UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE QUÍMICA

APLICACIÓN DE GAMS PARA RESOLVER PROBLEMAS DE OPTIMIZACIÓN

TESIS PROFESIONAL, MANCOMUNADA

QUE PARA OBTENER EL TÍTULO DE

INGENIERO QUÍMICO

PRESENTAN

CÉSAR ALBERTO NOGUEZ GARCÍA

LILIANA RIVERA ACEVEDO

MÉXICO, D.F.

2004

de la
so el
al.
Liliana Rivera Acevedo

9-Julio-2004
LILIANA RIVERA A.

Jurado asignado:

Presidente	Prof. Norma Gisela González Mariscal
Vocal	Prof. Celestino Montiel Maldonado
Secretario	Prof. Martín Rivera Toledo
1er. Suplente	Prof. Juan Carlos Jiménez Bedolla
2do. Suplente	Prof. Cesar Reyes Chávez

Sitio donde se desarrolló el tema: Facultad de Química, Edificio "D"

Asesor:

Ing. Martín Rivera Toledo

Sustentantes:

César Alberto Noguez García

Liliana Rivera Acevedo LILIANA RIVERA A.

de la
so el
al.
César Alberto Noguez
García
9-Julio-2004

A NUESTROS PADRES

Como mínimo tributo a su esfuerzo para
ver culmiada esta meta.

A LA FACULTAD DE QUÍMICA Y A LA
UNIVERISIDAD NACIONAL AUTÓNOMA DE MÉXICO

Por permitirnos ser parte y esencia de la misma

A NUESTRO ASESOR

Por sus observaciones y consejos.

A NUESTROS MAESTROS

Que en el transcurso de nuestros estudios
han dejado su huella en nosotros.

A todos los que de una forma u otra nos
ayudaron a concluir la meta fijada.
GRACIAS

ÍNDICE

1. *INTRODUCCIÓN*

Los modelos de programación han tenido un gran desarrollado en las áreas de Investigación operacional y de Ingeniería. Esto se debe principalmente por la elaboración de algoritmos de optimización capaces de resolver grandes problemas de optimización, por un lado, y a la evolución de los simuladores, por el otro.

Además, los simuladores de proceso en la actualidad se han convertido en las herramientas básicas en proyectos de ingeniería química. Dichos proyectos implican típicamente el uso de un simulador comercial o de un simulador académico tales como ASPENPLUS, ChemiCAD, CheShare, FLOWTRAN, HYSYS y ProIIw/PROVISION. Muchos libros de diseño, incluyen ejercicios preparados específicamente para un simulador particular. Por ejemplo, el texto escrito por Seider, Seader y el Lewin (2003) contiene ejemplos escritos para el uso con ASPENPLUS, HYSYS, GAMS y DYNAPLUS

Por esta razón, decidimos desarrollar un material de apoyo enfocado a GAMS (General Algebraic Modeling System), el cual lleva como título "Aplicación de GAMS para resolver problemas de optimización"; teniendo como objetivos:

- Recopilar material de apoyo para uso del programa GAMS como herramienta para resolver problemas de optimización.

- Aplicar las técnicas de programación matemática para resolver problemas de optimización enfocándose a la utilización del programa GAMS.

El presente trabajo presenta los temas siguientes:

Optimización: En este capitulo se explica de manera breve el concepto de optimización, tratando los puntos principales de un problema de optimización y los tipos de problemas de optimización que se pueden presentar.

Modelos de optimización: Este capitulo comienza dando el significado del modelo, por lo cual esta enfocado a tratara exclusivamente modelos de optimización, es decir, aquellos modelos donde existe un conjunto de variables de decisión que deben maximizar/minimizar una función objetivo sometidas a un conjunto de restricciones.

Se explican las etapas en el desarrollo de un modelo, así mismo se indican los aspectos importantes en el desarrollo de un modelo.

Codificación de problemas de optimización: En este capitulo trataremos las principales alternativas para el desarrollo de modelos de optimización.

Se incluye brevemente la historia del lenguaje de modelación, así como las principales alternativas para el desarrollo de modelos de optimización.

Por otra parte se tocara el tema de las características, ventajas y desventajas principales de los lenguajes algebraicos de modelado.

Dentro de este mismo capitulo se explicara el uso del programa GAMS como herramienta para resolver problemas de optimización.

Así mismo se anexa una serie de problemas de ingeniería resueltos en GAMS.

2. *OPTIMIZACIÓN*

2.1 Investigación de operaciones y optimización

Definir el término investigación de operaciones 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. Otra definición más extensa es la siguiente: la investigación operativa es la aplicación, por grupos interdisciplinarios, del método científico a los problemas complejos producidos en la dirección y gestión de grandes sistemas de hombres máquinas, etc. La principal característica consiste en construir un modelo científico del sistema del cual se pueden predecir y comparar los resultados de diversas estrategias, decisiones, incorporando medidas del azar y del riesgo. El objetivo es ayudar a los responsables a determinar su política y actuaciones en forma científica.

Los profesionales de la investigación operativa colaboran con las decisiones 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.

La optimización es una parte relevante dentro de la investigación de operaciones. 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 de programación lineal debido a Dantzig es de 1947, el principio de optimalidad de Bellman base de la programación dinámica se formuló en 1957. En la última década se han producido avances significativos generados por el desarrollo en 1984 por parte de Karmarkar de un método de punto interior para programación lineal. Por ejemplo, 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 150000 ecuaciones con 150000 variables y 1000000 de elementos no nulos en la matriz de restricciones en un PC con suficiente memoria principal.

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.

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.

2.2 Clasificación de los métodos de optimización

El problema general de optimización posee:

- *Función objetivo*
  Es el indicador de comportamiento del sistema, que deseamos optimizar, es decir, maximizar o minimizar, el cual puede depender de costo, rendimiento, producción, ganancia, etc. Como ejemplo de funciones objetivo se pueden mencionar: la minimización de los costes variables de operación de un sistema eléctrico, la maximización de los beneficios netos de venta de ciertos productos, la minimización del cuadrado de las desviaciones con respecto a unos valores observados, la minimización del material utilizado en la fabricación de un producto, etc.

- *Variables*
  Conjunto de parámetros que influyen en el comportamiento del proceso y puede ser ajustada por optimización. 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 proceso serán las condiciones de operación, por ejemplo la temperatura, presión, flujo, viscosidad, densidad, composición de la mezcla, la longitud y radio del reactor, entre otros. 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

  Son funciones que acotan la región factible para la identificación del óptimo. Por ejemplo, las potencias máximas y mínimas 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.

- Región factible

  Es el conjunto de puntos en los cuales todas las restricciones se cumplen. La región factible, como se muestra en las siguientes figuras puede ser: acotada, no acotada o vacío, es decir, que no haya ni un solo punto que verifique todas las restricciones al mismo tiempo.

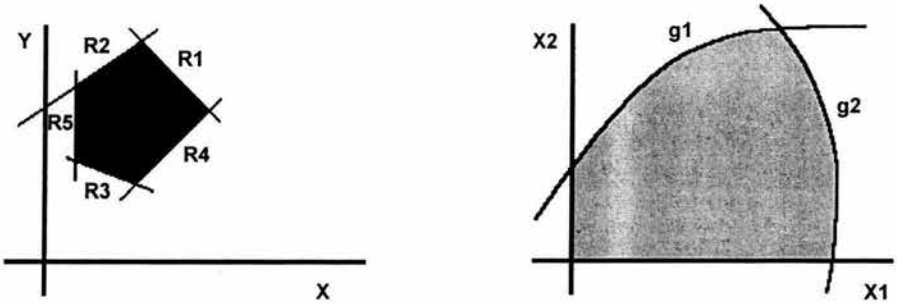


Fig. A Región factible acotada

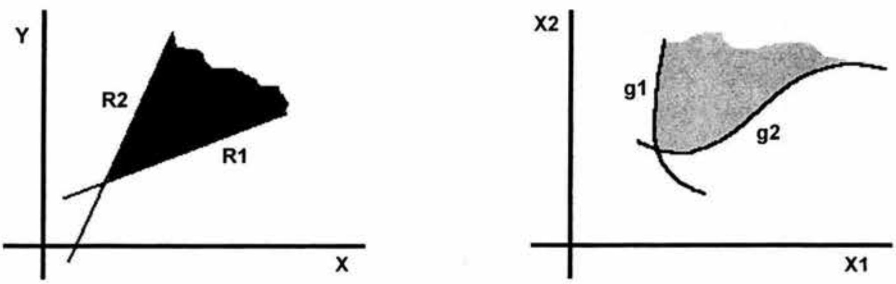


Fig. B Región factible no acotada

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.

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.

  La forma matemática general para este tipo de problema de optimización es:

  Programación Lineal (LP)

$$\begin{aligned} &\min_{x} c^T x \\ &Ax = b \\ &x \geq 0 \\ &x \in \boldsymbol{R}^n, c \in \boldsymbol{R}^n, A \in \boldsymbol{R}^{m \times n}, b \in \boldsymbol{R}^m \end{aligned}$$

  Programación No Lineal (NLP)

$$\begin{aligned} &\min_{x} f(x) \\ &g(x) = 0 \\ &h(x) \leq 0 \\ &l \leq x \leq u \\ &f : \boldsymbol{R}^n \rightarrow \boldsymbol{R} \\ &g,h : \boldsymbol{R}^n \rightarrow \boldsymbol{R}^m \end{aligned}$$

- 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.

  La forma matemática general para los problemas de optimización no lineal sin restricciones es:

Optimización No Lineal sin restricciones

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

$$f : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$$

- 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.

La forma matemática general para los problemas de optimización multiobjetivo es:

Programación multiobjetivo

$$\min_{x} (f_1(x), \ldots, f_k(x))$$

$$Ax = b$$

$$x \geq 0$$

$$x \in \boldsymbol{R}^n,\ c \in \boldsymbol{R}^n,\ A \in \boldsymbol{R}^{mxn},\ b \in \boldsymbol{R}^m$$

$$f_i(x) : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$$

Los métodos de optimización se clasifican en: métodos *clásicos* (que son los que habitualmente se explican en los libros de optimización) y métodos *metaheurísticos* (que aparecieron ligados a lo que se denominó inteligencia artificial). 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)* o las búsquedas heurísticas (método tabú, búsqueda aleatoria, etc.). 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	$\min_x c^T x$ $Ax = b$ $x \geq 0$ $x \in \mathbf{R}^n, c \in \mathbf{R}^n, A \in \mathbf{R}^{m \times n}, b \in \mathbf{R}^m$
Programación lineal entera mixta *(mixed integer programming)* MIP	$\min_x c^T x + d^T y$ $Ax + By = b$ $x, y \geq 0$ $x \in \mathbf{Z}^n, y \in \mathbf{R}^l, c \in \mathbf{R}^n, d \in \mathbf{R}^l,$ $A \in \mathbf{R}^{m \times n}, B \in \mathbf{R}^{m \times l}, b \in \mathbf{R}^m$
Programación cuadrática *(quadratic programming)* QP	$\min_x c^T x + \frac{1}{2} x^T Q x$ $Ax = b$ $x \geq 0$ $x \in \mathbf{R}^n, c \in \mathbf{R}^n, A \in \mathbf{R}^{m \times n}$ $Q \in \mathbf{R}^{n \times n}, b \in \mathbf{R}^m$
Programación no lineal *(non linear programming)* NLP	$\min_x f(x)$ $g(x) = 0$ $h(x) \leq 0$ $l \leq x \leq u$ $f : \mathbf{R}^n \longrightarrow \mathbf{R}$ $g,h : \mathbf{R}^n \longrightarrow \mathbf{R}^m$
Programación multiobjetivo *(multiobjective programming)*	$\min_x (f_1(x), \ldots, f_k(x))$ $Ax = b$ $x \geq 0$ $x \in \mathbf{R}^n, c \in \mathbf{R}^n, A \in \mathbf{R}^{m \times n}, b \in \mathbf{R}^m$ $f_i(x) : \mathbf{R}^n \longrightarrow \mathbf{R}$

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) o binarias (como localización de plantas o almacenes). Estos problemas se denominan, genéricamente, de programación lineal entera mixta, son problemas *lineales* donde algunas o todas las variables son enteras. Los problemas lineales con variables enteras se pueden clasificar en: PIP *(pure integer programming)* si todas las variables son enteras. BIP *(binary integer programming)* si todas son binarias o 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*, $[0,1] \in Z^n$, 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 la cota superior de x, $0 \leq x \leq u$, y estando u comprendida en el intervalo $2^N \leq u \leq 2^{N+1}$.

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

Optimización no lineal sin restricciones	$\min_x f(x)$ $f: R^n \rightarrow R$
Ajuste no lineal mínimo cuadrático	$\min_x f(x) = \sum_{i=1}^{N} F_i^2(x)$ $F_i: R^n \rightarrow R$
Problema mixto complementario *(mixed complementarity problem)* MCP	$\min z = \underline{c}^T x$ sujeta a $[A]\, x \leq \underline{b}$ $\max w = \underline{b}^T y$ sujeta a $[A]^T y \geq \underline{c}$

2.2.1 Técnicas de optimización

Las técnicas de optimización que se usan en la modelización de sistemas para los cuales se conoce un modelo expresado en forma de un algoritmo o una función matemática, se dividen en dos: Métodos variacionales y Métodos de Programación Matemática.

Los Métodos variacionales se basan en el principio variacional, y pretenden partiendo de una función de prueba, llegar a una función lo más próxima posible a la solución exacta del sistema en cuya solución estamos interesados. Se utilizan para resolver modelos de programación lineal.

Los Métodos de Programación Matemática, son los más utilizados en Ingeniería Química, ya que nos permiten resolver modelos de programación no lineal. Dentro de estos Métodos de Programación Matemática los más usuales son:

Multiplicadores de Lagrange

Este método nos permite resolver un problema con extremos restringidos, es decir, construye una función sin restricciones de modo que los óptimos de la función original se encuentran en los puntos críticos de la función de Lagrange.

Supóngase que queremos Optimizar $f(x_1, x_2,..., x_n)$ sujeto a $g_i (x_1,...,x_n)=b_i$, $i=1,...,m$ $(m<n)$. Introducimos una nueva variable λ_i llamada multiplicador de Lagrange por cada restricción g_i y formamos la función auxiliar L para la cual

$$L(x,\lambda) = f(x_1,\ldots,x_n) - \sum_{i=1}^{m} \lambda_i \left(g_i(x_1,\ldots,x_n) - b_i\right)$$

El problema, entonces, consiste en determinar los puntos críticos del lagrangiano ($\mathbf{x}^*$, λ^*).

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} = 0, \quad j = 1,\ldots,n$$

$$\frac{\partial L}{\partial \lambda_j} = b_j - g_j = 0, \quad j = 1,\ldots,m$$

Método del Gradiente Reducido Generalizado

El método consiste en resolver el sistema de ecuaciones no lineales que representan las restricciones para un subconjunto de K variables y luego generar un nuevo problema reducido en función de las N - K.

Supóngase que queremos minimizar una función no lineal con restricciones no lineales de igualdad. Esto es:

Minimizar f (x)

s.a. $h_k (x) = 0$, $k = 1, \ldots, K$

donde K es el número de restricciones del problema, $x = (x_1, x_2, \ldots, x_n)$ es el vector de variables de optimización y N es el número de variables del problema, siendo K < N.

Acontinuación se calcula el gradiente de f(x) reducido, $\nabla\tilde{f}(x)$ y se efectúa una búsqueda en línea en la dirección obtenida. El procedimiento se repite hasta lograr que $\|\nabla\tilde{f}(x)\|_2$ sea suficientemente pequeña.

Supongamos que se aproxima linealmente el sistema $h_k(x) = 0$ para k = 1, ..., K. Entonces, si x_1 es una solución del sistema, resulta

$$\tilde{h}_k(x) = h_k(x_1) + \nabla h_k(x_1)(x - x_1)$$

para k=1, ..., K. Si x_2 es otro punto factible, entonces,

$$\nabla h_k(x_1)(x - x_1) = 0$$

para k=1, ..., K.

Como este sistema tiene más incógnitas que ecuaciones, la técnica resuelve para K variables en términos de la N – K restantes. Para esto se parte al vector x en dos conjuntos de variables: $\hat{x} \in \Re^K$ contiene las variables denominadas básicas y $\bar{x} \in \Re^{N-K}$ contiene las restantes llamadas no básicas. Si de manera acorde se parte a $\nabla h(x)$ en $\nabla\hat{h}(x)$ y $\nabla\bar{h}(x)$, esto es $\nabla h_k(x) = (J(x)|C(x))$, donde

$$J(x) = \begin{pmatrix} \nabla\hat{h}_1(x) \\ \nabla\hat{h}_2(x) \\ \vdots \\ \nabla\hat{h}_K(x) \end{pmatrix} = \nabla\hat{h}(x)$$

$$C(x) = \begin{pmatrix} \nabla\bar{h}_1(x) \\ \nabla\bar{h}_2(x) \\ \vdots \\ \nabla\bar{h}_K(x) \end{pmatrix} = \nabla\bar{h}(x)$$

se puede despejar el sistema.

Suponiendo que se eligió una partición de x tal que J(x) sea no singular, se pueden despejar las variables básicas en función de las no básicas así

$$\hat{x} - \hat{x}_1 = -J^{-1}(x_1)C(x_1)(\bar{x} - \bar{x}_1)$$

Con esto se puede eliminar variables en la función f que se quiere minimizar, quedando

$$\tilde{f}(\hat{x}|\bar{x}) = f(x_1 - J^{-1}(x_1)C(x_1)(\bar{x} - \bar{x}_1)\bar{x})$$

de donde por la regla de la cadena resulta

$$\nabla\tilde{f}(x_1) = \nabla \bar{f}(x_1) - \nabla \hat{f}(x_1)J^{-1}(x_1)C(x_1)$$

donde $\nabla \hat{f} = (\partial f / \partial \hat{x})$ y $\nabla \bar{f} = (\partial f / \partial \bar{x})$.

Este vector se llama gradiente reducido.

<u>Método de Newton-Raphson</u>

Está basado en el uso de una línea tangente como aproximación de f(x), cerca de los puntos donde el valor de la función es cero.

1.- Escoger un número inicial (x_0)
2.- Calcular la siguiente aproximación de x_1 utilizando la fórmula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

3.- Si $| x_n - x_{n+1} | < \varepsilon$ entonces x_{n+1} es una raíz, de otra forma pasar al punto 2.

2.2.2 Condiciones necesarias y suficientes para un óptimo

En esta sección mencionaremos brevemente las condiciones necesarias y suficientes para que **x*** sea un mínimo local de un problema general de programación no lineal con restricciones.

$$\text{Minimizar } f(\mathbf{x}) \qquad \mathbf{x} = [\, x_1 \;\; x_2 \;\; \ldots \;\; x_n]^T$$

$$\text{s.a.}$$

$$h_j(\mathbf{x}) = 0 \qquad j = 1, \ldots, m$$

$$g_j(\mathbf{x}) \geq 0 \qquad j = m+1, \ldots, p$$

Las *condiciones necesarias* para que $\mathbf{x}^*$ sea un mínimo local de $f(\mathbf{x})$ son:

(a) $f(\mathbf{x})$, $h_j(\mathbf{x})$, $g_i(\mathbf{x})$ tienen segunda derivada en $\mathbf{x}^*$

(b) La llamada "restricción de calificación de segundo orden", sostiene que; las suficientes condiciones para este requisito son que los gradientes de las restricciones obligatorias ($g_j(\mathbf{x}^*)= 0$), $\nabla g_j(\mathbf{x}^*)$, y las restricciones de igualdad, $\nabla h_j(\mathbf{x}^*)$, porque $h_j(\mathbf{x}^*) = 0$, son independientemente lineales

(c) Los multiplicadores de Lagrange existen; si están sujetos a (b)

(d) Las restricciones son eficaces en $\mathbf{x}^*$ si:

$$h_j(\mathbf{x}^*) = 0$$
$$g_i(\mathbf{x}^*) \geq 0$$

(e) Los multiplicadores de Lagrange u_j^* (en $\mathbf{x}^*$) para las restricciones de desigualdad no son negativos (ω_j puede ser positivo o negativo)

$$u_j^* \geq 0$$

(f) Las restricciones de desigualdad activas siempre serán igual a cero; las restricciones de desigualdad inactivas son > 0 y la asociación de u_j´s son 0 en $\mathbf{x}^*$

$$u_j^* g_j(\mathbf{x}^*) = 0$$

(g) La función de Lagrange está en un punto estacionario

$$\nabla L_x(\mathbf{x}^*, \boldsymbol{\omega}^*, \mathbf{u}^*) = 0$$

(h) La Matriz Hessiana de L esta semidefinida como positiva para aquellas **v**´s, para las cuales $\mathbf{v}^T \nabla g_j(\mathbf{x}^*)= 0$, y $\mathbf{v}^T \nabla h_j(\mathbf{x}^*)= 0$, es decir, para todas las restricciones activas

$$\mathbf{v}^T \nabla^2 [\, L(\mathbf{x}^*, \boldsymbol{\omega}^*, \mathbf{u}^*)]\mathbf{v} \geq 0$$

Las *condiciones suficientes* para que **x*** sea un mínimo local son:

1.- Debe cumplir con las condiciones necesaria (a), (b), (c), (d), (e), (f), y (g)
2.- Mas la condición necesaria (h) pero modificada de la siguiente manera:

La Matriz Hessiana de L se define positiva para aquellos vectores **v**, tales que

$$\left.\begin{array}{l} \mathbf{v}^T \nabla g_j(\mathbf{x}^*) = 0 \\ \mathbf{v}^T \nabla h_j(\mathbf{x}^*) = 0 \end{array}\right\} \quad \text{para las restricciones obligatorias}$$

$$\mathbf{v}^T \nabla g_j(\mathbf{x}^*) \geq 0 \qquad \text{para las restricciones inactivas}$$

$$\mathbf{v}^T \nabla^2 [\, L(\mathbf{x}^*, \boldsymbol{\omega}^*, \mathbf{u}^*)]\mathbf{v} > 0$$

3. MODELOS DE OPTIMIZACIÓN

3.1 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.

Un modelo es una representación matemática simplificada de una realidad compleja. Modelar es la acción de construir un modelo, lo que implica la relación entre dos grupos: 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 multidisciplinario 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, aquéllos 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, diseño de procesos, forestal, agrónoma, comercio internacional, desarrollo económico. etc.

En Ingeniería Química se utilizan más los modelos de programación no lineal, debido a que es común manejar restricciones de desigualdad, sobretodo en el diseño de proyectos.

3.2 Etapas en el desarrollo de un modelo

En 1988, Kapur, indica que los principios para el modelo matemático son:

- El modelo matemático puede ser solo una aproximación de los procesos reales, los cuales son a menudo muy complejos y a veces parcialmente comprendidos. Es posible desarrollar varios modelos diferentes para un proceso y tener la posibilidad de mejorar la descripción global del mismo.

- El modelado es un proceso de desarrollo continuo. Se empieza a desarrollar un modelo simple hasta irlo haciendo más complejo. La forma final del modelo deberá proporcionar una descripción razonable del proceso y debe ser capaz de ser usado.

- Un modelo matemático es un arte, pero también un proceso muy importante de aprendizaje. El proceso de modelado sugiere también la necesidad de menos datos o necesidad de la experimentación para descubrir varios aspectos del comportamiento del proceso, que no están bien comprendidos.

- Los modelos deben ser robustos y reales.

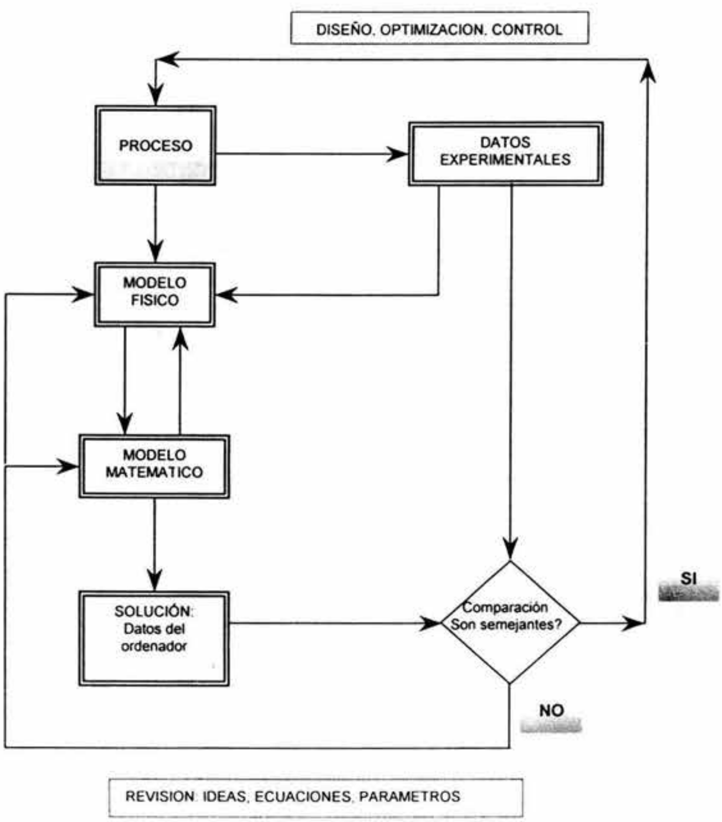
La persona que modela un proceso esta obligado a considerar la naturaleza de todos los parámetros importantes del proceso, sus efectos, el mismo y como cada parámetro puede ser definido en términos cuantitativos.

Una vez establecido el modelo, entonces puede ser utilizado, con relativa confianza, para predecir el comportamiento del proceso bajo diferentes condiciones y usarlo para el diseño del proceso, su optimización y control.

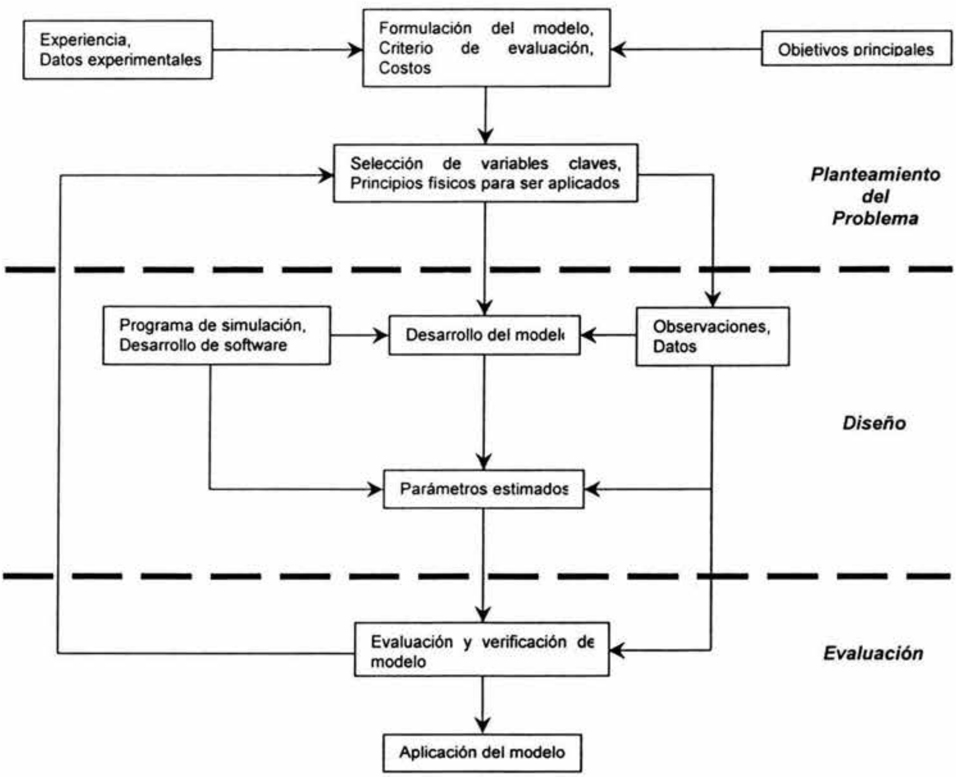
Aspectos importantes en el desarrollo de un "modelo" son:

- Formulación de ecuaciones del balance de materia y energía.
- Ecuaciones cinéticas apropiadas para las reacciones químicas.
- Velocidades de transferencia de masa y calor que representan los cambios de las propiedades del sistema, equilibrio de fase, y aplicación de control.

El procedimiento general para realizar un modelo matemático es el siguiente:



Para ser más precisos el procedimiento para modelizar se puede dividir en cuatro etapas, como se muestra en la siguiente figura: la definición y formulación del problema, un preliminar y análisis detallado, evaluación y por ultimo interpretación/aplicación.



Un uso del "modelo" del proceso es el análisis de datos experimentales, utilizando este hecho para caracterizar el proceso asignando valores numéricos a variables importantes del proceso.

La aplicación combinada de las aproximaciones de modelado y simulación conllevan las siguientes ventajas:

- El modelado matemático mejora el entendimiento.
- Los "modelos" ayudan en el diseño experimental.
- Los "modelos" pueden ser utilizados de forma predictiva para el diseño y control.

- Los "modelos" pueden ser utilizados en la educación y entretenimiento
- Los "modelos" sirven para la optimización del proceso.

Las etapas que componen el desarrollo 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 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 deben 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. Hay que tener en cuenta, además, que existen diversas alternativas de modelado (especialmente en programación entera) que afectan de manera fundamental en la resolución del mismo. Existiendo un desarrollo cada vez mayor en la reformulación de problemas.

- 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 3.1 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 existen códigos de optimización lineal que implantan algoritmos muy eficientes. Son fiables y numéricamente robustos y algunos están ampliamente disponibles. (Véase, por ejemplo, Lorenz T. Biegler, Jorge Nocedal, Claudia Schmid y

David Ternet (2000). Numerical Experience with a Reduced Hessian Method for Large Scale Constraine Optimization. *Computacional Optimization and Applications*, Vol. 15, No. 1. ó el artículo, Ignacio E. Grossmann, José Antonio Caballero y Héctor Yeomans (2000). Advances in Mathematical Programming for Automated Design, Integration and Operation of Chemical Processes. *Department of Chemical Enginering*, Carnegie Mellon University Pittsburg, PA 15213, USA.)

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

Tabla 3.1 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, siendo 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 desea (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).

Esta etapa de verificación, validación, comprobación da lugar a nuevas necesidades de modelado para mejorar la capacidad de representación de la realidad, a nuevos refinamientos indicados por el usuario.

Interpretación y análisis de los resultados

Esta etapa consiste en proponer soluciones. Permite conocer en detalle el comportamiento del modelo al realizar un análisis de sensibilidad en los parámetros de entrada, estudiar diferentes escenarios plausibles de los parámetros, detectar soluciones cuasióptimas, es decir, aquellas soluciones que se aproximan a la solución optima, comprobar la robustez de la solución óptima.

Implantación, documentación y mantenimiento

Esta 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 de un modelo no está en el desarrollo sino en la fase de uso y mantenimiento.

4. CODIFICACIÓN DE PROBLEMAS DE OPTIMIZACIÓN

4.1 Lenguajes de modelado

4.1.1 Lenguajes de representación de modelos

Los lenguajes de representación de modelos son de un desarrollo relativamente reciente en el ámbito de la investigación operacional y programación matemática. Los factores que influyen en su aparición fueron el desarrollo de algoritmos de optimización por un lado, y la disminución del costo del poder computacional, por el otro.

El desarrollo de algoritmos de optimización capaces de resolver grandes problemas de optimización de la producción genero una demanda por resolver este tipo de modelos. Sin embargo, resulto que, si bien los programas de optimización eran capaces de resolver estos problemas que tienden a ser bastante grandes, usarlos resultaba bastante engorroso, ya que eran necesarios manipular enormes volúmenes de datos para generar la entrada requerida por estos programas.

Una vez resuelto el problema, era además necesario interpretar y analizar la solución, lo cual tampoco resultaba fácil, ya que la solución también era bastante voluminosa. Todo esto dio origen al uso de programación orientada a ecuaciones, conocido también como "Generadores de Matrices", que si bien facilitaban, en cierta forma, la aplicación de la programación lineal, todavía presentaban inconvenientes.

La disminución del costo del poder computacional, por otro lado abrió la oportunidad para que el computador remplazase algunas de las actividades realizadas por el hombre en el proceso descrito. Esto dio origen a los lenguajes de representación de modelos o lenguajes de modelación. En 1983, Fourer en su artículo "Modeling languages versus matrix generators for linear programming", publicado en la revista *ACM Transactions on Mathematical Software, No. 9*, describe en forma muy clara, las ventajas de usar lenguajes de modelación en vez de generadores de matrices. Sin embargo, él usó un lenguaje de modelación teórico (que llamo XML), dado que en ese momento los lenguajes de modelación sólo existían como prototipos de distribución limitada. Uno de los conceptos interesantes que introdujo Fourer en su artículo, es que existían distintas representaciones del modelo. Por ejemplo, la que el modelador escribe en forma algebraica, en un papel, la llamo "representación del modelador" mientras que la que requiere el programa de optimización, la llamó "representación del algoritmo". La idea de los lenguajes de modelación es permitir al modelador especificar el modelo de una forma lo más parecido posible a la representación del modelador y dejar que un programa computacional traduzca esta representación a la del algoritmo.

El primer lenguaje de modelación que se popularizó y usó en forma más extensa fue GAMS, desarrollado en el Banco Mundial como no lo explican Bischop, J. y Meeraus, A. en su artículo "On the Development of a General Algebraic Modeling System in a Strategic Planning Environment," publicado en 1982, por la revista *Mathematical Programming Study* en su volumen 20. En este artículo se compara además GAMS con otros lenguajes de modelación de la época. Meeraus en su artículo "An Algebraic Approach to Modeling," publicado en 1983, por la revista *Journal of Economic Dynamics and Control, Vol. 5*, analiza los problemas que se presentan al desarrollar y resolver modelos de gran tamaño en un entorno muy dinámico, como el del Banco Mundial, donde la rotación de las personas que desarrollan los modelos es muy alta. Algunos de los problemas mencionados son: documentación del modelo, comunicación del modelo, supuestos del modelo, diseminación del modelo, interfaz con el usuario y deformación del modelo. Esto último de refiere a la tendencia de algunos modeladores de hacer calzar el modelo con el algoritmo de solución. Esto a veces los lleva hacer supuestos poco realista. Finalmente, en este artículo se ilustra como GAMS ayuda a resolver los problemas mencionados anteriormente.

Roy, A., Lasdon, L. y Lordeman, J. en su artículo "Extending Planning Languages to Include Optimization Capabilities", publicado en 1986 por la revista *Management Science, Vol. 32*, le dan un enfoque distinto, ellos describen el diseño y la implementación de la extensión del conocido sistema IFPS, para agregarle la capacidad de optimizar. IFPS (Interactive Financial Planning System), es un lenguaje de planificación que en cierto sentido, es precursor de los programas de plantillas de cálculo, tan populares hoy en día. A pesar de esta popularidad, que ha eclipsado es cierta forma a IFPS, éste sigue siendo muy usado ya que presenta importantes ventajas sobre las plantillas de cálculo en la elaboración de modelos complejos y de gran tamaño. La extensión presentada por Roy et al. agrega unos pocos comandos básicos en IFPS y, en su mayor parte, es transparente para el usuario. Es decir, el usuario no sabe que está corriendo una compleja aplicación de programación matemática para encontrar la solución óptima al problema descrito en IFPS.

En 1987, Maturana por otro lado, analiza y compara los sistemas de modelación matemática más importantes en esa época. Los sistemas comparados son: GAMS, GINO, GXMP, HEQS e IFPS/OPTIMUM. Con el objeto de llevar a cabo esta comparación, se presenta y discute un conjunto de criterios de comparación.

Eventualmente GAMS se convirtió en un producto comercial y muy pronto hubo otros lenguajes de modelación, como LINGO, que también se convirtieron en productos comerciales.

El más reciente lenguaje de modelación en convertirse en producto comercial, es AMPL, que fue desarrollado por la AT&T (se empezó a comercializar en Noviembre de 1992).

R. Fourer, D.M. Gay y B.W. Kernighan en su artículo "A Modeling Language for Mathematical Programming", publicado en 1990 por la revista *Management Science*, describen el diseño e implementación de AMPL. En este artículo, además, se dan ejemplos y se compara con otros lenguajes de modelación, en especial con GAMS. También se proporcionan algunos antecedentes sobre la implementación del traductor del lenguaje, que se llevó a cabo usando herramientas de desarrollo de software muy modernas. Cabe hacer notar que gran parte del diseño del lenguaje está inspirado en XML, el prototipo de fourer mencionado anteriormente.

