

***GAMS. Instalación. Creación de  
ficheros y corrección de errores.  
Uso de las librerías***

GAMS son las iniciales de: GENERAL ALGEBRAIC MODELLING SYSTEM, un software de optimización desarrollado inicialmente por técnicos del Banco Mundial para evaluar los modelos de crecimiento de países en vías de desarrollo, y que posteriormente ha ido ampliando sus posibilidades y capacidades.

En la actualidad esta gestionado por una compañía denominada GAMS Corporation que comercializa este tipo de software y da soporte a los diferentes usuarios, existiendo una lista de consultas: GAMS-L, en donde se envían y resuelven cuestiones por parte de GAMS o de otros usuarios.

La practica totalidad de empresa dedicadas 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<sup>®</sup>, tanto para las versiones 95, 98 o 2000. 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, ahora 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.com/download/>

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 pagina de descarga de los archivos necesarios. En la misma pagina de registro se

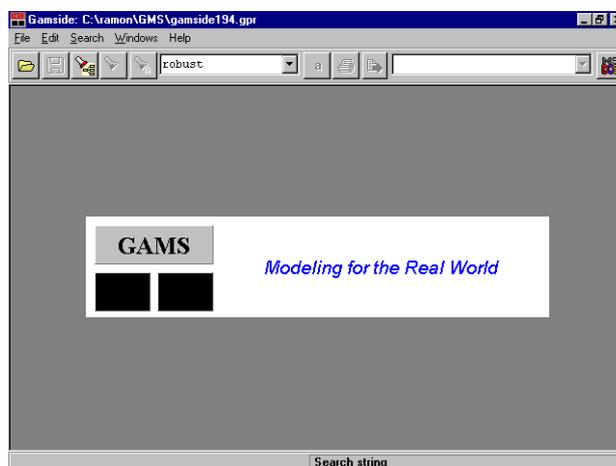
informa de las limitaciones de esta versión si es usada para construcción y resolución de modelos de optimización:

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

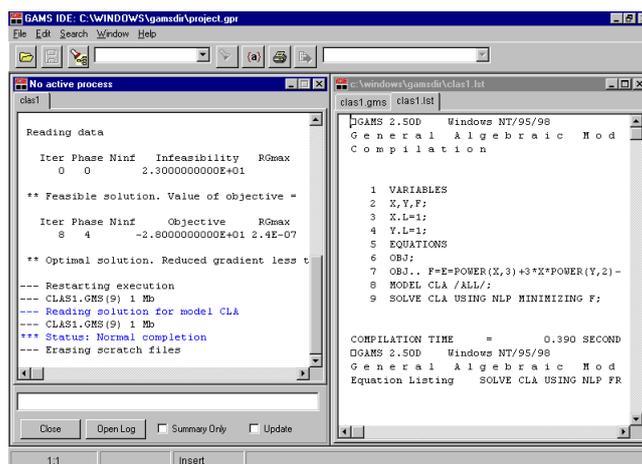
Una vez copiados los archivos en el disco, se procede a la instalación automática del programa. En esa instalación se crean el fichero de ayuda necesarias, así como una serie de manuales de ayuda (todos ellos en formato PDF). Estos manuales incluyen una guía de usuario de casi 300 paginas. Además incluye un “Tutorial” de 30 paginas, 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 con anterioridad, 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 practicas. 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 acudir 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 ficheros GMS y LST.



En los ficheros de modelos, 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 el fichero con una serie de indicaciones y comentarios respecto al 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 en 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, indicando que case son, que tipo de restricciones presentan, si tienen o no cotas, el punto de partida de cada uno de los

**Bloque de ecuaciones.** Este bloque ha de comenzar con el título 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 construcción de un fichero de datos, con los bloques obligatorios.

El fichero 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 ficheros de GAMS.

Comenzaremos la construcción del fichero con una serie de líneas de comentario 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 creaci3n de un fichero de datos que permita
    crear y resolver un problema clasico de maximizaci3n de la utilidad
    de un consumidor sujeto a una restriccion 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 funcion de utilidad del consumidor es:

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

    Por tanto, el problema a resolver sera:

    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 construccion 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 a~nadir 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 funci3n objetivo, no obstante, podemos relacionar m3s 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 de realiza como:

Punto de Partida	Nombre.L = {valor}
------------------	--------------------

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

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

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 fichero 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;
```

Con todo el fichero completo seria de la siguiente forma:

```
$ONTEXT  
  
    Se trata de la creaci3n de un fichero de datos que permita  
    crear y resolver un problema clasico de maximizaci3n de la utilidad  
    de un consumidor sujeto a una restricci3n 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 funci3n 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 construcci3n 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.
*IDENTIFICACION:

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.

