

# Práctica 1

Javier García Polo

fjgpolo@inf.uc3m.es

2.1B16

-

Grupo de Planificación y Aprendizaje (PLG)

Departamento de Informática

Universidad Carlos III de Madrid

27 de septiembre de 2020

- ▶ Tres partes:
  - ▶ Primera parte se resuelve con calc de LibreOffice (3 Puntos)
  - ▶ Segunda parte se resuelve con MathProg (5 Puntos)
  - ▶ Análisis de los resultados (2 puntos)
- ▶ Entrega hasta 8 de Noviembre a las 23:55.

## Primera parte

- ▶ Una compañía aérea:
  - ▶ Tres tipos de tarifas disponibles: estándar, leisure plus, y business plus
  - ▶ Cinco aviones con diferentes números de asientos y capacidad de equipaje
- ▶ **Objetivo:** determinar el número de billetes de cada tipo que se debe ofertar para cada avión de forma que se maximice el beneficio.
- ▶ Modelar el problema utilizando Calc y solucionarlo con el *Solver*.

## Segunda parte

- ▶ Asignación de pistas de aterrizaje para los aviones de la compañía
- ▶ Cada avión: tiempo aterrizaje, tiempo máximo, coste combustible
- ▶ *Slots* de tiempo de aterrizaje disponible
- ▶ Cada avión tiene que estar asignado a un único *slot*
- ▶ Un *slot* solo puede tener asignado como máximo un avión
- ▶ ...
- ▶ **Objetivo:** Determinar la asignación de aviones a slots de tiempo de aterrizaje de forma que se minimice el coste adicional de la compañía por los retrasos



## Tercera parte

- ▶ Análisis de los resultados
  - ▶ Descripción de la solución, restricciones, etc.
  - ▶ Número de variables y restricciones.
  - ▶ Modificar los parámetros añadiendo/quitando pistas, aviones
- ...

Geppetto Madera SA produce dos tipos de juguetes de madera: soldados y trenes. Cada soldado se vende por 27€ y usa 10€ de materias primas. Cada tren se vende por 21€ y usa 9€ de materia prima. Además cada soldado genera un sobrecoste y unos gastos adicionales por mano de obra de 14€, mientras que los sobrecostes por tren son de 10€. La producción de juguetes de madera requiere dos procesos: carpintería y acabado. Cada soldado requiere 1 hora de carpintería y 2 horas de acabado, mientras que cada tren requiere 1 y 1. Cada semana, el taller dispone de 80 horas para tareas de carpintería y 100 horas para tareas de acabado. La demanda de trenes es ilimitada, pero como mucho se venden 40 soldados por semana. Geppetto Madera SA nos pide calcular cuántos juguetes de cada tipo producir para maximizar su beneficio.

$$\text{máx } z = 3x_1 + 2x_2$$

$$2x_1 + x_2 \leq 100 \quad \text{Horas de acabado: 100}$$

$$x_1 + x_2 \leq 80 \quad \text{Horas de carpintería: 80}$$

$$x_1 \leq 40 \quad \text{Demanda soldados}$$

$$x_1, x_2 \geq 0 \quad \text{Unidades no negativas}$$

## Resolución en OpenOffice: Calc

- ▶ **Variables de decisión.** Una celda por cada variable de decisión.
- ▶ **Datos del problema.** Selección de una región donde almacenar los datos del problema.
- ▶ **Datos calculados.** Cálculos intermedios necesarios para solucionar el problema (opcional).
- ▶ **Función objetivo.** Una celda para la función objetivo.
- ▶ **Restricciones.** Selección de una región donde almacenar las restricciones.
- ▶ **Tools - Solver:**
  - ▶ Selección de qué celdas representa la función objetivo, las variables y restricciones.
  - ▶ Seleccionar el solver: Solver lineal.





U1

f(x) Σ =

	A	B	C	D
1	<b>Variables</b>			
2		Soldados	Trenes	
3	Unidades	20	60	
4				
5	<b>Datos</b>			
6		Soldados	Trenes	
7	Precio Venta	27	21	
8	Materias Primas	10	9	
9	Gastos Adicionales	14	10	
10	Tiempo Carpinteria	1	1	
11	Tiempo Acabado	2	1	
12	Demanda Máxima	40	-	
13				
14		Carpinteria	Acabado	
15	Horas Dedicadas	80	100	
16				
17				
18	<b>Datos calculados</b>			
19		Soldados	Trenes	
20	Beneficio	3	2	
21				
22	<b>Objetivo:</b>	180		
23				
24	<b>Restricciones</b>			
25		Izquierda		Derecha
26	Limite Horas Carpinteria	80 <=		80
27	Limite Horas Acabado	100 <=		100
28	Demanda Máxima Soldados	20 <=		40
29				

