GAMS, ejemplos introductorios

Héctor Manuel Mora Escobar
Marzo de 2009
`hectormora@yahoo.com`

El programa comercial Gams, General Algebraic Modeling System, es una herramienta de alto nivel para modelamiento y solución de problemas de optimización y programación matemática. Su página es `www.gams.com`. Allí se puede descargar un demo de Gams, la guía del usuario, un tutorial, ...

Su calidad, versatilidad y gran uso han hecho que se convierta en un estándar para la escritura de problemas de optimización. En NEOS, servidor para problemas de optimización, `www-neos.mcs.anl.gov`, la mayoría de los solucionadores ("*solvers*") tienen como uno de los formatos predeterminados el de Gams.

Es posible descargar e instalar Gams sin haber comprado la licencia, pero funciona como un demo de uso libre que tiene restricciones de tamaño. Para la mayoría de los ejemplos académicos es más que suficiente. Muchas gracias a los directivos de Gams. Los límites superiores de tamaño son, entre otros:

- Número de restricciones y variables: 300.
- Número de elementos no nulos: 2000.
- Número de variables discretas: 50.

Gams viene para muchas plataformas, Windows, Linux, Solaris, Mac, ..., 32 y 64 bits. A continuación hay indicaciones someras para los primeros pasos de Gams en Windows y Linux.

0.1 Windows

En Windows, Gams tiene un IDE (ambiente integrado de desarrollo) que permite, entre muchas cosas más, editar (escribir) el archivo y ejecutar Gams. Este archivo donde se escribe el problema tiene extensión `.gms`.

El archivo `.gms` es de tipo ASCII y puede ser escrito con cualquier editor para este tipo de archivos (Emacs, Bloc de notas, ...). El editor del ambiente Gams tiene una gran ventaja, resalta con diferente color las palabras específicas de Gams.

También desde el ambiente Gams se puede activar Gams mediante la tecla F9 o mediante el botón de la barra de menú *Run Gams*.

Gams mira el archivo `.gms` y si está bien escrito resuelve el problema. Gams envía algunos resultados al ambiente y crea un archivo `.lst` donde está la información sobre la solución.

Si en el archivo `.gms` hay errores, entonces en el archivo `.lst` aparece una transcripción del archivo `.gms`, con numeración de los renglones, e inmediatamente después de una línea errónea, aparece algo semejante a

```
****   $409
```

El valor 409 (u otro valor) es un código de error. Un poco más adelante, en el archivo `.lst`, aparece el significado de cada uno de los códigos de los errores ocurridos.

0.2 Linux

En Linux, Gams no viene con ambiente integrado. El archivo `.gms` se puede escribir con cualquier editor de texto (Emacs, vi, Kate, ...). Para invocar Gams, desde una ventana se da la orden

`gams` *archivo*`.gms`

También se puede dar la orden sin explicitar la extensión

`gams` *archivo*

De nuevo, se crea un archivo `.lst` donde está el resultado, bien sea la solución, o bien información sobre los errores de la misma manera que en Windows (ver sección anterior).

0.3 Un archivo explícito de datos

Los ejemplos de modelos en Gams de este documento están muy lejos de ser exhaustivos con el número de temas o con la profundidad utilizada. El propósito es explicar someramente algunos de los conceptos involucrados en un ejemplo. El lector interesado podrá encontrar información mas amplia y precisa en la guía del usuario de Gams.

Consideremos un problema de fabricación de sillas y escritorios, cuyo modelo sea:

$$
\begin{aligned}
\min\ z = -x_1 - 1.4x_2 & \\
x_1 + x_2 &\leq 400 \\
x_1 + 2x_2 &\leq 580 \\
x_1 &\leq 300 \\
x &\geq 0.
\end{aligned}
$$

El archivo de datos en Gams puede ser el siguiente:

```
* problema de OL, formulacion explicita
VARIABLES x1, x2, z;

POSITIVE VARIABLES x1, x2;

EQUATIONS obj, restr1, restr2;

obj..    z =e= -x1 - 1.4*x2;
restr1.. x1 + x2 =l= 400;
restr2.. x1 + 2*x2 =l= 580;
```

```
x1.UP = 300;

MODEL ejemplo /ALL/;
SOLVE ejemplo USING LP MINIMIZING z;
```

Supongamos que el archivo se llama `ej01.gms`. Al utilizar Gams, éste producirá un archivo con los resultados, llamado `ej01.lst`.

- En el archivo anterior las palabras propias de Gams están escritas en mayúsculas simplemente por presentación pero no es necesario.
- Algunas de estas palabras se pueden escribir en singular o plural, `SET` o `SETS`, `VARIABLE`, `EQUATION`, `SCALAR`, `PARAMETER`
- Si en la primera columna de una línea hay un asterisco, ésta se considera como una línea de comentarios.
- Las igualdades y desigualdades en las restricciones y función objetivo se escriben con `=e=` (para $=$), `=l=` (para $\leq$), `=g=` (para $\geq$) .
- Las cotas inferiores y superiores se escriben utilizando `.LO` y `.UP`
- Algunos de los procesos de solución previstos por Gams son:

`lp`	linear programming
`nlp`	nonlinear programming
`mip`	mixed integer programming
`rmip`	relaxed mixed integer programming
`minlp`	mixed integer nonlinear programming
`mcp`	mixed complementarity problems
`cns`	constrained nonlinear systems

- Si en el archivo de resultados, en el sitio donde debería aparecer un valor, vemos `.` , esto quiere decir que el valor es `0.0`.
- En la definición de las cotas no se usa `=e=`, se usa simplemente `=`.