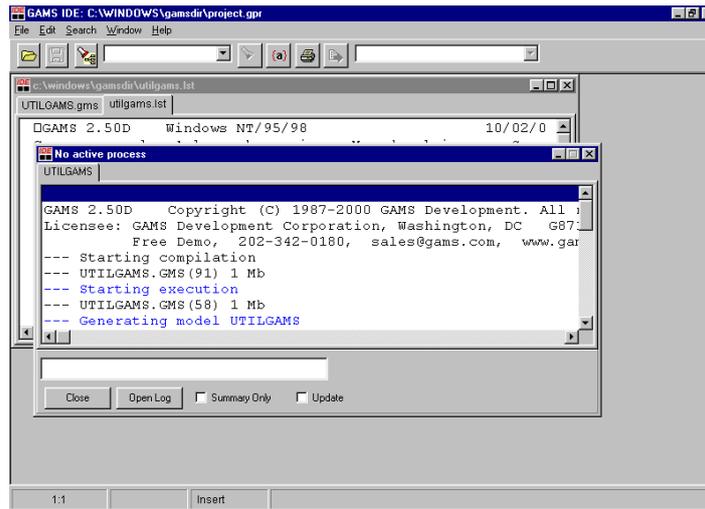
MODEL UTILGAMS/UTIL,RP/;

*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 fichero correspondiente, es necesario resolverlo. Para ello hay que ejecutar el programa GAMS. Para ello 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 fichero LST:

```

GAMS Rev 121 Windows NT/95/98 05/14/01 21:11:22 PAGE 1
General Algebraic Modeling System
Compilation

1

Se trata de la creaci3n de un fichero de datos que permita
crear y resolver un problema clasico de maximizaci3n de la utilidad
de un consumidor sujeto a una restricci3n 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 funci3n 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 construcci3n del problema, y mediante un asterisco *
iremos indicando los comentarios pertinente.

Este bloque de comentarios finalizar con la marca de:

34
35 *PRIMER BLOQUE OBLIGATORIO: BLOQUE DE VARIABLES
36 *IDENTIFICACION:
37
38 VARIABLES

```

```

39
40 * RELACION DE LAS VARIABLES A INCLUIR:
41
42 X      BIEN-1
43 Y      BIEN-2
44
45 * HAY QUE INCLUIR UNA VARIABLE PARA RECOGER EL VALOR DE LA
46 * FUNCION OBJETIVO. CADA GRUPO FINALIZAR CON UN ;
47
48 F;
49
50 *DENTRO DE ESTE MISMO BLOQUE, PODEMOS INCLUIR DOS APARTADOS:
51 *A) DEFINICION DE LA CLASE DE VARIABLES: NO NEGATIVAS X e Y.
52
53 POSITIVE VARIABLES X,Y;
54
55 *B) INDICACION DEL PUNTO DE PARTIDA DEL ALGORITMO.
56
57 X.L = 1.0;

```

```

GAMS Rev 121 Windows NT/95/98                05/14/01 21:11:22 PAGE    2
General Algebraic Modeling System
Compilation