Al popularizarse los lenguajes de modelación comenzó a prestarse más atención al proceso de la modelación. Uno de los primeros en explorar en forma un poco más rigurosa el proceso de modelación fue A. M. Geoffrion, en su artículo "An introduction to structured modeling", publicado en 1987 por la revista *Management Science, Vol. 33*, en el cual introdujo los conceptos básicos de lo que él llama "Structured Modeling", que es un intento de estructurar más el proceso de desarrollo de modelos cuantitativos. Es interesante hacer notar la analogía que existe entre las ideas de "Structured Modeling" y "Structured Programming". Ambas tratan de hacer que el trabajo de modelación, o de programación, sea más una ciencia y menos un arte. Es importante enfatizar que "Structured Modeling" no es un lenguaje de modelación sino un marco teórico sobre el cuál se puede definir lenguajes de modelación que tiene ciertas características deseables. Con el objeto de presentar y discutir ejemplos, Geoffrion diseño un lenguaje llamado SML ("Structured Modeling Languaje"), lo describe en su artículo "The SML language for structured modeling" publicado en 1992 por *Operations Research, Vol. 40*. Además, en su artículo "Integrated modeling systems", publicado en 1989b, por *Computer Science in Economics and Management*, presento los aspectos más formales de "Structured Modeling". Estos aspectos están basados, principalmente, en teoría de grafos los que se usan para definir rigurosamente los principales conceptos de "Structured Modeling".

Al analizar más de cerca el diseño e implementación de los lenguajes de modelación, surgen algunos aspectos que no están debidamente resueltos. En 1992 analiza algunos de estos aspectos y propone cierta forma de resolverlos usando lenguajes de modelación existentes para ilustrar los principales temas.

Otra consideración con respecto a los lenguajes de modelación, es el apoyo a interfaces gráficas. En general, la tendencia es ir, cada vez más hacia interfaces gráficas para los distintos programas. Por ejemplo Kendrick, en su artículo "A Graphical Interface for Production and Transportation System Modeling: PTS", publicado en 1991, en la revista *Computer Science in Economics and Management, Vol. 4*, presenta una interfaz gráfica para la modelación de sistemas de producción y transporte (PTS). Una vez especificado el modelo gráfica, un programa transforma esa representación en una especificación del modelo escrito en GAMS, lo que permite, a su vez resolver el modelo usando ese sistema.

Jones, que se ha especializado en la representación gráfica de modelos, presenta en sus artículos: "An Introduction to Graph-Based Modeling Systems, Part I: Overview" y "An Introduction to Graph-Based Modeling Systems, Part II: Graph Grammars and the Implementation", (ambos publicados en la revista *ORSA Journal on Computing* en 1990 y 1991, respectivamente), una introducción bastante completa a la implementación de sistemas basados en representaciones gráficas. En principio, las ideas de Jones se pueden aplicar a distintos lenguajes de modelación. En 1992, Jones, en su artículo "Attributed Graphs, Graph-Grammars, and Structured Modeling", publicado en la revista *Annals of Operations Research, Vol. 38*, explica cómo implementar el lenguaje de "Structured Modeling" como un sistema de representación gráfica que permita especificar, en forma gráfica, un modelo estructurado.

4.1.2 Lenguajes de modelado

En 1995, Sharda, en su artículo "Algebraic modeling languages on PCs" publicado por *OR/MS Today (Institute for Operations Research and the Management Sciences)*, explica las principales alternativas actuales para el desarrollo de modelos de optimización, las cuales suelen ser:

- *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.

- *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.)

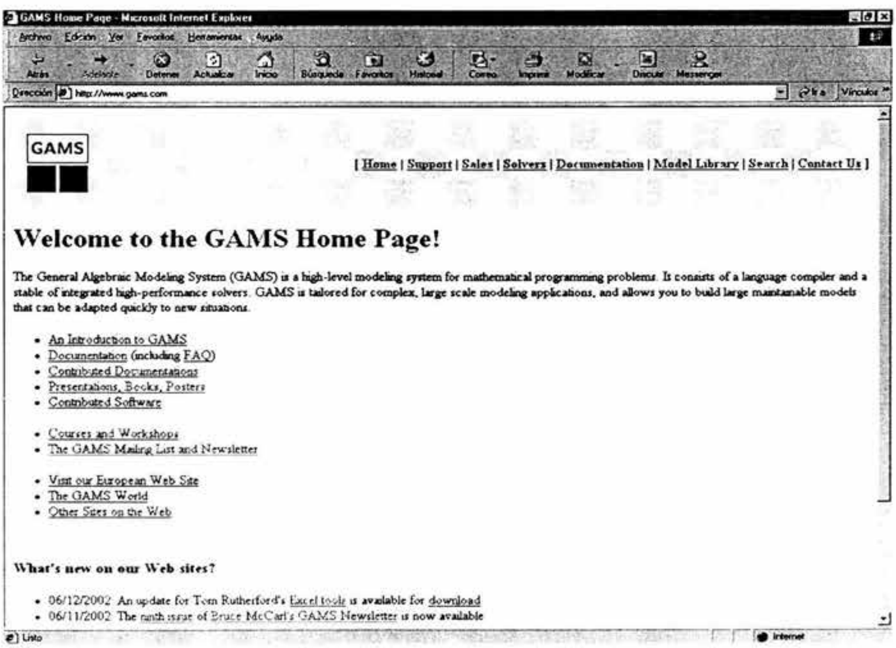
Los optimizadores de las hojas de cálculo, por ser aplicaciones muy comunes y conocidas. Pueden ser un vehículo eficaz de difusión de un modelo entre cierto tipo de usuarios y facilitan el manejo de datos que se encuentren ya en dicho formato [véase, Ragsdale, C. T., (1998), Spreadsheet modeling and decision analysis: a practical introduction to management science, *South-Western College. 2nd ed.*]. 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 [véase, Gass, S. I., Hirshfeld, D.S. and Wasil, E. A., (1995), "Model Word: The Spreadsheeting of OR/MS", *Interfaces*].

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.

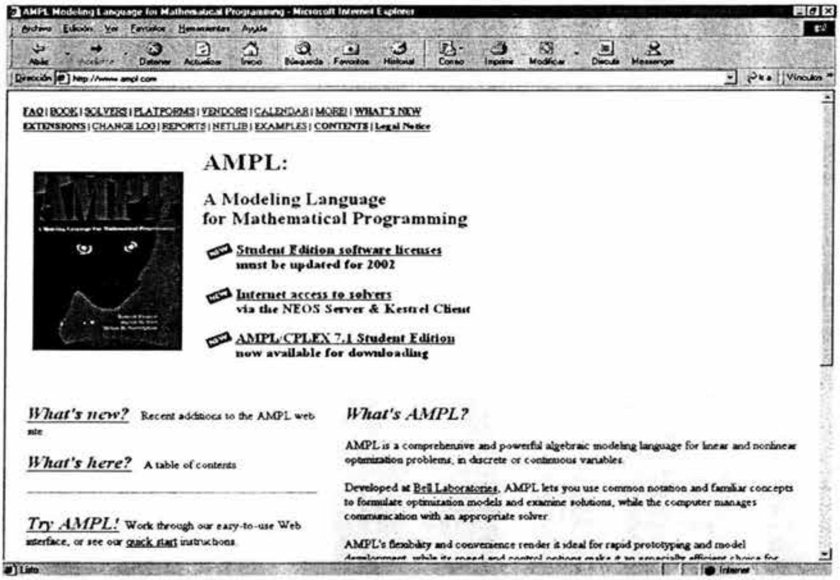
Todas estas alternativas pueden ser utilizadas para desarrollo rápido de un prototipo o una demostración ya que presentan capacidades de presentación gráfica que pueden ser aprovechadas. Son difícilmente utilizables cuando se plantean problemas de optimización de tamaño medio o superior.

- *Lenguajes algebraicos de modelado*

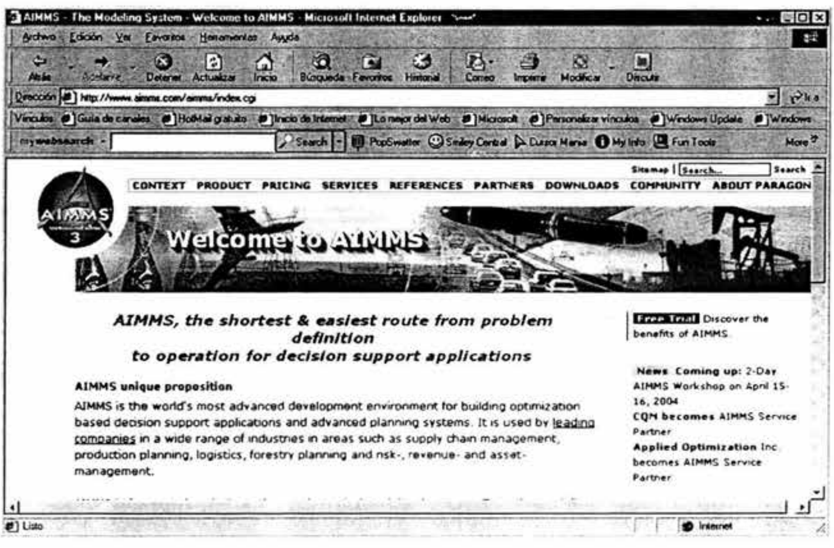
Son las alternativas más complejas y potentes por su capacidad de indexación de las variables y ecuaciones, permiten cambiar sin dificultad las dimensiones del modelo, de forma natural separan datos de resultados. Desde el punto de vista del modelador permiten la detección de errores de consistencia 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)



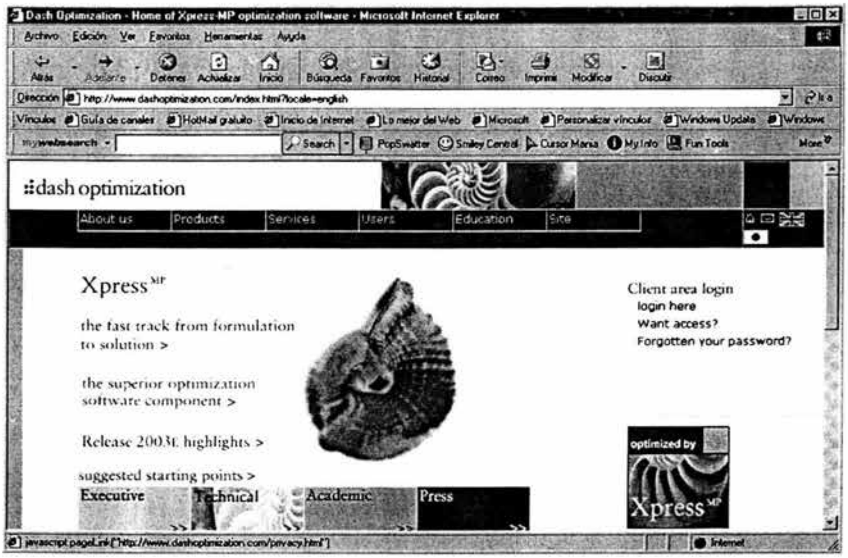
y AMPL (www.ampl.com) de origen estadounidense,



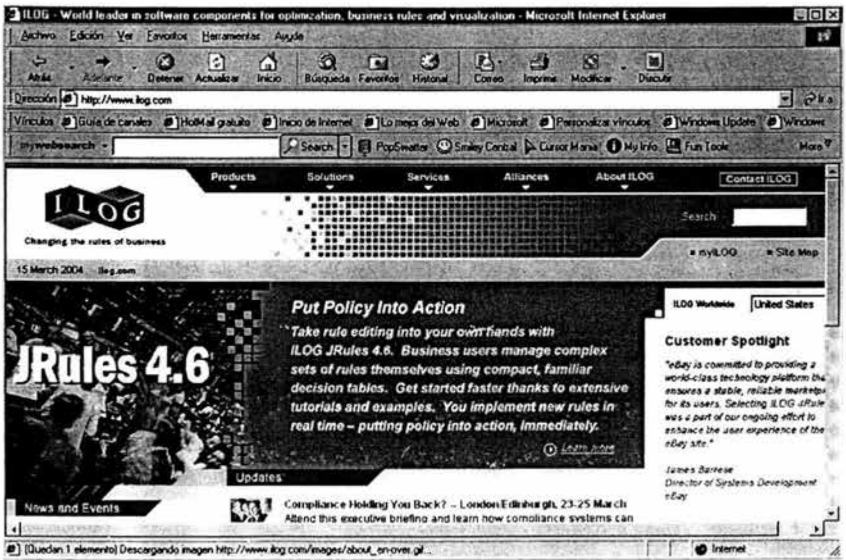
y AIMMS (www.aimms.com)



y XPRESS-MP (www.dashoptimization.com) de origen europeo, por citar algunos.

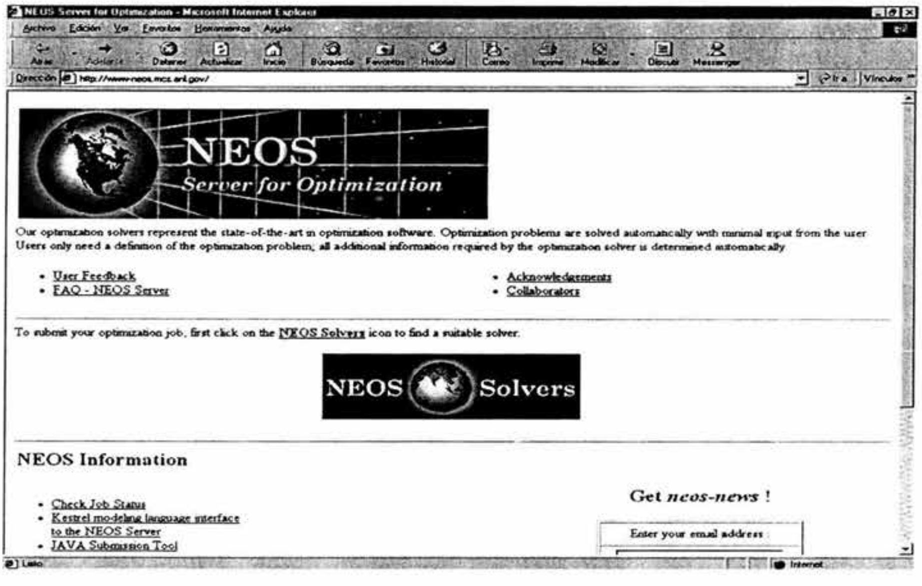


Existe una herramienta integrada denominada OPLStudio (www.ilog.com),



en la que se dispone de un lenguaje de modelado (OPL) y varios optimizadores dependiendo del modelo propuesto. Está especialmente desarrollada para problemas de planificación y programación de producción *(scheduling)*, 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, Planner, Solver, CPLEX, estando los tres primeros basados en la programación de restricciones y el último en programación matemática.

GAMS es el lenguaje más ampliamente difundido comercialmente con su propia lista de discusión de usuarios, mientras que AMPL se está potenciando mucho en las universidades estadounidenses. Existe un proyecto denominado NEOS (www.neos.mcs.anl.gov) 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 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 [véase, por ejemplo A. Brooke, D. Kendrick y A. Meeraus, (1998), *Gams: A User's Guide*. GAMS Development Co.], para AMPL, [véase, por ejemplo, R. Fourer, D. M. Gay and B. W. Kernighan, (2000), *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press. 2nd ed.], o para OPL [véase, por ejemplo, P. Van Hentenryck, (1999), *The OPL Optimization Programming*

Language. The MIT Press.]. Los campos de aplicación de estos lenguajes son tan amplios como los de la optimización propiamente dicha. Abarcan desde la micro y macroeconomía, a la economía de la energía, a la planificación energética o eléctrica, a la ingeniería química o forestal, a la planificación del desarrollo económico o del comercio internacional o a la cobertura de riesgos financieros. En el caso de la programación de restricciones ésta aparece especialmente en problemas combinatorios para modelar restricciones lógicas.

4.1.3 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
- Los optimizadores pueden ser intercambiados sin dificultad, se pueden probar nuevos optimizadores, nuevos métodos o nuevas versiones
- Por ejemplo, en el lenguaje GAMS se encuentran entre otros disponibles los optimizadores CPLEX, OSL, XA y XPRESS para problemas LP y MIP, MINOS y CONOPT para problemas NLP, DICOPT para problemas MINLP y MILES y PATH para problemas MCP.
- Permiten la realización de cambios en el modelo de manera sencilla y segura, es decir, se puede afrontar un refinamiento continuo en la formulación del problema

- Cualquier tipo de problemas de programación lineal, no lineal, flujos en redes o mixta complementaria resulta muy fácil implantar su formulación

- Permiten la implantación de algoritmos avanzados, que incluyan varias llamadas al optimizador o procedimientos específicos para el problema (como por ejemplo los métodos de descomposición)

- Permiten la portabilidad de los modelos entre plataformas y sistemas operativos

Como *desventajas* principales se pueden mencionar las siguientes:

- No son adecuados para la resolución de problemas de pequeño tamaño por parte de usuarios esporádicos por la barrera de entrada que supone el aprendizaje de un nuevo lenguaje

- No pueden utilizarse para la resolución directa de problemas gigantescos cuya formulación completa incluso no se puede realizar (por ejemplo, a partir de 1 millón de restricciones y/o variables)

- En la ejecución se incluye un tiempo de creación del modelo y de interfaz con el optimizador que ralentiza la obtención de la solución, por lo tanto no es recomendable cuando el tiempo de ejecución es un factor crítico.

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

- Interfaces de entrada y salida de datos más estrechamente relacionadas con bases de datos u hojas de cálculo

- El desarrollo de interfaces gráficas que faciliten al usuario la formulación visual y el entendimiento de problemas de optimización

- Interfaz con lenguajes de propósito general para la incorporación de funciones externas definidas por el usuario dentro de la optimización

- El avance en las capacidades de resolución directa de problemas estocásticos (con adición de características específicas en el propio lenguaje y uso de algoritmos de descomposición en el optimizador) o problemas no lineales complejos

- La posibilidad de ocultar el código fuente produciendo versiones ejecutables para usuarios finales

- La selección automática del método y optimizador

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

4.2 Modelado en GAMS

GAMS es un sistema de modelación para optimizaciones que provee una interfase con variedad de diferentes algoritmos. Los modelos son dados por el usuario a través de un archivo de entrada en la de ecuaciones algebraicas usando un lenguaje de alto nivel. Entonces GAMS compila el modelo y hace la interfase automáticamente con los resolvedores (p.e. un algoritmo de optimización). El modelo compilado así como la solución encontrada por el resolvedor son regresados al usuario a través de un archivo de salida. El diagrama de la Figura C, nos ilustra este proceso.

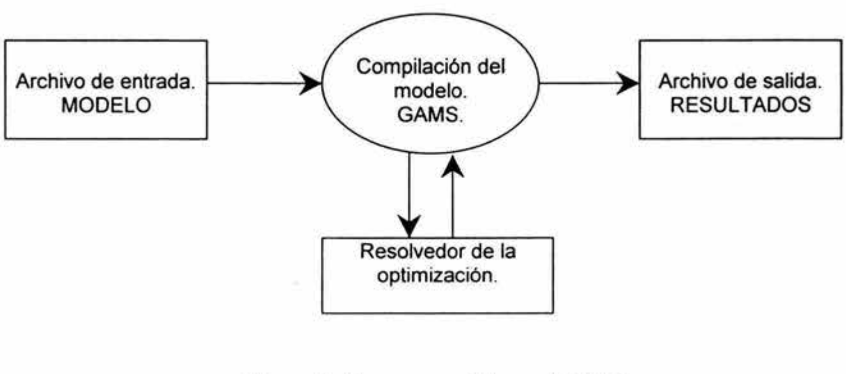
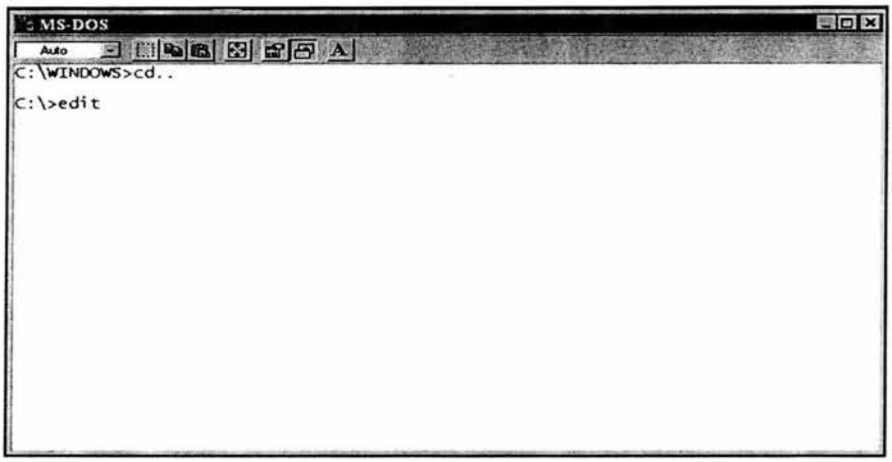


Figura C. Proceso seguido por el GAMS

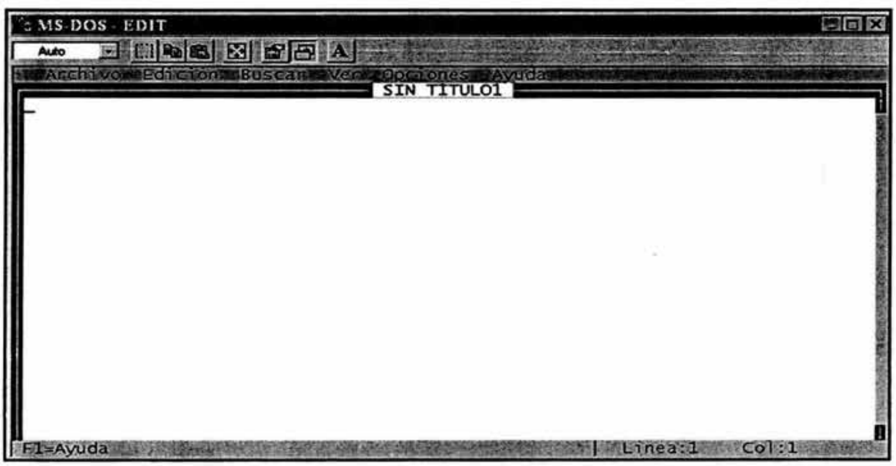
4.2.1 Guía general para el uso de GAMS

Todo problema a ser resuelto por GAMS, debe ser planteado en el MS-DOS Editor u otro editor de textos. Para ejecutar el MS-DOS Editor:

- Pasar de Windows a MS-DOS
- Escribir edit y dar enter, como se muestra acontinuación:

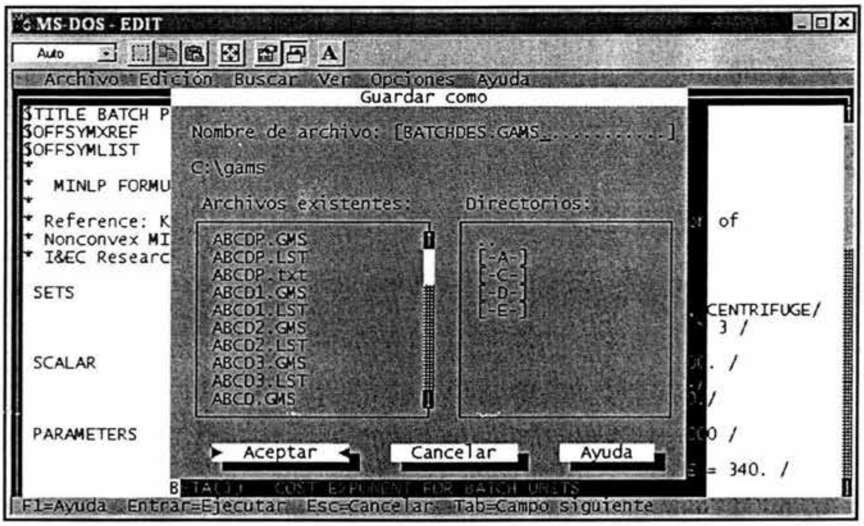


Al dar enter se despliega la pantalla siguiente:



- Oprimir la tecla Esc y comenzar la captura del programa.
- Ya capturado el programa, este se salvara con la extensión gms:

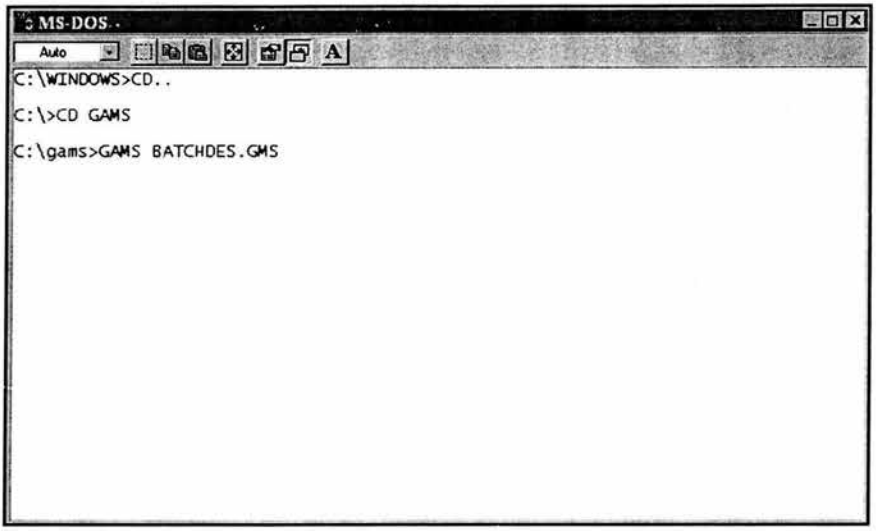
Nombre del archivo.gms



A continuación, para ejecutar el programa se tiene que introducir el siguiente comando:

C:\>cd gams

C:\ GAMS>Gams nombre del archivo.gms



dicho comando genera un archivo de salida con el nombre del archivo del programa y con la extensión lst, el cual se puede leer en cualquier editor de texto.

Estructura de los modelos de GAMS

Los componentes básicos de un modelo de GAMS se muestran en la Tabla 4.1.

Entradas	Salidas
Sets (Conjuntos) Declaración asignación de miembros	• Echo print • Mapas de referencia • Listado de ecuaciones • Reporte del estado • Resultados
Data (Datos) (parameters, Tables, Scalar) Declaración asignación de valores	
Variables Declaración asignación de tipo	
Asignación de fronteras valores iniciales (opcionales)	
Equations (ecuaciones) Declaración Definición	
Instrucciones de Model y Solve (modelo y resolver)	
Instrucción Display (desplegar), opcional	

Tabla 4.1 Componentes básicos de un modelo de GAMS.

Antes de revisar los componentes por separado, se deben tener presente los siguientes puntos:

1. Un modelo de GAMS es una colección de instrucciones en el lenguaje de GAMS. La única regla que gobierna el orden de estas instrucciones es que ninguna entidad puede ser referida antes de que se declare su existencia.

2. Las instrucciones de GAMS, fuera de errores tipográficos, pueden estar en cualquier estilo conforme al usuario; se permiten instrucciones en varias líneas, líneas en blanco o varias instrucciones en una línea.

3. Cada instrucción debe terminarse con un punto y coma *(;)*. La excepción ha esta regla es cuando la siguiente instrucción empieza con una palabra clave de GAMS.

4. El compilador de GAMS no distingue entre mayúsculas y minúsculas, por lo tanto el usuario es libre de usar cualquier forma.

5. La documentación es crucial en los modelos matemáticos, siendo más útil ponerla dentro del modelo que separadamente. Hay dos formas básicas para insertar documentación en un modelo de GAMS. La primera, cualquier línea que empieza con asterisco (*) en la columna I es tomada como un comentario por el compilador del GAMS. Y la segunda forma, quizá la más importante, texto documental puede ser insertada en instrucciones específicas de GAMS.

6. La creación de entidades de GAMS se lleva a cabo en dos pasos: una declaración y asignación o definición. "Declaración" significa declarar la existencia de algo y darle un nombre. "Asignación" o "Definición" significa darle a algo un valor específico o una forma específica. En el caso de ecuaciones, uno debe de hacer la declaración y definición en instrucciones separadas de GAMS. Para todas las demás entidades, sin embargo, se tiene la opción de hacer la declaración y asignación en la misma instrucción o separadamente.

7. Los nombres dados a las entidades del modelo deben empezar con una letra y pueden ser seguidas por un máximo de nueve letras o dígitos.

<u>Conjuntos (Sets)</u>

Los conjuntos son los bloques básicos de construcción de un modelo de GAMS, correspondiendo los índices exactamente a los de un modelo algebraico. Por ejemplo:

```
Sets
      I plantas      /Seattle, San-Diego/
      J mercadas  /New-York, Chicago, Topeka/;
```

El efecto de esta instrucción es probablemente por sí solo evidente. Se han declarado dos conjuntos, dándoles por nombre I y J, también se les ha asignado miembros como se muestra a continuación:

I={ Seattle, San Diego}
J={ New York, Chicago, Topeka}.

Hay que notar que la diferencia tipográfica entre el formato GAMS y el usado en las matemáticas para listar los elementos de un conjunto. GAMS usa '/' en vez de '{ }' para delinear el conjunto simplemente porque no todos los teclados de las computadoras tienen teclas para los corchetes. Hay que notar que los nombres con palabras múltiples en GAMS como 'New York' no son permitidas, así que se utiliza un guión.

Las palabras en minúsculas, en la instrucción de conjunto de arriba, son llamadas 'texto'. El texto es opcional, esta ahí solo como documentación interna, no teniendo un propósito formal en el modelo. El compilador del GAMS no hace ningún intento para interpretar este texto, pero salva el texto y lo regresa en los archivos de salida.

No es necesario el combinar la creación de los conjuntos I y J en una sola instrucción. Se podrían poner por separados como sigue:

```
Set  I   plantas    /Seattle, San-Diego/;
Set  J   mercados   /New-York, Chicago, Topeka/;
```

La colocación de espacios o líneas en blanco (así como el uso de mayúsculas y minúsculas) es opción del usuario.

Una característica conveniente de usar para la asignación de miembros en un conjunto es el asterisco. Se aplica en los casos en que los elementos siguen una secuencia. Por ejemplo:

```
Set  T   time periods     /1991*2000/;
Set  m   machine          /mach1 *mach24/:
```

El efecto de este es el siguiente:

T={ 1991,1992,1993,.... ,2000}
M={mach1,mach2,mach3,.....,mach24}

Hay que notar que los elementos del conjunto son caracteres alfanuméricos, así que los elementos de T NO son números ni pueden ser usados directamente como índices, es decir, no podemos referenciar 1991 usando T(I), como normalmente se haria en un lenguaje de programación.

Otra conveniente característica es la instrucción alias, la cual es usada para dar otro nombre a un conjunto previamente declarado. Por ejemplo:

```
Alias (T, Tp);
```

En este caso Tp es un nombre más para referenciar a T. Esta opción es útil en modelos que contienen interacciones de los elementos del mismo conjunto, normalmente utilizada en aplicaciones de programación dentro de GAMS.

Datos

En GAMS hay tres formatos para introducir datos: listas, tablas y asignación directa.

Listas

Este formato es ilustrado por el siguiente ejemplo:

```
Parameters

        A(I) capacidad de la planta I en barriles
              /Seattle        350
               San-Diego      600/
        B(J) demanda del mercado J en barriles
              /New-York       325
               Chicago        300
               Topeka         275/;
```

La instrucción declara la existencia de dos parámetros, dándoles los nombres A y B y declarándoles dentro del dominio de I y J, respectivamente. La instrucción también da texto documental para cada parámetro y les asigna valores de A(I) y B(J) para cada elemento de I y J. Está perfectamente aceptado el separar esta instrucción en dos, si uno prefiere.

Al usar el formato de lista, se debe tener presente los siguientes puntos:

1. La lista de los elementos del dominio y sus respectivos valores pueden ser puestas como uno quiera. La única regla es que la lista entera debe estar cerrada por '/' y cada pareja de elemento-valor deben estar separadas por ';' que no es ',' o escritas en diferentes líneas.

2. No hay un punto y coma separando la lista de elemento - valor de su nombre, dominio, y texto que le precede.

3. El compilador del GAMS tiene una característica inusual llamada 'verificado del

dominio', el cual verifica que cada elemento del dominio en la lista es en efecto un miembro del conjunto apropiado.

4. Cero es el valor por omisión para todos los parámetros. Por lo tanta, sólo es necesario incluir los valores diferentes a cero, y éstos pueden ser entrados en cualquier orden.

5. Un escalar es tomado como un parámetro sin dominio. Puede ser declarado y asignado con la instrucción Scalar conteniendo una lista de un solo valor.

Tablas

Este formato es ilustrado por el siguiente ejemplo:

Table D(I,J) distancia en miles de millas

	New-York	Chicago	Topeka
Seattle	2.5	1.7	1.8
San-Diego	2.5	1.8	1.4

El efecto de esta instrucción es el de declarar el parámetro D y especificar su dominio como el conjunto de parejas ordenadas en el producto cartesiana de (I,J). El valor de D también se da en la instrucción justo bajo su encabezado apropiado. Si hubiera un espacio en blanco, se tomaría como cera.

A lo igual que en el formato de lista, GAMS realiza la verificación del dominio para asegurarse que los nombres de los renglones y columnas de la tabla sean miembros del conjunto apropiado.

Asignación Directa

El método de asignación directa de entrada de datos difiere de los métodos de listas y tablas en que se divide la declaración de parámetros y asignación de parámetros en instrucciones separadas. Un ejemplo de esto sería:

```
Parameter C(I,J) costo de transporte en miles de dólares por barril;
C(I,J) = f*D(I,J)/l000;
```

Es importante enfatizar la presencia del punto y coma al final de la primera línea; sin él, el compilador del GAMS podría intentar interpretar ambas líneas como parte de la misma instrucción (por lo que GAMS marcaría un error).

El efecto de la primera instrucción es declarar el parámetro C, especificando el dominio (I,J), y proveyendo algún texto documental. La segunda instrucción asigna a C(I,J) el valor del producto del parámetro f y D(I,J). Naturalmente, esto es legal en GAMS solamente sí se ha asignado valores a f y D(I,J) en alguna instrucción previa.

La asignación directa de arriba se aplica a toda pareja (I,J) en el dominio de C. Si uno quiere asignar a valores a elementos específicos del dominio, se tiene que encerrar los nombres del elemento entre comillas, por ejemplo:

C('Seattle',New-York') = 0.40;

es una instrucción válida de GAMS. Por otro lado, C(I,I)=0.40 no seria una instrucción válida de GAMS.

Al mismo parámetro se le puede asignar valores más de una vez. Cada instrucción de asignación toma efecto inmediatamente y sobre un valor previo. (En contraste, el mismo parámetro no puede ser declarado más de una vez, un error en GAMS que hay que evitar es el usar accidentalmente el mismo nombre para dos cosas diferentes)

El lado derecho de una instrucción de asignación puede contener una gran variedad de expresiones matemáticas y de funciones.

Variables

Las variables de decisión de un modelo expresado en GAMS deben ser declaradas con la instrucción *Variables.* A cada variable se le da un nombre, un dominio si se requiere, y opcionalmente un texto. Un ejemplo de estos se muestra a continuación:

```
Variables
        X(I,J) Cantidades de embarcos por casos
        Z      Costo total de transportación en miles de dólares;
```

Esta instrucción resulta en la declaración de las variables "embarco" para cada pareja (I,J). La variable Z es declarada sin dominio porque es una cantidad escalar. Cada modelo de optimización de GAMS debe contener tantas variables que se utilizan como cantidades para ser minimizadas o maximizadas.

Declaradas las variables, se les asignara un tipo. Los tipos permisibles se muestran en la Tabla 4.2.

La variable que se usa como la cantidad a optimizar debe ser un escalar y ser de tipo free.

Tipo de variable	Rango permisible de la variable
Free (libre)	$-\infty$ a $+\infty$
Positive (Positiva)	0 a $+\infty$
Negative (Negativa)	$-\infty$ a 0
Binary (Binaria)	0 o 1
Integer (Entera)	0, 1, ..., 100

Tabla 4.2 Tipos de variables en GAMS

Ecuaciones

El poder de los lenguajes de modelación tal como el GAMS se nota más en la creación de las ecuaciones y desigualdades que comprenden el modelo bajo construcción. Es debido a que cuando un grupo de ecuaciones o desigualdades tienen la misma estructura algebraica, todos los miembros del grupo son creados simultáneamente, no individualmente.

