota el metro de la ciudad y planifica la asignación de turnos de su personal de operación. El convenio laboral exige que cada conductor disponga de dos días libres consecutivos por semana. La demanda de conductores para operar los trenes varía dependiendo del día, siendo, de lunes a domingo, de 18, 16, 15, 16, 19, 14 y 12 conductores respectivamente. El coste de personal varía a lo largo de la semana. Para un día cualquiera de lunes a viernes el coste es de 50 € por día, los sábados se pagan a 75 y los domingos a 90.

Además se puede recurrir a la contratación parcial de hasta 3 personas que trabajarían los viernes, sábados y domingos por un coste de 200 €. Determinar el esquema óptimo de turnos de trabajo de dichos conductores.

SOLUCIÓN:

X_i : número de conductores que empiezan su turno el día i

T_i : conductores que trabajan el día i

Y : número de personas contratadas a tiempo parcial

$$\min \quad \sum_i c_i T_i + 200Y$$

$$s.a. \quad T_i = \sum_{j \neq i-1, i-2} X_j \geq nec_i \quad \forall i \neq 5,6,7$$

$$T_i = \sum_{j \neq i-1, i-2} X_j + Y \geq nec_i \quad \forall i = 5,6,7$$

$$Y \leq 3$$

$$X_i, T_i, Y \geq 0$$

COSTE: 6730

L	M	X	J	V	S	D	Extra
8	2	2	4	3	3	0	0

IV.2.4. Caso 3: Producción

Una empresa puede fabricar 4 productos diferentes utilizando en su elaboración 5 tipos de materias primas. Las unidades requeridas de cada materia prima por cada unidad de cada producto se recogen en la tabla siguiente, así como el precio de venta de cada producto y la disponibilidad de cada materia prima:

	M1	M2	M3	M4	M5	Precio venta
P1	1	1		1		15
P2		1		1	1	20
P3	1	1	1			15
P4	1		1			10
Disponible	200	150	150	200	100	

El precio unitario de cada materia prima es de 2, 3, 4, 5 y 5, respectivamente, pero hay un coste adicional por hacer un pedido de cada materia prima, independiente de la cantidad que se pida, de 20, 25, 20, 25, 25 unidades, respectivamente. Determinar el plan de producción óptimo para maximizar los beneficios de la empresa.

SOLUCIÓN:

X_i : unidades producidas de producto Pi

$$M_j : \begin{cases} 1 & \text{si pide materia prima j} \\ 0 & \text{si no pide materia prima j} \end{cases}$$

$$\max \sum_i pv_i X_i - \sum_{i,j} a_{ij} c_j X_i - \sum_j f_j M_j$$

$$s.a. \quad \sum_i a_{ij} X_i \leq disp_j M_j \quad \forall j$$

$$X_i \geq 0, M_j \in \{0,1\}$$

Beneficio	1435
P1	50
P2	100
P3	0
P4	150

IV.3. Modelado en GAMS

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.

IV.3.1. 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)

      + 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
```

IV.3.2. 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 (t 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 t)
                 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
*              (t por mes)(€ por t)
      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 (t 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 (t 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
```

IV.3.3. 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
```

IV.3.4. 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

$$\begin{aligned}
&\min_{x_{ij},u_i} \sum_{i,j} c_{ij} x_{ij} \\
&\sum_i x_{ij} = 1 \quad \forall j \\
&\sum_j x_{ij} = 1 \quad \forall i \\
&u_i - u_j + [\mathrm{card}(i) - 1] x_{ij} \le \mathrm{card}(i) - 2 \quad \forall i,j \\
&x_{ij} \in \{0,1\}, u_i \ge 0
\end{aligned} \tag{1.15}$$

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
```

IV.3.5. 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
```

IV.3.6. 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 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 $\overline{F_{i\to j}}$.

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

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

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

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

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

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

En la primera formulación se tienen más variables pero menos restricciones[7] 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.

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

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

[7] 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.

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]
   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  / ;
* FC y FCA son dos formulaciones alternativas.
* Para este caso de estudio la segunda es menos costosa computacionalmente (un 10 %).
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
   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 ;
```

IV.4. Elementos de estilo de programación

> "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).

IV.4.1. 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*[8] 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[9] y la escritura de resultados deben separarse en diferentes ficheros que son convenientemente insertados mediante la instrucción `$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.

---

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

[9] 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.

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.

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

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

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

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

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[10] 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:

- 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.

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

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 (2003), 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.

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).

IV.4.2. 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[11]:

- 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 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

[11] 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.

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[12] 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[13] 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

[12] 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.

[13] 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.

Universidad Complutense Madrid

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 del flujo de cargas obsérvese que se consideran sólo las líneas eléctricas que existen y no cualquier posible conexión.

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[14]. 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

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)

[14] 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.

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.

IV.4.3. 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. (2003) "Linear Programming" *OR/MS Today*. December.

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., Muñoz, L., Rupérez, I., Martínez-Córcoles, F. and Martín-Corrochano, V. (1996) "Computational Experience with Optimization for a Bulk Production Cost Model" *12th PSCC*. Dresden, Germany.