

# SOFTWARE EN PROGRAMACIÓN MATEMÁTICA

Programación en GAMS

Several thin, white, parallel lines of varying lengths and slopes are positioned on the right side of the slide, extending from the top right towards the bottom left.



# Software en optimización

## Modelizadores:

*Reciben modelo en lenguaje matemático habitual y lo transforman a un formato reconocido por el optimizador.*

Entrada: Datos del modelo

Salida: Fichero MPS (*Mathematical Programming Standard*)

## Optimizadores:

*Reciben modelo en formato adecuado y lo resuelven.*

Entrada: Fichero MPS

Salida: Informe de la solución obtenida



# Software en optimización

## Modelizadores:

- GAMS                      AIMMS                      ...
- AMPL                      XPRESS

## Optimizadores:

- CPLEX
- FortMP                      MINOS                      ...
- OSL                      CONOPT

## Software modelizador y optimizador:

- Winqsb
- OPL (herramienta integrada)





# Software en optimización

## Modelizadores:

### Lenguajes de modelización:

- 1) Lenguajes de programación de propósito general.  
(pueden llamar a una biblioteca de optimización)
- 2) Lenguajes o entornos de cálculo numérico o simbólico  
(hojas de cálculo, MATLAB, Maple, Mathematica, etc...)
- 3) Lenguajes algebraicos de modelización  
(lenguajes de alto nivel diseñados específicamente para el desarrollo e implantación de modelos de optimización)



# Problema de mezclas

Una refinería de petróleo va a producir un nuevo tipo de gasolina mezclando las gasolinas que resultan de procesar diferentes tipos de crudo. Los crudos de origen son cuatro y tienen distinta composición.

Para simplificar el problema se supone que cada tipo de gasolina tiene un porcentaje distinto de los aditivos A, B y C. La tabla siguiente indica estos porcentajes y el precio unitario para los cuatro tipos de gasolina:





# Problema de mezclas

	Aditivo A	Aditivo B	Aditivo C	Precio Gasolina
Gasolina 1	80	10	10	43
Gasolina 2	30	30	40	31
Gasolina 3	70	10	20	47
Gasolina 4	40	50	10	37



# Problema de mezclas

Las exigencias del mercado imponen que la gasolina que se va a producir debe tener al menos

- el 60 % del aditivo A
- no más del 30 % del aditivo C.

Determinar la mezcla que producirá la gasolina con estas especificaciones y cuyo precio sea mínimo.





# Esquema básico de un programa en GAMS

- ***Datos del problema***
- ***Variables del problema***
  - Definición de las variables
  - Naturaleza de las variables
- ***Ecuaciones del problema***
  - Función objetivo
  - Restricciones
- ***Descripción del modelo***
- ***Llamada al software optimizador***





# Problema de mezclas

\$TITLE Hoja 1 problema 1

variables

x proporción de gas1 en la mezcla

y proporción de gas2 en la mezcla

z proporción de gas3 en la mezcla

t proporción de gas4 en la mezcla

obj objetivo;

positive variables x,y,z,t;



# Problema de mezclas

## equations

fobj coste de la mezcla

adA porcentaje aditivo A

adC porcentaje aditivo C

Total total de la mezcla;

fobj.. obj =E=  $43*x+31*y+47*z+37*t$ ;

adA..  $80*x+30*y+70*z+40*t =G= 60$ ;

adC..  $100*x+40*y+20*z+10*t =L= 30$ ;

total..  $x+y+z+t =E= 1$ ;

model mezcla /fobj,adA,adC, total/;

solve mezcla using LP minimizing obj;





# Problema del transporte

Se consideran dos conjuntos, uno de consumidores y otro de productores o factorías y se conocen las demandas de cada consumidor, la producción máxima de cada productor y el coste del transporte de unidades de producto entre productores y consumidores.

Se trata de determinar la cantidad de producto que enviar de cada productor a cada consumidor de tal forma que se satisfaga la demanda a coste mínimo.



# Modelo del problema del transporte

$x_{ij}$  = Cantidad de producto que se transporta de  $i$  a  $j$

$$\text{Min} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}$$

$$\text{s.a.:} \sum_{j=1}^n x_{ij} \leq a_i \quad \forall i = 1, \dots, n$$

$$\sum_{i=1}^m x_{ij} \geq b_j \quad \forall j = 1, \dots, m$$

$$x_{ij} \geq 0$$





# Ejemplo numérico

<b>Plantas</b>	<i>Distancias entre plantas y mercados</i>			<u><i>Producción</i></u>
	<b>Mercados</b>			
	<i>Nueva York</i>	<i>Chicago</i>	<i>Topeka</i>	
<i>Seattle</i>	2.5	1.7	1.8	<b>350</b>
<i>San Diego</i>	2.5	1.8	1.4	<b>600</b>
<u><i>Demandas</i></u>	<b>325</b>	<b>300</b>	<b>275</b>	

*Las distancias están en miles de millas. El coste por transportar una unidad de una ciudad a otra es de 90\$ por cada mil millas.*