Las ecuaciones deben ser declaradas y definidas en instrucciones separadas. El formato de la declaración es el mismo que para otras entidades del GAMS. Primero se utiliza la palabra clave "Equations" seguida por el nombre, dominio y texto de una ecuación o más grupos de ecuaciones o desigualdades como a continuación:

```
Equations
        Cost          Función objetivo
        Supply(I)     Límite observado de aprovisionamiento en la planta I;

Cost..        z =e= sum((I,J),C(I,J)*X(I,J));
Supply(I)..   sum(J,X(I,J)) =l= A(I);
```

Hay que tener en mente que la palabra Equation tiene un significado amplio en GAMS. Comprende tanto igualdades como desigualdades, y las ecuaciones de GAMS con un nombre sencillo pueden referirse a una o varias de estas relaciones. Por ejemplo, la ecuación Cost no tiene dominio por lo que es una ecuación sencilla, pero la ecuación Supply es un conjunto de desigualdades definidas sobre el dominio I.

En el ejemplo anterior I consistía de 2 elementos ('Seatle' y 'San-Diego') por lo que la ecuación Supply(I) representaría en realidad las dos desigualdades siguientes:

```
Supply('Seatle')..     sum(J,X('Seatle',J))=l=A('Seatle');
Supply('San-Diego)..   sum(J,X('San-Diego' ,J))=l=A('San-Diego');
```

La sumatoria en GAMS puede ser usada para expresiones simples y complejas. Su formato está basado en la idea de que siempre pensamos en la sumatoria como un operador con dos argumentos:

Sum(indice de la sumatoria, sumando)

Los dos argumentos son separados por una coma, y si el primer argumento requiere una coma entonces debe estar entre paréntesis. El segundo argumento puede ser cualquier expresión matemática inclusive otra sumatoria. Por ejemplo:

Sum (J, X(I,J)) equivalente a $\Sigma_J X_{I,J}$
Sum ((I,J), C(I,J)*X(I,J)) equivalente a $\Sigma_I \Sigma_J C_{I,J} X_{I,J}$

Él último ejemplo se podría también escribir como:

Sum (I, Sum(J, C(I,J)*X(I,J)))

Las multiplicatorias son definidas en GAMS usando exactamente el mismo formato que en la sumatoria, remplazando Sum por Prod. Por ejemplo:

Prod(J, X(I,J)) equivalente a $\Pi_J X_{I,J}$

Los operadores en la sumatoria y la multiplicatoria deben ser usados en una instrucción de asignación directa para parámetros. Por ejemplo:

Scalar totsupply Aprovisionamiento total de todas las plantas

Totsupply = sum(I,B(I));

La definición de las ecuaciones es la instrucción más compleja del GAMS en término de su variedad. Los componentes para definir una ecuación son, en orden:

- El nombre de la ecuación ha ser definida
- El dominio (conjunto de ecuaciones que representa)
- Condiciones de restricción del dominio (opcional)
- El símbolo '..'
- La expresión del lado izquierdo
- El operador de relación: =l=, =e=, ó =g=
- La expresión del lado derecho

La definición de la ecuación en GAMS se ilustrara con las siguientes instrucciones:

```
Cost..          z =e= sum ((I,J), C(I,J)*X(I,J));
Supply(I)..          sum (J, X(I,J)) =l= A(I);
Demand(J)..          sum (I, X(I,J)) =g= B(J);
```

A continuación se dan algunos puntos que recordar:

- El poder de crear múltiples ecuaciones con una instrucción sencilla de GAMS es controlada por el dominio. Por ejemplo, la definición de la restricción Demand resulta en la creación de una restricción por cada elemento del dominio de J.
- Los operadores de relación tienen el siguiente significado:

  =l= menor o igual que
  =g= mayor o igual que
  =e= igual a

  Es importante entender la diferencia entre el símbolo '=' y '=e='. El símbolo '=' es usado solamente en asignaciones directas, y el símbolo '=e=' es usado solamente en definición de ecuaciones. Una asignación directa nos da un valor deseado a un parámetro antes de que el resolvedor sea llamado. Una definición de ecuación también describe una relación deseada, pero no puede ser cumplida hasta después que el resolvedor es llamado. Además, la definición de ecuación debe contener variables y la asignación directa no.
- Las variables pueden aparecer en el lado derecho o izquierdo de la ecuación o en ambos. La misma variable puede aparecer en la ecuación más de una vez. En el procesador de GAMS convierte automáticamente las ecuaciones a la forma estándar equivalente hasta de llamar al resolvedor.
- En el caso de problemas lineales, la forma de despejar las ecuaciones no tiene ningún efecto en resolvedor. Sin embargo en el caso de optimización no lineal la forma como se definan las ecuaciones puede significar el éxito o fracaso en la solución del problema. Como recomendaciones generales, evítese dejar expresiones como dividendos y funciones que pudieran generar indeterminaciones numéricas (por ejemplo logaritmos de funciones que pudieran resultar iguales a cero) no solo tomando en cuenta las posibles soluciones de los modelos, sino todos los valores que pudieran adquirir estas funciones durante las iteraciones.
- La definición de ecuación puede aparecer en cualquier lado en la entrada de GAMS, dando la ecuación y todas las variables y todos los parámetros al cuál es referido son declarados previamente.

Expresiones matemáticas

Para asignar valores utilizando expresiones matemáticas hay que tener en cuenta las reglas siguientes:

1. El uso de índices en la asignación indica que la asignación se hace para todos sus posibles valores y combinaciones de ellos.

2. Se puede hacer una asignación específica dando los valores de los índices entre comillas:

c('p1','m1')=0.180;

3. Se pueden asignar mas de una vez valores a los escalares, parámetros y tablas. Los nuevos valores reemplazan a los antiguos.

4. Las expresiones matemáticas pueden incorporar funciones matemáticas estándar como se muestra en la siguiente tabla.

Función	Descripción
abs(x)	Valor absoluto de x
arctan(x)	Arco tangente (en radianes)
ceil(x)	Mínimo entero mayor o igual que x
cos(x)	Función coseno (x en radianes)
errorf(x)	Función de distribución de la normal N (0,1) en x
exp(x)	Función exponencial
floor(x)	Mayor entero menor o igual que x
log(x)	Logaritmo natural de x
log10(x)	Logaritmo en base 10 de x
mapval(x)	Función proyección
max(x_1, x_2, ...)	Máximo de una lista
min(x_1, x_2, ...)	Mínimo de una lista
mod(x,y)	Resto al dividir x por y
normal(x,y)	Número aleatorio de una variable normal con medida x y desviación típica y
power(x,y)	Función potencial x^y (donde y debe ser un entero)
x**y	Función potencial x^y (donde x debe ser positiva)
round(x)	Redondeo de x al entero mas cercano
round(x,y)	Redondea x a y decimales
sign(x)	Signo de x, 1 si es positivo, -1 si es negativo y 0 si es nulo
sin(x)	Función seno (en radianes)
sqr(x)	Cuadrado de x
sqrt(x)	Raíz cuadrada de x

Instrucción Model y Solve

La palabra *Model* tiene un significado muy preciso en GAMS. Es una simple colección de ecuaciones. El formato de la declaración es la palabra clave *Model,* el nombre del modelo que el usuario quiera definir, seguida por una lista de ecuaciones encerrada en diagonales '/'. Si todas las ecuaciones previamente definidas van a ser incluidas, se puede utilizar */all/* en lugar de listar las ecuaciones. Por ejemplo:

Model Transport /all/;

Esta instrucción puede verse superflua, pero es útil para usuarios avanzados que necesitan crear varios modelos en una corrida de GAMS. Si se quiere utilizar una lista explícita en vez de la palabra clave */all/*. La instrucción quedaría:

Model Transport /cost, supply, demand/;

El dominio es omitido de la lista de ecuaciones debido a que no son parte del nombre de la ecuación. La opción de la lista es usada solamente cuando se utiliza un subconjunto de las ecuaciones existentes que comprenden un modelo ha ser generado.

Una vez que el modelo ha sido declarado y las ecuaciones asignadas, estamos listo para llamar al resolvedor. Esto se hace con la instrucción *Solve,* por ejemplo:

Solve Transport using lp minimizing z;

El formato de la instrucción Solve es el siguiente:

1. La palabra clave Solve
2. El nombre del modelo a resolver
3. La palabra clave "using"
4. Un procedimiento de solución disponible. La lista completa de los procedimiento es (no todos los resolvedores están disponibles en nuestro sistema):

lp	para programación lineal
nlp para programación no lineal	
mip	para programación mixta entera
rmip	para programación mixta entera relajada
minlp	para programación mixta entera no lineal
rminlp	para programación mixta entera no lineal relajada
mcp	para problemas complementarios mixtos
cns	para sistemas de restricciones no lineales

5. La palabra clave minimizing o maximizing
6. El nombre de la variable a optimizar.

Instrucción Display

Para obtener los valores óptimos de las variables, se puede observar la salida del resolvedor, o si se quiere, se puede pedir que se desplieguen estos valores del GAMS. Por ejemplo:

Display x.l, x.m;

Esto provoca una impresión de los niveles (valores) finales (x.l) y marginales (x.m) de la variable x. GAMS automáticamente da formato a esta impresión.

GAMS fue diseñado con un pequeño sistema de base de datos en el cual los registros se mantienen para las variables y ecuaciones. Existen cuatro campos en dichos registros:

.lo	=	limite inferior
.l	=	valor de nivel o primal
.up	=	limite superior
.m	=	valor marginal o dual

El formato para reverenciar estas cantidades es el nombre de la variable o ecuación seguida por el nombre del campo, seguida (si es necesario) por el dominio (o un elemento del dominio).

GAMS permite al usuario un acceso completo a la base de datos para leer y escribir.

<u>Asignación de valores iniciales y límites a las variables</u>

El limite superior e inferior de una variable es establecida automáticamente de acuerdo al tipo de variable (free, positive, negative, binary, o integer), pero estos límites pueden ser sobrescritos por el usuario de GAMS. Algunos ejemplos de esto se muestran a continuación:

```
x.up(I,J)   =   capacity(I,J);
x.lo(I,J)   =    10.0;
x.up('seattle' ,'New-York') = 1.2 * capacity('seattle', 'new-york');
```

En el primer y tercer ejemplo, se supone que "capacity" es un parámetro previamente declarado y con valores asignados. Esta instrucción debe aparecer después de la declaración de variable y antes de la instrucción solve. Todas las expresiones matemáticas disponibles para asignación directa son usadas en el lado derecho.

En la programación es muy importante para el modelador ayudar al resolvedor con la especificación de un rango corto como sea posible entre los limites superiores e inferiores. Es también bastante útil la especificación de una solución inicial de la cual el resolvedor pueda empezar la búsqueda del óptimo.

Es importante entender que los campos .lo y .up son enteramente controlados por el usuario del GAMS, en contraste con los campos .l y .m los cuales el usuario solo puede inicializar pero el resolvedor es el que los controla.

__Salida del GAMS__

La salida por omisión de una corrida en GAMS es muy extensa e informativa, consistente de las siguientes secciones:

- Impresión del listado
- Mensajes de error
- Mapas de referencia
- Listado de ecuaciones
- Estadísticas del modelo
- Estado del reporte
- Reporte de la solución

Existan o no existan errores en el problema de optimización que se está resolviendo, la primera sección de la salida de una corrida de GAMS es la impresión del archivo de entrada. GAMS pone número de líneas en la parte izquierda de cada renglón.

Cuando el compilador del GAMS encuentra un error en el archivo de entrada inserta un código de mensaje de error debajo de la línea en que se encuentra el error. Este mensaje siempre empieza con **** y contiene un $ directamente en el punto donde el compilador piensa que está el error. El $ es seguido por un código numérico de error, el cual es explicado después de la impresión del listado.

La siguiente sección de la salida, la cual es la última si un error es encontrado, es una pareja de mapas de referencias que contiene un resumen y análisis del archivo de entrada con el propósito de depuración y documentación.

Si no se encontró ningún error, al final del archivo de salida se encuentra el Reporte de la Solución. En esta sección se define primero el estado del resolvedor (por ejemplo, "optimo global encontrado" para problemas lineales), se dan estadística del modelo matemático y de su solución y finalmente se presentan los valores finales de las variables (valores óptimos y marginales) y de las ecuaciones (valores óptimos y marginales).

4.2.2 GAMS – IDE v.19.4

La práctica totalidad de empresa dedicada a desarrollar software van actualizando progresivamente las versiones de sus productos haciéndolas cada vez accesibles, tanto en el uso como en las prestaciones. GAMS Corporation no podía ser menos, y por ello durante los últimos años ha estado desarrollado una versión de su producto en un entorno integrado, es decir, que sea capaz de aprovechar todas las prestaciones que en la actualidad dispone el sistema operativo Windows®. Ese esfuerzo ha desembocado en las sucesivas versiones del GAMS-IDE (Integrated Development Enviroment).

Durante mucho tiempo GAMS ha desarrollado su potencial bajo los sistemas DOS y UNIX mientras una parte de su "competencia" LINDO, GINO, etc. desarrollaba sus aplicaciones para Windows. Por ello, en esta breve referencia nos referiremos a la nueva versión de GAMS-IDE.

Lo primero que necesitamos saber es como poder disponer de una versión del programa. Para obtener una copia de este programa, es necesario conectar con la siguiente dirección:

http://www.gams.de/5download/cd.htm

En esa pagina web (web site) es necesario registrarse con una dirección de correo electrónico. Una vez registrados se recibe una contraseña para poder acceder a la página de descarga de los archivos necesarios. En la misma página de registro se informa de las limitaciones de esta versión si es usada para construcción y resolución de modelos de optimización:

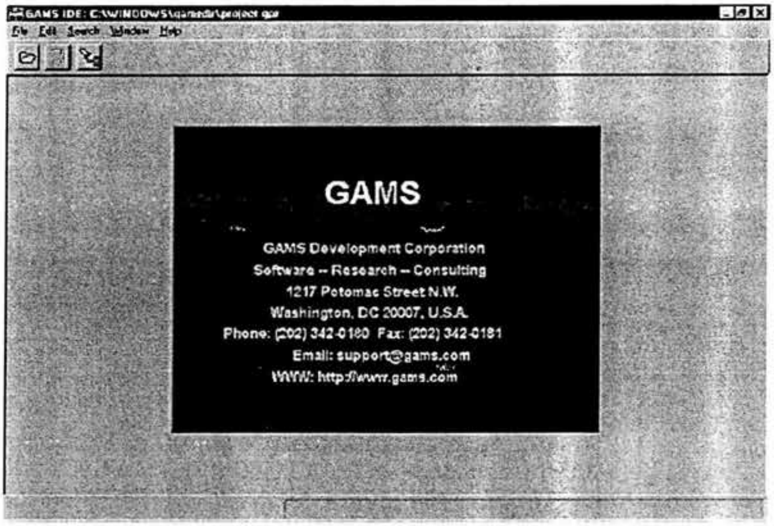
Máximo número de filas: 300
Máximo número de columnas: 300
Máximo número de elementos distintos de cero: 2000
Máximo número de elementos no lineales: 1000
Máximo número de variables discretas: 50

Una vez copiados los archivos en el disco, se procede a la instalación automática del programa. En esa instalación se crea un subdirectorio conteniendo una serie de manuales de ayuda (todos ellos en formato PDF). Estos manuales incluyen una guía de usuario. Además incluye un "Tutorial" , además de una breve referencia en la ayuda del programa. Pero no solo tiene ayuda sobre el programa, sino también tiene referencias y guías sobre los solvers que se usan como MINOS, CPLEX, OSL, etc.

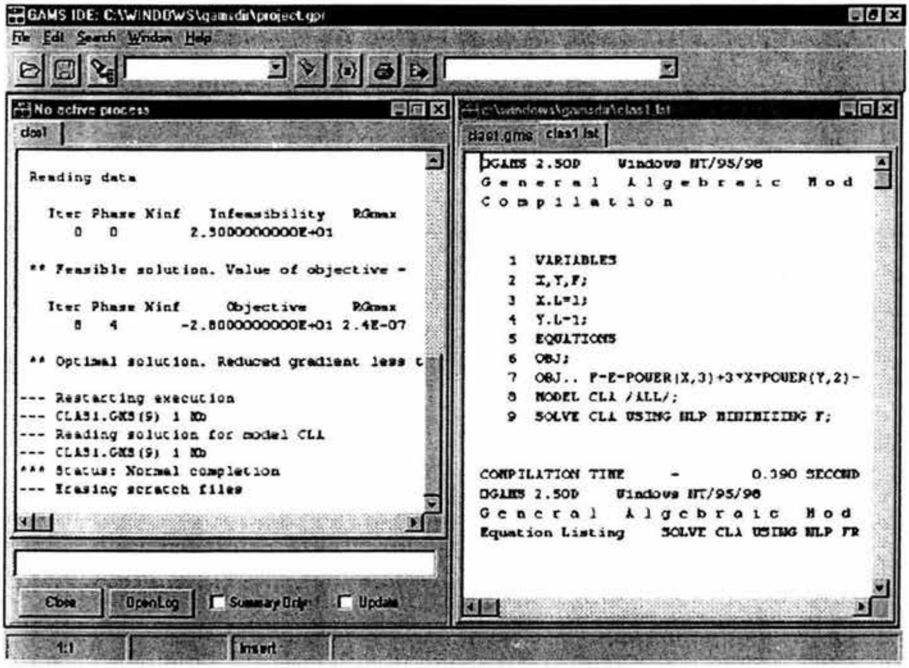
Una vez instalado se puede ejecutar alguno de los modelos de la librería de programas, ya sea lineal, no lineal, entero, PNL no diferenciable, Programación complementaria, etc.

Como se ha comentado, la versión que se obtiene gratuitamente es una versión "demo" o estudiante, que es perfectamente aplicable a las explicaciones docentes o de apoyo a las clases prácticas. Si se requiere resolver modelos de mayor tamaño, la solución inmediata consiste en adquirir una versión "académica" que no tiene limite de capacidad, ya que la única limitación es la disposición del sistema operativo del ordenador sobre el que esta instalado.

Esto ocurre con todos los programas "comerciales", que disponen de una versión "reducida" y si se quiere ampliar es necesario adquirir la licencia correspondiente. Sin embargo, si se quiere ampliar la potencia de resolución del programa puede acudirse a una segunda opción, que se comentará más adelante.



En las imágenes anteriores se puede observar la primera portada del programa (dependiendo de la versión), mientras que en la siguiente podemos ver los resultados de la ejecución y los archivos GMS y LST.



Al codificar en GAMS, hay que organizar una serie de bloques que son obligatorios y otros bloques que son opcionales. Nos centraremos en los bloques obligatorios, nombrando solamente los optativos. Los bloques obligatorios son:

Variables	VARIABLES
Ecuaciones	EQUATIONS
Modelo	MODEL
Solución SOLVE	

Los bloques optativos son:

Conjuntos	SET
Datos	DATA
Visualización	DISPLAY

Antes de explicar brevemente cada uno de los bloques obligatorios, siempre comenzaremos la codificación con una serie de indicaciones y comentarios respecto a los problemas que vamos a crear.

Estas líneas de comentario o explicación se pueden realizar de dos formas: a) comenzando cada línea con un asterisco (*), en este caso hay que tomar en consideración que ciertos símbolos están prohibidos, como por ejemplo los acentos, b) bien definiendo un bloque de comentarios que comienzan el la primera línea con la marca $ONTEXT y finaliza con la marca de $OFFTEXT, y entre ambas marcas se puede escribir cualquier expresión.

Bloque de variables. Este bloque debe comenzar con la palabra VARIABLES. Dentro de este bloque se han de definir las variables que se van a usar en el modelo.

Bloque de ecuaciones. Este bloque ha de comenzar con él titulo EQUATIONS. En este bloque hay que declarar y definir las ecuaciones que se van a usar en el modelo.

Bloque de modelo. En este grupo se ha de definir las ecuaciones que componen el modelo. No es obligatorio definir todas las ecuaciones utilizadas. Este bloque tiene que comenzar con el nombre de MODEL.

Bloque de solución. En este bloque hay que indicar que tipo de algoritmo deseamos usar para poder resolver el modelo que se ha definido previamente. A la hora de inicializar este bloque ha de aparecer la palabra SOLVE.

Además de estos cuatro bloques obligatorios, y como ya se ha indicado con anterioridad, se pueden definir otros tres bloques de carácter opcional:

Bloque de Conjuntos, SET. Consiste en definir una serie de conjuntos, por lo general índices, que asignarles unos valores a estos conjuntos.

Bloque de Datos, DATA. No se trata de un único bloque, sino que puede contener diferentes grupos. Se usa para definir una serie de datos fijos dentro del modelo, así podemos definir parámetros (PARAMETERS), tablas (TABLES) y escalares (SCALARS).

Bloque de visualización, DISPLAY. Este bloque permite indicar la clase de salida de datos, y formato, deseamos para el problema. Posteriormente haremos referencia a las diferentes opciones, en principio nos limitaremos a comentarla salida estándar (por defecto) que proporciona GAMS.

Vamos a comenzar por explicar la creación de un archivo de datos, con los bloques obligatorios.

El archivo lo crearemos utilizando un editor utilizando el nombre de UTILGAMS.GMS. La extensión GMS es la que por defecto se usa para identificar a los archivos de GAMS. Comenzaremos la codificación con una serie de líneas de comentarios que nos permitan identificar el problema que vamos a resolver.

Los comentarios pueden ser como los recogidos en el bloque siguiente:

```
$ONTEXT

        Se trata de la creación de un archivo de datos que permita
crear y resolver un problema clásico de maximización de la utilidad
de un consumidor sujeto a una restricción presupuestaria.

         Para definir el problema, comenzaremos por considerar dos
bienes ( denominaremos X e Y ), cuyos precios de mercado son de
4 y 6 unidades monetarias respectivamente.

       La cantidad total disponible por el consumidor es de 130 u.m.


       La función de utilidad del consumidor es:

       U(X,Y) = ( X + 2) * ( Y + 1)

       Por tanto, el problema a resolver será:

       Max U(X,Y)
       s.a:
       Px * X + Py * Y <= D

        En este problema consideraremos, que nos se pueden adquirir
cantidades negativas de ambos bienes, por tanto, las dos variables
estarán sujetas a las condiciones de no negatividad

A lo largo de la construcción del problema, y mediante un asterisco *
iremos indicando los comentarios pertinente.

        Este bloque de comentarios finalizar con la marca de:

$OFFTEXT
```

El siguiente bloque obligatorio es el bloque de variables. Como su nombre indica este bloque debe ser identificado con la palabra VARIABLES, y dentro de este grupo podemos diferenciar varios apartados.

a) Nombre de las variables del problema. (Obligatorio). El nombre de las variables puede ser arbitrario, es decir, x1, Y, alfa, etc., pero siempre de hasta ocho caracteres. Junto al nombre podemos añadir los comentarios pertinentes. Conviene hacer notar que a la hora de definir las variables hemos de definir las variables del modelo, incluyendo la variable que represente el valor de la función objetivo, no obstante, podemos relacionar más variables aunque no se utilicen con posterioridad.

Al final de este bloque, al igual que en todos los bloques, se ha de indicar que ha finalizado mediante un punto y coma (;).

b) Clase de variables. Una vez definidas las variables podemos asociar a que clase pertenecen, es decir, si se trata de variables no negativas (POSITIVE), variables libres (FREE), variables binarias (BINARY), variables enteras (INTEGER), etc. El valor por defecto es de variables libre, es decir, que sino indicamos nada de cada variable, se consideran variables libres.

c) Cotas sobre las variables. Podemos restringir los valores que pueden tomar las variables introduciendo un valor para las cotas superiores o inferiores de las variables. En caso de no definir las cotas, la opción por defecto, es que las variables pueden tomar cualquier valor entre menos infinito e infinito. Para definir las cotas usaremos:

Cota Superior	Nombre.UP = {valor}
Cota Inferior	Nombre.LO = {valor}

d) Valores iniciales de las variables. Aunque posteriormente haremos referencia a la conveniencia de introducir unos valores iniciales de las variables. Todos los algoritmos de resolución necesitan un punto inicial en donde comenzar el proceso iterativo de búsqueda de la solución, por ello es conveniente fijar un punto de partida. En caso de no definir este punto inicial el algoritmo asume que se inicia el proceso en el origen de coordenadas, es decir, (0,0,..0).

La definición se realiza como:

Punto de Partida	Nombre.L = {valor}

Vamos a ilustrar lo anterior con el bloque de variables del archivo UTILGAMS.GMS:

```
*PRIMER BLOQUE OBLIGATORIO: BLOQUE DE VARIABLES
*IDENTIFICACION:

VARIABLES

* RELACION DE LAS VARIABLES A INCLUIR:

X      BIEN-1
Y      BIEN-2

* HAY QUE INCLUIR UNA VARIABLE PARA RECOGER EL VALOR DE LA
* FUNCION OBJETIVO. CADA GRUPO FINALIZAR CON UN ;

F;
```

```
*DENTRO DE ESTE MISMO BLOQUE, PODEMOS INCLUIR DOS APARTADOS:
*A) DEFINICION DE LA CLASE DE VARIABLES: NO NEGATIVAS X e Y.

POSITIVE VARIABLES X,Y;

*B) INDICACION DEL PUNTO DE PARTIDA DEL ALGORITMO. PUNTO (1,1)

X.L = 1.0;
Y.L = 1.0;

*FINAL DEL BLOQUE DE VARIABLES.
```

Las dos variables introducidas (X e Y) representan la cantidad a adquirir de cada uno de los dos bienes, mientras que la variable F representa el valor de la función de utilidad. Tal como recoge el enunciado del problema, las cantidades a adquirir de los dos bienes no pueden ser negativas, por tanto, se definen ambas variables como POSITIVE VARIABLES, mientras que la variable F no se clasifica, con lo que se asume que es una variable libre.

Alternativamente a esto se podrían haber utilizado las definiciones de las cotas de las dos variables, en lugar de la etiqueta de variables positivas. Es decir, bastaría con definir una cota inferior de cero para cada variable, y considerar a estas como variables libres.
Evidentemente, la formulación elegida es más cómoda a la hora de "visualizar" el problema. La alternativa sería:

```
* RELACION DE LAS VARIABLES A INCLUIR:

X      BIEN-1
Y      BIEN-2

* HAY QUE INCLUIR UNA VARIABLE PARA RECOGER EL VALOR DE LA
* FUNCION OBJETIVO. CADA GRUPO FINALIZAR CON UN ;

F;

*DENTRO DE ESTE MISMO BLOQUE, PODEMOS INCLUIR DOS APARTADOS:
*A) DEFINICION DE COTAS PARA LAS VARIABLES

X.LO = 0.0;
Y.LO = 0.0;
```

Una vez definido el bloque de variables, hemos de incluir el bloque de ecuaciones. Bloque que ha de comenzar con el identificativo EQUATIONS.

Dentro de este bloque podemos considerar dos partes:

a) Nombre de las funciones o ecuaciones. Esta primera parte del bloque sirve para relacionar el nombre de todas las funciones que se han de utilizar en el modelo. El nombre de las ecuaciones puede ser arbitrario, y como máximo de ocho caracteres, pudiendo añadirse los comentarios pertinentes. Este grupo finaliza con un punto y coma (;).

b) Definición de las funciones. En este apartado hemos de relacionar algebraicamente las variables para formar las funciones. Inicialmente usaremos la notación de:

suma	+
diferencia	-
producto	*
cociente	/
exponente	** ó POWER(X,n)

Para indicar la relación entre la función y los términos independientes de las restricciones usaremos los siguientes símbolos:

igualdad	=e=
menor-igual	=l=
mayor-igual	=g=

Al final de cada ecuación hemos de poner la marca de final, es decir, un punto y coma.
Vamos a ver como hemos construido este bloque en el archivo UTILGAMS.GMS

```
*SEGUNDO BLOQUE OBLIGATIRIO: ECUACIONES.
*INDENTIFICACION:

EQUATIONS

* PRIMERA PARTE DEL BLOQUE: DEFINICION DE LAS FUNCIONES:

UTIL  FUNCION DE UTILIDAD
RP    RESTRICCION PRESUPUESTARIA;

*SEGUNDA PARTE: DECLARACION DE LAS FUNCIONES:

UTIL.. F =E= (X+2)*(Y+1);

RP.. 4 * X + 6 * Y =L= 130;
```

A partir del enunciado del problema, hemos definido la función de utilidad con el nombre de UTIL y la restricción presupuestaria con el nombre RP.

La función de utilidad la hemos definido tal como aparece en el enunciado del problema, mientras que la restricción presupuestaria se ha obtenido sin más que multiplicar la cantidad a adquirir de cada bien (variables) por el precio correspondiente. El límite de gasto es de 130 unidades monetarias, pero dado que no se exige gastar toda la cantidad disponible, la restricción es de la forma menor-igual.

Nótese que al final de cada ecuación se ha incluido la marca (;).

El tercer bloque obligatorio corresponde al grupo MODEL. En este bloque hay que asignar un nombre al modelo que queremos resolver, así como relacionar las ecuaciones que forman parte del modelo. En el caso que se usen todas las ecuaciones definidas, se puede sustituir el nombre de cada una de ellas por la palabra ALL, es decir, todas las ecuaciones. En este archivo hemos preferido relacionar el nombre todas las ecuaciones.

Así, el formato sería:

```
* TERCER BLOQUE OBLIGATORIO: BLOQUE DE MODELO.
* HAY QUE INDICAR EL NOMBRE DEL MODELO Y LAS ECUACIONES QUE
* FORMAN PARTE DEL MISMO.

MODEL UTILGAMS/UTIL,RP/;
```

El cuarto bloque obligatorio a relacionar es del bloque de SOLVE. Aquí hay que indicar lo que queremos hacer con el modelo, es decir, resolverlo. Este bloque consta de tres grupos, aunque se relacionan uno a continuación del otro, indicando:

a) El nombre del modelo a resolver.
b) La clase de modelo de que se trata: LP, NLP. MIP, etc.
c) La dirección de optimización de la función objetivo, es decir, maximizar o minimizar.

El formato sería:

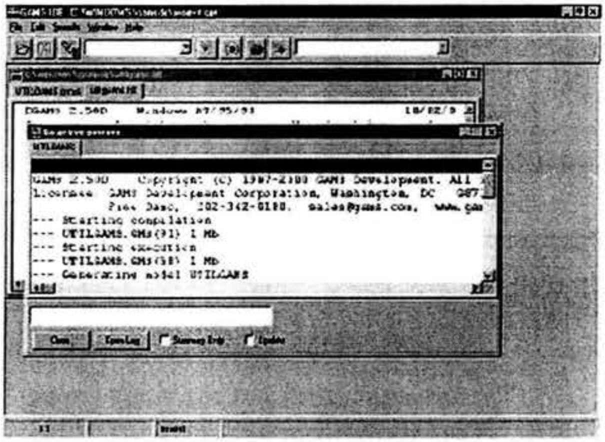
```
* CUARTO BLOQUE OBLIGATORIO: BLOQUE DE SOLUCION.
* DADO QUE SE TRATA DE UN PROBLEMA NO LINEAL, HEMOS DE INDICAR
* QUE HEMOS DE USAR UN PROGRAMA DE PNL (NLP).
* TAMBIEN HEMOS DE INDICAR LA DIRECCION DE OPTIMIZACION:
* MAXIMIZACION O MINIMIZACION DE LA FUNCION OBJETIVO.

SOLVE UTILGAMS USING NLP MAXIMIZING F;
```

La codificación en GAMS completa seria de la siguiente forma:

```
$ONTEXT

        Se trata de la creación de un archivo de datos que permita
crear y resolver un problema clásico de maximización de la utilidad
de un consumidor sujeto a una restricción presupuestaria.

        Para definir el problema, comenzaremos por considerar dos
bienes ( denominaremos X e Y ), cuyos precios de mercado son de
4 y 6 unidades monetarias respectivamente.

La cantidad total disponible por el consumidor es de 130 u.m.

       La función de utilidad del consumidor es:

       U(X,Y) = ( X + 2) * ( Y + 1)

       Por tanto, el problema a resolver ser:

       Max U(X,Y)
       s.a:
       Px * X + Py * Y <= D

       En este problema consideraremos, que nos se pueden adquirir
cantidades negativas de ambos bienes, por tanto, las dos variables
estaran sujetas a las condiciones de no negatividad

A lo largo de la construcción del problema, y mediante un asterisco *
iremos indicando los comentarios pertinente.

       Este bloque de comentarios finalizar con la marca de:

$OFFTEXT

*PRIMER BLOQUE OBLIGATORIO: BLOQUE DE VARIABLES
*IDENTIFICACION:

VARIABLES

* RELACION DE LAS VARIABLES A INCLUIR:


X      BIEN-1
Y      BIEN-2

* HAY QUE INCLUIR UNA VARIABLE PARA RECOGER EL VALOR DE LA
* FUNCION OBJETIVO. CADA GRUPO FINALIZAR CON UN;

F;

*DENTRO DE ESTE MISMO BLOQUE, PODEMOS INCLUIR DOS APARTADOS:
```

```
*A) DEFINICION DE LA CLASE DE VARIABLES: NO NEGATIVAS X e Y.

POSITIVE VARIABLES X,Y;

*B) INDICACION DEL PUNTO DE PARTIDA DEL ALGORITMO.

X.L = 1.0;
Y.L = 1.0;

*FINAL DEL BLOQUE DE VARIABLES.

*SEGUNDO BLOQUE OBLIGATIRIO: ECUACIONES.
*INDENTIFICACION:

EQUATIONS

* PRIMERA PARTE DEL BLOQUE: DEFINICION DE LAS FUNCIONES:

UTIL FUNCION DE UTILIDAD
RP RESTRICCION PRESUPUESTARIA;

*SEGUNDA PARTE: DECLARACION DE LAS FUNCIONES:

UTIL.. F =E= (X+2)*(Y+1);

RP.. 4 * X + 6 * Y =L= 130;

*TERCER BLOQUE OBLIGATORIO: BLOQUE DE MODELO.
* HAY QUE INDICAR EL NOMBRE DEL MODELO Y LAS ECUACIONES QUE
* FORMAN PARTE DEL MISMO.

MODEL UTILGAMS/UTIL,RP/;

* CUARTO BLOQUE OBLIGATORIO: BLOQUE DE SOLUCION.
* DADO QUE SE TRATA DE UN PROBLEMA NO LINEAL, HEMOS DE INDICAR
* QUE HEMOS DE USAR UN PROGRAMA DE PNL (NLP).
* TAMBIEN HEMOS DE INDICAR LA DIRECCION DE OPTIMIZACION:
* MAXIMIZACION O MINIMIZACION DE LA FUNCION OBJETIVO.

SOLVE UTILGAMS USING NLP MAXIMIZING F;
```

Una vez creado, y archivado, el código fuente correspondiente, es necesario resolverlo. Para ello hay que ejecutar el programa GAMS. Se elige la opción RUN en FILE o pulsando F9, con ello se obtiene una pantalla como la siguiente:



De esa pantalla, lo más importante es el archivo con extensión LST:

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

1

      Se trata de la creación de un archivo de datos que permita
   crear y resolver un problema clásico de maximización de la utilidad
   de un consumidor sujeto a una restricción presupuestaria.

      Para definir el problema, comenzaremos por considerar dos
   bienes ( denominaremos X e Y ), cuyos precios de mercado son de
   4 y 6 unidades monetarias respectivamente.

      La cantidad total disponible por el consumidor es de 130 u.m.

   La función de utilidad del consumidor es:

   U(X,Y) = ( X + 2) * ( Y + 1)

   Por tanto, el problema a resolver ser:

   Max U(X,Y)
   s.a:
   Px * X + Py * Y <= D

   En este problema consideraremos, que nos se pueden adquirir
  cantidades negativas de ambos bienes, por tanto, las dos variables
```

```
 estarán sujetas a las condiciones de no negatividad

 A lo largo de la construcción del problema, y mediante un asterisco *
 iremos indicando los comentarios pertinente.

      Este bloque de comentarios finalizar con la marca de:


*PRIMER BLOQUE OBLIGATORIO: BLOQUE DE VARIABLES
*IDENTIFICACION:

VARIABLES

* RELACION DE LAS VARIABLES A INCLUIR:

X         BIEN-1
Y         BIEN-2

* HAY QUE INCLUIR UNA VARIABLE PARA RECOGER EL VALOR DE LA
* FUNCION OBJETIVO. CADA GRUPO FINALIZAR CON UN ;

F;

*DENTRO DE ESTE MISMO BLOQUE, PODEMOS INCLUIR DOS APARTADOS:
*A) DEFINICION DE LA CLASE DE VARIABLES: NO NEGATIVAS X e Y.

POSITIVE VARIABLES X,Y;

*B) INDICACION DEL PUNTO DE PARTIDA DEL ALGORITMO.

X.L = 1.0;

GAMS 2.50D Windows NT/95/98          10/02/00 09:20:31 PAGE 2
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m
C o m p i l a t i o n

Y.L = 1.0;

*FINAL DEL BLOQUE DE VARIABLES.