Las variables no tienen que llamarse `x1`, `x2`, etc. El nombre de una variable (el identificador) puede tener hasta 63 caracteres alfanuméricos (letras o dígitos) sin espacios. Aparentemente también se puede utilizar el guión bajo (`_`) pero no puede ser el primer caracter del identificador.

```
$ONTEXT
Problema de OL.
Estos son renglones de comentarios.
$OFFTEXT

VARIABLES Sillas, escrit, z;
```

```
POSITIVE VARIABLES /ALL/;

EQUATIONS obj, madera, m_obra;

obj.. z =e= sillas + 1.4*escrit;
madera.. sillAs + escrit =l= 400;
m_obra..  sillas + 2*escrit =l= 580;
sillas.UP = 300;

MODEL ejemplo /obj, madera, m_obra/;
SOLVE ejemplo USING LP MAXIMIZING z;
```

Observaciones:

- Gams no diferencia entre minúsculas y mayúsculas, pero, por ejemplo, el nombre "principal" de la primera variable será `Sillas` (el primero que aparece). Las otras escrituras equivalentes se le asimilarán.
- `$ONTEXT` y `$OFFTEXT` sirven para empiezo y fin de comentarios. Estas dos palabras deben empezar en la primera columna.
- Cuando se define el modelo `ejemplo`, es posible considerar todas las ecuaciones con `ALL` o explicitarlas todas una por una o solamente algunas de ellas.

0.4 Estructura de un modelo en Gams

Un modelo en Gams es una sucesión de comandos o enunciados ("*statement*") en lenguaje Gams. El orden no importa pero, cualquier entidad (variable, parámetro, ...) debe ser declarada antes de ser usada.

Es posible tener renglones en blanco y varios comandos en el mismo renglón. Usualmente cada comando acaba con punto y coma.

En el archivo de entrada `.gms` los comandos son:

- `SETS`
- Datos: `PARAMETERS`, `TABLES`, `SCALARS`.
- `VARIABLES` : declaración y asignación de tipo.
- Asignación de cotas [y valores iniciales].
- `EQUATIONS`: declaración y definición.
- `MODEL` y `SOLVE`.
- `DISPLAY` (opcional).

En los modelos que siguen hay varios ejemplos de los comandos.

0.5 Un problema de transporte

Consideremos un problema clásico de transporte con 3 fábricas y 4 destinos. En la siguiente tabla están los costos unitarios de transporte (en miles de pesos), las capacidades máximas de producción de cada fábrica y los pedidos o demandas de cada destino.

	1	2	3	4	cap. max.
Macondo	23	29	19	31	200
Cali	12	16	20	10	180
Faca	11	13	17	19	100
demanda	105	80	99	135	

Usualmente el planteamiento de este problema se hace con las variables x_{ij}, $i = 1, ..., 3$, $j = 1, ..., 4$ que indican en número de unidades que van de la fábrica i al destino j. Denotemos por c_{ij} los costos unitarios de transporte, d_j las demandas en los destinos y p_i las capacidades máximas de producción en las fábricas. El modelo matemático es:

$$
\begin{aligned}
\min\ z = \sum_{i=1}^{3}\sum_{j=1}^{4} c_{ij}x_{ij}& \\
\sum_{j=1}^{4} x_{ij} \leq p_i, &\quad i = 1, ..., 3 \\
\sum_{i=1}^{3} x_{ij} = d_j, &\quad j = 1, ..., 4 \\
x_{ij} \geq 0 &\quad \forall i, j\ .
\end{aligned}
$$

A continuación están dos formulaciones del problema en Gams. La primera formulación es más explícita y no es estructurada.

```
$ONTEXT
  Problema de transporte,
  formulacion NO estructurada.
$OFFTEXT

VARIABLES x11, x12, x13, x14, x21, x22, x23, x24, x31, x32,
    x33, x44, z;

POSITIVE VARIABLES x11, x12, x13, x14, x21, x22, x23, x24,
    x31, x32, x33, x34;

EQUATIONS obj, f1, f2, f3, d1, d2, d3, d4;

obj.. z =e=  1000*( 23*x11 + 29*x12 + 19*x13 + 31*x14 +
```

```
  12*x21 + 16*x22 + 20*x23 + 10*x24 +
  11*x31 + 13*x32 + 17*x33 + 19*x34 );

f1.. x11 + x12 + x13 + x14 =l= 200;
f2.. x21 + x22 + x23 + x24 =l= 180;
f3.. x31 + x32 + x33 + x34 =l= 100;

d1.. x11 + x21 + x31 =e= 105;
d2.. x12 + x22 + x32 =e=  80;
d3.. x13 + x23 + x33 =e=  99;
d4.. x14 + x24 + x34 =e= 135;

MODEL transporte0 /ALL/;
SOLVE transporte0 USING LP MINIMIZING z;
```

Esta segunda formulación aprovecha las facilidades algebraicas de Gams.

```
$ONTEXT
  Problema de transporte,
  formulacion estructurada.
$OFFTEXT

SETS

  i fabricas u origenes /Macondo, Cali, Faca/
  j destinos            /1*4/;

PARAMETERS

  p(i) capacidad de produccion de las fabricas
    / Macondo  200
      Cali     180
      Faca     100 /

  d(j) demanda en los destinos
    /  1  105
       2   80
       3   99
       4  135 /;

TABLE

  c(i,j) costo unitario entre una fabrica y un destino
          1    2    3    4
Macondo   23   29   19   31
Cali      12   16   20   10
Faca      11   13   17   19  ;
```

```
SCALAR k coeficiente para costos unitarios /1000/;

VARIABLES
  x(i,j) numero de unidades de una fabrica a un destino
  z      costo total de transporte;

POSITIVE VARIABLES x;

EQUATIONS
  costo         costo total de transporte
  capacidad(i)  capacidad en cada fabrica
  demanda(j)    demanda en cada destino;

costo..        z =e= k*sum( (i,j), c(i,j)*x(i,j) );
capacidad(i).. sum(j, x(i,j) ) =l= p(i);
demanda(j)..   sum(i, x(i,j) ) =e= d(j);

MODEL transporte /ALL/;

SOLVE transporte USING LP MINIMIZING z;

DISPLAY x.l, x.m;
```

- En la orden `display`, para mostrar resultados, `x.l` se refiere al valor ("*level*") de las variables `x`; `x.m` se refiere al costo reducido o valor marginal de las variables `x`.
- `PARAMETER` sirve para definir arreglos en una dimensión (vectores). `TABLE` sirve para definir arreglos de dos o más dimensiones (matrices, ...).
- Es permitido, después de un identificador (conjunto, variable, ecuación, parámetro, tabla, ...), colocar texto explicativo. Por ejemplo,

  ```
  p(i) capacidad de produccion de las fabricas
  ```

