

GENERALIDADES DEL LENGUAJE DE PROGRAMACIÓN GAMS

Asignatura: investigación de Operaciones I.

Profesora: Patricia Jaramillo Álvarez

Departamento de Ciencias de la Computación y de la Decisión

Universidad Nacional de Colombia

PUEDE BAJAR UN DEMO DE GAMS en <http://gams.com/>

Video interesante en youtube

https://www.youtube.com/watch?v=sLxb0JsSs1U&index=5&list=PLGBbVX_WvN7ZhPW_A90iORZVL1yogAQ4W

GAMS es un lenguaje de programación matemática. Los lenguajes de programación son lenguajes que permiten la formulación de problemas de optimización complejos. Son muy potentes por su capacidad de resolución y la indexación de las variables y ecuaciones. 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.dash.co.uk) de origen europeo, por citar algunos.

GAMS es el lenguaje más ampliamente difundido comercialmente con su propia lista de discusión de usuarios (gams-l@listserv.gmd.de) mientras que AMPL se está potenciando mucho en las universidades estadounidenses. Existe un proyecto denominado **NEOS** (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. GAMS es un lenguaje de modelización, más que un programa para resolver problemas de optimización. La ventaja que presenta este programa GAMS, es que junto al módulo de modelización (base) incorpora diferentes solver (algoritmos de resolución de problemas) tanto de programación no lineal, como lineal y entera.

INFORMACION BASICA

Para resolver un problema de optimización mediante GAMS es necesario preparar un proyecto. Para ello elige "file" "Project" "New Project". A este proyecto se vincularán todos los archivos que tengan que ver con el proyecto: archivo de modelos, archivos de datos, archivos de opciones, etc que se deberán almacenar todos en la misma carpeta.

Los modelos se editan en un "input file" en donde se definen todos los elementos básicos: VARIABLES, EQUATIONS, MODEL, SOLVE, etc.

En este archivo pueden ir todos los datos y opciones, o estos pueden estar en otros archivos a los que se vincula con el código: \$include *Nombredelarchivo*

Para construir el archivo, se abre GAMSIDE y en la página en blanco que ofrece, se escribe, no el problema de forma explícita sino la siguiente información porque luego, cuando se compila, GAMS formula el problema en forma explícita (estándar). En la página en blanco es necesario declarar:

1. Los **sets** (subíndices: i, j, etc.)
2. **scalars**, parámetros del problema
3. **Parameters**, datos en matrices de una dimensión
4. **table**, datos en matrices de varias dimensiones
5. Las **variables** que aparecen en el problema
6. **tipo** de las variables (no negativas, enteras, binarias, etc.).
7. El nombre de las ecuaciones: función objetivo y restricciones (**equations**)
8. la expresión matemática de estas ecuaciones (incluye a la función objetivo y a las restricciones)
9. El nombre del modelo general y las ecuaciones que forman el modelo (*all* si son todas).
10. La declaración de resolución, el solver (NLP, LP, MIP, ...que significan no lineal, lineal programación mixta, respectivamente) y si se quiere maximizar o minimizar la variable correspondiente a la función objetivo
11. Declaración de cómo se desea que muestre los resultados.

En las ecuaciones los signos relacionales usados son: **=e=** (igual), **=l=** (menor o igual que), **=g=** (mayor o igual que), **=lt=** (menor que). Los tipos de variables posibles son: **Free, positive, negative, binary e integer**. Para expresar el tipo de variable debe colocarse, en la sección de tipo de variables, por ejemplo: **"Positive variable X"; o "Binary Variable Y"**.

Los operadores más usados son: **+** (suma), **-** (resta), ***** (multiplicación), **/** (división) y ******(Exponenciación).

sum (i,función) o sum((i,j), función), prod()

Complementos del lenguaje:

1. Cuando se desea introducir en una función todos los valores de un subíndice, en la función correspondiente se introduce el nombre del subíndice, por ejemplo $x(i)$
2. Cuando se desea introducir un subíndice específico se pone entre comillas, por ejemplo $x('3')$.
3. Cuando se desea establecer una condición se hace mediante el signo \$. Por ejemplo, si se desea hacer una sumatoria de las variables de decisión X_{ij} pero solo con los términos en los que i menor que j se expresa de la siguiente manera: $sum((i,j)\$(ord(i) lt ord(j)), x(j,j))$.
4. Si se desea hacer restricciones para cada i , siempre que $i > 2$, entonces se nombra a la restricción res y se define así: $res(i)\$(ord(i) > 2)$.
5. Cuando se desea hacer un comentario, o volver inactivo provisionalmente una línea de código se pone al inicio de la línea el signo **"**"**
6. Si se desea hacer un *loop* de operaciones (incluyendo optimizaciones en varias iteraciones) se usa la función

Loop (

);

7. si se desea usar el valor arrojado para una variable de decisión $x(i)$ como parámetro de otra operación se usa la expresión $x./i$ (I de level)

Problema 1: Problema del transporte: Minimizar el costo de llevar una mercancía desde 2 plantas a 3 mercados.

Indices: i=plantas, J=mercados

Datos:

O_i = oferta de la planta i

D_j = demanda del mercado j

d_{ij} = distancia entre planta i y mercado j

Distancias (miles de millas)				
	Mercados			
Plantas	New York	Chicago	Topeka	Oferta
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
Demanda	325	300	275	

El costo por artículo por mil millas es $F=\$90$.