*SEGUNDO BLOQUE OBLIGATIRIO: ECUACIONES.
*INDENTIFICACION:

EQUATIONS

* PRIMERA PARTE DEL BLOQUE: DEFINICION DE LAS FUNCIONES:

UTIL FUNCION DE UTILIDAD
RP RESTRICCION PRESUPUESTARIA;

*SEGUNDA PARTE: DECLARACION DE LAS FUNCIONES:

```

```
74 UTIL.. F =E= (X+2)*(Y+1);
75

76 RP.. 4 * X + 6 * Y =L= 130;
77
78 *TERCER BLOQUE OBLIGATORIO: BLOQUE DE MODELO.
79 * HAY QUE INDICAR EL NOMBRE DEL MODELO Y LAS ECUACIONES QUE
80 * FORMAN PARTE DEL MISMO.
81
82 MODEL UTILGAMS/UTIL,RP/;
83
84 *CUARTO BLOQUE OBLIGATORIO: BLOQUE DE SOLUCION.
85 * DADO QUE SE TRATA DE UN PROBLEMA NO LINEAL, HEMOS DE INDICAR
86 * QUE HEMOS DE USAR UN PROGRAMA DE PNL (NLP).
87 * TAMBIEN HEMOS DE INDICAR LA DIRECCION DE OPTIMIZACION:
88 * MAXIMIZACION O MINIMIZACION DE LA FUNCION OBJETIVO.
89
90 SOLVE UTILGAMS USING NLP MAXIMIZING F;
91


COMPILATION TIME    =      0.050 SECONDS      0.7 Mb      WIN-19-113
=====================================================================
GAMS 2.50D      Windows NT/95/98             10/02/00 09:20:31       PAGE 3
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
Equation Listing SOLVE UTILGAMS USING NLP FROM LINE 90

---- UTIL =E= FUNCION DE UTILIDAD


UTIL.. - (2)*X - (3)*Y + F =E= 0 ; (LHS = -6, INFES = 6 ***)


---- RP =L= RESTRICCION PRESUPUESTARIA


RP.. 4*X + 6*Y =L= 130 ; (LHS = 10)
GAMS 2.50D Windows NT/95/98 10/02/00 09:20:31 PAGE 4
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m
Column Listing SOLVE UTILGAMS USING NLP FROM LINE 90


---- X BIEN-1

X
                 (.LO, .L, .UP = 0, 1, +INF)
      (-2)       UTIL
       4         RP

---- Y BIEN-2
```

```
Y
                (.LO, .L, .UP = 0, 1, +INF)
       (-3)     UTIL
        6       RP



---- F

F
                    (.LO, .L, .UP = -INF, 0, +INF)
         1       UTIL
==========================================================================
GAMS 2.50D Windows NT/95/98            10/02/00 09:20:31     PAGE    5
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

Model Statistics SOLVE UTILGAMS USING NLP FROM LINE 90



MODEL STATISTICS

BLOCKS OF EQUATIONS      2     SINGLE EQUATIONS          2
BLOCKS OF VARIABLES      3     SINGLE VARIABLES          3
NON ZERO ELEMENTS        5     NON LINEAR N-Z            2
DERIVATIVE POOL          5     CONSTANT POOL             8
CODE LENGTH              18


GENERATION TIME    =     0.110 SECONDS      1.4 Mb      WIN-19-113
EXECUTION TIME     =     0.110 SECONDS      1.4 Mb      WIN-19-113
==========================================================================
GAMS 2.50D  Windows NT/95/98            10/02/00 09:20:31   PAGE   6
G e n e r a l  A l g e b r a i c  M o d e l i n g  S y s t e m

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

     MODEL UTILGAMS    OBJECTIVE F
     TYPE        NLP        DIRECTION MAXIMIZE
     SOLVER      MINOS      FROM LINE 90

**** SOLVER STATUS             1 NORMAL COMPLETION
**** MODEL STATUS              2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE                 216.0000

RESOURCE USAGE, LIMIT          0.270        1000.000
ITERATION COUNT, LIMIT         3           10000
EVALUATION ERRORS              0               0


     GAMS/MINOS Wintel version 5.5-002-037
```

```
      B. A. Murtagh, University of New South Wales
      P. E. Gill, University of California at San Diego,
      W. Murray, M. A. Saunders, and M. H. Wright,
      Systems Optimization Laboratory, Stanford University

You do not have a full license for this solver,
Continue to run in demonstration mode.
The following size restrictions apply:
      Maximum equations              : 300
      Maximum variables              : 300
      Maximum nonzero elements              : 2000
      Maximum Non-linear non-zeroes : 1000



Work space allocated -- 1.67 Mb

EXIT - Optimal Solution found.

Major, Minor Iterations         1        3
Funobj, Funcon calls            9        0
Superbasics                     1
Aggregations                    0
Interpreter Usage            0.00      0.0%

Work space used by solver   --    0.96 Mb

                          LOWER      LEVEL      UPPER   MARGINAL

---- EQU UTIL               .          .          .        1.000
---- EQU RP               -INF    130.000    130.000      3.000

UTIL FUNCION DE UTILIDAD
RP RESTRICCION PRESUPUESTARIA
==========================================================================
GAMS 2.50D Windows NT/95/98 10/02/00 09:20:31 PAGE 7
G e n e r a l A l g e b r a i c M o d e l i n g S y s t e m

                          LOWER      LEVEL      UPPER     MARGINAL

---- VAR X                  .        16.000      +INF        EPS
---- VAR Y                  .        11.000      +INF         .
---- VAR F                -INF      216.000      +INF         .

X      BIEN-1
Y      BIEN-2
F

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

```
EXECUTION TIME      =          0.050    SECONDS    0.7 Mb        WIN-19-113

USER: GAMS Development Corporation, Washington, DC G871201:0000XX-XXX
       Free Demo, 202-342-0180, sales@gams.com, www.gams.com DC9999

**** FILE SUMMARY

INPUT       C:\WINDOWS\GAMSDIR\UTILGAMS.GMS
OUTPUT      C:\WINDOWS\GAMSDIR\UTILGAMS.LST
```

Las primeras páginas, en este caso las páginas 1 y 2, corresponden al archivo de datos (UTILGMS.GMS). En el caso que se haya producido errores en la codificación, el programa los detecta y señala adecuadamente, así como da una breve indicación del procedimiento de solución.

En este caso no se han producido errores, y por ello, recoge el archivo de datos original, sin más que numerar las filas del texto fuera del espacio marcado con ONTEXT-OFFTEXT. Esta parte se conoce con el nombre de *compilación de datos.*

También se indica el tiempo, en segundos, que se ha tardado en realizar el proceso de compilación.

Una vez relacionado el archivo de datos, GAMS realiza un análisis de las ecuaciones y de las variables del modelo. Las páginas 4 y 5 recogen estos análisis.

La pagina 3 se dedica a relacionar las ecuaciones utilizadas, con los coeficientes, evaluados en el punto de partida, y analizando si se cumplen o no las ecuaciones en ese punto. Así por ejemplo, en el caso de la restricción presupuestaria, donde los coeficientes de las variables son constantes (4 y 6) están asociados a cada una de las variables, y en el punto de partida (1,1), el valor de la función es de 10, frente a 130 que vale el termino independiente, por ello no advierte que tres asteriscos que no se cumple la ecuación, pero no tiene mayor importancia, ya que el propio algoritmo buscara un punto factible posteriormente.

En la pagina 4 se recoge la información relativa a las variables, así por ejemplo, referente a la variable X, o BIEN-1, aparece por una parte: (.LO, .L, .UP = 0, 1, +INF), esta expresión nos informa de las cotas y el punto de partida de la variable. La cota inferior (.LO) es cero, ya que la variable ha sido declarada como no negativa, la cota superior (.UP) no ha sido definida y por tanto el valor que toma por defecto es más infinito. El punto de partida que hemos elegido es 1, por ello (L = 1).

Además, aparecen los coeficientes asociados a cada una de las variables en las diferentes ecuaciones. Aquí, conviene advertir que los que aparecen asociados a la función objetivo llevan el signo de minimización (si el problema es de máximo se le cambia el signo) y el valor inicial asociado a las restantes variables.

El contenido de estas páginas es el siguiente:

La página 5 informa sobre la estadística del modelo ejecutado, es decir, nos informa sobre el número de variables y ecuaciones, del número de elementos distintos de cero, etc. Asimismo, nos muestra el tiempo empleado en general el código compilado y el tiempo de ejecución del problema.

El formato de esta página es el siguiente:

De todas las páginas de la solución la más importante de todas ellas es la pagina titulada REPORT SOLUTION. Como indica su nombre nos informa sobre la solución del problema.

Dentro de este informe podemos distinguir claramente tres partes:

a) Resumen de la solución (SOLVE SUMMARY).
En esta parte se informa del comportamiento de la solución, del tipo de modelo usado, la dirección de la optimización, etc. Es importante prestar atención a las líneas inicializadas con cuatro asteriscos (****), en este caso nos advierten que se ha completado correctamente el proceso y que la solución es optima (en esta caso localmente optima)

Este primer bloque se recoge a continuación, junto a la información del algoritmo usado (MINOS5).

b) El segundo bloque corresponde a la solución propiamente dicha. Aparece en primer lugar el comportamiento de las ecuaciones, es decir, el valor que toma cada una de las restricciones y la función objetivo.

La información respecto a la función objetivo es prácticamente nula. La restricción RP es una restricción de la forma menor-igual, por tanto esta acotada superiormente por el valor del término independiente, mientras que la cota inferior es de menos infinito. El valor de la restricción en el punto solución optima es el mismo que el del termino independiente - se comporta como una igualdad, esta saturada-, además hay que destacar que la columna MARGINAL recoge el valor de los multiplicadores de las restricciones, es decir, las variables duales.
Cuando aparece un punto (.) significa que el valor es cero, mientras que si aparece el símbolo EPS (epsilon) el valor es prácticamente cero pero no es nulo.

La información de las variables es similar, al de las ecuaciones, aparecen las cotas inferiores y superiores, así como el valor de la solución (LEVEL), es este caso podemos comprobar que los valores son X = 16, Y = 11, siendo el valor de la función objetivo de F = 216.

Este grupo se recoge en el siguiente bloque:

c) El ultimo bloque esta destinado a indicarnos los errores aparecidos durante la ejecución de la búsqueda de la solución. En este caso observamos que en el bloque comienza con cuatro asteriscos, y por tanto es importante hacer una referencia a esta información. Los resultados nos indican el número de variables no óptimas, infactibles o no acotadas que se han encontrado, así como el número de errores. Al ser todos ellos 0, la solución es correcta.

Aparece información referente al tiempo de ejecución, el nombre del usuario, y de los archivos de datos y solución.

Con lo que acabamos de exponer podemos crear y resolver un modelo con GAMS.
Aunque ya se ha expuesto con anterioridad es bastante fácil cometer errores, sobre todo al inicio del uso de GAMS, en la codificación. Para ello, vamos a dar una serie de indicaciones de como detectar, y corregir, algunos de los errores que se comenten con mayor frecuencia.

Comenzaremos por crear un archivo de datos, que vamos a denominar: UTILERR.GMS. Este archivo es similar al que hemos creado con anterioridad.

El contenido del archivo de datos es el siguiente:

```
VARIABLES

X      BIEN-1
Y      BIEN-2
F;

POSITIVE VARIABLES X,Y

X.L = 1.0;
Y.L = 1.0;

EQUATIONS

UTIL         FUNCION DE UTILIDAD
RP           RESTRICCION PRESUPUESTARIA;

UTIL.. F = (X+2)*(Y+1;

RP.. 4X + 6 * Y =L= 130;

MODEL UTILGAMS/UTIL,RP1/;

SOLVE UTILGAMS USING NLP MAXIMIZING F;
```

Al ejecutar este archivo de datos con GAMS, nos proporciona el archivo de resultados, obviamente con errores, pero además nos informa sobre la naturaleza de los errores cometidos y/o la forma de subsanarlos.

Vamos a analizar línea a línea los errores. Obsérvese que debajo de cada línea aparecen cuatro asteriscos así como una serie de número precedidos por el símbolo $.

```
GAMS 2.50D Windows NT/95/98 10/02/00                09:26:54    PAGE      1

G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n
1
2 VARIABLES
3
4
5 X          BIEN-1
6 Y          BIEN-2
7 F;
8
9 POSITIVE VARIABLES X,Y
10
11 X.L = 1.0;
**** $142$97
12 Y.L = 1.0;
13
14 EQUATIONS

15
16 UTIL      FUNCION DE UTILIDAD

17 RP        RESTRICCION PRESUPUESTARIA;
18
19 UTIL.. F = (X+2)*(Y+1;
****                     $8,37
20
21 RP.. 4X + 6 * Y =L= 130;
****              $37,409
22
23 MODEL UTILGAMS/UTIL,RP1/;
****                     $140
24
25 SOLVE UTILGAMS USING NLP MAXIMIZING F;
****                                 $257

GAMS 2.50D Windows NT/95/98 10/02/00                  09:26:54    PAGE    2
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m

Error Messages

   8 ')' expected
  37 '=l=' or '=e=' or '=g=' operator expected
  97 Explanatory text can not start with '$', '=', or '..'
     (-or- check for missing ';' on previous line)
 140 Unknown symbol
 142 No suffix allowed here - suffix ignored
 257 Solve statement not checked because of previous errors
```

```
409 Unrecognizable item - skip to find a new statement
     looking for a ';' or a key word to get started again


**** 8 ERROR(S) 0 WARNING(S)


COMPILATION TIME = 0.060 SECONDS 0.7 Mb WIN-19-113

USER: GAMS Development Corporation, Washington, DC G871201:0000XX-XXX
        Free Demo, 202-342-0180, sales@gams.com, www.gams.com DC9999

**** FILE SUMMARY


INPUT C:\WINDOWS\GAMSDIR\UTILERR.GMS
OUTPUT C:\WINDOWS\GAMSDIR\UTILERR.LST

**** USER ERROR(S) ENCOUNTERED
```

La primera línea con indicación de errores es la línea 11. Los números de codificación de los errores corresponden al 142 y 97. En la pagina 2, estos números corresponden a los mensajes de error que informan que en la línea anterior falta la marca de final (;). Es decir, en la línea 8 la instrucción POSITIVE VARIABLES X,Y ha de finalizar con un punto y coma.

En la línea 19, nos indica dos errores -los números 8 y 37-, que corresponden a: Falta un paréntesis y el operador =E=. Evidentemente falta el paréntesis final y el signo de igualdad en las ecuaciones no es = sino =E=.

A veces aparecen varios números de errores, sin aparente relación con el error cometido originalmente, pero es necesario analizarlos todos. Así por ejemplo, en la línea 21, los mensajes de error son los números: 37 y 409. Siendo el verdadero error el que se ha omitido el signo * entre el coeficiente y la variable X.

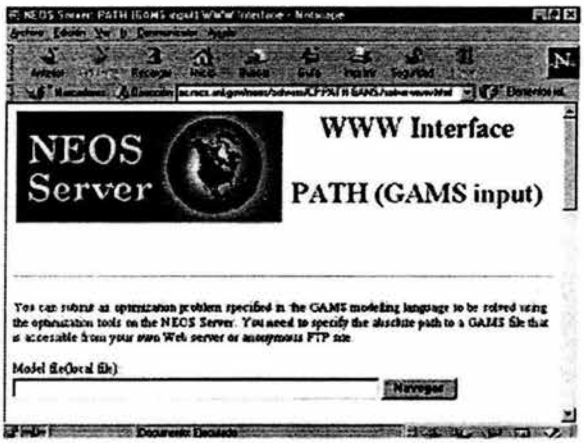
En la línea 23, aparece el error 140 -símbolo desconocido-, ello se debe a que hemos declarado que el modelo esta formado por una ecuación que no hemos definido en el bloque de variables.

Siempre que aparecen errores de usuario, también se especifica el error 257, que significa que no se ha resuelto el modelo porque se han detectado errores anteriores. Una vez analizados los errores en el archivo *.LST, hay que volver a editar el archivo *.GMS y corregir en este los errores detectados. Una vez corregidos hay que volver a ejecutar el archivo *.GMS y comprobar que se han subsanado los mismos, ya que en caso contrario hay que volver a repetir la operación de análisis de los errores.

A veces el tamaño del problema hace difícil resolver el problema por las limitaciones del solver se hace necesario usar un segundo procedimiento y es a través de Internet.

La segunda posibilidad es crear el archivo con el editor del GAMS-IDE, y una vez creado (con todas las opciones necesarias) se puede enviar vía web a la siguiente dirección:

http://www-neos.mcs.anl.gov/neos/solvers/CP:PATH-GAMS/solver-www.html



Se trata de una página del NEOS Server que admite la posibilidad de resolver en sus ordenadores (del tipo Sun Solaris) los archivos de GAMS que no es posible resolver con la versión de estudiante. La ventaja es que las instrucciones para crear los archivos son las mismas en entorno Windows o UNIX, y por tanto la única diferencia es la forma de crear el fichero. El archivo es enviado, junto con el fichero de opciones (sí es necesario, este fichero de opciones es necesario si, por ejemplo, se quiere hacer una análisis de sensibilidad de los coeficientes del modelo). Junto con el archivo hay que adicionar una dirección de correo electrónico, en donde recibir el archivo de la solución.

La pagina de NEOS, permite además el enviar los archivos vía e-mail o con has aplicaciones en JAVA (disponibles en el servidor como instalaciones cliente).

Una vez resuelto el programa, se recibe vía web (o mejor, vía e-mail) la solución en un archivo con extensión LST, tal como se hubiera resuelto en el ordenador local.

Esta es pues, una ventaja adicional de GAMS respecto de otros programas, pero al final debe ser el usuario-destinatario quien debe decidir la bondad o no este programa frente a otros de similares prestaciones y características. No hace falta reiterar que el que suscribe esta recensión es un usuario de este software, pero en todo caso, debe ser el usuario final el que debe analizar las ventajas e inconvenientes de este tipo de programas de ayuda a la docencia, y que si se pretende explicar o ayudar a la resolución de problemas de optimización clásica, no lineal, lineal y entera, hay pocos programas que permitan realizarlo con un mismo software, y además que sean de contrastada solvencia y garantía.

4.3 Elementos de estilo de programación

4.3.1 Generales

La programación la podemos considerar como *ciencia* y *arte*. Es ciencia en la medida que se pueden implantar modelos matemáticos complejos, donde 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* del código desarrollado. Piense 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 dato sí 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.

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 estudio 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 no lineal

1.-

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

$$\sum_{j=1}^{n} x_j = 1$$

$$\sum_{j=1}^{n} r_j x_j = r_0$$

2.-

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

$$\sum_{j=1}^{n} x_j = 1$$

$$\sum_{j=1}^{n} r_j x_j = r_0$$

3.-

$$\min \sum_{i=1}^{n} x_i w_i$$

$$W_i = \sum_{j=i+1}^{n} q_{ij} x_j$$

$$\sum_{j=1}^{n} X_j = 1$$

$$\sum_{j=1}^{n} r_j X_j = r_0$$

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 formulación 3 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. Es por esto que la formulación 3 resulta ser la más eficiente.

En la formulación del problema la opción profile muestra el tiempo y memoria consumida 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.

  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 nucas 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.

- Escalamiento 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=l) 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 muchos *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.

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 4.3 de comparación entre varios optimizadores y métodos de optimización. En la tabla 4.4 se muestra la diferencia de funcionamiento entre las opciones de preproceso de dos optimizadores.

		Caso1			Caso2		
		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 4.3 Comparación entre diferentes optimizadores en problemas LP.

	Caso1			Caso2		
	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 4.4 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.

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.

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.

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).

4.3.2. ESPECÍFICOS DE GAMS

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 incorrecta de un problema de optimización puede llevar a 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.

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:

- 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 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. Estos 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.

Desventajas

Una de las desventajas que posee el programa GAMS es que se lleva demasiado tiempo codificar las instrucciones. 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.

Con el objetivo de consumir el menor tiempo al momento de escribir las instrucciones se recomienda lo siguiente:

- 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 $) c ontroladas preferentemente mediante conjuntos dinámicos (mejor que con valores de parámetros), es decir, activar sólo las variables y restricciones necesarias. En el ejemplo previo del flujo de cargas se observa que se consideran sólo las líneas eléctricas que existen y no cualquier posible conexión entre dos nudos.

Tamaños máximos alcanzados

Como referencia final una indicación sobre el tamaño de los problemas que se están resolviendo y a los que se ha llegado aplicando estas recomendaciones. Se han podido resolver sin dificultad problemas de 117000 restricciones por 225000 variables con 655000 elementos no nulos en la matriz de restricciones en 240 segundos.

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. 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 ejecución de GAMS

```
gams nombre.gms suppress 1
```

Además, también se puede suprimir la información en pantalla que produce el optimizador con los siguientes parámetros

```
gams nombre.gms 110 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. 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 asincrónicamente. Una vez terminadas todas las ejecuciones se integran los resultados obtenidos.

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;
```

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 MIPS 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 . grns

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.

5. PROBLEMAS

PROBLEMA I

Cierta empresa se ha planteado el compromiso de maximizar sus ganancias al producir el químico C a partir de los procesos II o III (dichos procesos son excluyentes), ambos utilizan el químico B como materia prima. El químico B se puede comprar al producir por medio del proceso I. En base a las especificaciones indicadas en la siguiente tabla, formule el problema de programación lineal mixto entero para decidir:

a. ¿Qué proceso se debe construir (II o III)?
b. ¿Cómo se obtiene el químico B?
c. ¿Qué cantidad deberá producirse del químico C para maximizar las ganancias?

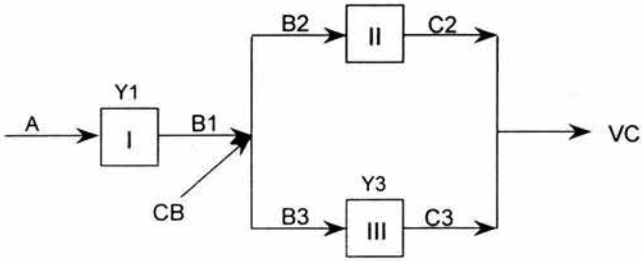
Considere que la demanda máxima de C es de 15 ton/hr con un precio de venta de $1800 / ton

Proceso	Costo fijo ($/hr)	Costo Variable ($/Ton Mat Prima)	Conversión
I	1000	250	90% $A \rightarrow B$
II	1500	400	82% $B \rightarrow C$
III	2000	550	95% $B \rightarrow C$

PRECIO DE VENTA PARA MATERIA PRIMA ($/TON); 500 (A) Y 950 (B)
LA CAPACIDAD MAXIMA DE ALMACENAMIENTO DE (A) ES DE 16 TON/h

Solución:

Representación esquemática del proceso.



Declaración de variables

B1: Representa la producción de químico B, en el proceso I. (Ton/hr)

B2: Representa el consumo de químico B, en el proceso II. (Ton/hr)

B3: Representa el consumo de químico B, en el proceso III. (Ton/hr)

C2: Representa la producción de químico C, en el proceso II. (Ton/hr)

C3: Representa la producción de químico C, en el proceso III. (Ton/hr)

A: Representa la cantidad de químico A comprada. (Ton/hr)

CB: Representa la cantidad de químico B comprada. (Ton/hr)

G : Ganancias. ($/hr)

VC: Representa la cantidad total de químico C producida. (Ton/hr)

Y1: Representa la selección del proceso I, cuando tenga el valor 1.

Y2: Representa la selección del proceso II, cuando tenga el valor de 1.

Y3: Representa la selección del proceso III, cuando tenga el valor de 1.

Formulación de la función objetivo

La ecuación que representa las ganancias generadas para la producción de químico C, es la siguiente:

Ganancias = -[Costo fijo proceso I + (Costo variable de A en el proceso I)(Cantidad de químico A comprado) + Costo fijo proceso II + (Costo variable de B en el proceso II)(Cantidad consumida de químico B en el proceso II) + Costo fijo proceso III + (Costo variable de B en el proceso III)(Cantidad consumida de químico B en el proceso III)] – (Precio de venta del químico A)(Cantidad de químico A comprado) - (Precio de venta del químico B)(Cantidad de químico B comprado) + (Precio de venta del químico C)(Cantidad total de químico C producido)

Finalmente la función objetivo queda de la siguiente forma:

$$G = -[1000*Y1 + 250*A + 1500*Y2 + 400*B2 + 2000*Y3 + 550*B3] - 500*A - 950*CB + 1800*VC$$

Agrupando terminos:

$$G = 1800*VC - [1000*Y1 + 1500*Y2 + 2000*Y3 + 500*A + 950*CB + 250*A + 400*B2 + 550*B3]$$

Restricciones de igualdad

- Balance de masa para el químico B

Sabemos que:

Proceso I	$A \rightarrow B$	
Proceso II	$B \rightarrow C$	
Proceso III		$B \rightarrow C$

En consecuencia el balance de masa es

$$B1 + CB = B2 + B3$$

- Balance de masa para el químico C

Sabemos que:

Proceso II	$B \rightarrow C$	
Proceso III		$B \rightarrow C$

En consecuencia el balance de masa es

$$C2 + C3 = VC$$

- Balance de masa para el proceso I

Sabemos que:

Proceso I $A \rightarrow B$

Como la conversión es del 90%, el balance de masa seria

$$B1 = 0.9*A$$

- Balance de masa para el proceso II

Sabemos que:

Proceso II $B \rightarrow C$

Como la conversión es del 82%, el balance de masa seria

$$C2 = 0.82*B2$$

- Balance de masa para el proceso III

Sabemos que:

Proceso III $B \rightarrow C$

Como la conversión es del 95%, el balance de masa seria

$$C3 = 0.95*B3$$

Restricciones de desigualdad

La selección del proceso esta sujeta a la siguiente restricción:

$$Y2 + Y3 \leq 1$$

Otra de las restricciones que se debe tomar encuenta es que la capacidad máxima de almacenamiento de químico A es de 16 Ton/hr, en consecuencia la restricción quedaria:

La cantidad de químico A comprada debe ser menor o igual a 16 Ton/hr, para el proceso I es decir

$$A \leq 16*YI$$

Se indica que la demanda máxima de químico C es de 15 Ton/hr. Por lo tanto

$$VC \leq 15 \text{ Ton/hr}$$

En base a lo anterior se formulan las siguientes restricciones para el consumo de químico B en los procesos II y III:

$$B2 \leq (15 / 0.82)*Y2$$

$$B3 \leq (15 / 0.95)*Y3$$

Codificación en GAMS

```
* inicio del archivo procsel1.gms

* asignacion de un titulo para el problema
$TITLE seleccion de procesos

$offsymxref offsymlist
*filename:    procsel1.gms
option optcr=0, limrow=0, limcol=0;
* declaracion de la naturaleza de las vartiables
BINARY VARIABLES y1, y2, y3;

POSITIVE VARIABLES A, B1,B2,B3,C2,C3,VC,CB;

* declaracion de la variable de la funcion objetivo
* VARIABLE DE LA GANACIA
free VARIABLE  G;
* declaracion  del nombre de las ecuaciones
EQUATIONS obj,
       dmax, cp1, exc, conv1, conv2,conv3,
       bm1, bm2, f1,f2,f3;

* declaracion de las expresiones de las ecuaciones
obj..   G =e= 1800*VC-(1000*y1+1500*y2+2000*y3 +
         500*A+950*CB +
```

```
        250*A+400*B2+550*B3);
dmax..  VC =l= 15;
cp1..   A =l= 16;
exc..   y2+y3 =e= 1;
conv1.. C2 =e= 0.82*B2;
conv2.. C3 =e= 0.95*B3;
conv3.. B1 =e= 0.90*A;
bm1..   B1 + CB =e= B2 + B3;
bm2..   C2 + C3 =e= VC;
f1..    A =l= 16*y1;
f2..    B2 =l= 15/0.82*y2;
f3..    B3 =l= 15/0.95*y3;
* asignacion de nombre del problema
MODEL procsel1 /ALL/ ;
* declaracion del tipo de modelo
SOLVE procsel1 USING MIP MAXIMIZING G;
* fin del archivo procsel1.gms;
```

En el punto 8.1 del Apéndice, se muestra el desplegado del archivo de salida, PROCSEL1.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Y1	.	1.000	1.000	680.000
---- VAR Y2	.	.	1.000	.
---- VAR Y3	.	1.000	1.000	.
---- VAR A	.	16.000	+INF	.
---- VAR B1	.	14.400	+INF	.
---- VAR B2	.	.	+INF	-27.930
---- VAR B3	.	15.789	+INF	.
---- VAR C2	.	.	+INF	.
---- VAR C3	.	15.000	+INF	.
---- VAR VC	.	15.000	+INF	.
---- VAR CB	.	1.389	+INF	.
---- VAR G	-INF	1995.789	+INF	.

Entonces:

a. ¿Qué proceso se construye (II o III)?

En base a los resultados obtenidos se nos indica que para poder maximizar las ganancias se debe llevar acabo el proceso III.

b. ¿Cómo se obtiene el químico B?

Se debe obtener el químico B, mediante el proceso I ya que es más económico producirlo que comprarlo.

c. ¿Qué cantidad deberá producirse del químico C?

Para poder incrementar las ganancias, se recomienda producir 15 ton/h del químico C.

PROBLEMA II

Bosqueje la superestructura para la separación de cuatro componentes A, B, C & D y formule el problema de MILP para minimizar el costo total para:

a. La mejor secuencia
b. La segunda mejor secuencia
c. La tercera mejor secuencia
d. La peor secuencia

Considere que la alimentación es de 1000 kgmol / h con una fracción de 0.15 de A, 0.30 de B, 0.35 de C & 0.20 de D y los datos económicos y de carga térmica se encuentran en la tabla siguiente:

		Costo Fijo	Costo Variable	Coeficiente de carga térmica
k	Separador	α_k (10^3 \$/año)	β_k (10^3 \$h / kgmol año)	K_k (10^6 kJ / kgmol)
1	A/BCD	145	0.42	0.028
2	AB/CD	52	0.12	0.042
3	ABC/D	76	0.25	0.054
6	A/BC	125	0.78	0.024
7	AB/C	44	0.11	0.039
4	B/CD	38	0.14	0.040
5	BC/D	66	0.21	0.047
10	A/B	112	0.39	0.022
9	B/C	37	0.08	0.036
8	C/D	58	0.19	0.44

Costos de servicios:

Agua de enfriamiento $C_C = 1.3*10^3$ \$/10^6 kJ / año
Vapor de calentamiento $C_H = 34*10^3$ \$/10^6 kJ / año

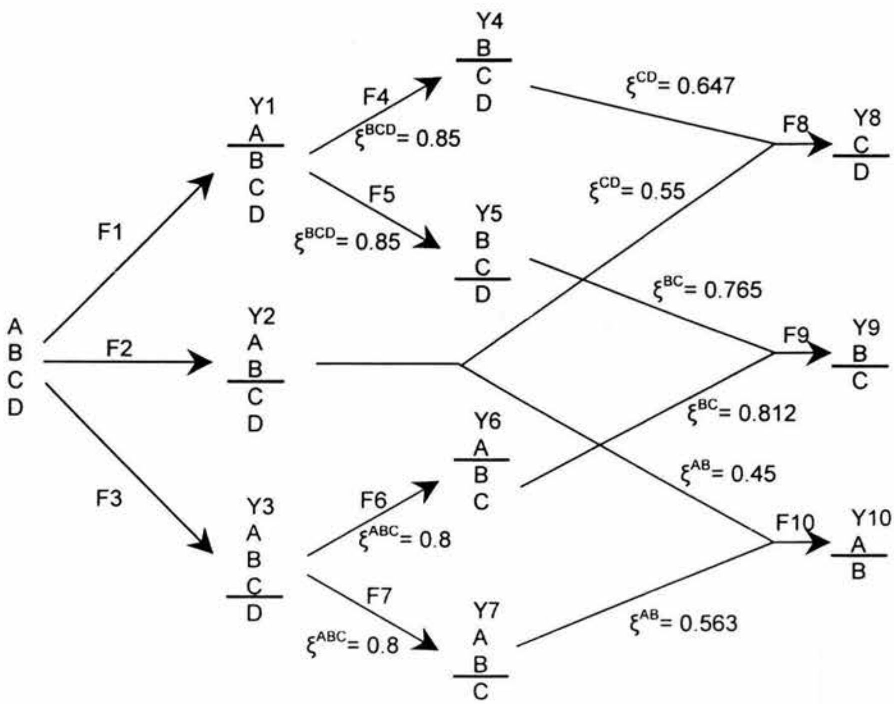
Fracciones de recuperación:

$\xi_1^{A} = 0.15$	$\xi_3^{D} = 0.2$	$\xi_6^{A} = 0.188$	$\xi_8^{D} = 0.364$
$\xi_1^{BCD} = 0.85$	$\xi_4^{B} = 0.353$	$\xi_6^{BC} = 0.812$	$\xi_9^{B} = 0.462$
$\xi_2^{AB} = 0.45$	$\xi_4^{CD} = 0.647$	$\xi_7^{AB} = 0.563$	$\xi_9^{C} = 0.538$
$\xi_2^{CD} = 0.55$	$\xi_5^{BC} = 0.765$	$\xi_7^{C} = 0.437$	$\xi_{10}^{A} = 0.333$
$\xi_3^{ABC} = 0.8$	$\xi_5^{D} = 0.235$	$\xi_8^{C} = 0.636$	$\xi_{10}^{B} = 0.667$

Solución:

Representación Esquemática del Proceso

Basándonos en los datos proporcionados se procede a dibujar la superestructura, la cual nos es de gran ayuda para diseñar las ecuaciones.



Declaración de variables

α_k = Coeficiente de costo fijo. (10^3 \$ / año)

β_k = Coeficiente de costo variable. (10^3 \$hr / kgmol año)

C = Costo total. (10^3 \$ / año)

C_C = Costo de servicio de agua de enfriamiento (10^3 \$/10^6 kJ / año)

C_H = Costo de servicio de Vapor de calentamiento (10^3 \$/10^6 kJ / año)

F_k = Flujo en la corriente k. (kgmol / hr)

K_k = Coeficiente de carga térmica en la corriente k. (10^6 kJ / kgmol)

k = No. de corriente. (1,2,...., 10)

Q_k = Calor. (10^6 kJ / hr)

y_k = Fracción molar

Formulación de la función objetivo

Como el objetivo consiste en seleccionar el mínimo costo total del proceso, entonces:

$$\min C = \sum_{k=1}^{10}(\alpha_k y_k + \beta_k F_k) + (C_H + C_C)\sum_{k=1}^{10} Q_k$$

Sustituyendo los costos de servicios, finalmente la función objetivo queda:

$$\min C = \sum_{k=1}^{10}(\alpha_k y_k + \beta_k F_k) + (34 + 1.3)\sum_{k=1}^{10} Q_k$$

Sabemos que el calor es igual al coeficiente de carga térmica por el flujo, por lo tanto la ecuación general queda:

$$Q_k = K_k * F_k$$

Restricciones de igualdad

En base al flujo total de 1000 kgmol/h y al nodo inicial de la red tenemos:

$$F1 + F2 + F3 = 1000$$

Basándonos en las fracciones de recuperación mostradas en la superestructura, los balances de masas para cada producto intermedio son:

1.- Intermedio (BCD)

$$F4 + F5 = 0.85*F1$$

igualando a cero la ecuación:

$$F4 + F5 - 0.85*F1 = 0$$

2.- Intermedio (ABC)

$$F6 + F7 = 0.8*F3$$

igualando a cero la ecuación:

$$F6 + F7 - 0.8*F3 = 0$$

3.- Intermedio (AB)

$$F10 = 0.45*F2 + 0.563*F7$$

igualando a cero la ecuación:

$$F10 - 0.45*F2 - 0.563*F7 = 0$$

4.- Intermedio (BC)

$$F9 = 0.765*F5 + 0.812*F6$$

igualando a cero la ecuación:

$$F9 - 0.765*F5 - 0.812*F6 = 0$$

5.- Intermedio (CD)

$$F8 = 0.55*F2 + 0.647*F4$$

igualando a cero la ecuación:

$$F8 - 0.55*F2 - 0.647*F4 = 0$$

Restricciones de desigualdad

Se deben tomar en cuenta las siguientes restricciones:

$k = 1, \dots 10$

$Y_k = [0, 1]$

$F_k \geq 0$

$F_k - (1000*y_k) \leq 0$

$y_2 + y_8 + y_{10} \leq 2$

$y_1 + y_4 + y_8 \leq 2$

a.- Mejor solución

Codificación en GAMS

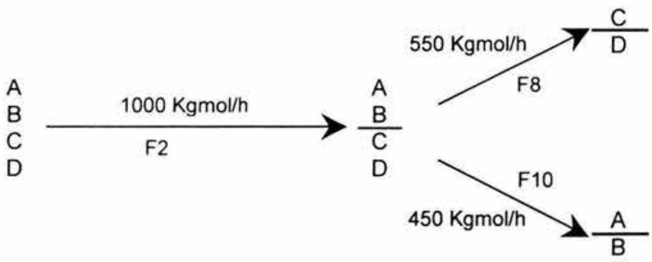
```
$ TITLE abcd1
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
        Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;

* variable de costos totales
free variable CT;

* nombre de las ecuaciones
```

```
equations obj,
        bm1,bm2,bm3,bm4,bm5,bm6,
        be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
        fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
        exc1,exc2,exc3;

*expresiones de las ecuaciones
obj.. CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112*Y10+

0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*F9+0.39*F10
        +35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
fl9.. F9=L=1000*Y9;
fl10.. F10=L=1000*Y10;
exc1.. Y1+Y2+Y3=L=1;
exc2.. Y4+Y5=L=1;
exc3.. Y6+Y7=L=1;

* nombre del problema
model abcd /all/;
solve abcd using mip minimizing CT;
* fin del archivo abcd1.gms
```

En el punto 8.2 del Apéndice, se muestra el desplegado del archivo de salida, ABCD1.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR F1	.	.	+INF	.
---- VAR F2	.	1000.000	+INF	.
---- VAR F3	.	.	+INF	.
---- VAR F4	.	.	+INF	1.268
---- VAR F5	.	.	+INF	0.210
---- VAR F6	.	.	+INF	0.780
---- VAR F7	.	.	+INF	0.767
---- VAR F8	.	550.000	+INF	.
---- VAR F9	.	.	+INF	0.080
---- VAR F10	.	450.000	+INF	.
---- VAR Q1	.	.	+INF	35.300
---- VAR Q2	.	42.000	+INF	.
---- VAR Q3	.	.	+INF	35.300
---- VAR Q4	.	.	+INF	35.300
---- VAR Q5	.	.	+INF	35.300
---- VAR Q6	.	.	+INF	35.300
---- VAR Q7	.	.	+INF	35.300
---- VAR Q8	.	24.200	+INF	.
---- VAR Q9	.	.	+INF	35.300
---- VAR Q10	.	9.900	+INF	.
---- VAR QT	.	76.100	+INF	.
---- VAR Y1	.	.	1.000	-2521.330
---- VAR Y2	.	1.000	1.000	52.000
---- VAR Y3	.	.	1.000	-2760.330
---- VAR Y4	.	.	1.000	38.000
---- VAR Y5	.	.	1.000	66.000
---- VAR Y6	.	.	1.000	125.000
---- VAR Y7	.	.	1.000	44.000
---- VAR Y8	.	1.000	1.000	58.000
---- VAR Y9	.	.	1.000	37.000
---- VAR Y10	.	1.000	1.000	112.000
---- VAR CT	-INF	3308.330	+INF	.