  El nombre del parámetro es `p(i)`. Lo que sigue es un texto explicativo. No debe tener más de 254 caracteres. No debe tener tildes ni `/` ni comas ni símbolos raros, o sea, preferiblemente debe tener únicamente letras, dígitos y espacios. Por ejemplo, no es correcto escribir `toneladas/mes`, es preferible ton por mes. Sin embargo, es posible tener un texto explicativo más complejo si se coloca todo entre comillas sencillas o dobles. Por ejemplo,

  ```
  cp(i) 'capacidad de produccion en ton/mes'
  ```

0.6 Un problema de vigilantes

Una compañía de vigilancia evaluó sus necesidades de vigilantes, por periodos de 4 horas, en un gran conjunto residencial, de la siguiente manera:

Periodo			Cantidad
2 a.m.	a	6 a.m	84
6 a.m.	a	10 a.m	48
10 a.m.	a	2 p.m	37
2 p.m.	a	6 p.m	35
6 p.m.	a	10 p.m	32
10 p.m.	a	2 a.m	30

Cada vigilante trabaja 8 horas al día, pero de manera continua. La compañía desea organizar la distribución de sus vigilantes de tal forma que el número total de vigilantes sea mínimo. Plantee el anterior problema de OL.

Variables:
x_1 el número de vigilantes que empiezan su turno a las 2,
x_2 el número de vigilantes que empiezan su turno a las 6, ...,
x_6 el número de vigilantes que empiezan su turno a las 22,

Para hacer más compacto el planteamiento, denotemos por v_1 (un dato) el número mínimo de vigilantes necesarios en el primer periodo (2 a 6), ..., v_6 el número mínimo de vigilantes necesarios en el sexto periodo (22 a 2).

$$\begin{aligned} \min\ z = \sum_{i=1}^{6} x_i & \\ x_1 + x_6 &\geq v_1, \\ x_i + x_{i-1} &\geq v_i, \quad i = 2, ..., 6, \\ x &\geq 0. \end{aligned}$$

El problema se puede escribir en Gams así:

```
$ONTEXT
  Problema de vigilantes,
  formulacion estructurada.
  uso del simbolo pesos  y  ord
$OFFTEXT

SETS
  i  periodos  /2-6, 6-10, 10-14, 14-18, 18-22, 22-2/;

PARAMETERS
  v(i) necesidad de vigilantes
  / 2-6    84
    6-10   48
    10-14  37
    14-18  35
    18-22  32
    22-2   30 /;
```

```
VARIABLES
  x(i) numero de vigilantes que empiezan su trabajo
*      en el periodo  i
  ntv  numero total de vigilantes;

POSITIVE VARIABLES x;

EQUATIONS objetivo, numVig1, numVigi(i);

objetivo..  ntv =e= sum(i, x(i) );

numVig1..     x('2-6') + x('22-2') =g= v('2-6');
numVigi(i)$(ord(i) gt 1 ).. x(i) + x(i-1) =g= v(i);

MODEL vigilantes /ALL/;
SOLVE vigilantes USING LP MINIMIZING ntv;

DISPLAY x.l;
```

La solución es

```
----     37 VARIABLE x.L  numero de vigilantes que empiezan su trabajo

2-6   84.000,    6-10   4.000,    10-14 33.000,    14-18  2.000,    18-22 30.000
```

El símbolo $, que se puede leer como *tales que*, permite tener restricciones u operaciones para elementos de un conjunto que cumplen cierta condición. En el ejemplo anterior, la restricción `numVigi(i)` se define únicamente para los elementos del conjunto `i` tales que su ordinal es mayor que 1, es decir, para los elementos de ordinal 2, ..., 5.

También se puede hacer usando el símbolo `--` que considera un conjunto como una lista circular (cuando se resta 1 al primer elemento, se toma el último):

```
$ONTEXT
  Problema de vigilantes,
  formulacion estructurada,
  uso de --
  FUNCIONA BIEN
$OFFTEXT

SETS
  i  periodos  /2-6, 6-10, 10-14, 14-18, 18-22, 22-2/;

PARAMETERS
  v(i) necesidad de vigilantes
  / 2-6    84
    6-10   48
```

```
    10-14  37
    14-18  35
    18-22  32
    22-2   30 /;

VARIABLES
  x(i) numero de vigilantes que empiezan su trabajo
*      en el periodo  i
  ntv  numero total de vigilantes;

POSITIVE VARIABLES x;

EQUATIONS objetivo, numVig(i);

objetivo..  ntv =e= sum(i, x(i) );

numVig(i).. x(i) + x(i--1) =g= v(i);

MODEL vigilantes /ALL/;
SOLVE vigilantes USING LP MINIMIZING ntv;

DISPLAY x.l;
```

0.7 Un problema de dieta

Un ama de casa desea hacer un almuerzo equilibrado utilizando los siguientes productos: carne, papas, habichuela, leche y guayaba. Los precios por kilo de estos alimentos son respectivamente: \$700, \$80, \$250, \$70 y \$80. Aquí estamos suponiendo que la leche se vende por kilos, o lo que es aproximadamente lo mismo, que un litro de leche pesa un kilo. La familia está compuesta por 6 personas y cada persona debe consumir 800 calorías (en el almuerzo). Para que la alimentación sea equilibrada debe estar compuesta, idealmente, de 25% de proteínas, 25% de grasas, 50% de glúcidos o carbohidratos. En la práctica, los porcentajes reales no deben diferir en más de 5% de los porcentajes ideales. Estos porcentajes están dados con respecto a la materia seca, es decir, sin tener en cuenta el agua contenida en los alimentos. Obviamente, hay muchas más condiciones que se deben tener en cuenta y aquí se hace una simplificación para facilitar el planteamiento del problema. En la siguiente tabla se expresa la composición de cada alimento y su aporte calórico. Se supone que fuera de proteina, grasa y carbohidratos, solamente hay agua con el porcentaje restante.

	% Proteínas	% Grasas	% Glúcidos	Calorías por kilo
Carne	10	10	0	1300
Papas	2	0	20	880
Habichuelas	1	0	5	240
Leche	5	3	5	670
Guayaba	1	0	15	640

El ama de casa desea saber cómo organizar su mercado de tal forma que se cumplan las restricciones nutricionales y que, además, se minimice el costo.