	A	B	C	D
1	<b>Variables</b>			
2		Soldados	Trenes	
3	Unidades	20	60	
4				
5	<b>Datos</b>			
6		Soldados	Trenes	
7	Precio Venta	27	21	
8	Materias Primas	10	9	
9	Gastos Adicionales	14	10	
10	Tiempo Carpinteria	1	1	
11	Tiempo Acabado	2	1	
12	Demanda Máxima	40	-	
13				
14		Carpinteria	Acabado	
15	Horas Dedicadas	80	100	
16				
17				
18	<b>Datos calculados</b>			
19		Soldados	Trenes	
20	Beneficio	3	2	
21				
22	<b>Objetivo:</b>	180		
23				
24	<b>Restricciones</b>			
25		Izquierda		Derecha
26	Limite Horas Carpinteria	80 <=		80
27	Limite Horas Acabado	100 <=		100
28	Demanda Máxima Soldados	20 <=		40
29				
30				

**Solver**

Target cell:

Optimize result to:

- ☒ Maximum
- ☐ Minimum
- ☐ Value of

By changing cells:

Limiting conditions:

Cell reference	Operator	Value
<input type="text" value="\$B\$26:\$B\$28"/>	<input "="" type="text" value="&lt;="/>	<input type="text" value="\$D\$26:\$D\$28"/>
<input type="text"/>	<input "="" type="text" value="&lt;="/>	<input type="text"/>
<input type="text"/>	<input "="" type="text" value="&lt;="/>	<input type="text"/>
<input type="text"/>	<input "="" type="text" value="&lt;="/>	<input type="text"/>

Options... Help Close Solve

**Options**

Solver engine:

Settings:

- ☐ Assume variables as integer
- ☒ Assume variables as non-negative
- Epsilon level (0-3):
- ☒ Limit branch-and-bound depth
- Solving time limit (seconds):

Edit

## Ejemplo de formalización

- ▶ Coches =  $\{Laguna, A3, Passat\}$ , Países =  $\{Spain, Italy, Germany\}$
- ▶ Variable unidades de cada coche que se venden en cada país.
  - ▶  $x_{ij}$  Número de unidades del coche  $i$  que se venden en el país  $j$ .
- ▶ Las ventas de cada coche en cada país no debe superar los 100.000€
  - ▶  $\forall i, j \ x_{ij} \times c_{ij} \leq 100000$  donde  $i = \{Laguna, A3, Passat\}$  y  $j = \{Spain, Italy, Germany\}$ .
- ▶ Las ventas totales debe ser superior a 1.000.000€
  - ▶  $\sum_{i=\{Laguna, A3, Passat\}} \sum_{j=\{Spain, Italy, Germany\}} x_{ij} \times c_{ij} \geq 1000000$
- ▶ Las ventas totales en España debe ser inferior a 500.000€.
  - ▶  $\sum_{i=\{Laguna, A3, Passat\}} \sum_{j=\{Spain\}} x_{ij} \times c_{ij} \leq 500000$

# MathProg

- ▶ Conjuntos: Colección de objetos del problema.

```
set TOYS;
```

- ▶ Parámetros: Los datos del problema se ponen como parámetros.

```
param Finishing_hours {i in TOY};  
param Carpentry_hours {i in TOY};  
param Demand_toys      {i in TOY};  
param Profit_toys      {i in TOY};
```

- ▶ Variables de decisión:

```
var units{i in TOYS}, >= 0;
```

- ▶ Función objetivo:

```
maximize Profit: sum{i in TOY} Profit_toys[i]*units[i];
```

- ▶ Restricciones:

- ▶  $2x_1 + x_2 \leq 100$

- s.t. Fin\_hours : sum{i in TOY} Finishing\_hours[i]\*units[i] <= 100;

- ▶  $x_1 + x_2 \leq 80$

- s.t. Carp\_hours : sum{i in TOY} Carpentry\_hours[i]\*units[i] <= 80;

- ▶  $x_1 \leq 40$

- s.t. Dem {i in TOY} : units[i] <= Demand\_toys[i];

## En fichero aparte los datos del problema

- ▶ Conjuntos:

```
set TOY := soldier train;
```

- ▶ Parámetros:

```
param Finishing_hours:=  
soldier 2  
train 1;
```

- ▶ Ejecución

```
glpsol -m model.dat -d data.dat -o output.txt
```

- Colección de objetos:

```
set CARS;  
set COUNTRY;
```

- Variables en matrices bidimensionales:

```
var ventas{i in CARS, j in COUNTRY} integer, >= 0;
```

- Las ventas de cada coche en cada país no debe superar los 100.000€

```
s.t. benefits {i in CAR, j in COUNTRY}: ventas[i,j] * price[i,j] >= 100000;
```

- Las ventas totales debe ser superior a 1.000.000€

```
s.t. benefits: sum{i in CAR, j in COUNTRY} ventas[i,j] * price[i,j] >= 1000000;
```