Los resultados obtenidos por GAMS nos indican que el proceso óptimo para la separación de los cuatro componentes A, B, C & D, debe seguir la secuencia siguiente:



Con esta secuencia se obtiene un costo total de $3308.330 x 10^3 / año.

b.- Segunda mejor solución

En base a la secuencia de separación obtenida en el inciso (a), podemos ingresar una restricción que involucre a y_2, y_8, y_{10}. Sabemos que el valor máximo de una fracción mol es de 1 entonces:

$$y_2 = y_8 = y_{10} = 1$$

Por lo tanto la restricción a ingresar es:

$$y_2 + y_8 + y_{10} \leq 2$$

<u>*Codificación en GAMS*</u>

```
$ TITLE abcd2
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
        Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;
```

```
* variable de costos totales
free variable CT;

* nombre de las ecuaciones
equations obj,
        bm1,bm2,bm3,bm4,bm5,bm6,
        be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
        fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
        exc1,exc2,exc3,
        exc4;

*expresiones de las ecuaciones
obj.. CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112*Y10+

0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*F9+0.39*F10
        +35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
fl9.. F9=L=1000*Y9;
fl10.. F10=L=1000*Y10;
exc1.. Y1+Y2+Y3=L=1;
exc2.. Y4+Y5=L=1;
exc3.. Y6+Y7=L=1;
exc4.. Y2+Y8+Y10=L=2;
```

```
* nombre del problema
model abcd /all/;
solve abcd using mip minimizing CT;

* fin del archivo abcd2.gms
```

En el punto 8.3 del Apéndice, se muestra el desplegado del archivo de salida, ABCD2.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR F1	.	1000.000	+INF	.
---- VAR F2	.	.	+INF	.
---- VAR F3	.	.	+INF	.
---- VAR F4	.	850.000	+INF	.
---- VAR F5	.	.	+INF	.
---- VAR F6	.	.	+INF	0.780
---- VAR F7	.	.	+INF	0.110
---- VAR F8	.	549.950	+INF	.
---- VAR F9	.	.	+INF	0.080
---- VAR F10	.	.	+INF	0.390
---- VAR Q1	.	28.000	+INF	.
---- VAR Q2	.	.	+INF	35.300
---- VAR Q3	.	.	+INF	35.300
---- VAR Q4	.	34.000	+INF	.
---- VAR Q5	.	.	+INF	35.300
---- VAR Q6	.	.	+INF	35.300
---- VAR Q7	.	.	+INF	35.300
---- VAR Q8	.	24.198	+INF	.
---- VAR Q9	.	.	+INF	35.300
---- VAR Q10	.	.	+INF	35.300
---- VAR QT	.	86.198	+INF	.
---- VAR Y1	.	1.000	1.000	145.000
---- VAR Y2	.	.	1.000	-2555.513
---- VAR Y3	.	.	1.000	-3360.273
---- VAR Y4	.	1.000	1.000	38.000
---- VAR Y5	.	.	1.000	-2403.850
---- VAR Y6	.	.	1.000	125.000
---- VAR Y7	.	.	1.000	44.000
---- VAR Y8	.	1.000	1.000	58.000
---- VAR Y9	.	.	1.000	37.000
---- VAR Y10	.	.	1.000	112.000
---- VAR CT	-INF	3927.273	+INF	.

Podemos ver que al ingresar la restricción, $y_2 + y_8 + y_{10} \leq 2$, los resultados obtenidos por GAMS nos indican que podemos tener un segundo proceso óptimo para la separación de los cuatro componentes A, B, C & D, el cual debe seguir la secuencia siguiente:

$$\begin{array}{c}A\\B\\C\\D\end{array} \xrightarrow[F1]{1000\ Kgmol/h} \begin{array}{c}A\\ \hline B\\C\\D\end{array} \xrightarrow[F4]{850\ Kgmol/h} \begin{array}{c}B\\ \hline C\\D\end{array} \xrightarrow[F8]{549.95\ Kgmol/h} \begin{array}{c}C\\ \hline D\end{array}$$

Con esta secuencia se obtiene un costo total de $\$3927.273 \times 10^3$ / año.

c.- Tercera mejor solución

En base a la secuencia de separación obtenida en el inciso (b), podemos ingresar otra restricción que involucre a y_1, y_4, y_8. Sabemos que el valor máximo de una fracción mol es de 1 entonces:

$$y_1 = y_4 = y_8 = 1$$

Por lo tanto la restricción a ingresar es:

$$y_1 + y_4 + y_8 \leq 2$$

<u>*Codificación en GAMS*</u>

```
$ TITLE abcd3
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;

* variable de costos totales
free variable CT;

* nombre de las ecuaciones
```

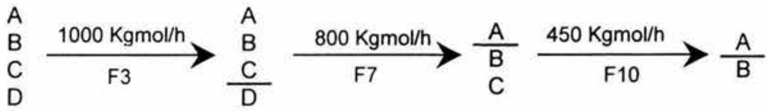
```
equations obj,
bm1,bm2,bm3,bm4,bm5,bm6,
be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
exc1,exc2,exc3,
exc4,exc5;

*expresiones de las ecuaciones
obj.. CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112*Y10+

0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*F9+0.39*F10
        +35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
fl9.. F9=L=1000*Y9;
fl10.. F10=L=1000*Y10;
exc1.. Y1+Y2+Y3=L=1;
exc2.. Y4+Y5=L=1;
exc3.. Y6+Y7=L=1;
exc4.. Y2+Y8+Y10=L=2;
exc5.. Y1+Y4+Y8=L=2;

* nombre del problema
model abcd /all/;
solve abcd using mip minimizing CT;
```

* fin del archivo abcd3.gms

En el punto 8.4 del Apéndice, se muestra el desplegado del archivo de salida, ABCD3.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR F1	.	.	+INF	.
---- VAR F2	.	.	+INF	.
---- VAR F3	.	1000.000	+INF	.
---- VAR F4	.	.	+INF	0.140
---- VAR F5	.	.	+INF	0.210
---- VAR F6	.	.	+INF	.
---- VAR F7	.	800.000	+INF	.
---- VAR F8	.	.	+INF	0.190
---- VAR F9	.	.	+INF	0.080
---- VAR F10	.	450.400	+INF	.
---- VAR Q1	.	.	+INF	35.300
---- VAR Q2	.	.	+INF	35.300
---- VAR Q3	.	54.000	+INF	.
---- VAR Q4	.	.	+INF	35.300
---- VAR Q5	.	.	+INF	35.300
---- VAR Q6	.	.	+INF	35.300
---- VAR Q7	.	31.200	+INF	.
---- VAR Q8	.	.	+INF	35.300
---- VAR Q9	.	.	+INF	35.300
---- VAR Q10	.	9.909	+INF	.
---- VAR QT	.	95.109	+INF	.
---- VAR Y1	.	.	1.000	-3305.997
---- VAR Y2	.	.	1.000	-3174.027
---- VAR Y3	.	1.000	1.000	76.000
---- VAR Y4	.	.	1.000	38.000
---- VAR Y5	.	.	1.000	66.000
---- VAR Y6	.	.	1.000	-1238.496
---- VAR Y7	.	1.000	1.000	44.000
---- VAR Y8	.	.	1.000	58.000
---- VAR Y9	.	.	1.000	37.000
---- VAR Y10	.	1.000	1.000	112.000
---- VAR CT	-INF	4102.997	+INF	.

Podemos ver que al ingresar la restricción, $y_1 + y_4 + y_8 \leq 2$, los resultados obtenidos por GAMS nos indican que puede existir un tercer proceso óptimo para la separación de los cuatro componentes A, B, C & D, el cual debe seguir la secuencia siguiente:



Con esta secuencia se obtiene un costo total de $4102.997 x 10^3 / año.

d.- La peor solución

Para obtener la peor solución en lugar de ingresar las restricciones siguientes:

$$y_2 + y_8 + y_{10} \le 2$$
$$y_1 + y_4 + y_8 \le 2$$

adicionamos

$$y_1+y_2+y_3+y_4+y_5+y_6+y_7+y_8+y_9+y_{10} \le 3$$

Codificación en GAMS

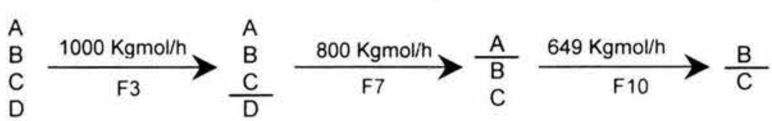
```
$ TITLE abcdp
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;

* variable de costos totales
free variable CT;

* nombre de las ecuaciones
equations obj,
bm1,bm2,bm3,bm4,bm5,bm6,
be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
exc1,exc2,exc3,exc4;
```

```
*expresiones de las ecuaciones
obj.. CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112*Y10+

0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*F9+0.39*F10
          +35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
fl9.. F9=L=1000*Y9;
fl10.. F10=L=1000*Y10;
exc1.. Y1+Y2+Y3=L=1;
exc2.. Y4+Y5=L=1;
exc3.. Y6+Y7=L=1;
exc4.. Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8+Y9+Y10=L=3;

* nombre del problema
model abcd /all/;
solve abcd using mip maximizing CT;

* fin del archivo abcdp.gms
```

En el punto 8.5 del Apéndice, se muestra el desplegado del archivo de salida, ABCDP.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR F1	.	.	+INF	-0.460
---- VAR F2	.	.	+INF	-1.249
---- VAR F3	.	1000.000	+INF	.
---- VAR F4	.	.	+INF	-0.223
---- VAR F5	.	.	+INF	.
---- VAR F6	.	800.000	+INF	.
---- VAR F7	.	.	+INF	-0.581
---- VAR F8	.	.	+INF	.
---- VAR F9	.	649.600	+INF	.
---- VAR F10	.	.	+INF	.
---- VAR Q1	.	.	+INF	.
---- VAR Q2	.	.	+INF	.
---- VAR Q3	.	54.000	+INF	.
---- VAR Q4	.	.	+INF	.
---- VAR Q5	.	.	+INF	.
---- VAR Q6	.	19.200	+INF	.
---- VAR Q7	.	.	+INF	.
---- VAR Q8	.	.	+INF	.
---- VAR Q9	.	23.386	+INF	.
---- VAR Q10	.	.	+INF	.
---- VAR QT	.	96.586	+INF	.
---- VAR Y1	.	.	1.000	145.000
---- VAR Y2	.	.	1.000	52.000
---- VAR Y3	.	1.000	1.000	76.000
---- VAR Y4	.	.	1.000	38.000
---- VAR Y5	.	.	1.000	66.000
---- VAR Y6	.	1.000	1.000	125.000
---- VAR Y7	.	.	1.000	44.000
---- VAR Y8	.	.	1.000	58.000
---- VAR Y9	.	1.000	1.000	37.000
---- VAR Y10	.	.	1.000	112.000
---- VAR CT	-INF	4573.440	+INF	.

Podemos ver que al ingresar la restricción, $y_1+y_2+y_3+y_4+y_5+y_6+y_7+y_8+y_9+y_{10} \leq 3$, el resultado obtenido por GAMS nos indica que en caso de realizar el proceso de separación de los cuatro componentes A, B, C & D, con la secuencia siguiente:



se tendría un costo total de $4573.440 x 10^3 / año, siendo un costo muy elevado.

En resumen:

Opción	*Costo Total (10^3 $/Año)*	*Flujo Masico Total (Kgmol/h)*
Primera	3,308.330	2000
Segunda	3,927.273	2399.95
Tercera	4,102.997	2250
Peor	4,573.440	2449

Al comparar el costo total con el flujo podemos corroborar que la mejor secuencia es la primera opción además de que posee el menor costo también tiene el mínimo flujo masico total.

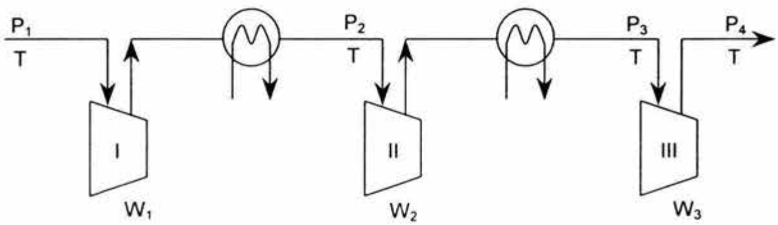
PROBLEMA III

Se desea calcular el trabajo mínimo del sistema de compresión de metano desde 15 Psia y 70°F hasta1000 Psia para un sistema de tres etapas, indique los valores de las presiones intermedias P_2 y P_3

Datos:
γ = 1.32   R= 10.731 Psia ft^3/ lbmol R

Solución:

Representación esquemática del proceso.



Declaración de variables

ρ = Densidad. (Lb/ft^3)

ρ_s = Densidad inicial. (Lb/ft^3)

ρ_x = Densidad intermedia o final. (Lb/ft^3)

γ = Es el cociente de los calores específicos de los gases a presión y volumen constante, 1.32 para el metano.

M = Peso molecular del metano, 16 Lb/Lbmol.

P = Presión del sistema. (Psia)

$P_s = P_1$ =Presión inicial. (Psia)

P_x = Presión intermedia o final. (Psia)

R = Constante Universal de los Gases, 10.731 Psia ft^3/ lbmol R.

T = Temperatura del sistema. (R)

W_x = Trabajo del sistema de compresión en la etapa x. (Psia ft^3 / lb)

W_T = Trabajo Total del sistema de tres etapas de compresión. (Psia ft^3 / lb)

x = 1,2,3

Formulación de la función objetivo

Con el objetivo de obtener el trabajo mínimo de un sistema de tres etapas de compresión de metano y los valores de las presiones intermedias, se partirá de un balance de energía y de un balance adiabático.

El trabajo generado por este sistema de compresión esta en función al balance energético:

$$-dW_x = \frac{dP_x}{d\rho_x}$$

Como se trata de un proceso adiabático, el balance para este proceso quedaría:

$$\frac{P_s}{\rho_s^{\gamma}} = \frac{P_x}{\rho_x^{\gamma}}$$

Además conocemos la ecuación del gas ideal:

$$\rho = \frac{M * P}{R * T}$$

Despejando ρ_x de la ecuación del balance adiabático tendríamos:

$$\rho_x^{\gamma} = \left(\frac{P_x}{P_s}\right) * \rho_s^{\gamma}$$

$$\rho_x = \sqrt[\gamma]{\left(\frac{P_x}{P_s}\right) * \rho_s^{\gamma}}$$

Simplificando, se obtiene:

$$\rho_x = \left(\frac{P_x}{P_s}\right)^{\frac{1}{\gamma}} * \rho_s$$

sustituyendo la expresión anterior, en la ecuación del balance energético:

$$-dW_x = \frac{dP_x}{\left(\frac{P_x}{P_s}\right)^{\frac{1}{\gamma}} * \rho_s}$$

Simplificando, se obtiene:

$$-dW_x = \left[\left(\frac{P_x}{P_s}\right)^{-\left(\frac{1}{\gamma}\right)} * \frac{1}{\rho_s}\right] dP_x$$

Al integrar esta ecuación se obtiene la siguiente función:

$$-\int dW_x = \int\left[\left(\frac{P_x}{P_s}\right)^{-\left(\frac{1}{\gamma}\right)} * \frac{1}{\rho_s}\right] dP_x$$

$$-W_x = \left[\frac{\left(\frac{P_x}{P_s}\right)^{\left(-\frac{1}{\gamma}+1\right)}}{\left(-\frac{1}{\gamma}+1\right)}\right]\left(\frac{P_x}{\rho_s}\right) - \left[\left(\frac{1}{\left(-\frac{1}{\gamma}+1\right)}\right)\left(\frac{1}{\rho_s}\right)\right] * P_x$$

$$-W_x = \left[\frac{\left(\frac{P_x}{P_s}\right)^{\left(\frac{1-\gamma}{-\gamma}\right)}}{\left(\frac{1-\gamma}{-\gamma}\right)}\right]\left(\frac{P_x}{\rho_s}\right) - \left(\frac{1}{\left(\frac{1-\gamma}{-\gamma}\right) * \rho_s}\right) * P_x$$

$$-W_x = \left[\left(\frac{-\gamma}{1-\gamma}\right)\left(\frac{P_x}{P_s}\right)^{\frac{1-\gamma}{-\gamma}} * \left(\frac{P_x}{\rho_s}\right)\right] - \left[\left(\frac{-\gamma}{1-\gamma}\right)\left(\frac{1}{\rho_s}\right)\right] * P_x$$

Multiplicamos por -1, obtenemos:

$$W_x = \left[\left(\frac{\gamma}{\gamma-1}\right)\left(\frac{P_x}{P_s}\right)^{\frac{\gamma-1}{\gamma}} * \left(\frac{P_x}{\rho_s}\right)\right] - \left[\left(\frac{\gamma}{\gamma-1}\right)\left(\frac{P_x}{\rho_s}\right)\right]$$

De la ecuación del gas ideal despejamos $\frac{P}{\rho}$ y lo sustituimos en la ecuación anterior:

$$\frac{P}{\rho} = \frac{R * T}{M}$$

Sustituyendo este termino en la ecuación de W_x:

$$W_x = \left[\left(\frac{\gamma}{\gamma - 1} \right) \left(\frac{P_x}{P_s} \right)^{\frac{\gamma - 1}{\gamma}} * \left(\frac{RT}{M} \right) \right] - \left(\frac{RT\gamma}{M(\gamma - 1)} \right)$$

Considerando que el término $\frac{RT\gamma}{M(\gamma - 1)}$, es constante, la ecuación anterior quedaría:

$$W_x = \frac{RT\gamma}{M(\gamma - 1)} \left[\left(\frac{P_x}{P_s} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right]$$

Como se trata de un sistema de tres etapas de compresión de metano, el trabajo total será:

$$W_T = \left[\frac{RT\gamma}{M(\gamma - 1)} \left(\left(\frac{P_2}{P_1} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right) \right] + \left[\frac{RT\gamma}{M(\gamma - 1)} \left(\left(\frac{P_3}{P_2} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right) \right] + \left[\frac{RT\gamma}{M(\gamma - 1)} \left(\left(\frac{P_4}{P_3} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right) \right]$$

Finalmente la función objetivo es:

$$W_T = \frac{RT\gamma}{M(\gamma-1)}\left[\left(\frac{P_2}{P_1}\right)^{\frac{\gamma-1}{\gamma}} + \left(\frac{P_3}{P_2}\right)^{\frac{\gamma-1}{\gamma}} + \left(\frac{P_4}{P_3}\right)^{\frac{\gamma-1}{\gamma}} - 3\right]$$

Sustituyendo los valores conocidos, la ecuación de W_T quedaría:

$$W_T = \frac{(10.731)(70+460)(1.32)}{16(0.32)}\left[\left(\frac{P_2}{15}\right)^{0.24} + \left(\frac{P_3}{P_2}\right)^{0.24} + \left(\frac{1000}{P_3}\right)^{0.24} - 3\right]$$

Designaremos que:

$$rc_1 = \frac{P_2}{50} \qquad (5.3\text{-}1)$$

$$rc_2 = \frac{P_3}{P_2} \qquad (5.3\text{-}2)$$

$$rc_3 = \frac{1000}{P_3} \qquad (5.3\text{-}3)$$

Por lo tanto la ecuación de W_T quedaría:

$$W_T = \frac{(10.731)(70+460)(1.32)}{16(0.32)}\left[(rc_1)^{0.24} + (rc_2)^{0.24} + (rc_3)^{0.24} - 3\right]$$

Para calcular las presiones intermedias, despejamos P_2 de la ecuación 5.3-1 y P_3 de la ecuación 5.3-2, las cuales quedan:

$$P_2 = 15 * rc_1$$

$$P_3 = P_2 * rc_2$$

Y por último despejamos la presión final de la ecuación 5.3-3, la cual queda:

$$1000 = P_3 * rc_3$$

Restricciones

La presión 2 tendrá como limite inferior 45 y la presión 3 tendrá como limite inferior 180.

Codificación en GAMS

```
* inicio del archivo trabajo.gms
* modelo para la minimizacion del trabajo
* para un sistema de tres etapas de compresion
$ TITLE compresion
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;

* declaracion de las variables del problema
positive variables P2,P3, rc1, rc2, rc3;
* declaracion de la variable de la funcion objetivo
free variable Wt;
* delcaracion del conjunto de ecuaciones del modelo
```

```
equations obj, erc1, erc2, erc3;
* declaracion de las expresiones de las ecuaciones

*expresion de la funcion objetivo para el trabajo minimo
erc1.. 15*rc1=e=P2;
erc2.. P2*rc2=e=P3;
erc3.. P3*rc3=e=1000;
obj.. Wt =e= 10.731*(70+460)*1.32/(16*0.32)*
        (rc1**0.2424 + rc2**0.2424 + rc3**0.2424 - 3);
P2.lo=45;
P3.lo=150;

* asignacion del nombre al problema
model compresion /all/;
* declaracion de solucion del problema
solve compresion using NLP minimizing Wt;
*fin del archivo trabajo.gms
```

En el punto 8.6 del Apéndice, se muestra el desplegado del archivo de salida, TRABAJO.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
VAR P2	60.000	60.822	+INF	EPS
VAR P3	180.000	246.621	+INF	EPS
VAR RC1	.	4.055	+INF	.
VAR RC2	.	4.055	+INF	.
VAR RC3	.	4.055	+INF	.
VAR WT	-INF	1777.211	+INF	.

Al checar los resultados obtenidos, el programa nos reporta que el trabajo mínimo para un sistema de tres etapas de compresión de metano será de 1777.211 [Psia ft^3 / lb], el cual se llevara acabo con presiones intermedias de 60.822 Psia para P_2 y de 246.621 Psia para P_3.

PROBLEMA IV

El objetivo de este problema consiste en calcular el valor óptimo de la ecuación siguiente:

$$\min F = x_1^2 + x_2^2 + x_3^2$$

la cual se encuentra sujeta a las siguientes restricciones:

$$
\begin{aligned}
&g1: -2x_1 - x_2 \geq -5 \\
&g2: -x_1 - x_3 \geq -2 \\
&g3: -x_1 - 2x_2 - x_3 \geq -10 \\
&h1: 2x_1 - 2x_2 + x_3 = -2 \\
&h2: 10x_1 + 8x_2 - 14x_3 = 26 \\
&h3: -4x_1 + 5x_2 - 6x_3 = 6
\end{aligned}
\qquad \text{con } x_0=[1\ 2\ 0]^T
$$

Solución:

En este caso, como ya se nos indica la función objetivo, se procede a realizar la codificación en GAMS, para obtener el valor óptimo de una manera rápida.

Codificación en GAMS

```
$ TITLE minf
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;

* declaracion de las variables del problema
positive variables x1, x2, x3;
* declaracion de la variable de la funcion objetivo
free variable f;
* delcaracion del conjunto de ecuaciones del modelo
equations obj, g1, g2, g3, h1, h2, h3;
* declaracion de las expresiones de las ecuaciones
obj.. f =e= x1*x1 + x2*x2 + x3*x3;
g1.. -2*x1 - x2 =g= -5;
g2 .. -x1 - x3 =g= -2;
g3.. -x1 - 2*x2 - x3 =g= -10;
h1.. 2*x1 - 2*x2 + x3 =e= -2;
h2.. 10*x1 + 8*x2 - 14*x3 =e= 26;
h3.. -4*x1 + 5*x2 - 6*x3 =e= 6;
* declaracion de los limites de las variables
x1.l = 1.;
x2.l = 2.;
```

```
x3.l = 0.;
* asignacion del nombre al problema
model minf /all/;
* declaracion de solucion del problema
solve minf using NLP minimizing f;
*fin de la codificacion del archivo minf.gms
```

En el punto 8.7 del Apéndice, se muestra el desplegado del archivo de salida, MINF.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
VAR X1	.	1.000	+INF	.
VAR X2	.	2.000	+INF	.
VAR X3	.	.	+INF	5.333
VAR F	-INF	5.000	+INF	.

Por lo tanto los valores óptimos que cumplen con las restricciones asignadas son:

$x_1 = 1$
$x_2 = 2$
$x_3 = 0$

Con estos valores obtenemos una "f" igual a 5.

PROBLEMA V

Se desea calcular el costo mínimo para el consumo de servicios considerando un acercamiento mínimo de 20 °C para el conjunto de corrientes calientes y frías siguiente:

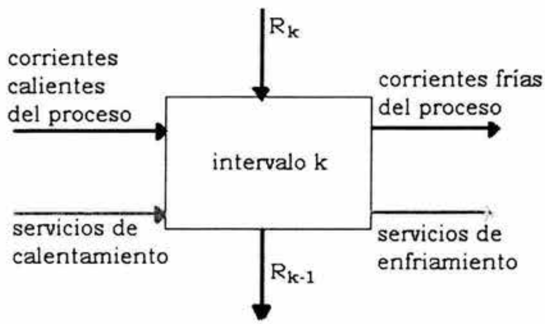
CORRIENTES	FCp [MW/°C]	T_{ENT} [°C]	T_{SAL} [°C]
H1	1.0	400	120
H2	2.0	340	120
C1	1.5	160	400
C2	1.3	100	250

Los costos específicos para el vapor de calentamiento y el agua de enfriamiento son de $80/kw-año y $20/kw-año, respectivamente
Reporte

a. La temperatura(s) del punto de pliegue
b. El consumo de servicios
c. El costo total

Solución

<u>Representación esquemática del proceso.</u>



<u>Declaración de variables</u>

C_m = Costo Especifico para el Servicio de Calentamiento. ($/MW*año)

C_n = Costo Especifico para el Servicio de Enfriamiento. ($/MW*año)

i = Es el número de la corriente del vapor de calentamiento. (1 ó 2)

j = Es el número de la corriente del agua de enfriamiento. (1 ó 2)

k = Es el número de intervalo. (1, 2, ..., K intervalos)

m = Es el número del servicio de calentamiento. (1, 2, ... # servicios de calentamiento en el intervalo k)

n = Es el número del servicio de enfriamiento. (1, 2, ... # servicios de enfriamiento en el intervalo k)

Q_{Hik} = Consumo de la corriente i del vapor de calentamiento en el intervalo k. (MW)

Q_{Cjk} = Consumo de la corriente j del agua de enfriamiento en el intervalo k. (MW)

Q_m^s = Corriente m para Servicios de Calentamiento. (MW)

Q_n^w = Corriente n para Servicios de enfriamiento. (MW)

Q_s = Consumo total del vapor de calentamiento. (MW)

Q_W = Consumo total de agua de enfriamiento. (MW)

R_k = Corriente a la salida del intervalo k. (MW)

R_1 = Consumo a la salida del intervalo 1. (MW)

R_2 = Consumo a la salida del intervalo 2. (MW)

R_3 = Consumo a la salida del intervalo 3. (MW)

R_4 = Consumo a la salida del intervalo 4. (MW)

Z = Costo Total. ($/año)

Formulación de la función objetivo

El método que se utilizara para resolver este tipo de problemas es el de síntesis secuencial, el cual nos permite dar solución a un conjunto de subproblemas como son:

a) Costo mínimo de servicios
b) Costo mínimo de servicios con restricciones en las combinaciones de las corrientes
c) Predicción de combinaciones para minimizar el número de unidades

Para una red de intercambio de calor el modelo condensado del transbordo (Papoulias & Grossmann -1983-) es el siguiente:

$$MinZ = \sum_{m=1}^{\substack{\#\text{ servicios} \\ \text{de calentamie nto}}} C_m Q_m{}^s + \sum_{n=1}^{\substack{\#\text{ servicios} \\ \text{de enfriamien to}}} C_n Q_n{}^w$$

sujeta a los balances de energía.

Tomando como referencia esta ecuación para elaborar la función objetivo y sustituyendo en ella los costos específicos del vapor de calentamiento y del agua de enfriamiento, el modelo de programación matemática dado por el problema de transbordo que nos permite calcular el costo mínimo para el consumo de servicios, quedaría:

$$MinZ = 80\ 000 * Q_s + 20\ 000 * Q_w$$

Restricciones de igualdad

Recordemos que el costo mínimo para el consumo de servicios esta sujeta al balance de energía para cada intervalo:

$$R_k - R_{k-1} - \sum_{m=1}^{\substack{\#\ \text{servicios} \\ \text{de calentamiento} \\ \text{en el intervalo } k}} Q_m{}^{s} + \sum_{n=1}^{\substack{\#\ \text{servicios} \\ \text{de enfriamiento} \\ \text{en el intervalo } k}} Q_n{}^{w} = \sum_{i=1}^{\substack{\#\ \text{corrientes} \\ \text{calientes} \\ \text{en el intervalo } k}} Q_{Hik} - \sum_{j=1}^{\substack{\#\ \text{corrientes} \\ \text{frias} \\ \text{en el intervalo } k}} Q_{Cjk}$$

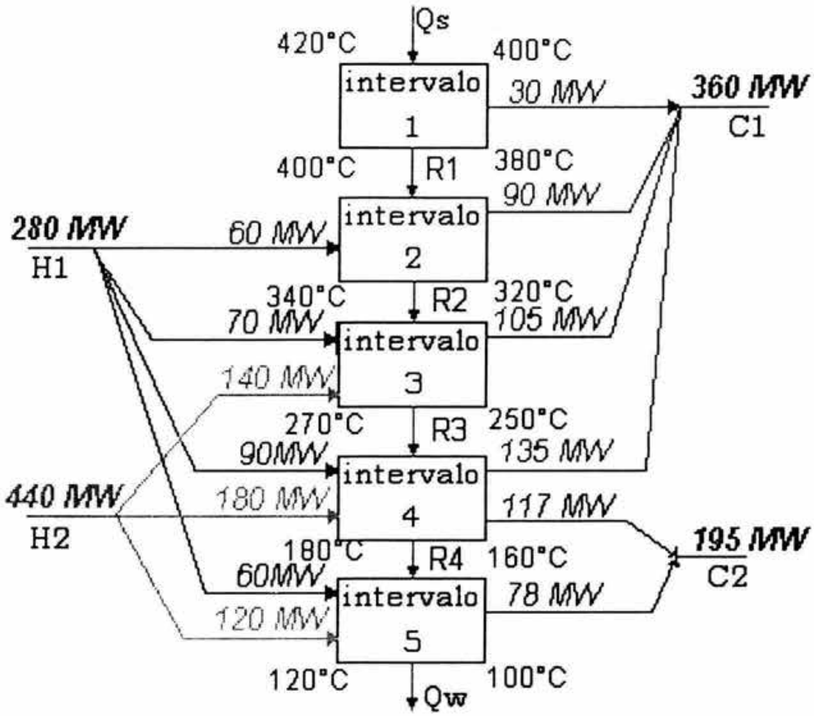
$$k = 1,2,\ldots$$

El balance de energía y el número de intervalos se realizo mediante el programa de matlab, la codificación se encuentra en el punto 8.8 del Apéndice.

El resultado del cálculo del balance de energía para cada uno de los intervalos, reportado por el programa matlab, es el siguiente:

INTERVALO k	T [°C]	QH1 [MW]	QH2 [MW]	QC1 [MW]	QC2 [MW]	Q [MW]
1	20	0	0	30	0	-30
2	60	60	0	90	0	-30
3	70	70	140	105	0	105
4	90	90	180	135	117	18
5	60	60	120	0	78	102

En base al balance de energía se procede a realizar la representación en una cascada de calor:



Balances de energía:

<u>Intervalo 1</u>

$$R_1 - Q_s = -Q_{C11}$$

Sustituyendo los valores conocidos, tenemos:

$$R_1 - Q_s = -30$$

<u>Intervalo 2</u>

$$R_2 - R_1 = Q_{H12} - Q_{C12}$$

Sustituyendo los valores conocidos, tenemos:

$$R_2 - R_1 = 60 - 90$$

entonces

$$R_2 - R_1 = -30$$

Intervalo 3

$$R_3 - R_2 = (Q_{H13} + Q_{H23}) - Q_{C13}$$

Sustituyendo los valores conocidos, tenemos:

$$R_3 - R_2 = (70 + 140) - 105$$

entonces

$$R_3 - R_2 = 105$$

Intervalo 4

$$R_4 - R_3 = (Q_{H14} + Q_{H24}) - (Q_{C14} + Q_{C24})$$

Sustituyendo los valores conocidos, tenemos:

$$R_4 - R_3 = (90 + 180) - (135 + 117)$$

entonces

$$R_4 - R_3 = 18$$

Intervalo 5

$$-R_4 + Q_W = (Q_{H15} + Q_{H25}) - Q_{C25}$$

Sustituyendo los valores conocidos, tenemos:

$$Q_W - R_4 = (60 + 120) - 78$$

entonces

$$Q_W - R_4 = 102$$

Restricción de desigualdad

$Q_S, Q_W, R_1, R_2, R_3 \& R_4 \geq 0$

Codificación en GAMS

```
* inicio del archivo utility.gms
* modelo para la minimizacion del trabajo
* para un sistema de tres etapas de compresion
* calculo de costo minimo de servicios
$ TITLE utility
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;

* declaracion de las variables del problema
positive variables Qs, Qw, R1, R2, R3, R4;
* declaracion de la variable de la funcion objetivo [$/a&o]
free variable Z;
* delcaracion del conjunto de ecuaciones del modelo
equations obj,
        beint1, beint2, beint3, beint4, beint5;

*expresion de la funcion objetivo para el costo minimo
obj..    Z =e= 80000 * Qs + 20000 * Qw;
* declaracion de las expresiones de balances
* de enrgia en los intervalos [MW]
```

```
beint1..  R1 - Qs =e= -30;
beint2..  R2 - R1 =e= -30;
beint3..  R3 - R2 =e= 105;
beint4..  R4 - R3 =e= 18;
beint5..  Qw - R4 =e= 102;

* asignacion del nombre al problema
model utility /all/;
* declaracion de solucion del problema
solve utility using LP minimizing Z;
* fin del archivo utility.gms
```

En el punto 8.9 del Apéndice, se muestra el desplegado del archivo de salida, UTILITY.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
VAR Qs	.	60.000	+INF	.
VAR Qw	.	225.000	+INF	.
VAR R1	.	30.000	+INF	.
VAR R2	.	.	+INF	1.0000E+5
VAR R3	.	105.000	+INF	.
VAR R4	.	123.000	+INF	.
VAR Z	-INF	9.3000E+6	+INF	.