Las variables pueden ser: x_i: cantidad de kilos del alimento i que hay que comprar para el almuerzo, $i = 1, ..., 5$

Para facilitar el planteamiento, introduzcamos unos nombres, unos valores intermedios y una variable adicional:

c_i = costo de un kilo del alimentos i,
r_j = requerimiento ideal, en porcentaje, del componente $j = 1, ..., 3$,
$u_j = r_j - 5$ = requerimiento mínimo, en porcentaje, del componente j,
$v_j = r_j + 5$ = requerimiento máximo, en porcentaje, del componente j,
p_{ij} = porcentaje, en el alimento i, del componente j,
a_i = aporte calórico de un kilo del alimento i,
$s_i = \sum_{j=1}^{3} p_{ij}$ = porcentaje de materia seca en el alimento i,
y = cantidad (kilos) de materia seca de los alimentos comprados.

$$\begin{aligned}
\min\ z = \sum_{i=1}^{5} c_i x_i& \\
\sum_{i=1}^{5} a_i x_i &= 6 \times 800 \\
\sum_{i=1}^{5} 0.01 s_i x_i &= y \\
\sum_{i=1}^{5} 0.01 p_{ij} x_i &\geq 0.01 u_j y, \quad j = 1, ..., 3, \\
\sum_{i=1}^{5} 0.01 p_{ij} x_i &\leq 0.01 v_j y, \quad j = 1, ..., 3, \\
x &\geq 0
\end{aligned}$$

Este modelo se puede escribir enGams así:

```
$ONTEXT
  Problema de dieta,
  formulacion estructurada,
$OFFTEXT

SETS
  i alimentos   /carn, papa, habi, lech, guay/
  j componentes /prot, gras, carb/;

VARIABLES
  x(i) peso en kilos del alimento i
  y    peso de la materia seca en kilos
```

```
  z    costo del almuerzo;

POSITIVE VARIABLES x, y;

TABLE p(i,j) porcentaje en el alimento i del componente j
           prot  gras   carb
  carn     10    10      0
  papa      2     0     20
  habi      1     0      5
  lech      5     3      5
  guay      1     0     15   ;

PARAMETER
  r(j) requerimientos ideales de cada componente en porcentaje
  / prot   25
    gras   25
    carb   50 /

  c(i) precio de un kilo de cada alimento
  / carn     700
    papa      80
    habi     250
    lech      70
    guay      80 /

  ac(i)  aporte calorico de un kilo de cada alimento
  / carn    1300
    papa     880
    habi     240
    lech     670
    guay     640 /;

SCALAR
  k     coeficiente para convertir a fraccion /0.01/
  n     numero de personas /6/
  calp  calorias por persona /800/
  delta diferencia permitida en requerimientos /5/;


PARAMETER  pms(i)  porcentaje de materia seca de cada alimento;
  pms(i) = sum(j, p(i,j)) ;

PARAMETER  u(j)  cota inferior para el porcentaje de j;
  u(j) = r(j) - delta;

PARAMETER  v(j)  cota superior para el porcentaje de j;
  v(j) = r(j) + delta;
```

```
DISPLAY pms, u, v;

EQUATIONS
  obj       funcion objetivo
  peso_ms   peso materia seca
  calorias  total de calorias
  reqmin(j) requerimiento minimo
  reqmax(j) requerimiento maximo;

obj..       z =e= sum(i, c(i)*x(i));

peso_ms..   y =e= k*sum(i, pms(i)*x(i));

calorias..  sum(i, ac(i)*x(i) ) =e= n*calp;

reqmin(j).. k*sum(i, p(i,j)*x(i) ) =g= k*u(j)*y;
reqmax(j).. k*sum(i, p(i,j)*x(i) ) =l= k*v(j)*y;

MODEL dieta /ALL/;
SOLVE dieta USING LP MINIMIZING z;
DISPLAY x.l;
```

La solución es

$$x = (0.768,\ 0,\ 0,\ 3.84,\ 1.92), \quad z = 960.$$

0.8 Optimización entera

Consideremos el siguiente problema

$$\begin{array}{rl} \max & z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ & 6.7x_1 + 10x_2 + 5.5x_3 + 3.4x_4 \leq 19 \\ & x_j \in \{0,1\}\,, \quad j = 1, ..., 4. \end{array} \tag{1}$$

Al resolver la siguiente relajación lineal

$$\begin{array}{rl} \max & z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ & 6.7x_1 + 10x_2 + 5.5x_3 + 3.4x_4 \leq 19 \\ & x \geq 0\,, \end{array}$$

escrita en Gams así:

```
$ONTEXT
  Ejemplo de Cornuejols, pag. 193
```

```
  sin considerar integralidad de variables
  ni cotas superiores
$OFFTEXT

VARIABLES x1, x2, x3, x4, z;
POSITIVE VARIABLES x1, x2, x3, x4;
EQUATIONS obj, restr;
obj..  z =e= 8*x1 + 11*x2 + 6*x3 + 4*x4;
restr.. 6.7*x1 + 10*x2 + 5.5*x3 + 3.4*x4 =l= 19;
MODEL pag193 /ALL/;
SOLVE pag193 USING LP MAXIMIZING z;
```

La solución obtenida es

$$x = (\,2.8358,\ 0,\ 0,\ 0\,), \quad z = 22.6866\,,$$

muy diferente de lo esperado. Coloquemos ahora cotas superiores:

```
$ONTEXT
  Ejemplo de Cornuejols, pag. 193
  sin considerar integralidad de variables.
$OFFTEXT

VARIABLES x1, x2, x3, x4, z;
POSITIVE VARIABLES x1, x2, x3, x4;
EQUATIONS obj, restr;
obj..  z =e= 8*x1 + 11*x2 + 6*x3 + 4*x4;
restr.. 6.7*x1 + 10*x2 + 5.5*x3 + 3.4*x4 =l= 19;
x1.UP = 1;
x2.UP = 1;
x3.UP = 1;
x4.UP = 1;
MODEL pag193 /ALL/;
SOLVE pag193 USING LP MAXIMIZING z;
```