```

```

58 Y.L = 1.0;
59
60 *FINAL DEL BLOQUE DE VARIABLES.
61
62 *SEGUNDO BLOQUE OBLIGATIRIO: ECUACIONES.
63 *IDENTIFICACION:
64
65 EQUATIONS
66
67 * PRIMERA PARTE DEL BLOQUE: DEFINICION DE LAS FUNCIONES:
68
69 UTIL      FUNCION DE UTILIDAD
70 RP        RESTRICCIÓN PRESUPUESTARIA;
71
72 *SEGUNDA PARTE: DECLARACION DE LAS FUNCIONES:
73
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 = 0.000 SECONDS 0.7 Mb WIN200-121

=====

GAMS Rev 121 Windows NT/95/98 05/14/01 21:11:22 PAGE 3  
General Algebraic Modeling System  
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 Rev 121 Windows NT/95/98 05/14/01 21:11:2 PAGE 4  
General Algebraic Modeling System  
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 Rev 121 Windows NT/95/98 05/14/01 21:11:22 PAGE 5  
General Algebraic Modeling System  
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 = 0.050 SECONDS 1.9 Mb WIN200-121

EXECUTION TIME = 0.050 SECONDS 1.9 Mb WIN200-121

=====

GAMS Rev 121 Windows NT/95/98 05/14/01 21:11:22 PAGE 6  
General Algebraic Modeling System

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

MODEL UTILGAMS OBJECTIVE F  
TYPE NLP DIRECTION MAXIMIZE  
SOLVER MINOS5 FROM LINE 90

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

RESOURCE USAGE, LIMIT 0.109 1000.000  
ITERATION COUNT, LIMIT 3 10000  
EVALUATION ERRORS 0 0

MINOS5 Mar 21, 2001 WIN.M5.M5 20.0 108.043.039.WAT

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.

Work space allocated -- 0.04 Mb

EXIT -- OPTIMAL SOLUTION FOUND  
MAJOR ITNS, LIMIT 1 200  
FUNOBJ, FUNCON CALLS 8 0  
SUPERBASICS 1  
INTERPRETER USAGE 0.00  
NORM RG / NORM PI 6.280E-16

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU UTIL	.	.	.	1.000
---- EQU RP	-INF	130.000	130.000	3.000

UTIL FUNCION DE UTILIDAD  
RP RESTRICCIÓN PRESUPUESTARIA

	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
=====
GAMS Rev 121 Windows NT/95/98 05/14/01 21:11:22 PAGE 7
General Algebraic Modeling System

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

EXECUTION TIME      =          0.000 SECONDS      0.7 Mb      WIN200-121

USER: Ramon Sala Garrido          G010426:1108AV-
WIN

**** FILE SUMMARY

INPUT      D:\UTILGAMS.GMS
OUTPUT    C:\UTILGAMS.LST

```

Las primeras paginas, en este caso las paginas 1 y 2, recogen el fichero de datos (UTILGMS.GMS). En el caso que se haya producido errores en el fichero original, 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 fichero de datos original, sin mas que numerar las filas del texto fuera del espacio marcado con ONTEXT-OFFTEXT. Esta parte del fichero 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 fichero de datos, GAMS realiza un análisis de las ecuaciones y de las variables del modelo. Las paginas 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 (sí el problema es de máximo se le cambia el signo) y el valor inicial asociado a las restantes variables.

El contenido de estas paginas es el siguiente:

La pagina 5 informa sobre la estadística del modelo ejecutado, es decir, nos informa sobre el número de variables y ecuaciones, del numero 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 pagina es el siguiente:

De todas las paginas 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 hacer notar, tanto aquí como a lo largo del fichero, que hay que prestar una extraordinaria 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 termino 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 (épsilon) 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 las 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 numero de variables no optimas , infactibles o no acotadas que se han encontrado, así como el numero 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 ficheros 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 creación de los ficheros de datos. 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 fichero de datos, que vamos a denominar: **UTILERR.GMS**. Este fichero es similar al que hemos creado con anterioridad.

El contenido del fichero 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        RESTRICCIÓN 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 fichero de datos con GAMS, nos proporciona el fichero 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 numero precedidos por el símbolo \$.

```
GAMS Rev 121 Windows NT/95/98 05/14/01 21:16:51 PAGE 1
General Algebraic Modeling System
Compilation

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 RESTRICION 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 Rev 121 Windows NT/95/98 05/14/01 21:16:51 PAGE 2
General Algebraic Modeling System
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 = 0.050 SECONDS 0.7 Mb WIN200-121
USER: Ramon Sala Garrido G010426:1108AV-WIN

**** FILE SUMMARY
INPUT D:\UTILERR.GMS
OUTPUT C:\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 - 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 en el fichero, 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 fichero \*.LST, hay que volver a editar el fichero \*.GMS y corregir en este fichero los errores detectados. Una vez corregidos hay que volver a ejecutar el fichero \*.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.

A veces es posible acudir a ejemplo o modelos ya desarrollados, es decir, hacer uso de las librerías de modelos. Por ello vamos a exponer brevemente el uso de las

librerías de modelos

GAMS dispone de la posibilidad de consultar unas librerías de modelos ya contruidos. Las librerías que pueden consultarse son la de GAMS o las que defina el propio usuario.