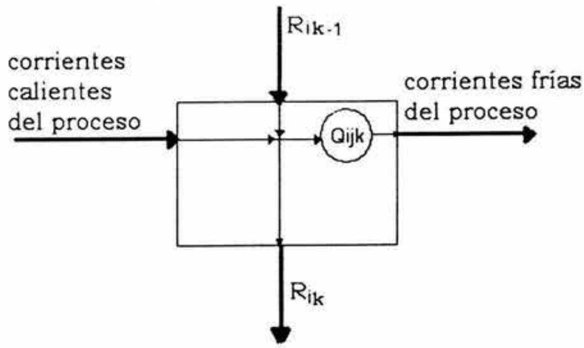
Podemos concluir que se puede obtener un costo total minimo de $ 9.3 millones /año con un consumo de servicios de calentamiento de 60 MW y de enfriamiento de 225 MW. Como R_2 es igual a cero el punto de pliegue se encuentra en 340°C - 320°C.

PROBLEMA VI

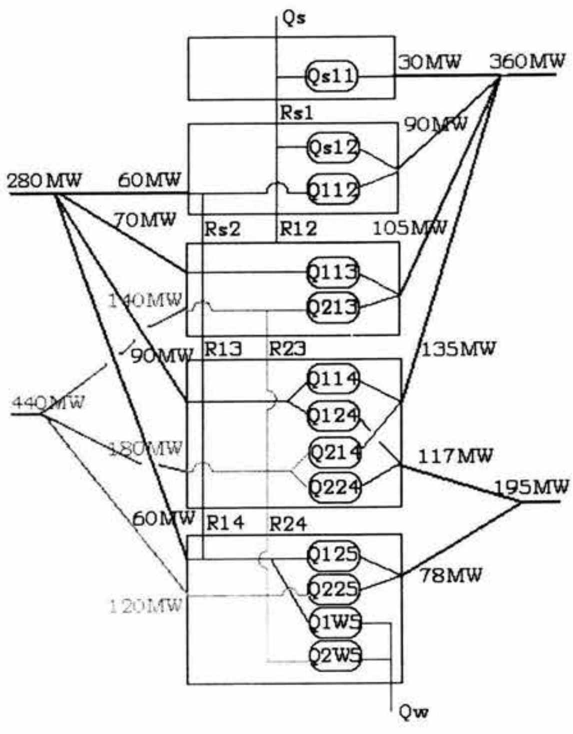
- Con los datos del problema V, calcule el consumo mínimo de servicios aplicando el método de transbordo expandido.
- Trace la superestructura del problema V, para la formulación del problema de transbordo expandido para las combinaciones de las corrientes indicadas en la tabla del balance de energía.

Solución

Representación esquemática del proceso.



La superestructura es la siguiente:



Declaración de variables

C_m = Costo Especifico para el Servicio de Calentamiento. ($/MW*año)

C_n = Costo Especifico para el Servicio de Enfriamiento. ($/MW*año)

i = Es el número de la corriente caliente. (1, 2, ... # corrientes calientes en el intervalo k)

j = Es el número de la corriente fría. (1, 2, ..., # corrientes frías en el intervalo k)

k = Es el número de intervalo. (1, 2, ..., K intervalos)

m = Es el número del servicio de calentamiento. (1, 2, ..., # servicios de calentamiento en el intervalo k)

n = Es el número del servicio de enfriamiento. (1, 2, ..., # servicios de enfriamiento en el intervalo k)

Q_{Hik} = Consumo de la corriente i del vapor de calentamiento en el intervalo k. (MW)

Q_{Cjk} = Consumo de la corriente j del agua de enfriamiento en el intervalo k. (MW)

Q_m^s = Corriente m para Servicios de Calentamiento. (MW)

Q_n^w = Corriente n para Servicios de enfriamiento. (MW)

Z = Costo Total. ($/año)

Formulación de la función objetivo

Para una red de intercambio de calor el modelo expandido del transbordo (Papoulias & Grossmann -1983-) es el siguiente:

$$\mathrm{MinZ} = \sum_{m=1}^{\substack{\#\text{ servicios}\\ \text{de calentamie nto}}} C_m Q_m{}^s + \sum_{n=1}^{\substack{\#\text{ servicios}\\ \text{de enfriamien to}}} C_n Q_n{}^w$$

sujeta a las restricciones de balance de energía, los cuales se plantean mas adelante.

Tomando como referencia esta ecuación para elaborar la función objetivo y sustituyendo en ella los costos especificos del vapor de calentamiento y del agua de enfriamiento, el modelo de programación matemática dado por el problema de transbordo que nos permite calcular el costo minimo para el consumo de servicios, quedaría:

$$MinZ = 80\,000 * Q_s + 20\,000 * Q_w$$

Restricciones de igualdad

Para el modelo expandido del transbordo las restricciones de balance de energía son:

Para las corrientes calientes

$$R_{ik} - R_{ik-1} + \sum_{j=1}^{\substack{\#\,\text{corrientes} \\ \text{frias} \\ \text{en el intervalo } k}} Q_{ijk} + \sum_{n=1}^{\substack{\#\,\text{servicios} \\ \text{de enfriamiento} \\ \text{en el intervalo } k}} Q_{ink} = Q_{Hik}$$

i = 1, 2, … # corrientes calientes en el intervalo k.

Para los servicios de calentamiento

$$R_{mk} - R_{mk-1} + \sum_{j=1}^{\substack{\#\text{corrientes} \\ \text{frias} \\ \text{en el intervalo } k}} Q_{mjk} - Q_m{}^S = 0$$

m = 1, 2, …, # servicios de calentamiento en el intervalo k

Para las corrientes frías

$$\sum_{i=1}^{\substack{\#\,\text{corrientes} \\ \text{calientes} \\ \text{en el intervalo } k}} Q_{ijk} + \sum_{j=1}^{\substack{\#\,\text{corrientes} \\ \text{frias} \\ \text{en el intervalo } k}} Q_{mjk} = Q_{Cjk}$$

j = 1, 2, ..., # corrientes frías en el intervalo k

Para los servicios de enfriamiento

$$\sum_{i=1}^{\substack{\text{\# corrientes}\\ \text{calientes}\\ \text{en el intervalo k}}} Q_{ink} - Q_n^{w} = 0$$

n = 1, 2, ..., #servicios de enfriamiento en el intervalo k

todo esto para k=1,2,...K intervalos

y las restricciones inherentes

$$Q_m^{s},\quad Q_n^{w},\ R_{ik}, R_{mk},\ Q_{ijk}, Q_{ink}, Q_{mjk}\ \geq 0$$
$$R_{i0} = 0,\quad \&\quad R_{ik} = 0$$

Entonces los balances de energía para cada intervalo quedan de la siguiente forma:

Intervalo 1

bes1: $R_{s1} + Q_{s11} = Q_s$

bec11 : $Q_{s11} = Q_{c11}$

Sustituyendo el valor de Q_{c11} en bec11:

bec11 : $Q_{s11} = 30$

Intervalo 2

bes2 : $R_{s2} - R_{s1} + Q_{s12} = 0$
beh12 : $R_{12} + Q_{112} = Q_{H12}$

Sustituyendo el valor de Q_{H12} en beh12:

beh12 : $R_{12} + Q_{112} = 60$

bec12: $Q_{s12} + Q_{112} = Q_{C12}$

Sustituyendo el valor de Q_{C12} en bec12:

bec12: $Q_{s12} + Q_{112} = 90$

punto de pliegue: $R_{12} = 0$ & $R_{s2} = 0$

Intervalo 3

beh13 : $R_{13} - R_{12} + Q_{113} = Q_{H13}$

Sustituyendo el valor de Q_{H13}, en beh13:

beh13 : $R_{13} - R_{12} + Q_{113} = 70$

beh23: $R_{23} + Q_{213} = Q_{H23}$

Sustituyendo el valor de Q_{H23} en beh23:

beh23: $R_{23} + Q_{213} = 140$

bec13: $Q_{113} + Q_{213} = Q_{C13}$

Sustituyendo el valor de Q_{C13}, en bec13:

bec13: $Q_{113} + Q_{213} = 105$

Intervalo 4

beh14 : $R_{14} - R_{13} + Q_{114} + Q_{124} = Q_{H14}$

Sustituyendo el valor de Q_{H14} en beh14:

beh14 : $R_{14} - R_{13} + Q_{114} + Q_{124} = 90$

beh24 : $R_{24} - R_{23} + Q_{214} + Q_{224} = Q_{H24}$

Sustituyendo el valor de Q_{H24} en beh24:

beh24 : $R_{24} - R_{23} + Q_{214} + Q_{224} = 180$

bec14 : $Q_{114} + Q_{214} = Q_{C14}$

Sustituyendo el valor de Q_{C14} en bec14:

bec14 : $Q_{114} + Q_{214} = 135$

bec24 : $Q_{124} + Q_{224} = Q_{C24}$

Sustituyendo el valor de Q_{C24} en bec24:

bec24 : $Q_{124} + Q_{224} = 117$

Intervalo 5

beh15 : $-R_{14} + Q_{125} + Q_{1w5} = Q_{H15}$

Sustituyendo el valor de Q_{H15} en beh15:

beh15 : $-R_{14} + Q_{125} + Q_{1w5} = 60$

beh25: $-R_{24} + Q_{225} + Q_{2w5} = Q_{H25}$

Sustituyendo el valor de Q_{H25} en beh25:

beh25: $-R_{24} + Q_{225} + Q_{2w5} = 120$

bec25: $Q_{125} + Q225 = Q_{C25}$

Sustituyendo el valor de Q_{C25} en bec25:

bec25: $Q_{125} + Q225 = 78$

bew5 : $Q_{1w5} + Q_{2w5} = Q_w$

Restricción de desigualdad

Considerando que Q_m^s , Q_n^w , R_{ik}, R_{sk}, Q_{ijk}, Q_{sjk}, Q_{iwk} ≥ 0, entonces

Q_s, Q_w, Q_{s11}, Q_{s12}, R_{s1}, R_{s2}, Q_{112}, Q_{113}, Q_{114}, Q_{124}, Q_{125}, R_{12}, R_{13}, R_{14}, Q_{213}, Q_{214}, Q_{224}, Q_{225}, R_{23}, R_{24}, Q_{1w5}, $Q_{2w5} \geq 0$

Codificación en GAMS

```
* inicio del archivo tutility.gms
* calculo del consumo minimo de servicios aplicando
* el metodo de transbordo expandido
$ TITLE servicios
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion del conjunto de variables del problema
positive variables Qs, Qw,
                Qs11, Qs12, Rs1, Rs2,
                Q112, Q113, Q114, Q124, Q125, R12, R13, R14,
                Q213, Q214, Q224, Q225, R23, R24,
                Q1w5, Q2w5;
* declaracion de la variables de la funcion
free variable z;
* declaracion de las ecuaciones del modelo
equations obj,
                bes1, bec11,
                bes2, beh12, bec12,
                beh13, beh23, bec13,
                beh14, beh24, bec14, bec24,
                beh15, beh25, bec25, bew5;
* expresiones del modelo
obj.. z =e= 80000 * Qs + 20000 * Qw;
* balance de energia para el intervalo 1
bes1.. Rs1 + Qs11 =e= Qs;
bec11.. Qs11 =e= 30;
* balance de energia para el intervalo 2
bes2.. Rs2 - Rs1 + Qs12 =e= 0;
beh12.. R12 + Q112 =e= 60;
bec12.. Qs12 + Q112 =e= 90;
* punto de pliegue
R12.l = 0;
Rs2.l = 0;
* balance de energia para el intervalo 3
beh13.. R13 - R12 + Q113 =e= 70;
beh23.. R23 + Q213 =e= 140;
bec13.. Q113 + Q213 =e= 105;
* balance de energia para el intervalo 4
```

```
beh14.. R14 - R13 + Q114 + Q124 =e= 90;
beh24.. R24 - R23 +Q214 + Q224 =e= 180;
bec14.. Q114 + Q214 =e= 135;
bec24.. Q124 + Q224 =e= 117;
* balance de energia para el intervalo 5
beh15.. -R14 + Q125 + Q1w5 =e= 60;
beh25.. -R24 + Q225 + Q2w5 =e= 120;
bec25.. Q125 + Q225 =e= 78;
bew5.. Q1w5 + Q2w5 =e= Qw;

* asignacion del nombre al problema
model tutility /all/;
* declaracion de solucion del problema
solve tutility using lp minimizing z;
*fin del archivo tutility.gms
```

En el punto 8.10 del Apéndice, se muestra el desplegado del archivo de salida, TUTILITY.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR QS	.	60.000	+INF	.
---- VAR QW	.	225.000	+INF	.
---- VAR QS11	.	30.000	+INF	.
---- VAR QS12	.	30.000	+INF	.
---- VAR RS1	.	30.000	+INF	.
---- VAR RS2	.	.	+INF	80000.000
---- VAR Q112	.	60.000	+INF	.
---- VAR Q113	.	.	+INF	EPS
---- VAR Q114	.	.	+INF	EPS
---- VAR Q124	.	37.000	+INF	.
---- VAR Q125	.	.	+INF	EPS
---- VAR R12	.	.	+INF	1.0000E+5
---- VAR R13	.	70.000	+INF	.
---- VAR R14	.	123.000	+INF	.
---- VAR Q213	.	105.000	+INF	.
---- VAR Q214	.	135.000	+INF	.
---- VAR Q224	.	80.000	+INF	.
---- VAR Q225	.	78.000	+INF	.
---- VAR R23	.	35.000	+INF	.
---- VAR R24	.	.	+INF	EPS
---- VAR Q1W5	.	183.000	+INF	.
---- VAR Q2W5	.	42.000	+INF	.
---- VAR Z	-INF	9.3000E+6	+INF	.

Podemos concluir que se puede obtener un costo total mínimo de $ 9.3 millones /año con un consumo de servicios de calentamiento de 60 MW y de enfriamiento de 225 MW.

Se aprecia que estos resultados también fueron obtenidos con el método de transbordo condensado, por lo tanto los dos métodos son confiables. Solo que con el método de transbordo expandido podremos obtener la red de intercambio de calor.

PROBLEMA VII

Para el problema VI, calcular el número mínimo de unidades y dibuje la red de intercambio de calor.

Solución

Declaración de variables

i = Es el número de la corriente caliente. (1, 2, ... # corrientes calientes)

j = Es el número de la corriente fría. (1, 2, ... # corrientes frías)

P = Número de unidades

q = Región (a ó b)

Q_{Hik} = Consumo de la corriente i del vapor de calentamiento. (MW)

Q_{Cjk} = Consumo de la corriente j del agua de enfriamiento. (MW)

Uij =Es el valor mínimo entre la corriente i del vapor de calentamiento y la corriente j del agua de enfriamiento. (MW)

Y_{ij}^{q} = Variable binaria

Formulación de la función objetivo

Para la formulación del modelo para la minimización del número de unidades es necesario definir el conjunto de variables binarias como sigue

$$Y_{ij}^{q} = \begin{Bmatrix} 1 \text{ si la corriente i intercambia calor con la corriente j en la region q} \\ 0 \text{si la corriente i no intercambia calor con la corriente j en la region q} \end{Bmatrix}$$

así que el problema a resolver es

$$\mathrm{min}P = \sum_{i=1}^{\substack{\#\,\text{corrientes}\\ \text{calientes}}} \; \sum_{j=1}^{\substack{\#\,\text{corrientes}\\ \text{frias}}} Y_{ij}^{q}$$

sujeta a las restricciones de balance de energía en cada uno de los intervalos, como en el caso del modelo del transbordo expandido, pero además se tendrán que incluir las restricciones lógicas siguientes

$$\sum_{k=1}^{\substack{\#\text{intervalo de}\\ \text{combinacion}\\ \text{para i \& j en}\\ \text{la region q}}} Q_{ijk} \leq U_{ij} Y_{ij}^{q} \qquad \text{con } Uij = \min(Q_i^{H}, Q_j^{C})$$

En base a lo anterior la función objetiva que nos permite calcular el número mínimo de unidades para este problema, quedaría:

$$\text{Min P} = Y_{s1}^{a} + Y_{11}^{a} + Y_{11}^{b} + Y_{12}^{b} + Y_{1w}^{b} + Y_{21}^{b} + Y_{22}^{b} + Y_{2w}^{b}$$

<u>Restricciones de igualdad</u>

La funcion objetiva esta sujeta a las restricciones de balance de energía de cada intervalo que se obtuvieron en el problema VI.

<u>Restricción de desigualdad</u>

Restricciones lógicas:

Arriba del punto de pliegue

Qs11 + Qs12 $\leq$ Us1 Ys1a con Us1 = min (Qs, Q1C) = min (60,120)MW

Us1 = 60 MW

$Q112 \leq U11\ Y11a$ con $U11 = \min(Q1H, Q1C) = \min(60,120)MW$

$U11 = 60\ MW$

Abajo del punto de pliegue

$Q112 + Q113 + Q114 \leq U11\ Y11b$ con $U11 = \min(Q1H, Q1C) = \min(220,240)MW$

$U11 = 220\ MW$

$Q124 + Q125 \leq U12\ Y12b$ con $U12 = \min(Q1H, Q2C) = \min(220,195)MW$

$U12 = 195\ MW$

$Q1w5 \leq U1w\ Y1wb$ con $U1w = \min(Q1H, Qw) = \min(220,225)MW$

$U1w = 225\ MW$

$Q213 + Q214 \leq U21\ Y21b$ con $U21 = \min(Q2H, Q1C) = \min(440,240)MW$

$U21 = 240\ MW$

$Q224 + Q225 \leq U22\ Y22b$ con $U22 = \min(Q2H, Q2C) = \min(440,195)MW$

$U22 = 195\ MW$

$Q2w5 \leq U2w\ Y_{2w}{}^{b}$ con $U2w = \min(Q2^{H}, Q_{w}) = \min(440,225)MW$

$U2w = 225\ MW$

<u>Codificación en GAMS</u>

```
* inicio del archivo units1.gms
* calculo del numero minimo de unidades

$ TITLE minimo numero de unidades
$ OFFSYMXREF
$ OFFSYMLIST

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion del conjunto de variables del problema
positive variables Qs, Qw,
            Qs11, Qs12, Rs1, Rs2,
            Q112, Q113, Q114, Q124, Q125, R12, R13, R14,
            Q213, Q214, Q224, Q225, R23, R24,
            Q1w5, Q2w5;
* declaracion de la variables de la funcion
free variable p;
* declaracion de las ecuaciones del modelo
* variables binarias
binary variables Ys1a, Y11a,
            Y11b, Y12b, Y1wb, Y21b, Y22b, Y2wb;
```

```
equations obj,
        bes1, bec11,
        bes2, beh12, bec12,
        beh13, beh23, bec13,
        beh14, beh24, bec14, bec24,
        beh15, beh25, bec25, bew5,
        rls1a, rl11a,
        rl11b, rl12b, rl1wb, rl21b, rl22b, rl2wb;
* expresiones del modelo
obj.. p =e= Ys1a + Y11a + Y11b + Y12b + Y1wb + Y21b + Y22b + Y2wb;
* valores del consumo minimo de servicios
Qs.l=60;
Qw.l=225;
* balance de energia para el intervalo 1
bes1.. Rs1 + Qs11 =e= Qs;
bec11.. Qs11 =e= 30;
* balance de energia para el intervalo 2
bes2.. Rs2 - Rs1 + Qs12 =e= 0;
beh12.. R12 + Q112 =e= 60;
bec12.. Qs12 + Q112 =e= 90;
* punto de pliegue
R12.l = 0;
Rs2.l = 0;
* balance de energia para el intervalo 3
beh13.. R13 - R12 + Q113 =e= 70;
beh23.. R23 + Q213 =e= 140;
bec13.. Q113 + Q213 =e= 105;
* balance de energia para el intervalo 4
beh14.. R14 - R13 + Q114 + Q124 =e= 90;
beh24.. R24 - R23 +Q214 + Q224 =e= 180;
bec14.. Q114 + Q214 =e= 135;
bec24.. Q124 + Q224 =e= 117;
* balance de energia para el intervalo 5
beh15.. -R14 + Q125 + Q1w5 =e= 60;
beh25.. -R24 + Q225 + Q2w5 =e= 120;
bec25.. Q125 + Q225 =e= 78;
bew5.. Q1w5 + Q2w5 =e= Qw;

* expresiones de las restricciones logicas
rls1a.. Qs11 + Qs12 =l= 60* Ys1a;
rl11a.. Q112 =l= 60* Y11a;
rl11b.. Q112 + Q113 + Q114 =l= 220* Y11b;
rl12b.. Q124 + Q125 =l= 195* Y12b;
rl1wb.. Q1w5 =l= 220* Y1wb;
rl21b.. Q213 + Q214 =l= 240* Y21b;
rl22b.. Q224 + Q225 =l= 195* Y22b;
rl2wb.. Q2w5 =l= 225* Y2wb;
* asignacion del nombre al problema
model unidades /all/;
```

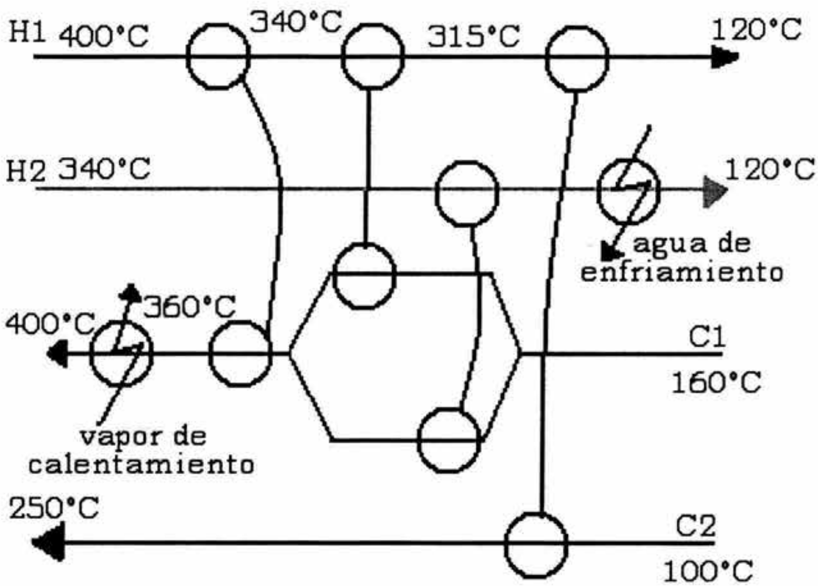
```
* declaracion de solucion del problema
solve unidades using mip minimizing p;
*fin del archivo units1.gms
```

En el punto 8.11 del Apéndice, se muestra el desplegado del archivo de salida, UNITS1.LST, del cual se tomaron los siguientes resultados reportados por GAMS:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR QS	.	60.000	+INF	.
---- VAR QW	.	225.000	+INF	.
---- VAR QS11	.	30.000	+INF	.
---- VAR QS12	.	30.000	+INF	.
---- VAR RS1	.	30.000	+INF	.
---- VAR RS2	.	.	+INF	EPS
---- VAR Q112	.	60.000	+INF	.
---- VAR Q113	.	.	+INF	EPS
---- VAR Q114	.	25.000	+INF	.
---- VAR Q124	.	117.000	+INF	.
---- VAR Q125	.	78.000	+INF	.
---- VAR R12	.	.	+INF	EPS
---- VAR R13	.	70.000	+INF	.
---- VAR R14	.	18.000	+INF	.
---- VAR Q213	.	105.000	+INF	.
---- VAR Q214	.	110.000	+INF	.
---- VAR Q224	.	.	+INF	.
---- VAR Q225	.	.	+INF	EPS
---- VAR R23	.	35.000	+INF	.
---- VAR R24	.	105.000	+INF	.
---- VAR Q1W5	.	.	+INF	EPS
---- VAR Q2W5	.	225.000	+INF	.
---- VAR P	-INF	6.000	+INF	.
---- VAR YS1A	.	1.000	1.000	1.000
---- VAR Y11A	.	1.000	1.000	1.000
---- VAR Y11B	.	1.000	1.000	1.000
---- VAR Y12B	.	1.000	1.000	1.000
---- VAR Y1WB	.	.	1.000	1.000
---- VAR Y21B	.	1.000	1.000	1.000
---- VAR Y22B	.	.	1.000	1.000
---- VAR Y2WB	.	1.000	1.000	1.000

En base a los resutados obtenidos por GAMS, para este problema el número mínimo de unidades deben ser 6.

La red de intercambio de calor correspondiente es:


H1 400°C
340°C
315°C
120°C
H2 340°C
120°C
agua de
enfriamiento
400°C
360°C
C1
160°C
vapor de
calentamiento
250°C
C2
100°C

6. CONCLUSIONES

En conclusión los modelos matemáticos ofrecen el modo más conveniente y compacto de reducir la información experimental, y un medio para la simulación de experiencias hipotéticas a partir de muy pocos datos experimentales. El modelado es justamente el proceso mediante el cual se construye una función o algoritmo, capaz de explicar el comportamiento observado del sistema y de predecir respuestas desconocidas. Una vez establecido el modelo general del sistema en estudio, se estará en condiciones de ajustar los parámetros del mismo a los datos experimentales.

Para la construcción de un modelo se necesita tener un conocimiento detallado tanto de cada una de las partes individuales que forman el sistema como de las interacciones existentes entre ellas, aproximándose tanto más el modelo a la realidad cuanto más detallado sea dicho conocimiento. Sin embargo, en muchos casos, para que un modelo sea manejable es necesario renunciar a algunos elementos que componen el sistema y/o a algunas de las interrelaciones entre ellos.

Si los elementos que se han conservado y las interrelaciones que se han establecido son correctas y abarcan los aspectos más relevantes del sistema, el modelo será útil.

Por contra, si se han identificado incorrectamente los elementos, o han sido mal definidos, o si se han olvidado algunas de las relaciones importantes, el modelo estará deformado y será inútil. Así pues, puede decirse que la clave para la construcción de un modelo radica, esencialmente, en identificar de manera adecuada y sin ambigüedades los elementos cruciales, definirlos de forma precisa y operativa y establecer las principales relaciones entre ellos.

El término optimización se aplica a aquellos casos en que se maximiza o minimiza la función objetivo.

Los problemas de optimización, por comodidad, se suelen enfocar sólo hacia búsquedas de mínimos. Cualquier máximo de la función modelo se puede considerar siempre como el opuesto de un mínimo, lo que permite la búsqueda de máximos con los mismos algoritmos que para mínimos: simplemente basta con cambiar el signo de la función objetivo. En los problemas de optimización es frecuente encontrar funciones que contengan varios máximos y mínimos. Se considera óptimo al más favorable de los mínimos relativos (o máximos, si se maximiza la respuesta), de entre todos los encontrados en el espacio de los parámetros. En general, el más favorable suele ser el más profundo de todos los mínimos o el máximo más elevado, aunque en algunas ocasiones hay que tener en cuenta otros factores que consideren la realidad experimental antes de su elección.

Para la resolución de problemas de optimización se recomienda utilizar el programa GAMS (General Algebraic Modeling System). Debido a su capacidad para resolver problemas pequeños (docenas de variables y restricciones) y grandes problemas (miles de variables y restricciones) escribiendo básicamente el mismo programa. Dispone de una forma compacta y eficiente para escribir bloques de ecuaciones similares sin más que escribir una de ellas.

El usuario de GAMS formula el modelo consistentemente, y una vez expresado en notación GAMS, uno de los programas disponibles se encarga de generar la solución. Como resultado, el usuario se centra en el modelado, sin ser perturbado por los problemas técnicos de los algoritmos de resolución. Esto hace posible un proceso de modelado muy sencillo y agradable.

Una vez entendido los conceptos básicos de este programa más los conocimientos de ingeniería, GAMS permite la resolución de problemas de Optimización del tipo lineal, no lineales, enteros, etc., y también el estudio de los diferentes aspectos del problema, con solo pequeños cambios en el archivo de entrada.

GAMS prácticamente reproduce la descripción del problema de programación matemática. Como resultado, el código GAMS es casi auto-explicativo para los estudiantes que tengan un mínimo conocimiento de optimización.

Además podemos ver que la última versión de GAMS para ambiente Windows® aporta mejoras de tiempo o funcionalidad con respecto a la versión previa.

En particular, una característica muy atractiva de GAMS 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.

Por todo esto el programa GAMS es de gran ayuda, para los estudiantes de Ingeniería Química, para la correcta resolución e interpretación de los problemas de Optimización.

7. BIBLIOGRAFÍA

- Edgar, Himmelblau & Lasdon (2001). Optimization of chemical processes. 2nd Edition McGraw-Hill

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

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

- Ignacio E. Grossmann, José Antonio Caballero y Héctor Yeomans (2000). Advances in Mathematical Programming for Automated Design, Integration and Operation of Chemical Processes. Department of Chemical Enginering, Carnegie Mellon University Pittsburg, PA 15213, USA.

- Lorenz T. Biegler, Jorge Nocedal, Claudia Schmid y David Ternet (2000). Numerical Experience with a Reduced Hessian Method for Large Scale Constraine Optimization. *Computacional Optimization and Applications*, Vol. 15, No. 1.

- Cutlip, M. B., and M. Shacham (1999). Problem Solving in Chemical Engineering with Numerical Methods. Prentice may PTR.

- Fourer. R. (1999). Linear Programming. *OR/MS Today, (Institute for Operations Research and the Management Sciences).*

- Nicolás J. Scenna y col. (1999). Modelado, Simulación y Optimización de Procesos Químicos.

- Van Hentenryck, P. (1999). The OPL Optimization Programming Language. The MIT Press, Cambridge, Mass.

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

- Bazaraa, M.; Jarvis, J.; Sherali, H., (1998). Programación Lineal y Flujo en Redes. Limusa.

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

- McCarl. B. A. (1998). So Your G.4MS Model Didn't Work Right. A Guide to Model Repair. Technical Report.

- McCarl. B. A. and Spreen. Th. H. (1998). Applied Mathematical Programming using

Algebraic Systems. Technical Report.

- Nocedal, J. and S. Wright (1998). Numerical Optimization. Springer.
- Ragsdale, C. T. (1998). Spreadsheet modeling and decision analysis: a practical introduction to management science. South-Western College. 2nd ed.
- Taha. H.A. (1998). Investigación de operaciones. Una introducción. Prentice Hall.
- Biegler, Grossmann & Westerberg (1997). Systematic Methods for Chemical Engineering Design. Prentice-Hall
- Hillier, F.S. Lieberman, G.J. (1997). Introducción a la Investigación de Operaciones. McGraw Hill.
- Pascal Van Hentenryck, Laurent Michel, and Yves Deville, (1997). Numerica: A Modeling Language for Global Optimization, The MIT Press, Cambridge, Mass.
- Mocholí, M. y Sala, R. (1996). Decisiones de optimización. Tirant lo Blanch. Valencia.
- Ramos. A. et al. (1996). Computational Experience with Optimization for a Bulk Production Cost Model. I2th PSCC. Dresden, Germany.
- Sarabia, A. (1996). La Investigación Operativa. UPCO (Universidad Pontificia de Comillas)
- C. Floudas, (1995). Non linear and Mixed-Integer Optimization. Oxford University press.
- Gass, S. I., Hirshfeld, D.S. and Wasil, E. A., (1995). Model Word: The Spreadsheeting of OR/MS. *Interfaces.*
- Sharda, R. and Rampal, G. (1995). Algebraic Modeling Languages on PCs. *OR/MS Today (Institute for Operations Research and the Management Sciences).*
- Winston, W.L. (1994). Investigación de Operaciones. Aplicaciones y Algoritmos. Grupo Editorial Iberoamericana.
- Bazaraa, Sherali & Shetty, (1993). Nonlinear Programming. Theory and Algorithms. 2nd edition, Wiley.
- R. Fourer, D. M. Gay and B. W. Kernighan. (1993). AMPL: A Modeling Language for Mathematical Progrmming. South San Francisco: The Scientific Press
- M. S. Bazaraa, H. D. Sherali y M. Shetty (1993). Nonlinear Programming: Theory and

Algorithms. Wiley, New Cork, EUA.

- A. Brooke, D. Kendrick y A. Meeraus (1992). GAMS: A User´s Guide, Release 2.25. The Scientific Press, South San Francisco, EUA.

- A. M. Geoffrion. (1992). The SML language for structured modeling. *Operations Research*, Vol. 40, No.1, págs. 38-75.

- Jones, C. (1992). Attributed Graphs, Graph-Grammars, and Structured Modeling. *Annals of Operations Research*, Vol. 38, págs. 281-324.

- Louis Leithold (1992). El Cálculo con Geometría Analitica. Sexta edición, HARLA.

- Jones, C. (1991). An Introduction to Graph-Based Modeling Systems, Part II: Graph Grammars and the Implementation. *ORSA Journal on Computing,* Vol. 3, págs. 180-206.

- Kendrick, D. (1991) "A Graphical Interface for Production and Transportation System Modeling: PTS", *Computer Science in Economics and Management*, Vol. 4, págs. 229-236.

- Jones, C. (1990). An Introduction to Graph-Based Modeling Systems, Part I: Overview. *ORSA Journal on Computing*, Vol. 2, págs.136-151.

- R. Fourer, D. M. Gay y B. W. Kernighan (1990). A modeling language for mathematical programming. *Management Science*, Vol. 36, No. 5, págs. 519-554.

- A. M. Geoffrion. (1989b). Integrated modeling systems. *Computer Science in Economics and Management*, Vol. 2, págs. 3-15.

- Douglas, J. M. (1988). Conceptual Design of Chemical Process. McGRaw-Hill.

- G. E. Pales and C. A. Floudas (1989). APROS: Algorithmic Development Methodology for Discrete-Continuous Optimization Problems. *Operations Research,* Vol. 37, No. 6, págs. 902-915.

- G. R. Kocis and I.E. Grossman (1988). Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis. *Ind. Eng. Chem. Res.* No. 27.

- Papalambros, P. Y. and D. J. Wilde (1988). Principles of Optimal Design. Cambridge Press.

- A. M. Geoffrion. (1987). An introduction to structured modeling. *Management Science*, Vol. 33, págs. 547-588.

- G. R. Kocis and I.E. Grossman (1987). Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Ind. Eng. Chem. Res.* No. 26.

- Lohner R. J. (1987). Enclosing the Solutions of ordinary inicial and boundary value problems. In *Computer Arithmetic: Scientific Computation and Programming Languages.* Wiley.

- Store J., Bulirsch R. (1987). Introduction to Numerical Analysis. Springer-Verlag.

- Bartels, R. H., and N. Mahdavi-Amiri (1986). On Generating Test Problems for Nonlinear Programming Algorithms. *SIAM J. Sci. Stat. Comput.*, No. 7.

- Roy, A., Lasdon, L. y Lordeman, J. (1986). Extending Planning Languages to Include Optimization Capabilities. *Management Science*, Vol. 32, No. 3, 360-373.

- Luenberger, D. G., (1984). Linear and Nonlinear Programming, 2d. ed., Addison Wesley.

- Dennis, J. E. and R. B. Schnabel (1983). Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice Hall, reprinted by SIAM.

- Meeraus, A. (1983). An Algebraic Approach to Modeling. *Journal of Economic Dynamics and Control*, Vol. 5, págs. 81-108.

- Papoulias&Grossmann (1983) A structural optimization approach to process synthesis – II. Heat recovery networks. Computer and Chemical Engineering.

- R. Fourer. (1983) Modeling languages versus matrix generators for linear programming. *ACM Transactions on Mathematical Software*, Vol. 9, págs. 143-183.

- Reklaitis G. V., Ravindran A. y Ragsdell K. M., (1983). Engineering Optimization-Methods and Applications. John Wiley&Sons.

- Bischop, J. y Meeraus, A. (1982). On the Development of a General Algebraic Modeling System in a Strategic Planning Environment. *Mathematical Programming Study*, Vol. 20, págs. 1-29.

- Ray, W. H. (1982). Advanced Process Control, McGraw-Hill.

- Schittkowski, K. (1981). The Nonlinear Programming Method of Wilson, Han and Powell with an Augmented Lagrangian Type Line Search Function. *Numerische Math.*, Vol. 38.

- Van der Hoek, G. (1980). Reduction Methods in Nonlinear Programming. Math. Centrum, Ámsterdam.

- Luyben, W. L. (1973). Process Modeling, Simulation, and Control for Chemical Engineers. McGraw-Hill.

* Beveridge, G. S., and R. S. Schechter (1967). Optimization: Theory and Practice. McGraw-Hill.