Se obtiene

$$x = (\,1,\ 0.89,\ 0,\ 1\,), \quad z = 21.7900\,,$$

que casi cumple con las restricciones de (1). Finalmente al escribir en Gams la información sobre integralidad de las variables y diciendo que se trata de un problema MIP (mixed integer programming),

```
$ONTEXT
  ejemplo de Cornuejols, pag. 193
  con variables enteras.
$OFFTEXT
```

```
VARIABLES x1, x2, x3, x4, z;
INTEGER VARIABLES x1, x2, x3, x4;
EQUATIONS obj, restr;
obj..  z =e= 8*x1 + 11*x2 + 6*x3 + 4*x4;
restr.. 6.7*x1 + 10*x2 + 5.5*x3 + 3.4*x4 =l= 19;
x1.UP = 1;
x2.UP = 1;
x3.UP = 1;
x4.UP = 1;
MODEL pag193 /ALL/;
SOLVE pag193 USING MIP MAXIMIZING z;
```

se obtiene

$$x = (\,0,\ 1,\ 1,\ 1), \quad z = 20.$$

El problema (1) no solamente es de optimización entera, también es un problema de variables binarias. Entonces en Gams se puede escribir más corto

```
$ONTEXT
  ejemplo de Cornuejols, pag. 193
  con variables binarias.
$OFFTEXT

VARIABLES x1, x2, x3, x4, z;
BINARY VARIABLES x1, x2, x3, x4;
EQUATIONS obj, restr;
obj..  z =e= 8*x1 + 11*x2 + 6*x3 + 4*x4;
restr.. 6.7*x1 + 10*x2 + 5.5*x3 + 3.4*x4 =l= 19;
MODEL pag193 /ALL/;
SOLVE pag193 USING MIP MAXIMIZING z;
```

En los problemas de optimización entera es muy importante tener en cuenta un valor producido por Gams, se trata de `Relative gap`. Normalmente Gams no realiza un estudio exhaustivo del árbol (del método de ramificación y acotamiento). Sea δ el valor absoluto de la diferencia relativa entre el mejor valor de z obtenido para un punto entero y el valor estimado del mejor valor posible de z. Gams se detiene cuando δ es menor que un valor predeterminado. Si en los resultados `Relative gap` es cero, todo funcionó bien y el valor de x y z producidos por Gams son óptimos. Cuando el valor no es cero, no hay seguridad sobre el resultado obtenido. En el último ejemplo de Gams el valor de `Relative gap` es cero y se obtuvo la solución.

Consideremos un problema parecido a (1):

$$\begin{aligned} \max \quad z = 8x_1 + 11x_2 + 5x_3 + 4x_4 \\ 6.7x_1 + 10x_2 + 5.5x_3 + 3.4x_4 \le 19 \\ x \in \{0,1\}^4. \end{aligned} \tag{2}$$

Al resolverlo con

```
VARIABLES x1, x2, x3, x4, z;
BINARY VARIABLES x1, x2, x3, x4;
EQUATIONS obj, restr;
obj..  z =e= 8*x1 + 11*x2 + 5*x3 + 4*x4;
restr.. 6.7*x1 + 10*x2 + 5.5*x3 + 3.4*x4 =l= 19;
MODEL pag193 /ALL/;
SOLVE pag193 USING MIP MAXIMIZING z;
```

se obtiene

$$x = (\,1,\ 1,\ 0,\ 0), \quad z = 19,$$

pero

```
Best possible:          20.000000
Absolute gap:            1.000000
Relative gap:            0.052632
```

El valor $0.052632 = 1/(20-1)$ deja dudas sobre el x obtenido. Entonces es mejor redefinir el parámetro `OPTCR` con un valor pequeño:

```
$ONTEXT
  Optimizacion con variables binarias.
  y definiendo OPTCR
$OFFTEXT

OPTION OPTCR = 0.0001;

VARIABLES x1, x2, x3, x4, z;
BINARY VARIABLES x1, x2, x3, x4;
EQUATIONS obj, restr;
obj..  z =e= 8*x1 + 11*x2 + 5*x3 + 4*x4;
restr.. 6.7*x1 + 10*x2 + 5.5*x3 + 3.4*x4 =l= 19;
MODEL pag193b /ALL/;
SOLVE pag193b USING MIP MAXIMIZING z;
```

Así se obtiene

$$x = (\,1,\ 1,\ 1,\ 0), \quad z = 20,$$

solución óptima ya que el valor de `Relative gap` es cero.

0.9 Un ejemplo de optimización estocástica

Tomado de [BiL97]. Un granjero cultiva trigo, maíz y remolacha en un terreno de 500 acres. El desea saber, en el invierno, cuantos acres dedicar a cada cultivo el año que viene. Para sus animales el necesita por lo menos 200 T (toneladas) de trigo y 240 T de maíz. Si su producción no es suficiente, el puede comprar cada tonelada a US$238 y US$210 respectivamente. Si sobra de su producción, el puede vender a US$170 y US$150 cada tonelada. Cada tonelada de remolacha la puede vender a US$36 pero hasta un máximo de 600 T. Por encima de esa cantidad, cada tonelada la puede vender a $10. El costo total de cultivar un acre es de US$150, US$230 y US$260 para trigo, maíz y remolacha.

En un planteamiento determinista se conoce el rendimiento de cada cultivo en toneladas por acre: 2.5, 3 y 20 para trigo, maíz y remolacha respectivamente. ¿Cuántos acres debe dedicar el granjero a cada cultivo para maximizar su beneficio?