# Estructura programa en Gams



Inputs	Outputs
<ul style="list-style-type: none"><li>• <b>Sets</b> Declaration Assignment of members</li><li>• <b>Data</b> (Parameters, Tables, Scalar) Declaration Assignment of values</li><li>• <b>Variables</b> Declaration Assignment of type Assignment of bounds and/or initial values (optional)</li><li>• <b>Equations</b> Declaration Definition</li><li>• <b>Model and Solve statements</b></li><li>• <b>Display statement</b> (optional)</li></ul>	<ul style="list-style-type: none"><li>• <b>Echo Print</b></li><li>• <b>Reference Maps</b></li><li>• <b>Equation Listings</b></li><li>• <b>Status Reports</b></li><li>• <b>Results</b></li></ul>





# Datos del problema I

## Sets

i plantas de producción / Seattle, San-Diego /  
j mercados / New-York, Chicago, Topeka / ;

*Conjuntos de  
índices*

## Parameters

a(i) capacidad de la planta i

/ Seattle 350  
San-Diego 600 /

b(j) demanda del mercado j

/ New-York 325  
Chicago 300  
Topeka 275 / ;

*Parámetros*



# Datos del problema II

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 ;

*Tabla de datos*

Scalar  $f$  coste por miles de millas /90/ ;

*Escalares*

Parameter  $c(i,j)$  coste del transporte por planta y mercado ;

*Parámetros*

$$c(i,j) = f*d(i,j)/1000 ;$$





# Variables del problema

## Variables

$x(i,j)$  Uds de producto que se envían de  $i$  a  $j$   
 $z$  coste total del transporte en miles de dólares ;

*Definición de las variables*

Positive variable  $x$  ;

*Naturaleza de las variables*



# Ecuaciones del problema

## Equations

cost	función objetivo del problema
supply(i)	no se supera tope de producción
demand(j)	se satisface la demanda;

*Definición de las ecuaciones*

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) ;

*Expresiones formales de las ecuaciones*

Código de desigualdades:

=e=  $\rightarrow$  =

=g=  $\rightarrow$   $\geq$

=l=  $\rightarrow$   $\leq$





# Definición del modelo, llamada al optimizador y datos de salida

```
Model transport /all/ ;
```

```
solve transport using lp minimizing z ;
```

```
display x.l, x.m ;
```

*Definición del modelo*

*Llamada al software optimizador*

*Petición de datos de salida*

## Códigos solve:

lp	Programación lineal
dnlp	Programación no lineal con derivadas discontinuas
rmip	Programación lineal relajada entera-mixta
rminlp	Programación no lineal relajada entera-mixta
mpec	Problemas matemáticos con restricciones de equilibrio

nlp	Programación no lineal
mip	Programación lineal entera-mixta
minlp	Programación no lineal entera-mixta
mcp	Problemas complementarios mixtos
cns	Sistemas no lineales acotados

## Display:

.l	Solución óptima
.m	Solución marginal o costes reducidos



# Otros aspectos a tener en cuenta

- Definición *alias*

Sets

```
i  aviadores / Espanol, Frances, Italiano, Griego, Portugues /  
k  puesto   / piloto, copiloto / ;  
alias (i,j)
```

- Operador Condicional \$

*Código de desigualdades:*

$< lt$	$\leq le$
$> gt$	$\geq ge$
$= eq$	$\neq ne$

- Asterisco \*

- Producto \*
- Exponenciación \*\*
- Comentarios
- Definición de conjuntos  $1^*5 \rightarrow 1,2,3,4,5$





# Esquema de un fichero de salida

- Compilación y conteo de líneas
- Código MPS (ecuaciones escritas de forma explícita)
- Coeficientes de las variables y ecuaciones en las que aparecen éstas
- Resumen del modelo con número de ecuaciones, variables y otros datos
- Resolución del problema
- Datos de salida pedidos en el código

# Resolución de problemas en NEOS



NEOS Solvers - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección <http://www-neos.mcs.anl.gov/neos/solvers/index.html> Ir Links

Y! Guardar en Mi Web Entrar Correo Mi Yahoo! Y! Respuestas Noticias Cine Música Deportes Abrir

Choose a solver for the type of optimization problem that you want to solve. If you are not sure of the type of optimization problem, consult the [Optimization Tree](#) of the NEOS Guide for information on optimization problems. The choice of solver is then dictated by the language used to define the optimization problem.

Each solver has sample problems and background information on the solver. Be sure to submit a sample problem to get a feel for how to submit optimization problems to NEOS. If you encounter problems, consult the [NEOS Server FAQ](#), or send mail to us by clicking on the **Comments and Questions** icon at the bottom of the page.

- **Administration**
  - [view job queue](#)
  - [view job status/results](#)
  - [kill/dequeue job](#)
- **Bound Constrained Optimization**
  - BLMVM [[ampl Input](#)][[C Input](#)][[Fortran Input](#)]
  - L-BFGS-B [[AMPL Input](#)]
  - TRON [[AMPL Input](#)][[FORTRAN Input](#)]
- **Combinatorial Optimization**
  - concorde [[TSP Input](#)]
- **Complementarity Problems**

Internet