8. APÉNDICE

8.1 Al ejecutar el archivo PROCSEL1.gms en GAMS, este da origen al archivo de salida, PROCSEL1.lst. , que se muestra acontinuación. Con los resultados reportados se da solución a las preguntas planteadas en el problema I.

```
GAMS 2.25  PC AT/XT                                06/19/04 09:42:04  PAGE      1
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n


* inicio del archivo procsel1.gms

* asignacion de un titulo para el problema
GAMS 2.25  PC AT/XT                                06/19/04 09:42:04  PAGE      2
seleccion de procesos



*filename:     procsel1.gms
option optcr=0, limrow=0, limcol=0;
* declaracion de la naturaleza de las vartiables
BINARY VARIABLES y1, y2, y3;

POSITIVE VARIABLES A, B1,B2,B3,C2,C3,VC,CB;

* declaracion de la variable de la funcion objetivo
* VARIABLE DE LA GANACIA
free VARIABLE  G;
* declaracion  del nombre de las ecuaciones
EQUATIONS obj,
           dmax, cp1, exc, conv1, conv2,conv3,
           bm1, bm2, f1,f2,f3;

* declaracion de las expresiones de las ecuaciones
obj..     G =e= 1800*VC-(1000*y1+1500*y2+2000*y3 +
             500*A+950*CB +
             250*A+400*B2+550*B3);
dmax..   VC =l= 15;
cp1..    A =l= 16;
exc..    y2+y3 =e= 1;
conv1..  C2 =e= 0.82*B2;
conv2..  C3 =e= 0.95*B3;
conv3..  B1 =e= 0.90*A;
bm1..    B1 + CB =e= B2 + B3;
bm2..    C2 + C3 =e= VC;
f1..     A =l= 16*y1;
f2..     B2 =l= 15/0.82*y2;
f3..     B3 =l= 15/0.95*y3;
* asignacion de nombre del problema
MODEL procsel1 /ALL/                                                   ;
* declaracion del tipo de modelo
SOLVE procsel1 USING MIP MAXIMIZING G;
* fin del archivo procsel1.gms                           ;
```

```
COMPILATION TIME     =         0.000 SECONDS         VERID TP5-00-038
GAMS 2.25  PC AT/XT                               06/19/04 09:42:04  PAGE      3
seleccion de procesos
Model Statistics    SOLVE PROCSEL1 USING MIP FROM LINE 40


MODEL STATISTICS

BLOCKS OF EQUATIONS       12     SINGLE EQUATIONS        12
BLOCKS OF VARIABLES       12     SINGLE VARIABLES        12
NON ZERO ELEMENTS         32     DISCRETE VARIABLES       3

GENERATION TIME      =         0.000 SECONDS


EXECUTION TIME       =         0.000 SECONDS         VERID TP5-00-038
GAMS 2.25  PC AT/XT                               06/19/04 09:42:04  PAGE      4
seleccion de procesos
Solution Report     SOLVE PROCSEL1 USING MIP FROM LINE 40


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

     MODEL   PROCSEL1              OBJECTIVE  G
     TYPE    MIP                   DIRECTION  MAXIMIZE
     SOLVER  ZOOM                  FROM LINE  40

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE               1995.7895

 RESOURCE USAGE, LIMIT          0.000      1000.000
 ITERATION COUNT, LIMIT        12          1000


 Z O O M / X M P    ---   PC Version 2.2 Nov 1990

 Dr Roy E. Marsten and Dr Jaya Singhal,
 XMP Optimization Software Inc.
 Tucson, Arizona


 D E M O N S T R A T I O N   M O D E
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements: 1000
      Total discrete variables: 20


 Estimate work space needed     --     16 Kb
 Work space allocated           --    237 Kb

               Iterations      Time
 Initial LP            12       .00
 Heuristic              0       .00
 Branch and bound       0       .00
 Final LP               0       .00
```

```
                               LOWER      LEVEL      UPPER     MARGINAL

---- EQU OBJ                     .          .          .         1.000
---- EQU DMAX                 -INF       15.000     15.000        .
---- EQU CP1                  -INF       16.000     16.000        .
---- EQU EXC                  1.000      1.000      1.000     1315.789
---- EQU CONV1                  .          .          .       1800.000
---- EQU CONV2                  .          .          .       1800.000
---- EQU CONV3                  .          .          .        950.000
---- EQU BM1                    .          .          .       -950.000
---- EQU BM2                    .          .          .      -1800.000
---- EQU F1                   -INF         .          .        105.000
---- EQU F2                   -INF         .          .        153.930
---- EQU F3                   -INF         .          .        210.000

                               LOWER      LEVEL      UPPER     MARGINAL

---- VAR Y1                     .         1.000      1.000     680.000
---- VAR Y2                     .          .         1.000        .
---- VAR Y3                     .         1.000      1.000        .
---- VAR A                      .        16.000      +INF         .
---- VAR B1                     .        14.400      +INF         .
---- VAR B2                     .          .         +INF      -27.930
GAMS 2.25  PC AT/XT                                  06/19/04 09:42:04  PAGE      5
seleccion de procesos
Solution Report     SOLVE PROCSEL1 USING MIP FROM LINE 40

                               LOWER      LEVEL      UPPER     MARGINAL

---- VAR B3                     .        15.789      +INF         .
---- VAR C2                     .          .         +INF         .
---- VAR C3                     .        15.000      +INF         .
---- VAR VC                     .        15.000      +INF         .
---- VAR CB                     .         1.389      +INF         .
---- VAR G                    -INF     1995.789      +INF         .


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


EXECUTION TIME       =            0.060 SECONDS          VERID TP5-00-038


USER: CACHE DESIGN CASE STUDIES SERIES                    G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT       C:\GAMS\PROCSEL1.GMS
OUTPUT      C:\GAMS\PROCSEL1.LST
```

8.2 Al ejecutar el archivo ABCD1.GMS en GAMS, este da origen al archivo de salida, ABCD1.LST, que se muestra acontinuación. Con los resultados reportados podemos saber la mejor secuencia que debe seguir el proceso óptimo para la separación de los cuatro componentes, planteado en el problema II.

```
GAMS 2.25  PC AT/XT                               03/25/04 10:38:11  PAGE      1
abcd1


OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;
* variable de costos totales
free variable CT;
* nombre de las ecuaciones
equations obj,
bm1,bm2,bm3,bm4,bm5,bm6,
be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
exc1,exc2,exc3;
*expresiones de las ecuaciones
obj..CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112
                                                                    *Y10+
0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08
                                                            *F9+0.39*F10+
35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10..  Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
fl9.. F9=L=1000*Y9;
```

```
  48  f110.. F10=L=1000*Y10;
  49  exc1.. Y1+Y2+Y3=L=1;
  50  exc2.. Y4+Y5=L=1;
  51  exc3.. Y6+Y7=L=1;
  52  * nombre del problema
  53  model abcd /all/;
  54  solve abcd using mip minimizing CT;
  55  * fin del archivo abcd.gms
  56
  57
  58
  59
  60
  61
  62
GAMS 2.25  PC AT/XT                              03/25/04 10:38:11  PAGE       2
abcd1


COMPILATION TIME     =         0.060 SECONDS         VERID TP5-00-038
GAMS 2.25  PC AT/XT                              03/25/04 10:38:11  PAGE       3
abcd1
Model Statistics    SOLVE ABCD USING MIP FROM LINE 54



MODEL STATISTICS

BLOCKS OF EQUATIONS       31      SINGLE EQUATIONS        31
BLOCKS OF VARIABLES       32      SINGLE VARIABLES        32
NON ZERO ELEMENTS         98      DISCRETE VARIABLES      10

GENERATION TIME      =         0.110 SECONDS


EXECUTION TIME       =         0.160 SECONDS         VERID TP5-00-038
GAMS 2.25  PC AT/XT                              03/25/04 10:38:11  PAGE       4
abcd1
Solution Report     SOLVE ABCD USING MIP FROM LINE 54


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

     MODEL   ABCD                 OBJECTIVE  CT
     TYPE    MIP                  DIRECTION  MINIMIZE
     SOLVER  ZOOM                 FROM LINE  54

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       8 INTEGER SOLUTION
**** OBJECTIVE VALUE              3308.3300

 RESOURCE USAGE, LIMIT          0.160       1000.000
 ITERATION COUNT, LIMIT        62           1000

 Z O O M / X M P    ---   PC Version 2.2 Nov 1990
```

Dr Roy E. Marsten and Dr Jaya Singhal,
XMP Optimization Software Inc.
Tucson, Arizona

D E M O N S T R A T I O N   M O D E
You do not have a full license for this program.
The following size retrictions apply:
Total nonzero elements: 1000
Total discrete variables: 20

Estimate work space needed -- 33 Kb
Work space allocated -- 271 Kb

Note: The stopping tolerance is satisfied,
but the solution may not be optimal.

No better solution than : 3220.6300

	Relative	Absolute
Actual distance	2.72306971E-02	87.700000
Tolerances	.10000000	.00000000

	Iterations	Time
Initial LP	37	.06
Heuristic	15	.05
Branch and bound	0	.00
Final LP	10	.00

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU OBJ	.	.	.	1.000
---- EQU BM1	.	.	.	.
---- EQU BM2	.	.	.	.
---- EQU BM3	.	.	.	1.167
---- EQU BM4	.	.	.	.
---- EQU BM5	.	.	.	1.743
---- EQU BM6	1000.000	1000.000	1000.000	3.086
---- EQU BE1	.	.	.	.
---- EQU BE2	.	.	.	35.300
---- EQU BE3	.	.	.	.
---- EQU BE4	.	.	.	.
---- EQU BE5	.	.	.	.

GAMS 2.25 PC AT/XT 03/25/04 10:38:11 PAGE 5
abcd1
Solution Report SOLVE ABCD USING MIP FROM LINE 54

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU BE6	.	.	.	.
---- EQU BE7	.	.	.	.

---- EQU BE8	.	.	.	35.300
---- EQU BE9	.	.	.	.
---- EQU BE10	.	.	.	35.300
---- EQU BET	.	.	.	35.300
---- EQU FL1	-INF	.	.	-2.666
---- EQU FL2	-INF	.	.	.
---- EQU FL3	-INF	.	.	-2.836
---- EQU FL4	-INF	.	.	.
---- EQU FL5	-INF	.	.	.
---- EQU FL6	-INF	.	.	.
---- EQU FL7	-INF	.	.	.
---- EQU FL8	-INF	-450.000	.	.
---- EQU FL9	-INF	.	.	.
---- EQU FL10	-INF	-550.000	.	.
---- EQU EXC1	-INF	1.000	1.000	.
---- EQU EXC2	-INF	.	1.000	.
---- EQU EXC3	-INF	.	1.000	.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR F1	.	.	+INF	.
---- VAR F2	.	1000.000	+INF	.
---- VAR F3	.	.	+INF	.
---- VAR F4	.	.	+INF	1.268
---- VAR F5	.	.	+INF	0.210
---- VAR F6	.	.	+INF	0.780
---- VAR F7	.	.	+INF	0.767
---- VAR F8	.	550.000	+INF	.
---- VAR F9	.	.	+INF	0.080
---- VAR F10	.	450.000	+INF	.
---- VAR Q1	.	.	+INF	35.300
---- VAR Q2	.	42.000	+INF	.
---- VAR Q3	.	.	+INF	35.300
---- VAR Q4	.	.	+INF	35.300
---- VAR Q5	.	.	+INF	35.300
---- VAR Q6	.	.	+INF	35.300
---- VAR Q7	.	.	+INF	35.300
---- VAR Q8	.	24.200	+INF	.
---- VAR Q9	.	.	+INF	35.300
---- VAR Q10	.	9.900	+INF	.
---- VAR QT	.	76.100	+INF	.
---- VAR Y1	.	.	1.000	-2521.330
---- VAR Y2	.	1.000	1.000	52.000
---- VAR Y3	.	.	1.000	-2760.330
---- VAR Y4	.	.	1.000	38.000
---- VAR Y5	.	.	1.000	66.000
---- VAR Y6	.	.	1.000	125.000
---- VAR Y7	.	.	1.000	44.000
---- VAR Y8	.	1.000	1.000	58.000
---- VAR Y9	.	.	1.000	37.000
---- VAR Y10	.	1.000	1.000	112.000
---- VAR CT	-INF	3308.330	+INF	.

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

```
abcd1
Solution Report     SOLVE ABCD USING MIP FROM LINE 54


EXECUTION TIME       =          0.110 SECONDS        VERID TP5-00-038

USER: CACHE DESIGN CASE STUDIES SERIES                 G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT      C:\GAMS\ABCD1.GMS
OUTPUT     C:\GAMS\ABCD1.LST
```

8.3 Al ejecutar el archivo ABCD2.GMS en GAMS, este da origen al archivo de salida, ABCD2.LST, que se muestra acontinuación. Con los resultados reportados podemos saber la segunda mejor secuencia que debe seguir el proceso óptimo para la separación de los cuatro componentes, planteado en el problema II.

```
GAMS 2.25  PC AT/XT                             03/26/04 16:18:17  PAGE      1
abcd2

OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;
* variable de costos totales
free variable CT;
* nombre de las ecuaciones
equations obj,
bm1,bm2,bm3,bm4,bm5,bm6,
be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
exc1,exc2,exc3,
exc4;
*expresiones de las ecuaciones
obj.. CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+
                                                                   112*Y10+
0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*
                                                                  F9+0.39*F10+
35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
```

```
f18.. F8=L=1000*Y8;
f19.. F9=L=1000*Y9;
f110.. F10=L=1000*Y10;
exc1.. Y1+Y2+Y3=L=1;
exc2.. Y4+Y5=L=1;
exc3.. Y6+Y7=L=1;
exc4.. Y2+Y8+Y10=L=2;
* nombre del problema
model abcd /all/;
solve abcd using mip minimizing CT;
* fin del archivo abcd2.gms

GAMS 2.25  PC AT/XT                          03/26/04 16:18:17  PAGE      2
abcd2


COMPILATION TIME     =          0.000 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                          03/26/04 16:18:17  PAGE      3
abcd2
Model Statistics    SOLVE ABCD USING MIP FROM LINE 56




MODEL STATISTICS

BLOCKS OF EQUATIONS       32     SINGLE EQUATIONS        32
BLOCKS OF VARIABLES       32     SINGLE VARIABLES        32
NON ZERO ELEMENTS        101     DISCRETE VARIABLES      10

GENERATION TIME      =          0.170 SECONDS


EXECUTION TIME       =          0.170 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                          03/26/04 16:18:17  PAGE      4
abcd2
Solution Report     SOLVE ABCD USING MIP FROM LINE 56

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

     MODEL    ABCD                 OBJECTIVE  CT
     TYPE     MIP                  DIRECTION  MINIMIZE
     SOLVER   ZOOM                 FROM LINE  56

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      8 INTEGER SOLUTION
**** OBJECTIVE VALUE             3927.2728
```

```
RESOURCE USAGE, LIMIT           0.219       1000.000
ITERATION COUNT, LIMIT         97           1000


Z O O M / X M P    ---   PC Version 2.2 Nov 1990

Dr Roy E. Marsten and Dr Jaya Singhal,
XMP Optimization Software Inc.
Tucson, Arizona


D E M O N S T R A T I O N    M O D E
  You do not have a full license for this program.
  The following size retrictions apply:
     Total nonzero elements: 1000
     Total discrete variables: 20


Estimate work space needed     --     34 Kb
Work space allocated           --    271 Kb

Note: The stopping tolerance is satisfied,
     but the solution may not be optimal.


No better solution than :   3921.5728
                  Relative             Absolute
Actual distance    1.45349844E-03       5.7000000
Tolerances          .10000000           .00000000

The branch and bound tree contained        16 nodes (max.      7087 nodes).

               Iterations      Time
Initial LP             37       .06
Heuristic              31       .05
Branch and bound       18       .06
Final LP               11       .00


                        LOWER      LEVEL      UPPER    MARGINAL

---- EQU OBJ              .          .          .        1.000
---- EQU BM1              .          .          .        2.680
---- EQU BM2              .          .          .         .
---- EQU BM3              .          .          .         .
---- EQU BM4              .          .          .         .
---- EQU BM5              .          .          .        1.743
---- EQU BM6          1000.000   1000.000   1000.000     3.686
---- EQU BE1              .          .          .       35.300
---- EQU BE2              .          .          .         .
---- EQU BE3              .          .          .         .

GAMS 2.25  PC AT/XT                             03/26/04 16:18:17  PAGE       5
abcd2
Solution Report     SOLVE ABCD USING MIP FROM LINE 56


                        LOWER      LEVEL      UPPER    MARGINAL
```

```
---- EQU BE4          .          .          .        35.300
---- EQU BE5          .          .          .          .
---- EQU BE6          .          .          .          .
---- EQU BE7          .          .          .          .
---- EQU BE8          .          .          .        35.300
---- EQU BE9          .          .          .          .
---- EQU BE10         .          .          .          .
---- EQU BET          .          .          .        35.300
---- EQU FL1        -INF         .          .          .
---- EQU FL2        -INF         .          .        -2.608
---- EQU FL3        -INF         .          .        -3.436
---- EQU FL4        -INF     -150.000       .          .
---- EQU FL5        -INF         .          .        -2.470
---- EQU FL6        -INF         .          .          .
---- EQU FL7        -INF         .          .          .
---- EQU FL8        -INF     -450.050       .          .
---- EQU FL9        -INF         .          .          .
---- EQU FL10       -INF         .          .          .
---- EQU EXC1       -INF        1.000      1.000       .
---- EQU EXC2       -INF        1.000      1.000       .
---- EQU EXC3       -INF         .         1.000       .
---- EQU EXC4       -INF        1.000      2.000       .

                    LOWER      LEVEL      UPPER    MARGINAL

---- VAR F1           .      1000.000      +INF        .
---- VAR F2           .          .         +INF        .
---- VAR F3           .          .         +INF        .
---- VAR F4           .       850.000      +INF        .
---- VAR F5           .          .         +INF        .
---- VAR F6           .          .         +INF       0.780
---- VAR F7           .          .         +INF       0.110
---- VAR F8           .       549.950      +INF        .
---- VAR F9           .          .         +INF       0.080
---- VAR F10          .          .         +INF       0.390
---- VAR Q1           .        28.000      +INF        .
---- VAR Q2           .          .         +INF      35.300
---- VAR Q3           .          .         +INF      35.300
---- VAR Q4           .        34.000      +INF        .
---- VAR Q5           .          .         +INF      35.300
---- VAR Q6           .          .         +INF      35.300
---- VAR Q7           .          .         +INF      35.300
---- VAR Q8           .        24.198      +INF        .
---- VAR Q9           .          .         +INF      35.300
---- VAR Q10          .          .         +INF      35.300
---- VAR QT           .        86.198      +INF        .
---- VAR Y1           .         1.000      1.000    145.000
---- VAR Y2           .          .         1.000  -2555.513
---- VAR Y3           .          .         1.000  -3360.273
---- VAR Y4           .         1.000      1.000     38.000
---- VAR Y5           .          .         1.000  -2403.850
---- VAR Y6           .          .         1.000    125.000
---- VAR Y7           .          .         1.000     44.000
---- VAR Y8           .         1.000      1.000     58.000
---- VAR Y9           .          .         1.000     37.000
---- VAR Y10          .          .         1.000    112.000
---- VAR CT         -INF     3927.273      +INF        .
```

```
GAMS 2.25  PC AT/XT                                  03/26/04 16:18:17  PAGE      6
abcd2
Solution Report     SOLVE ABCD USING MIP FROM LINE 56


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


EXECUTION TIME       =          0.110 SECONDS        VERID TP5-00-038


USER: CACHE DESIGN CASE STUDIES SERIES                  G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT       C:\GAMS\ABCD2.GMS
OUTPUT      C:\GAMS\ABCD2.LST
```

8.4 Al ejecutar el archivo ABCD3.GMS en GAMS, este da origen al archivo de salida, ABCD3.LST, que se muestra acontinuación. Con los resultados reportados podemos saber la tercera mejor secuencia que debe seguir el proceso óptimo para la separación de los cuatro componentes, planteado en el problema II.

```
GAMS 2.25  PC AT/XT                              03/26/04 16:38:51   PAGE      1
abcd3


OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;
* variable de costos totales
free variable CT;
* nombre de las ecuaciones
equations obj,
bm1,bm2,bm3,bm4,bm5,bm6,
be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
exc1,exc2,exc3,
exc4,exc5;
*expresiones de las ecuaciones
obj..CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112
                                                                     *Y10+
0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*
                                                              F9+0.39*F10+
35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
```

```
  48  f19.. F9=L=1000*Y9;
  49  f110.. F10=L=1000*Y10;
  50  exc1.. Y1+Y2+Y3=L=1;
  51  exc2.. Y4+Y5=L=1;
  52  exc3.. Y6+Y7=L=1;
  53  exc4.. Y2+Y8+Y10=L=2;
  54  exc5.. Y1+Y4+Y8=L=2;
  55  * nombre del problema
  56  model abcd /all/;
  57  solve abcd using mip minimizing CT;
  58  * fin del archivo abcd3.gms
  59
  60
  61
  62

GAMS 2.25  PC AT/XT                           03/26/04 16:38:51  PAGE      2
abcd3


  63
  64
  65
  66


COMPILATION TIME     =        0.060 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                           03/26/04 16:38:51  PAGE      3
abcd3
Model Statistics    SOLVE ABCD USING MIP FROM LINE 57


MODEL STATISTICS

BLOCKS OF EQUATIONS       33     SINGLE EQUATIONS        33
BLOCKS OF VARIABLES       32     SINGLE VARIABLES        32
NON ZERO ELEMENTS        104     DISCRETE VARIABLES      10

GENERATION TIME      =        0.110 SECONDS


EXECUTION TIME       =        0.160 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                           03/26/04 16:38:51  PAGE      4
abcd3
Solution Report     SOLVE ABCD USING MIP FROM LINE 57


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

     MODEL   ABCD                OBJECTIVE  CT
     TYPE    MIP                 DIRECTION  MINIMIZE
     SOLVER  ZOOM                FROM LINE  57

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE             4102.9966
```

```
RESOURCE USAGE, LIMIT            0.332        1000.000
ITERATION COUNT, LIMIT         163            1000


Z O O M / X M P     ---   PC Version 2.2 Nov 1990

Dr Roy E. Marsten and Dr Jaya Singhal,
XMP Optimization Software Inc.
Tucson, Arizona


D E M O N S T R A T I O N     M O D E
  You do not have a full license for this program.
  The following size retrictions apply:
     Total nonzero elements: 1000
     Total discrete variables: 20


Estimate work space needed     --      34 Kb
Work space allocated           --     271 Kb

The branch and bound tree contained        36 nodes (max.      6957 nodes).

                 Iterations      Time
Initial LP            37          .06
Heuristic             48          .05
Branch and bound      67          .11
Final LP              11          .06


                        LOWER      LEVEL      UPPER     MARGINAL

---- EQU OBJ              .          .          .         1.000
---- EQU BM1              .          .          .          .
---- EQU BM2              .          .          .         2.143
---- EQU BM3              .          .          .         1.167
---- EQU BM4              .          .          .          .
---- EQU BM5              .          .          .          .
---- EQU BM6          1000.000   1000.000   1000.000      3.871
---- EQU BE1              .          .          .          .
---- EQU BE2              .          .          .          .
---- EQU BE3              .          .          .        35.300
---- EQU BE4              .          .          .          .
---- EQU BE5              .          .          .          .
---- EQU BE6              .          .          .          .
---- EQU BE7              .          .          .        35.300
---- EQU BE8              .          .          .          .
---- EQU BE9              .          .          .          .
---- EQU BE10             .          .          .        35.300
---- EQU BET              .          .          .        35.300
---- EQU FL1            -INF         .          .        -3.451


GAMS 2.25  PC AT/XT                       03/26/04 16:38:51  PAGE       5
abcd3
Solution Report      SOLVE ABCD USING MIP FROM LINE 57
```

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU FL2	-INF	.	.	-3.226
---- EQU FL3	-INF	.	.	.
---- EQU FL4	-INF	.	.	.
---- EQU FL5	-INF	.	.	.
---- EQU FL6	-INF	.	.	-1.363
---- EQU FL7	-INF	-200.000	.	.
---- EQU FL8	-INF	.	.	.
---- EQU FL9	-INF	.	.	.
---- EQU FL10	-INF	-549.600	.	.
---- EQU EXC1	-INF	1.000	1.000	.
---- EQU EXC2	-INF	.	1.000	.
---- EQU EXC3	-INF	1.000	1.000	.
---- EQU EXC4	-INF	1.000	2.000	.
---- EQU EXC5	-INF	.	2.000	.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR F1	.	.	+INF	.
---- VAR F2	.	.	+INF	.
---- VAR F3	.	1000.000	+INF	.
---- VAR F4	.	.	+INF	0.140
---- VAR F5	.	.	+INF	0.210
---- VAR F6	.	.	+INF	.
---- VAR F7	.	800.000	+INF	.
---- VAR F8	.	.	+INF	0.190
---- VAR F9	.	.	+INF	0.080
---- VAR F10	.	450.400	+INF	.
---- VAR Q1	.	.	+INF	35.300
---- VAR Q2	.	.	+INF	35.300
---- VAR Q3	.	54.000	+INF	.
---- VAR Q4	.	.	+INF	35.300
---- VAR Q5	.	.	+INF	35.300
---- VAR Q6	.	.	+INF	35.300
---- VAR Q7	.	31.200	+INF	.
---- VAR Q8	.	.	+INF	35.300
---- VAR Q9	.	.	+INF	35.300
---- VAR Q10	.	9.909	+INF	.
---- VAR QT	.	95.109	+INF	.
---- VAR Y1	.	.	1.000	-3305.997
---- VAR Y2	.	.	1.000	-3174.027
---- VAR Y3	.	1.000	1.000	76.000
---- VAR Y4	.	.	1.000	38.000
---- VAR Y5	.	.	1.000	66.000
---- VAR Y6	.	.	1.000	-1238.496
---- VAR Y7	.	1.000	1.000	44.000
---- VAR Y8	.	.	1.000	58.000
---- VAR Y9	.	.	1.000	37.000
---- VAR Y10	.	1.000	1.000	112.000
---- VAR CT	-INF	4102.997	+INF	.

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

```
EXECUTION TIME        =        0.110 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                          03/26/04 16:38:51  PAGE      6
abcd3


USER: CACHE DESIGN CASE STUDIES SERIES                 G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT       C:\GAMS\ABCD3.GMS
OUTPUT      C:\GAMS\ABCD3.LST
```

8.5 Al ejecutar el archivo ABCDP.GMS en GAMS, este da origen al archivo de salida, ABCDP.LST, que se muestra acontinuación. Con los resultados reportados podemos ver la secuencia que no debe seguir el proceso para la separación de los cuatro componentes, planteado en el problema II.

```
GAMS 2.25  PC AT/XT                              03/26/04 17:00:45  PAGE      1
abcdp


OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion de variables
positive variables F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,
Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,Q10,QT;
binary variables Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10;
* variable de costos totales
free variable CT;
* nombre de las ecuaciones
equations obj,
bm1,bm2,bm3,bm4,bm5,bm6,
be1,be2,be3,be4,be5,be6,be7,be8,be9,be10,bet,
fl1,fl2,fl3,fl4,fl5,fl6,fl7,fl8,fl9,fl10,
exc1,exc2,exc3,exc4;
*expresiones de las ecuaciones
obj..CT=E=145*Y1+52*Y2+76*Y3+38*Y4+66*Y5+125*Y6+44*Y7+58*Y8+37*Y9+112
                                                                   *Y10+
0.42*F1+0.12*F2+0.25*F3+0.14*F4+0.21*F5+0.78*F6+0.11*F7+0.19*F8+0.08*
                                                              F9+0.39*F10+
35.3*QT;
bm1.. F4+F5=E=0.85*F1;
bm2.. F6+F7=E=0.8*F3;
bm3.. F10=E=0.45*F2+0.563*F7;
bm4.. F9=E=0.765*F5+0.812*F6;
bm5.. F8=E=0.55*F2+0.647*F4;
bm6.. F1+F2+F3=E=1000;
be1.. Q1=E=0.028*F1;
be2.. Q2=E=0.042*F2;
be3.. Q3=E=0.054*F3;
be4.. Q4=E=0.040*F4;
be5.. Q5=E=0.047*F5;
be6.. Q6=E=0.024*F6;
be7.. Q7=E=0.039*F7;
be8.. Q8=E=0.044*F8;
be9.. Q9=E=0.036*F9;
be10.. Q10=E=0.022*F10;
bet.. QT=E=Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10;
fl1.. F1=L=1000*Y1;
fl2.. F2=L=1000*Y2;
fl3.. F3=L=1000*Y3;
fl4.. F4=L=1000*Y4;
fl5.. F5=L=1000*Y5;
fl6.. F6=L=1000*Y6;
fl7.. F7=L=1000*Y7;
fl8.. F8=L=1000*Y8;
fl9.. F9=L=1000*Y9;
```

```
  48  fl10.. F10=L=1000*Y10;
  49  exc1.. Y1+Y2+Y3=L=1;
  50  exc2.. Y4+Y5=L=1;
  51  exc3.. Y6+Y7=L=1;
  52  exc4.. Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8+Y9+Y10=L=3;
  53  * nombre del problema
  54  model abcd /all/;
  55  solve abcd using mip maximizing CT;
  56  * fin del archivo abcdp.gms
```

```
COMPILATION TIME     =         0.050 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                           03/26/04 17:00:45  PAGE      2
abcdp
Model Statistics    SOLVE ABCD USING MIP FROM LINE 55


MODEL STATISTICS

BLOCKS OF EQUATIONS       32     SINGLE EQUATIONS        32
BLOCKS OF VARIABLES       32     SINGLE VARIABLES        32
NON ZERO ELEMENTS        108     DISCRETE VARIABLES      10

GENERATION TIME      =         0.220 SECONDS
EXECUTION TIME       =         0.280 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                           03/26/04 17:00:45  PAGE      3
abcdp
Solution Report     SOLVE ABCD USING MIP FROM LINE 55


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

     MODEL   ABCD                 OBJECTIVE  CT
     TYPE    MIP                  DIRECTION  MAXIMIZE
     SOLVER  ZOOM                 FROM LINE  55

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE             4573.4397

 RESOURCE USAGE, LIMIT          0.988      1000.000
 ITERATION COUNT, LIMIT       141          1000


 Z O O M / X M P    ---   PC Version 2.2 Nov 1990

 Dr Roy E. Marsten and Dr Jaya Singhal,
 XMP Optimization Software Inc.
 Tucson, Arizona


 D E M O N S T R A T I O N   M O D E
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements: 1000
      Total discrete variables: 20
```

Estimate work space needed	--	34 Kb	
Work space allocated	--	271 Kb	

The branch and bound tree contained 16 nodes (max. 7085 nodes).

	Iterations	Time
Initial LP	41	.06
Heuristic	44	.33
Branch and bound	32	.16
Final LP	24	.17

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU OBJ	.	.	.	1.000
---- EQU BM1	.	.	.	2.902
---- EQU BM2	.	.	.	2.724
---- EQU BM3	.	.	.	1.167
---- EQU BM4	.	.	.	1.351
---- EQU BM5	.	.	.	1.743
---- EQU BM6	1000.000	1000.000	1000.000	4.335
---- EQU BE1	.	.	.	35.300
---- EQU BE2	.	.	.	35.300
---- EQU BE3	.	.	.	35.300
---- EQU BE4	.	.	.	35.300
---- EQU BE5	.	.	.	35.300
---- EQU BE6	.	.	.	35.300
---- EQU BE7	.	.	.	35.300
---- EQU BE8	.	.	.	35.300
---- EQU BE9	.	.	.	35.300
---- EQU BE10	.	.	.	35.300
---- EQU BET	.	.	.	35.300
---- EQU FL1	-INF	.	.	.

GAMS 2.25 PC AT/XT 03/26/04 17:00:45 PAGE 4
abcdp
Solution Report SOLVE ABCD USING MIP FROM LINE 55

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU FL2	-INF	.	.	.
---- EQU FL3	-INF	.	.	.
---- EQU FL4	-INF	.	.	.
---- EQU FL5	-INF	.	.	.
---- EQU FL6	-INF	-200.000	.	.
---- EQU FL7	-INF	.	.	.
---- EQU FL8	-INF	.	.	.
---- EQU FL9	-INF	-350.400	.	.
---- EQU FL10	-INF	.	.	.
---- EQU EXC1	-INF	1.000	1.000	.
---- EQU EXC2	-INF	.	1.000	.
---- EQU EXC3	-INF	1.000	1.000	.
---- EQU EXC4	-INF	3.000	3.000	.