Para facilitar el planteamiento y hacerlo más general, denotemos los datos de la siguiente manera:

$$
\begin{aligned}
s_i = \quad & \text{costo total de sembrar un acre con el cultivo } i = 1, 2, 3, \\
v_i = \quad & \text{precio de venta de una tonelada del cultivo } i = 1, 2, 3, \\
\bar{v}^r = \quad & \text{segundo precio venta para una tonelada de remolacha}, \\
c_i = \quad & \text{precio de compra de una tonelada del cultivo } i = 1, 2, \\
d_i = \quad & \text{necesidad o demanda, en T, del cultivo } i = 1, 2, \\
r_i = \quad & \text{rendimiento, en toneladas por acre, del cultivo } i.
\end{aligned}
$$

El problema se puede plantear únicamente con las variables relativas al número de acres por cultivo, pero puede ser más claro si se introducen otras variables intermedias:

$$
\begin{aligned}
x_i = \quad & \text{número de acres dedicados al cultivo } i = 1, 2, 3, \\
y_i = \quad & \text{número de toneladas compradas del cultivo } i = 1, 2, \\
w_i = \quad & \text{número de toneladas vendidas del cultivo } i = 1, 2, 3, \\
\bar{w}^r = \quad & \text{número de toneladas de remolacha vendidas al segundo precio}, \\
p_i = \quad & \text{número toneladas producidas del cultivo } i = 1, 2, 3, \\
z^i = \quad & \text{ingreso por ventas de los tres cultivos}, \\
z^g = \quad & \text{gastos por siembra y compras}
\end{aligned}
$$

$$
\begin{aligned}
\max \ z = z^i - z^g & \\
z^i &= \sum_{i=1}^{3} v_i w_i + \bar{v}^r \bar{w}^r \\
z^g &= \sum_{i=1}^{3} s_i x_i + \sum_{i=1}^{2} c_i y_i \\
\sum_{i=1}^{3} x_i &\le 500 \\
p_i &= r_i x_i, \quad i = 1, ...3, \\
p_i + y_i - w_i &= d_i, \quad i = 1, 2 \\
p_3 - w_3 - \bar{w}^r &= 0 \\
w_3 &\le 600 \\
x, y, w, \bar{w}^r &\ge 0
\end{aligned}
$$

En Gams, se puede esribir así:

```
$ONTEXT
  ejemplo del granjero,
  Birge Louveaux, p. 4
  modelo determinista
$OFFTEXT

SETS
  i    clase de cultivo  /tr, mz, rm/
  j(i) granos            /tr, mz/;

PARAMETERS
  r(i) "rendimiento del cultivo en ton/acre"
    /  tr  2.5
       mz  3
       rm  20 /

  s(i) costo de siembra en dolares por ton
    /  tr  150
       mz  230
       rm  260 /

  v(i) precio de venta de cultivos en dolares por ton
    /  tr  170
       mz  150
       rm  36  /

  c(j) precio de compra de granos en dolares por ton
    /  tr  238
       mz  210/

  d(j) requerimiento o demanda de granos en toneladas
    /  tr  200
       mz  240  /;

SCALARS
  at    area total disponible en acres /500/
  tmax  tonelaje maximo de remolachas para primer precio /6000/
  pr2   segundo precio de venta de la remolacha /10/  ;

VARIABLES
  x(i)  numero de acres sembrados con el cultivo  i
  p(i)  produccion en toneladas del cultivo  i
  y(j)  numero de ton compradas de grano  j
  w(i)  numero de ton vendidas del cultivo i
  wr2   num de ton de rem vendidas al segundo precio
  egr   gastos o egresos
  ing   ingresos
```

```
  z     beneficio total;

POSITIVE VARIABLES
  x, y, w, wr2;

EQUATIONS
  benef
  gastos
  ingresos
  requer(j) requerimiento del cultivo j
  produc(i) produccion del cultivo j
  area
  remo  ;

benef..     z =e= ing - egr;
gastos..    egr =e= sum(i, s(i)*x(i) ) + sum(j, c(j)*y(j) );
ingresos..  ing =e= sum(i, v(i)*w(i) ) + pr2*wr2;
produc(i).. p(i) =e= r(i)*x(i);
requer(j).. p(j) + y(j) - w(j) =g= d(j);
area..      sum(i,x(i)) =l= at;
remo..      w('rm') + wr2 =e= p('rm');

w.UP('rm') = tmax;

MODEL granjero /ALL/;
SOLVE granjero USING LP MAXIMIZING z;
DISPLAY x.l, p.l, y.l, w.l, wr2.l, z.l;
```

El resultado es:

```
----     79 VARIABLE x.L  numero de acres sembrados con el cultivo i

tr 120.000,    mz  80.000,    rm 300.000


----     79 VARIABLE p.L  produccion en toneladas del cultivo  i

tr  300.000,    mz  240.000,    rm 6000.000


----     79 VARIABLE y.L  numero de ton compradas de grano j

                      ( ALL       0.000 )


----     79 VARIABLE w.L  numero de ton vendidas del cultivo i

tr  100.000,    rm 6000.000


----     79 VARIABLE wr2.L                 =        0.000  num de ton de rem vendidas al segundo precio

            VARIABLE z.L                   =   118600.000  beneficio total
```

En el modelo estocástico los rendimientos no son deterministas. El granjero prevee tres posibles escenarios en función del tiempo de los últimos tres meses anteriores a la cosecha: rendimiento bajo, medio y alto:

	alto	medio	bajo
trigo	3	2.5	2
maíz	3.6	3	2.4
remolacha	24	20	16

Estos valores los podemos denotar r_{ie} = rendimiento del cultivo i en el escenario e.

El granjero cree que cada uno de los tres escenarios tiene una probabilidad de 1/3. Sean π_i los valores de la tres probabilidades. Ahora se desea maximizar el valor esperado del beneficio.