Abra archivo **problema de tte.gms**. Al correr GAMS (**File, Run** o usando el icono con una flecha roja), y si no se encuentra algún error se genera automáticamente el problema formulado se compila y se resuelve (en el caso de que encuentre algún error el aclara en letra roja la causa. Haciendo clic en el texto rojo lo redirecciona al lugar donde hay un error. Cuidar que no falte ; o espacios, o que escriba correctamente los nombres de las variables y las ecuaciones. No llame con el mismo nombre a variables y a ecuaciones). Cierre la información de compilación (**Close**) y mire el archivo **problema de tte.gms.lst**. Tanto el problema formulado como su solución se guardan en este archivo y en él se indica (entre muchas más cosas pero solo concéntrese en lo siguiente):

- inicialmente el mismo código que usted introdujo en **problema de tte.gms**
- luego la formulación del problema de forma estándar ($\text{Max } Z(x)$, sujeto a : $g(x)$).
- SOLVE SUMMARY:** aquí aparece la solución óptima del problema. En **OBJECTIVE VALUE** indica el valor obtenido para la función objetivo, luego aparecen los valores que toman todas las restricciones, y luego los valores que toman las variables de decisión. Para cada valor muestra cuatro columnas: **LOWER, LEVEL, UPPER, MARGINAL**. Cuando el valor de la variable se ha condicionado a un mínimo o a un máximo, aparece ese valor en la columna LOWER y UPPER respectivamente, de lo contrario aparece un punto (.). **LEVEL** es el valor que toma la restricción o la variable de decisión en la solución óptima. **MARGINAL** se refiere los coeficientes de lagrange de las restricciones o a los costos marginales para las variables de decisión.

Respecto a los costos marginales de las variables de decisión:

- Si el problema es para maximizar, salen negativos. En GAMS se interpreta de la siguiente manera. Si los costos marginales son cero es porque la variable salió básica (>0) en la solución óptima. Si los costos marginales son diferentes de cero es porque la variable salió No básica ($=0$) en la solución óptima (es decir, no justifica hacer de ella

porque seguramente sus beneficios unitarios son bajos), así el valor absoluto es la cantidad que tendría que aumentar el Ci (coeficiente de la Función objetivo) para que la variable Xi se vuelva básica.

- Si el problema es para minimizar, salen positivos. En GAMS se interpreta de la siguiente manera. Si los costos marginales son cero es porque la variable salió básica (>0) en la solución óptima. Si los costos marginales son diferentes de cero es porque la variable salió No básica ($=0$) en la solución óptima (es decir, no justifica hacer de ella porque seguramente sus costos unitarios son muy altos), así, el valor absoluto es la cantidad que tendría que disminuir el Ci (coeficiente de la Función objetivo) para que la variable Xi se vuelva básica.

Respecto a los precios sombra de las restricciones. Se pueden interpretar así:

- Si el problema es para maximizar y la restricción es tipo \leq , ellos son positivos, lo que se puede interpretar como que si los recursos disponibles aumentan, la Z aumenta. Si la restricción es tipo \geq ellos son negativos, lo que se puede interpretar como que si la b aumenta, toca incurrir en mas costos y los beneficios disminuyen.
- Si el problema es para minimizar y la restricción es tipo \leq , ellos son negativos, lo que se puede interpretar como que si los recursos disponibles aumentan, los costos disminuyen porque puede haber mejor uso y eficiencia en la producción. Si la restricción es tipo \geq ellos son positivos, lo que se puede interpretar como que si la b aumenta, toca incurrir en mas costos por lo que Z aumenta.

	RESTRICCIÓN	VARIABLE
MARGINAL	Precio sombra	Costo reducido
MINIMIZAR	Tipo \leq , son negativos Si los recursos disponibles (b) aumentan, los costos disminuyen debido a un mejor uso y eficiencia de la producción, la Z disminuye.	Salen positivos SI = 0, \rightarrow Var. Básica SI \neq 0, \rightarrow Var. No Básica (no justifica producir) valor \rightarrow cantidad que tendría que disminuir el coeficiente de la variable en la F.O. para que la variable se vuelva básica.
	Tipo \geq , son positivos, si los recursos disponibles (b) aumentan, los costos aumentan debido a un mal uso y deficiencia de la producción, la Z aumenta.	
MAXIMIZAR	Tipo \leq , son positivos Si los recursos disponibles (b) aumentan, la Z aumenta.	Salen negativos SI = 0, \rightarrow Var. Básica SI \neq 0, \rightarrow Var. No Básica (no justifica producir) valor \rightarrow cantidad que tendría que aumentar el coeficiente de la variable en la F.O. para que la variable se vuelva básica.
	Tipo \geq , son negativos, si los recursos disponibles (b) aumentan, la Z disminuye.	

Para Mostrar Resultados “amigables al usuario”

Uno puede hacer que se genere un archivo donde se escriban en la forma que el modelador desee los resultados.

Para ello, debe indicar el nombre del archivo y luego mediante la palabra **put**, escribe lo que desea y como lo desee en él.

Algunas Ayudas para esto son :

Escribir una palabra con un tab de 20.

Put @20, 'palabra';

Pasar dos lineas sin texto

Put //;

Mostrar los valores de una variable $x(i,j)$

Put x.l(i,j);

Mostrar los valores de los precios sombra (o variables duales)

Put nombre_de_equation.m;