	LOWER	LEVEL	UPPER	MARGINAL

```
---- VAR F1            .          .         +INF      -0.460
---- VAR F2            .          .         +INF      -1.249
---- VAR F3            .      1000.000      +INF        .
---- VAR F4            .          .         +INF      -0.223
---- VAR F5            .          .         +INF        .
---- VAR F6            .       800.000      +INF        .
---- VAR F7            .          .         +INF      -0.581
---- VAR F8            .          .         +INF        .
---- VAR F9            .       649.600      +INF        .
---- VAR F10           .          .         +INF        .
---- VAR Q1            .          .         +INF        .
---- VAR Q2            .          .         +INF        .
---- VAR Q3            .        54.000      +INF        .
---- VAR Q4            .          .         +INF        .
---- VAR Q5            .          .         +INF        .
---- VAR Q6            .        19.200      +INF        .
---- VAR Q7            .          .         +INF        .
---- VAR Q8            .          .         +INF        .
---- VAR Q9            .        23.386      +INF        .
---- VAR Q10           .          .         +INF        .
---- VAR QT            .        96.586      +INF        .
---- VAR Y1            .          .         1.000    145.000
---- VAR Y2            .          .         1.000     52.000
---- VAR Y3            .         1.000      1.000     76.000
---- VAR Y4            .          .         1.000     38.000
---- VAR Y5            .          .         1.000     66.000
---- VAR Y6            .         1.000      1.000    125.000
---- VAR Y7            .          .         1.000     44.000
---- VAR Y8            .          .         1.000     58.000
---- VAR Y9            .         1.000      1.000     37.000
---- VAR Y10           .          .         1.000    112.000
---- VAR CT          -INF     4573.440      +INF        .


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


EXECUTION TIME       =       0.220 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                         03/26/04 17:00:45   PAGE       5
abcdp


USER: CACHE DESIGN CASE STUDIES SERIES               G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT      C:\GAMS\ABCDP.GMS
OUTPUT     C:\GAMS\ABCDP.LST
```

8.6 Al ejecutar el archivo TRABAJO.GMS en GAMS, este da origen al archivo de salida, TRABAJO.LST, que se muestra acontinuación. Con los resultados reportados podemos saber el trabajo mínimo y las presiones intermedias, como se pide en el problema III.

```
GAMS 2.25  PC AT/XT                                05/06/04 13:23:57  PAGE      1
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n

   1  * inicio del archivo trabajo.gms
   2  * modelo para la minimizacion del trabajo
   3  * para un sistema de tres etapas de compresion

GAMS 2.25  PC AT/XT                                05/06/04 13:23:57  PAGE      2
compresion


   7
   8  OPTION LIMROW=0;
   9  OPTION LIMCOL=0;
  10
  11  * declaracion de las variables del problema
  12  positive variables P2,P3, rc1, rc2, rc3;
  13  * declaracion de la variable de la funcion objetivo
  14  free variable Wt;
  15  * declaracion del conjunto de ecuaciones del modelo
  16  equations obj, erc1, erc2, erc3;
  17
  18  *expresion de la funcion objetivo para el trabajo minimo
  19  erc1.. 15*rc1=e=P2;
  20  erc2.. P2*rc2=e=P3;
  21  erc3.. P3*rc3=e=1000;
  22  obj.. Wt =e= 10.731*(70+460)*1.32/(16*0.32)*
  23               (rc1**0.2424 + rc2**0.2424 + rc3**0.2424 - 3);
  24  P2.lo=60;
  25  P3.lo=180;
  26
  27  * asignacion del nombre al problema
  28  model compresion /all/;
  29  * declaracion de solucion del problema
  30  solve compresion using NLP minimizing Wt;
  31  *fin del archivo trabajo.gms

COMPILATION TIME     =          0.060 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                                05/06/04 13:23:57  PAGE      3
compresion
Model Statistics    SOLVE COMPRESION USING NLP FROM LINE 30



MODEL STATISTICS
```

```
BLOCKS OF EQUATIONS          4     SINGLE EQUATIONS           4
BLOCKS OF VARIABLES          6     SINGLE VARIABLES           6
NON ZERO ELEMENTS           11     NON LINEAR N-Z             7
DERIVATIVE POOL              6     CONSTANT POOL              2
CODE LENGTH                 75

GENERATION TIME      =        0.490 SECONDS


EXECUTION TIME       =        0.490 SECONDS          VERID TP5-00-038

GAMS 2.25  PC AT/XT                                 05/06/04 13:23:57  PAGE        4
compresion
Solution Report     SOLVE COMPRESION USING NLP FROM LINE 30


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

     MODEL   COMPRESION            OBJECTIVE  WT
     TYPE    NLP                   DIRECTION  MINIMIZE
     SOLVER  MINOS5                FROM LINE  30

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE             1777.2112

 RESOURCE USAGE, LIMIT           0.277      1000.000
 ITERATION COUNT, LIMIT         31          1000
 EVALUATION ERRORS               0             0

     M I N O S     5.3     (Nov 1990)          Ver: 225-DOS-02
     = = = = =

     B. A. Murtagh, University of New South Wales
        and
     P. E. Gill,  W. Murray,  M. A. Saunders and M. H. Wright
     Systems Optimization Laboratory, Stanford University.

 D E M O N S T R A T I O N   M O D E
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements:  1000
      Nonlinear nonzero elements:  300


 Estimate work space needed     --     39 Kb
 Work space allocated           --    158 Kb


 EXIT -- OPTIMAL SOLUTION FOUND
 MAJOR ITNS, LIMIT              17       200
 FUNOBJ, FUNCON CALLS           88        88
 SUPERBASICS                     2
 INTERPRETER USAGE             .00
 NORM RG / NORM PI       4.995E-08

                       LOWER      LEVEL      UPPER     MARGINAL
```

```
---- EQU OBJ        -4398.872 -4398.872 -4398.872      1.000
---- EQU ERC1           .         .         .          8.205
---- EQU ERC2           .         .         .          2.023
---- EQU ERC3        1000.000  1000.000  1000.000      0.499

                      LOWER     LEVEL     UPPER    MARGINAL

---- VAR P2           60.000    60.822     +INF       EPS
---- VAR P3          180.000   246.621     +INF       EPS
---- VAR RC1            .        4.055     +INF        .
---- VAR RC2            .        4.055     +INF        .
---- VAR RC3            .        4.055     +INF        .
---- VAR WT           -INF    1777.211     +INF        .

GAMS 2.25  PC AT/XT                          05/06/04 13:23:57  PAGE      5
compresion
Solution Report     SOLVE COMPRESION USING NLP FROM LINE 30


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


EXECUTION TIME      =        0.060 SECONDS        VERID TP5-00-038

USER: CACHE DESIGN CASE STUDIES SERIES             G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT      C:\GAMS\TRABAJO.GMS
OUTPUT     C:\GAMS\TRABAJO.LST
```

8.7 Al ejecutar el archivo MINF.GMS en GAMS, este da origen al archivo de salida, MINF.LST, que se muestra acontinuación. Con los resultados reportados podemos saber los valores óptimos para la función objetivo, planteada en el problema IV.

```
GAMS 2.25  PC AT/XT                          03/29/04 13:32:42  PAGE      1
minf


   4
   5  OPTION LIMROW=0;
   6  OPTION LIMCOL=0;
   7
   8  * declaracion de las variables del problema
   9  positive variables x1, x2, x3;
  10  * declaracion de la variable de la funcion objetivo
  11  free variable f;
  12  * declaracion del conjunto de ecuaciones del modelo
  13  equations obj, g1, g2, g3, h1, h2, h3;
  14  * declaracion de las expresiones de las ecuaciones
  15  obj.. f =e= x1*x1 + x2*x2 + x3*x3;
  16  g1.. -2*x1 - x2 =g= -5;
  17  g2 .. -x1 - x3 =g= -2;
  18  g3.. -x1 - 2*x2 - x3 =g= -10;
  19  h1.. 2*x1 - 2*x2 + x3 =e= -2;
  20  h2.. 10*x1 + 8*x2 - 14*x3 =e= 26;
  21  h3.. -4*x1 + 5*x2 - 6*x3 =e= 6;
  22  * declaracion de los limites de las variables
  23  x1.l = 1.;
  24  x2.l = 2.;
  25  x3.l = 0.;
  26  * asignacion del nombre al problema
  27  model minf /all/;
  28  * declaracion de solucion del problema
  29  solve minf using NLP minimizing f;
  30  *fin de la codificacion del archivo minf.gms


COMPILATION TIME     =        0.000 SECONDS       VERID TP5-00-038

GAMS 2.25  PC AT/XT                          03/29/04 13:32:42  PAGE      2
minf


Model Statistics    SOLVE MINF USING NLP FROM LINE 29

MODEL STATISTICS

BLOCKS OF EQUATIONS        7     SINGLE EQUATIONS          7
BLOCKS OF VARIABLES        4     SINGLE VARIABLES          4
NON ZERO ELEMENTS         20     NON LINEAR N-Z            3
DERIVATIVE POOL            7     CONSTANT POOL             0
CODE LENGTH               45

GENERATION TIME      =        0.110 SECONDS
```

```
EXECUTION TIME       =        0.170 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                         03/29/04 13:32:42  PAGE      3
minf
Solution Report     SOLVE MINF USING NLP FROM LINE 29


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

     MODEL   MINF                    OBJECTIVE  F
     TYPE    NLP                     DIRECTION  MINIMIZE
     SOLVER  MINOS5                  FROM LINE  29

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      2 LOCALLY OPTIMAL
**** OBJECTIVE VALUE               5.0000

 RESOURCE USAGE, LIMIT          0.109      1000.000
 ITERATION COUNT, LIMIT         2          1000
 EVALUATION ERRORS              0             0

     M I N O S    5.3    (Nov 1990)         Ver: 225-DOS-02
     = = = = =

     B. A. Murtagh, University of New South Wales
       and
     P. E. Gill,  W. Murray,  M. A. Saunders and M. H. Wright
     Systems Optimization Laboratory, Stanford University.


D E M O N S T R A T I O N   M O D E
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements:  1000
      Nonlinear nonzero elements:  300


 Estimate work space needed     --     39 Kb
 Work space allocated           --    166 Kb


 EXIT -- OPTIMAL SOLUTION FOUND
 MAJOR ITNS, LIMIT                  1      200
 FUNOBJ, FUNCON CALLS               6        0
 SUPERBASICS                        0
 INTERPRETER USAGE                .00
 NORM RG / NORM PI         0.000E+00

                        LOWER      LEVEL      UPPER    MARGINAL

---- EQU OBJ              .          .          .        1.000
---- EQU G1            -5.000     -4.000      +INF         .
---- EQU G2            -2.000     -1.000      +INF         .
---- EQU G3           -10.000     -5.000      +INF         .
---- EQU H1            -2.000     -2.000     -2.000     -0.667
---- EQU H2            26.000     26.000     26.000      0.333
```

```
---- EQU H3                6.000      6.000      6.000        .

                          LOWER      LEVEL      UPPER     MARGINAL

---- VAR X1                 .        1.000      +INF         .
---- VAR X2                 .        2.000      +INF         .
---- VAR X3                 .          .        +INF        5.333
---- VAR F                -INF       5.000      +INF         .

GAMS 2.25  PC AT/XT                            03/29/04 13:32:42  PAGE      4
minf
Solution Report     SOLVE MINF USING NLP FROM LINE 29


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


EXECUTION TIME       =        0.060 SECONDS        VERID TP5-00-038


USER: CACHE DESIGN CASE STUDIES SERIES              G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT      C:\GAMS\MINF.GMS
OUTPUT     C:\GAMS\MINF.LST
```

8.8 Codificación en matlab para obtener el cálculo del balance de energía, el cual se utiliza en la solución del problema V.

Códigos [matlab]

1. Código pinch

```
%Rutina para el Calculo de Pinch Point
%Construccion de Red de Intercambio de Calor
%Victor Manuel Zavala Tejeda (c)
%Mayo 2003

clc
clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%Entrada de
Datos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pas=input('Introduccion de Datos? 1-En Linea   2-Datos.m --> ');
if pas==1
   n=input('Numero de Corrientes Calientes (H): ');
   m=input('Numero de Corrientes Frias (C): ');
   dt=input('Acercamiento Minimo (dTmin): ');
   for i=1:n
   fprintf('[Tin Tout] de la Corriente H%0.0f',i)
   TH1(i,:)=input(': ');
   end
   FcpH=input('FCps Corrientes Calientes [H1..Hn] ');
   for i=1:m
   fprintf('[Tin Tout] de la Corriente C%0.0f',i)
   TC1(i,:)=input(': ');
   end
   FcpC=input('FCps Corrientes Frias [C1..Cn] ');
else
  run Datos
end
%------------------------------------------------------------------------%

%Obtencion Vectores de Temperatura

TH=[TH1(:,1)' TH1(:,2)'];
TC=[TC1(:,1)' TC1(:,2)'];
Tcaldt=sort(TH-dt);
Tfrdt=sort(TC+dt);
vector1=sort([TH, Tfrdt])';
vector2=sort([TC, Tcaldt])';

%Eliminacion temperaturas repetidas
for i=1:2*(n+m)
el(i)=vector1(i);
if i==1
```

```
  i=i+1;
 else
  if (el(i)-el(i-1))<=1e-18
  vector1(i)=0;
 end
 end
 end
 vector1=sort(vector1);
 for i=1:2*(n+m)
 el(i)=vector2(i);
 if i==1
  i=i+1;
 else
  if (el(i)-el(i-1))<=1e-18
  vector2(i)=0;
 end
 end
 end
 vector2=sort(vector2);


 %Obtencion Intervalos de Temperatura
 for i=1:length(vector1)
   if i==1
   vectorH(i)=vector1(length(vector1));
   else
   vectorH(i)=vector1(length(vector1)-i+1);
   end
 end

 for i=1:2*(n+m)

   if vectorH(i)==0
   break
   else
   vectorHaux(i)=vectorH(i);
 end
 end
 vectorH=vectorHaux;


 for j=1:length(vector2)
   if i==1
   vectorC(j)=vector2(length(vector2));
   else
   vectorC(j)=vector2(length(vector2)-j+1);
   end
 end
```

```
for i=1:2*(n+m)

  if vectorC(i)==0
  break
  else
  vectorCaux(i)=vectorC(i);
end
end
vectorC=vectorCaux;

cond=length(vectorH);

for i=2:cond
   dtk(1)=0;
   dtk(i)=abs(vectorH(i)-vectorH(i-1));
end

for i=1:n
  limsupH(i)=max(TH1(i,:));
  liminfH(i)=min(TH1(i,:));
end
for i=1:m
  limsupC(i)=max(TC1(i,:));
  liminfC(i)=min(TC1(i,:));
end

for j=1:n
limsH(j)=find(vectorH==limsupH(j));
liminH(j)=find(vectorH==liminfH(j));
end

for j=1:m
limsC(j)=find(vectorC==limsupC(j));
liminC(j)=find(vectorC==liminfC(j));
end

% %Construccion flechas
for j=1:n
for i=1:length(vectorH)
   if i==limsH(j)
   flechasH(i,j)=0;
   end
   if i==liminH(j)
   flechasH(i,j)=1;
   end
   if i<limsH(j)
   flechasH(i,j)=0;
   end
   if i>liminH(j)
```

```
    flechasH(i,j)=0;
    end
    if i<liminH(j)
    if i>limsH(j)
      flechasH(i,j)=1;
    end
    end
end
end

for j=1:m
for i=1:length(vectorC)
    if i==limsC(j)
    flechasC(i,j)=0;
    end
    if i==liminC(j)
    flechasC(i,j)=1;
    end
    if i<limsC(j)
    flechasC(i,j)=0;
    end
    if i>liminC(j)
    flechasC(i,j)=0;
    end
    if i<liminC(j)
    if i>limsC(j)
      flechasC(i,j)=1;
    end
    end
end
end

for j=1:n
    for i=1:length(vectorH)
    if flechasH(i,j)==1
    QH(i,j)=FcpH(j)*dtk(i);
    else
    QH(i,j)=0;
    end
    end
end

for j=1:m
    for i=1:length(vectorH)
    if flechasC(i,j)==1
    QC(i,j)=FcpC(j)*dtk(i);
    else
    QC(i,j)=0;
    end
```

```
  end
end

for i=1:length(vectorH)
  QHtot(i)=sum(QH(i,:));
  QCtot(i)=sum(QC(i,:));
end
for i=1:length(vectorH)
  dQk(i)=QHtot(i)-QCtot(i);
end

for i=1:cond
  if i==1
  k(i)=0;
else
  k(i)=i-1;
end
end

for i=1:n
  QHtotal(i)=FcpH(i)*(max(TH1(i,:))-min(TH1(i,:)));
end
for i=1:m
  QCtotal(i)=FcpC(i)*(max(TC1(i,:))-min(TC1(i,:)));
end

for i=1:length(QH)
QHk(i)=sum(QH(i,:));
QCk(i)=sum(QC(i,:));
end
Dif=QCk-QHk;

%%%%%%%%%%%%%%%%%Representacion en
Cascada%%%%%%%%%%%%%%%%%%%%%
Qcal=0;
for i=1:length(dQk)
if i==1
  cascada1(i)=Qcal;
else
  cascada1(i)=dQk(i)+cascada1(i-1);
end
end
Pinch1=find(cascada1==0);

Qcal1=-min(cascada1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
for i=1:length(dQk)
```

```
if i==1
   cascada2(i)=Qcal1;
else
   cascada2(i)=dQk(i)+cascada2(i-1);
end
end
Pinch2=find(cascada2==0);
TPinch=(vectorH(Pinch2)+vectorC(Pinch2))/2;

for i=1:length(cascada2)
   if cascada2(i)<0
   Qcal2=-cascada2(i);
   break
   end
end
Qenf=cascada2(length(cascada2));

%%%%%Trazo Curvas Compuestas Caliente y Fria%%%%%%
for i=1:length(vectorH)
if  i==1
   CompH(i)=sum(QHk);
else
   CompH(i)=CompH(i-1)-QHk(i);
end
if i==length(vectorH)
  CompH(i)=0;
end
end

for i=1:length(vectorH)
  CompC(1)=sum(QCk)+Qenf;
  if i>1
  CompC(i)=CompC(i-1)-QCk(i);
  end
  CompC(length(vectorH))=Qenf;
end

plot(CompC,vectorC,'bo')
hold on
plot(CompH,vectorH,'redo')
hold on
plot(CompC,vectorC,'b')
hold on
plot(CompH,vectorH,'red')
ylabel('Temperatura')
xlabel('Q')
grid on
axis([0 max(CompC) min(vectorC) max(vectorH)])
clc
```

```
%Despliegue de Resultados
fprintf('---------------------------------------------------Despliegue de Resultados---------------------------------------------------------\n')
fprintf('\n')
fprintf('Tabla de Resultados:\n')
fprintf('k          HS          T          T          CS        dTk        QHS        QCS        dQk\n')
fprintf('-------------------------------------------------------------------------------------------------------\n')
Tabla=[k',flechasH, vectorH',vectorC', flechasC, dtk',QH,QC,dQk']
fprintf('Calor Aportado de H´s y C´s a Intervalos:\n')
fprintf('-----------------------')
Tabla1=[k',QH,k',QC]
fprintf('        k        QH        QC     Sale-Entra \n')
fprintf('    ---------------------------------------------  \n')
[k' QHk' QCk' Dif']
fprintf('Temperatura Pinch\n')
fprintf('-----------------------\n')
TPinch
fprintf('Qenf-Qcal Calculada\n')
fprintf('-----------------------\n')
difQ=sum(dQk)
fprintf('Qenf-Qcal Datos\n')
fprintf('-----------------------\n')
difQp=sum(QHtotal)-sum(QCtotal)
fprintf('Q total corrientes H\n')
fprintf('-----------------------\n')
QHtotal=QHtotal
fprintf('Q total corrientes C\n')
fprintf('-----------------------\n')
QCtotal=QCtotal
fprintf('Representacion en Cascada Qcal=0\n')
fprintf('-----------------------------------\n')
fprintf('       dQk\n')
[ dQk' cascada1']
fprintf('Representacion en Cascada Qcal=%0.2f\n',Qcal1)
fprintf('-----------------------------------\n')
fprintf('       dQk\n')
[ dQk' cascada2']

save Tabla.dat  Tabla -ascii
save Tabla1.dat  Tabla1 –ascii
```

8.9 Al ejecutar el archivo UTILITY.GMS en GAMS, este da origen al archivo de salida, UTILITY.LST, que se muestra acontinuación. En el se reporta el costo total y el consumo de servicios, solicitados en el problema V.

```
GAMS 2.25  PC AT/XT                              03/29/04 18:20:55  PAGE      1
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n


   1  * inicio del archivo utility.gms
   2  * modelo para la minimizacion del trabajo
   3  * para un sistema de tres etapas de compresion
   4  * calculo de costo minimo de servicios

GAMS 2.25  PC AT/XT                              03/29/04 18:20:55  PAGE      2
utility


   8
   9  OPTION LIMROW=0;
  10  OPTION LIMCOL=0;
  11
  12  * declaracion de las variables del problema
  13  positive variables Qs, Qw, R1, R2, R3, R4;
  14  * declaracion de la variable de la funcion objetivo [$/a&o]
  15  free variable Z;
  16  * delcaracion del conjunto de ecuaciones del modelo
  17  equations obj,
  18            beint1, beint2, beint3, beint4, beint5;
  19
  20  *expresion de la funcion objetivo para el costo minimo
  21  obj..      Z =e= 80000 * Qs + 20000 * Qw;
  22  * declaracion de las expresiones de balances
  23  * de enrgia en los intervalos [MW]
  24  beint1..   R1 - Qs =e= -30;
  25  beint2..   R2 - R1 =e= -30;
  26  beint3..   R3 - R2 =e= 105;
  27  beint4..   R4 - R3 =e= 18;
  28  beint5..   Qw - R4 =e= 102;
  29
  30  * asignacion del nombre al problema
  31  model utility /all/;
  32  * declaracion de solucion del problema
  33  solve utility using LP minimizing Z;
  34  * fin del archivo utility.gms


COMPILATION TIME     =        0.000 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                              03/29/04 18:20:55  PAGE      3
utility
Model Statistics    SOLVE UTILITY USING LP FROM LINE 33


MODEL STATISTICS
```

```
BLOCKS OF EQUATIONS          6     SINGLE EQUATIONS          6
BLOCKS OF VARIABLES          7     SINGLE VARIABLES          7
NON ZERO ELEMENTS           13

GENERATION TIME       =          0.160 SECONDS
EXECUTION TIME        =          0.220 SECONDS         VERID TP5-00-038

GAMS 2.25  PC AT/XT                                 03/29/04 18:20:55  PAGE       4
utility
Solution Report     SOLVE UTILITY USING LP FROM LINE 33


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

     MODEL   UTILITY                  OBJECTIVE  Z
     TYPE    LP                       DIRECTION  MINIMIZE
     SOLVER  BDMLP                    FROM LINE  33

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE             9300000.0000

 RESOURCE USAGE, LIMIT             0.047      1000.000
 ITERATION COUNT, LIMIT            2          1000

 BDM - LP  VERSION 1.01

 A. Brooke, A. Drud, and A. Meeraus,
 Analytic Support Unit,
 Development Research Department,
 World Bank,
 Washington, D.C. 20433, U.S.A.

 D E M O N S T R A T I O N   M O D E
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements: 1000


 Estimate work space needed     --     35 Kb
 Work space allocated           --    358 Kb

 EXIT -- OPTIMAL SOLUTION FOUND.


                        LOWER      LEVEL      UPPER     MARGINAL

---- EQU OBJ              .          .          .         1.000
---- EQU BEINT1        -30.000    -30.000    -30.000  -8.000E+4
---- EQU BEINT2        -30.000    -30.000    -30.000  -8.000E+4
---- EQU BEINT3        105.000    105.000    105.000  20000.000
---- EQU BEINT4         18.000     18.000     18.000  20000.000
---- EQU BEINT5        102.000    102.000    102.000  20000.000


                        LOWER      LEVEL      UPPER     MARGINAL

---- VAR QS               .        60.000     +INF         .
```

```
---- VAR QW               .         225.000        +INF        .
---- VAR R1               .          30.000        +INF        .
---- VAR R2               .            .           +INF   1.0000E+5
---- VAR R3               .         105.000        +INF        .
---- VAR R4               .         123.000        +INF        .
---- VAR Z             -INF       9.3000E+6        +INF        .


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


EXECUTION TIME       =        0.110 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                             03/29/04 18:20:55  PAGE       5
utility


USER: CACHE DESIGN CASE STUDIES SERIES                  G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT       C:\GAMS\UTILITY.GMS
OUTPUT      C:\GAMS\UTILITY.LST
```

8.10 Al ejecutar el archivo TUTILITY.GMS en GAMS, este da origen al archivo de salida, TUTILITY.LST, que se muestra acontinuación. En el se reporta el consumo mínimo de servicios, calculado mediante el método de transbordo expandido, con se pide en el problema VI.

```
GAMS 2.25  PC AT/XT                          05/09/04 15:29:57  PAGE      1
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n


* inicio del archivo tutility.gms
* calculo del consumo minimo de servicios aplicando
* el metodo de transbordo expandido

GAMS 2.25  PC AT/XT                          05/09/04 15:29:57  PAGE      2
servicios



OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion del conjunto de variables del problema
positive variables Qs, Qw,
Qs11, Qs12, Rs1, Rs2,
Q112, Q113, Q114, Q124, Q125, R12, R13, R14,
Q213, Q214, Q224, Q225, R23, R24,
Q1w5, Q2w5;
* declaracion de la variables de la funcion
free variable z;
* declaracion de las ecuaciones del modelo
equations obj,
bes1, bec11,
bes2, beh12, bec12,
beh13, beh23, bec13,
beh14, beh24, bec14, bec24,
beh15, beh25, bec25, bew5;
* expresiones del modelo
obj.. z =e= 80000 * Qs + 20000 * Qw;
* balance de energia para el intervalo 1
bes1.. Rs1 + Qs11 =e= Qs;
bec11.. Qs11 =e= 30;
* balance de energia para el intervalo 2
bes2.. Rs2 - Rs1 + Qs12 =e= 0;
beh12.. R12 + Q112 =e= 60;
bec12.. Qs12 + Q112 =e= 90;
* punto de pliegue
R12.l = 0;
Rs2.l = 0;
* balance de energia para el intervalo 3
beh13.. R13 - R12 + Q113 =e= 70;
beh23.. R23 + Q213 =e= 140;
bec13.. Q113 + Q213 =e= 105;
* balance de energia para el intervalo 4
beh14.. R14 - R13 + Q114 + Q124 =e= 90;
beh24.. R24 - R23 +Q214 + Q224 =e= 180;
bec14.. Q114 + Q214 =e= 135;
bec24.. Q124 + Q224 =e= 117;
```

```
  45  * balance de energia para el intervalo 5
  46  beh15.. -R14 + Q125 + Q1w5 =e= 60;
  47  beh25.. -R24 + Q225 + Q2w5 =e= 120;
  48  bec25.. Q125 + Q225 =e= 78;
  49  bew5.. Q1w5 + Q2w5 =e= Qw;
  50  * asignacion del nombre al problema
  51  model tutility /all/;
  52  * declaracion de solucion del problema
  53  solve tutility using lp minimizing z;
  54  *fin del archivo tutility.gms


COMPILATION TIME     =         0.060 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                               05/09/04 15:29:57  PAGE       3
servicios
Model Statistics    SOLVE TUTILITY USING LP FROM LINE 53



MODEL STATISTICS

BLOCKS OF EQUATIONS        17     SINGLE EQUATIONS          17
BLOCKS OF VARIABLES        23     SINGLE VARIABLES          23
NON ZERO ELEMENTS          44

GENERATION TIME      =         0.270 SECONDS


EXECUTION TIME       =         0.320 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                               05/09/04 15:29:57  PAGE       4
servicios
Solution Report     SOLVE TUTILITY USING LP FROM LINE 53


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

     MODEL   TUTILITY            OBJECTIVE  Z
     TYPE    LP                  DIRECTION  MINIMIZE
     SOLVER  BDMLP               FROM LINE  53

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE          9300000.0000

 RESOURCE USAGE, LIMIT           0.328      1000.000
 ITERATION COUNT, LIMIT          8          1000

 BDM - LP  VERSION 1.01

 A. Brooke, A. Drud, and A. Meeraus,
 Analytic Support Unit,
 Development Research Department,
 World Bank,
 Washington, D.C. 20433, U.S.A.

 D E M O N S T R A T I O N   M O D E
```

```
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements: 1000


 Estimate work space needed     --     42 Kb
 Work space allocated           --    350 Kb

 EXIT -- OPTIMAL SOLUTION FOUND.

                          LOWER     LEVEL     UPPER    MARGINAL

---- EQU OBJ                .         .         .         1.000
---- EQU BES1               .         .         .      -8.000E+4
---- EQU BEC11            30.000    30.000    30.000  80000.000
---- EQU BES2               .         .         .      -8.000E+4
---- EQU BEH12            60.000    60.000    60.000  -8.000E+4
---- EQU BEC12            90.000    90.000    90.000  80000.000
---- EQU BEH13            70.000    70.000    70.000  20000.000
---- EQU BEH23           140.000   140.000   140.000  20000.000
---- EQU BEC13           105.000   105.000   105.000  -2.000E+4
---- EQU BEH14            90.000    90.000    90.000  20000.000
---- EQU BEH24           180.000   180.000   180.000  20000.000
---- EQU BEC14           135.000   135.000   135.000  -2.000E+4
---- EQU BEC24           117.000   117.000   117.000  -2.000E+4
---- EQU BEH15            60.000    60.000    60.000  20000.000
---- EQU BEH25           120.000   120.000   120.000  20000.000
---- EQU BEC25            78.000    78.000    78.000  -2.000E+4
---- EQU BEW5               .         .         .      -2.000E+4

                          LOWER     LEVEL     UPPER    MARGINAL

---- VAR QS                 .       60.000     +INF        .
---- VAR QW                 .      225.000     +INF        .
---- VAR QS11               .       30.000     +INF        .
---- VAR QS12               .       30.000     +INF        .
---- VAR RS1                .       30.000     +INF        .
---- VAR RS2                .         .        +INF   80000.000
---- VAR Q112               .       60.000     +INF        .
GAMS 2.25  PC AT/XT                           05/09/04 15:29:57  PAGE      5
servicios
Solution Report     SOLVE TUTILITY USING LP FROM LINE 53


                          LOWER     LEVEL     UPPER    MARGINAL

---- VAR Q113               .         .        +INF       EPS
---- VAR Q114               .         .        +INF       EPS
---- VAR Q124               .       37.000     +INF        .
---- VAR Q125               .         .        +INF       EPS
---- VAR R12                .         .        +INF    1.0000E+5
---- VAR R13                .       70.000     +INF        .
---- VAR R14                .      123.000     +INF        .
---- VAR Q213               .      105.000     +INF        .
---- VAR Q214               .      135.000     +INF        .
---- VAR Q224               .       80.000     +INF        .
---- VAR Q225               .       78.000     +INF        .
---- VAR R23                .       35.000     +INF        .
```

```
---- VAR R24             .          .          +INF        EPS
---- VAR Q1W5            .       183.000       +INF         .
---- VAR Q2W5            .        42.000       +INF         .
---- VAR Z             -INF   9.3000E+6        +INF         .


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


EXECUTION TIME       =         0.280 SECONDS        VERID TP5-00-038


USER: CACHE DESIGN CASE STUDIES SERIES                G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION

**** FILE SUMMARY

INPUT      C:\GAMS\TUTILITY.GMS
OUTPUT     C:\GAMS\TUTILITY.LST
```

8.11 Al ejecutar el archivo UNITS1.GMS en GAMS, este da origen al archivo de salida, UNITS1.LST, que se muestra acontinuación. En el se reporta el número mínimo de unidades, con se pide en el problema VII.

```
GAMS 2.25  PC AT/XT                          05/10/04 17:26:22  PAGE      1
G e n e r a l   A l g e b r a i c   M o d e l i n g   S y s t e m
C o m p i l a t i o n

* inicio del archivo units1.gms
* calculo del numero minimo de unidades

GAMS 2.25  PC AT/XT                          05/10/04 17:26:22  PAGE      2
minimo numero de unidades


OPTION LIMROW=0;
OPTION LIMCOL=0;
* declaracion del conjunto de variables del problema
positive variables Qs, Qw,
Qs11, Qs12, Rs1, Rs2,
Q112, Q113, Q114, Q124, Q125, R12, R13, R14,
Q213, Q214, Q224, Q225, R23, R24,
Q1w5, Q2w5;
* declaracion de la variables de la funcion
free variable p;
* declaracion de las ecuaciones del modelo
* variables binarias
binary variables Ys1a, Y11a,
Y11b, Y12b, Y1wb, Y21b, Y22b, Y2wb;
equations obj,
bes1, bec11,
bes2, beh12, bec12,
beh13, beh23, bec13,
beh14, beh24, bec14, bec24,
beh15, beh25, bec25, bew5,
rls1a, rl11a,
rl11b, rl12b, rl1wb, rl21b, rl22b, rl2wb;
* expresiones del modelo
obj.. p =e= Ys1a + Y11a + Y11b + Y12b + Y1wb + Y21b + Y22b + Y2wb;
* valores del consumo minimo de servicios
Qs.l=60;
Qw.l=225;
* balance de energia para el intervalo 1
bes1.. Rs1 + Qs11 =e= Qs;
bec11.. Qs11 =e= 30;
* balance de energia para el intervalo 2
bes2.. Rs2 - Rs1 + Qs12 =e= 0;
beh12.. R12 + Q112 =e= 60;
bec12.. Qs12 + Q112 =e= 90;
* punto de pliegue
R12.l = 0;
Rs2.l = 0;
* balance de energia para el intervalo 3
beh13.. R13 - R12 + Q113 =e= 70;
```

```
  45  beh23.. R23 + Q213 =e= 140;
  46  bec13.. Q113 + Q213 =e= 105;
  47  * balance de energia para el intervalo 4
  48  beh14.. R14 - R13 + Q114 + Q124 =e= 90;
  49  beh24.. R24 - R23 +Q214 + Q224 =e= 180;
  50  bec14.. Q114 + Q214 =e= 135;
  51  bec24.. Q124 + Q224 =e= 117;
  52  * balance de energia para el intervalo 5
  53  beh15.. -R14 + Q125 + Q1w5 =e= 60;
  54  beh25.. -R24 + Q225 + Q2w5 =e= 120;
  55  bec25.. Q125 + Q225 =e= 78;
  56  bew5.. Q1w5 + Q2w5 =e= Qw;
  57  * expresiones de las restricciones logicas
  58  rlsla.. Qs11 + Qs12 =l= 60* Ysla;
  59  rl11a.. Q112 =l= 60* Y11a;
  60  rl11b.. Q112 + Q113 + Q114 =l= 220* Y11b;
  61  rl12b.. Q124 + Q125 =l= 195* Y12b;
  62  rl1wb.. Q1w5 =l= 220* Y1wb;
  63  rl21b.. Q213 + Q214 =l= 240* Y21b;
  64  rl22b.. Q224 + Q225 =l= 195* Y22b;
  65  rl2wb.. Q2w5 =l= 225* Y2wb;
  66  * asignacion del nombre al problema

GAMS 2.25  PC AT/XT                              05/10/04 17:26:22  PAGE      3
minimo numero de unidades


  67  model unidades /all/;
  68  * declaracion de solucion del problema
  69  solve unidades using mip minimizing p;
  70  *fin del archivo units1.gms


COMPILATION TIME     =          0.110 SECONDS        VERID TP5-00-038

GAMS 2.25  PC AT/XT                              05/10/04 17:26:22  PAGE      4
minimo numero de unidades
Model Statistics    SOLVE UNIDADES USING MIP FROM LINE 69


MODEL STATISTICS

BLOCKS OF EQUATIONS         25     SINGLE EQUATIONS         25
BLOCKS OF VARIABLES         31     SINGLE VARIABLES         31
NON ZERO ELEMENTS           72     DISCRETE VARIABLES        8

GENERATION TIME      =          0.270 SECONDS


EXECUTION TIME       =          0.380 SECONDS        VERID TP5-00-038
GAMS 2.25  PC AT/XT                              05/10/04 17:26:22  PAGE      5
minimo numero de unidades
Solution Report     SOLVE UNIDADES USING MIP FROM LINE 69


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

     MODEL   UNIDADES              OBJECTIVE  P
```

```
     TYPE    MIP                    DIRECTION  MINIMIZE
     SOLVER  ZOOM                   FROM LINE  69

**** SOLVER STATUS      1 NORMAL COMPLETION
**** MODEL STATUS       1 OPTIMAL
**** OBJECTIVE VALUE                 6.0000

 RESOURCE USAGE, LIMIT           1.047       1000.000
 ITERATION COUNT, LIMIT        146           1000


 Z O O M / X M P    ---   PC Version 2.2 Nov 1990

 Dr Roy E. Marsten and Dr Jaya Singhal,
 XMP Optimization Software Inc.
 Tucson, Arizona


 D E M O N S T R A T I O N   M O D E
   You do not have a full license for this program.
   The following size retrictions apply:
      Total nonzero elements: 1000
      Total discrete variables: 20


 Estimate work space needed     --     26 Kb
 Work space allocated           --    271 Kb

 The branch and bound tree contained       35 nodes (max.      9692 nodes).

                Iterations      Time
 Initial LP             32       .05
 Heuristic              37       .33
 Branch and bound       57       .39
 Final LP               20       .22
```

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU OBJ	.	.	.	1.000
---- EQU BES1	.	.	.	EPS
---- EQU BEC11	30.000	30.000	30.000	EPS
---- EQU BES2	.	.	.	EPS
---- EQU BEH12	60.000	60.000	60.000	EPS
---- EQU BEC12	90.000	90.000	90.000	EPS
---- EQU BEH13	70.000	70.000	70.000	EPS
---- EQU BEH23	140.000	140.000	140.000	EPS
---- EQU BEC13	105.000	105.000	105.000	EPS
---- EQU BEH14	90.000	90.000	90.000	EPS
---- EQU BEH24	180.000	180.000	180.000	EPS
---- EQU BEC14	135.000	135.000	135.000	EPS
---- EQU BEC24	117.000	117.000	117.000	EPS
---- EQU BEH15	60.000	60.000	60.000	EPS
---- EQU BEH25	120.000	120.000	120.000	EPS
---- EQU BEC25	78.000	78.000	78.000	EPS
---- EQU BEW5	.	.	.	EPS
---- EQU RLS1A	-INF	.	.	.
---- EQU RL11A	-INF	.	.	.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU RL11B	-INF	-135.000	.	.
---- EQU RL12B	-INF	.	.	.
---- EQU RL1WB	-INF	.	.	.
---- EQU RL21B	-INF	-25.000	.	.
---- EQU RL22B	-INF	.	.	EPS
---- EQU RL2WB	-INF	.	.	.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR QS	.	60.000	+INF	.
---- VAR QW	.	225.000	+INF	.
---- VAR QS11	.	30.000	+INF	.
---- VAR QS12	.	30.000	+INF	.
---- VAR RS1	.	30.000	+INF	.
---- VAR RS2	.	.	+INF	EPS
---- VAR Q112	.	60.000	+INF	.
---- VAR Q113	.	.	+INF	EPS
---- VAR Q114	.	25.000	+INF	.
---- VAR Q124	.	117.000	+INF	.
---- VAR Q125	.	78.000	+INF	.
---- VAR R12	.	.	+INF	EPS
---- VAR R13	.	70.000	+INF	.
---- VAR R14	.	18.000	+INF	.
---- VAR Q213	.	105.000	+INF	.
---- VAR Q214	.	110.000	+INF	.
---- VAR Q224	.	.	+INF	.
---- VAR Q225	.	.	+INF	EPS
---- VAR R23	.	35.000	+INF	.
---- VAR R24	.	105.000	+INF	.
---- VAR Q1W5	.	.	+INF	EPS
---- VAR Q2W5	.	225.000	+INF	.
---- VAR P	-INF	6.000	+INF	.
---- VAR YS1A	.	1.000	1.000	1.000
---- VAR Y11A	.	1.000	1.000	1.000
---- VAR Y11B	.	1.000	1.000	1.000
---- VAR Y12B	.	1.000	1.000	1.000
---- VAR Y1WB	.	.	1.000	1.000
---- VAR Y21B	.	1.000	1.000	1.000
---- VAR Y22B	.	.	1.000	1.000
---- VAR Y2WB	.	1.000	1.000	1.000

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


EXECUTION TIME      =        0.330 SECONDS        VERID TP5-00-038


USER: CACHE DESIGN CASE STUDIES SERIES              G911007-1447AX-TP5
      GAMS DEMONSTRATION VERSION
```

```
**** FILE SUMMARY

INPUT       C:\GAMS\UNITS1.GMS
OUTPUT      C:\GAMS\UNITS1.LST
```