Es cómodo plantear el problema con variables semejantes a las anteriores, pero todas, salvo las x_i, tendrán un subíndice adicional correspondiente al escenario: $y_{ie},\ w_{ie},\ \bar{w}^r_e,\ p_{ie},\ z^i_e,\ z^g_e,\ z_e$.

$$\max\ \zeta = \sum_{e=1}^{3} \pi_e z_e$$

$$\begin{aligned}
z_e &= z^i_e - z^g_e \\
z^i_e &= \sum_{i=1}^{3} v_i w_{ie} + \bar{v}^r \bar{w}^r_e\,, \quad e = 1, ..., 3 \\
z^g_e &= \sum_{i=1}^{3} s_i x_i + \sum_{i=1}^{2} c_i y_{ie}\,, \quad e = 1, ..., 3 \\
\sum_{i=1}^{3} x_i &\leq 500 \\
p_{ie} &= r_{ie} x_i, \quad i = 1, ...3, \ , \quad e = 1, ..., 3 \\
p_{ie} + y_{ie} - w_{ie} &= d_i, \quad i = 1, 2\,, \quad e = 1, ..., 3 \\
p_{3e} - w_{3e} - \bar{w}^r_e &= 0\,, \quad e = 1, ..., 3 \\
w_{3e} &\leq 600\,, \quad e = 1, ..., 3 \\
x, y, w, \bar{w}^r &\geq 0
\end{aligned}$$

En Gams:

```
$ONTEXT
  ejemplo del granjero,
  Birge Louveaux, p. 4
  modelo estocastico
  representacion de escenarios
$OFFTEXT

SETS
```

```
  i    clase de cultivo  /tr, mz, rm/
  j(i) granos            /tr, mz/
  e    escenarios para el rendimiento/ alto, medio, bajo/;

TABLE  r(i,e) rendimiento del cultivo  segun escenario
      alto medio bajo
tr    3    2.5   2
mz    3.6  3     2.4
rm    24   20    16  ;

PARAMETERS
  s(i) costo de siembra en dolares por ton
    /  tr  150
       mz  230
       rm  260 /

  v(i) precio de venta de cultivos en dolares por ton
    /  tr  170
       mz  150
       rm  36  /

  c(j) precio de compra de granos en dolares por ton
    /  tr  238
       mz  210/

  d(j) requerimiento o demanda de granos en toneladas
    /  tr  200
       mz  240  /

  pr(e) probabilidad de escenarios
    /  alto   0.333333
       medio  0.333333
       bajo   0.333333 /;
*    /  alto   0
*       medio  1
*       bajo   0 /;

SCALARS
  at    area total disponible en acres /500/
  tmax  tonelaje maximo de remolachas para primer precio /6000/
  pr2   segundo precio de venta de la remolacha /10/  ;

VARIABLES
  x(i)    numero de acres sembrados con el cultivo i
  p(i,e)  produccion del cultivo  i en el escenario  e
  y(j,e)  num de ton compradas de grano j en el escenario  e
  w(i,e)  num de ton vendidas del cultivo i en el escenario  e
```

```
  wr2(e)  num de ton de rem vendidas al sdo precio en el escenario  e
  egr(e)  gastos o egresos en el escenario  e
  ing(e)  ingresos en el escenario  e
  z(e)    beneficio en un escenario
  zz      beneficio esperado;

POSITIVE VARIABLES
  x, y, w, wr2;

EQUATIONS
      ben_esp
      ben_esc(e)  beneficio en un escenario
      gastos(e)
      ingresos(e)
      requer(j,e) requerimiento del cultivo j en un escenario
      produc(i,e) produccion del cultivo i
      area
      remo(e)  ;

ben_esp..     zz =e= sum(e, pr(e)*z(e) );
ben_esc(e)..  z(e) =e= ing(e) - egr(e);
gastos(e)..   egr(e) =e= sum(i, s(i)*x(i) ) + sum(j, c(j)*y(j,e) );
ingresos(e).. ing(e) =e= sum(i, v(i)*w(i,e) ) + pr2*wr2(e);
produc(i,e).. p(i,e) =e= r(i,e)*x(i);
requer(j,e).. p(j,e) + y(j,e) - w(j,e) =g= d(j);
area..        sum(i,x(i)) =l= at;
remo(e)..     w('rm',e) + wr2(e) =e= p('rm',e);

w.UP('rm',e) = tmax;

MODEL granjero /ALL/;
SOLVE granjero USING LP MAXIMIZING zz;
DISPLAY x.l, p.l, y.l, w.l, wr2.l, z.l, zz.l;
```

Resultados:

```
----     88 VARIABLE x.L  numero de acres sembrados con el cultivo i

tr 170.000,    mz  80.000,    rm 250.000


----     88 VARIABLE p.L  produccion del cultivo  i en el escenario  e

          alto       medio        bajo

tr     510.000     425.000     340.000
mz     288.000     240.000     192.000
rm    6000.000    5000.000    4000.000


----     88 VARIABLE y.L  num de ton compradas de grano j en el escenario  e

          bajo
```

```
mz      48.000


----     88 VARIABLE w.L  num de ton vendidas del cultivo i en el escenario  e

          alto       medio        bajo

tr     310.000     225.000     140.000
mz      48.000
rm    6000.000    5000.000    4000.000


----     88 VARIABLE wr2.L  num de ton de rem vendidas al sdo precio en el escenario  e

                     ( ALL       0.000 )


----     88 VARIABLE z.L  beneficio en un escenario

alto  167000.000,    medio 109350.000,    bajo   48820.000


----     88 VARIABLE zz.L                 =   108389.892  beneficio esperado
```

0.10 El problema lockbox

Tomado de [CoT07]. Una compañía de nivel nacional recibe cheques de todo el país. Por el correo mismo y por los trámites bancarios transcurre un tiempo (en días) entre el momento en que el cheque es recibido en la oficina de correo y el momento en que la compañía recibe en su cuenta el valor del cheque. Este tiempo de trámite varía dependiendo de las ciudades. La compañía desea abrir algunas oficinas ("lockboxes") para centralizar la recepción de cheques. Supongamos que hay cuatro regiones R_1, R_2, R_3 y R_4 y que habría oficinas centrales en cuatro ciudades C_1, C_2, C_3 y C_4. Los tiempos de trámite entre las regiones y las ciudades son:

	Ciudad 1	Ciudad 2	Ciudad 3	Ciudad 4	\$ 10^6
Región 1	2	4	6	6	1200
Región 2	4	2	5	5	480
Región 3	6	5	2	5	1440
Región 4	7	5	6	3	720

En la última columna aparece el promedio diario (en millones de pesos) de la suma total proveniente de cada una de las cuatro regiones. El manejo y administración anual de cada central es de 180 millones de pesos.

Supongamos por facilidad que, para calcular la pérdida de interés por la demora en el cheque, la tasa nominal es de 5% anual y que para estas cuentas todos los día del año son hábiles. Supongamos además que como el número de días es pequeño se puede usar la fórmula de interés simple.

Los cheques de una región se envían a una sola ciudad pero es posible enviar los cheques de dos regiones diferentes a la misma central. Se desea saber qué centrales abrir y a qué central se deben enviar los cheques de cada región.

Consideremos las siguientes variables, todas binarias:

- $y_j = 1$ si se abre la oficina central en la ciudad j; $y_j = 0$ en caso contrario.
- $x_{ij} = 1$ si los cheques de la región i se envían a la ciudad j.

Sea n el número de días en un año. Entonces la tasa diaria de interés es $0.05/n$. Así, por ejemplo, si los cheques de R_1 se envían a C_2, entonces diariamente se dejan de ganar

$$\$1.200'000.000 \; \frac{0.05}{n} \; 4$$

Al hacer la cuenta para todo el año, la pérdida es de

$$n \; \$1.200'000.000 \; \frac{0.05}{n} 4 = \$240'000.000$$

Así podemos construir una tabla donde están los valores p_{ij} de pérdidass anuales en millones de pesos.

	Ciudad 1	Ciudad 2	Ciudad 3	Ciudad 4
Región 1	120	240	360	360
Región 2	96	48	120	120
Región 3	432	360	144	360
Región 4	252	180	216	108

$$\begin{aligned}
\min \quad z = \sum_{i=1}^{4}\sum_{j=1}^{4} p_{ij}x_{ij} + 180 \sum_{j=1}^{4} y_j & \\
\sum_{j=1}^{4} x_{ij} &= 1, \quad i = 1, ..., 4 \\
\sum_{i=1}^{4} x_{ij} &\leq 4y_j\,, \quad j = 1, ..., 4 \\
x_{ij}, y_j &\in \{0, 1\}\,.
\end{aligned}$$

La segunda restricción quiere decir: si $y_j = 0$, entonces $x_{1j} = x_{2j} = x_{3j} = x_{4j} = 0$.

```
$ONTEXT
  problema lockbox, Cornuejols, p. 213-215
  escritura estructurada
$OFFTEXT

OPTION OPTCR = 0.0001;

SETS
    i regiones /r1*r4/
    j ciudades /c1*c4/ ;

TABLE p(i,j) perdida de intereses en millones
          c1    c2    c3    c4
```

```
r1        120   240   360   360
r2        96    48    120   120
r3        432   360   144   360
r4        252   180   216   108;

SCALAR coc de una oficina central /180/;

VARIABLES
    x(i,j) los cheques de la region ri van a la ciudad cj
    y(j)   la central en la ciudad cj se abre
    z      costo total en millones;

BINARY VARIABLES x, y;

EQUATIONS
    costo    funcion ojetivo
    sale(i)  lo que sale de una region
    llega(j) lo que llega a una ciudad;

costo..    z =e= sum( (i,j), p(i,j)*x(i,j) ) + coc*sum(j,
y(j) );
sale(i)..  sum( j, x(i,j) ) =e= 1;
llega(j).. sum(i, x(i,j) ) - 4*y(j) =l= 0;

MODEL lockbox /ALL/;
SOLVE lockbox USING MIP MINIMIZING z;
```

0.11 Subasta combinatoria

Tomado de [CoT07]. En algunas subastas los elementos aparecen agrupados por paquetes no necesariamente disyuntos y la oferta que hace el posible comprador por cada paquete no es la suma de los precios de sus elementos, puede ser mayor o menor.

Sea $M = \{1, 2, ..., m\}$ el conjunto de elementos. El subastador recibe n ofertas, cada una es una pareja $B_j = (S_j, p_j)$, donde S_j es un conjunto de elementos ($S_j \subseteq M$) y p_j es el valor propuesto por el posible comprador. El subastador desea saber cuales ofertas aceptar para maximizar los ingresos.

Sea x_j una variable que vale 1 si el subastador acepta la oferta j y vale cero en caso contrario.

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} p_j x_j \\ & \sum_{j\,:\,i \in S_j} x_j \le 1, \quad i = 1, ..., m \\ & x_j \in \{0, 1\}, \quad j = 1, .., n. \end{aligned}$$

Ejemplo: Sean 4 elementos y 6 ofertas

$$
\begin{aligned}
B_1 &= (\{1\},\ 6)\\
B_2 &= (\{2\},\ 3)\\
B_3 &= (\{3,4\},\ 12)\\
B_4 &= (\{1,3\},\ 12)\\
B_5 &= (\{2,4\},\ 8)\\
B_6 &= (\{1,3,4\},\ 16)
\end{aligned}
$$

$$
\begin{array}{lccccccl}
\max & 6x_1 & +3x_2 & +12x_3 & +12x_4 & +8x_5 & +16x_6 & \\
 & x_1 & & & +x_4 & & +x_6 & \leq 1\\
 & & x_2 & & & +x_5 & & \leq 1\\
 & & & x_3 & +x_4 & & +x_6 & \leq 1\\
 & & & x_3 & & +x_5 & +x_6 & \leq 1\\
 & & & & & & x_j & \in \{0,1\}, \quad j = 1, ..., 6.
\end{array}
$$

En Gams

```
$ONTEXT
  Subasta combinatoria, Cornuejols, p. 213
$OFFTEXT

VARIABLES x1, x2, x3, x4, x5, x6, z;
BINARY VARIABLES x1, x2, x3, x4, x5, x6;
EQUATIONS obj, el1, el2, el3, el4;
obj..  z =e= 6*x1 + 3*x2 + 12*x3 + 12*x4 + 8*x5 + 16*x6;
el1.. x1 + x4 + x6 =l= 1;
el2.. x2 + x5      =l= 1;
el3.. x3 + x4 + x6 =l= 1;
el4.. x3 + x5 + x6 =l= 1;
MODEL pag213 /ALL/;
SOLVE pag213 USING MIP MAXIMIZING z;
```

La solución es

$$x = (1, 1, 1, 0, 0, 0), \quad z = 21.$$

REFERENCIAS

[Bil97] Birge J.R., Louveaux F., *Introducción to stochastic Programming*, Spinger, New York, 1997.

[CoT07] Cornuejols G., Tutuncu R., *Optimization Methods in Finance*, Cambridge Univ. Press, 2007.