Ejercicios de Laboratorio PLIE

Grupo Aula E

(Elena Fernández)

Planificació de la producció:

	Consum unitari mà obra (h)	Consum unitari fusta (kg)	Consum plastic (kg)	Benefici unitari (€)	x^* (kg)
Producte A	1	3	2	300	50
Producte B	2	2	-	250	50
Disponibilitat	150h/dia	300kg/dia	100kg/dia		

- Formuleu la modelització matemàtica parametritzada
- Implementeu i resoleu amb OPTMODEL
- Indiqueu els valors de la solució òptima: B^*, x_B^*, N^*, r^* i λ^*

Parámetros

Recursos

Consumo (por kg producido)

Disponibilidad

Productos

Beneficios

	Producte A	Producte B	Disponibilitat
Consum unitari mà obra (h)	1	2	150h/dia
Consum unitari fusta (kg)	3	2	300kg/dia
Consum plastic (kg)	2	-	100kg/dia
Benefici unitari (€)	300	250	
x^* (Kg)			

OPTMODEL

```
set<string> PRODUCTE = {'A','B'};
set<string> RECURS = {'ma', 'fusta', 'plastic'};
number consum{ RECURS, PRODUCTE} = [ 1  2
                                     3  2
                                     2  0 ];
number disp{RECURS} = [150 300 100];
number benefici{PRODUCTE} = [300 250];
```

Modelo Optimización (papel)

	Producte A	Producte B	Disponibilitat
Consum unitari mà obra (h)	1	2	150h/dia
Consum unitari fusta (kg)	3	2	300kg/dia
Consum plastic (kg)	2	-	100kg/dia
Benefici unitari (€)	300	250	
x^* (Kg)			

Variables de decisión: x_j: cantidad fabricada de producto j (en Kg), $j \in \{A, B\}$

Restricciones:

$x_A + 2\,x_B \leq 150$ (disp. mano obra)

$3\,x_A + 2\,x_B \leq 300$ (disp. madera)

$2\,x_A \leq 100$ (disp. pástico)

$\sum_{j \in Prod} a_{ij} x_j \leq b_i \quad i \in Recursos$

Dominio de las variables: $x_A, x_B \geq 0$

Función Objetivo: Max $300\,x_A + 250\,x_B$

Max $\sum_{j \in Prod} p_j x_j$

Modelo Optimización: OPTMODEL

Variables de decisión: x_j: cantidad fabricada de producto j (en Kg), $j \in \{A, B\}$

Restricciones: $\sum_{j \in Prod} a_{ij} x_j \leq b_i, \quad i \in Recursos$

Dominio de las variables: $x_A, x_B \geq 0$

Función Objetivo: Max $\sum_{j \in Prod} p_j x_j$

OPTMODEL

```
var Produc {PRODUCTE} >= 0;
max Total_benefici = sum {j in PRODUCTE} benefici[j]*Produc[j];
con Consum_recurs {i in RECURS}: sum {j in PRODUCTE} consum[i,j]*Produc[j] <= disp[i];
```

```
/* Paràmetres */
set<string> PRODUCTE = {'A','B'};
set<string> RECURS = {'ma', 'fusta', 'plastic'};
number consum{ RECURS, PRODUCTE} = [ 1 3 2 2 2 0 ];
number disp{RECURS} = [150 300 100];
number benefici{PRODUCTE} = [300 250];

/* Model d'optimització */
var Produc {PRODUCTE} >= 0;
max Total_benefici = sum {j in PRODUCTE} benefici[j]*Produc[j];
con Consum_recurs {i in RECURS}: sum {j in PRODUCTE} consum[i,j]*Produc[j] <= disp[i];
```

```
/* Model extens */
expand;

/* Optimització i resultats */
solve;
```

```
print Produc.lb Produc.sol Produc.ub Produc.rc Produc.status;
print Consum_recurs.lb Consum_recurs.body Consum_recurs.ub Consum_recurs.dual
Consum_recurs.status;
```

The OPTMODEL Procedure

```
Var Produc[A] >= 0
Var Produc[B] >= 0
Maximize Total_benefici=300*Produc[A] + 250*Produc[B]
Restricción Consum_recurs[ma]: Produc[A] + 2*Produc[B] <= 150
Restricción Consum_recurs[fusta]: 3*Produc[A] + 2*Produc[B] <= 300
Restricción Consum_recurs[plastic]: 2*Produc[A] <= 100
```

The OPTMODEL Procedure

Resumen del problema	
Objective Sense	Maximization
Objective Function	Total_benefici
Objective Type	Linear
Number of Variables	2
Bounded Above	0
Bounded Below	2
Bounded Below and Above	0
Free	0
Fixed	0
Number of Constraints	3
Linear LE (<=)	3
Linear EQ (=)	0
Linear GE (>=)	0
Linear Range	0
Coeficientes de restricción	5

Información de rendimiento	
Execution Mode	On Client
Number of Threads	1

The OPTMODEL Procedure

Resumen de la solución	
Solver	LP
Algorithm	Dual Simplex
Objective Function	Total_benefici
Solution Status	Optimal
Objective Value	27500
Iterations	4
Primal Infeasibility	0
Dual Infeasibility	0
Bound Infeasibility	0

[1]	Produc.LB	Produc.SOL	Produc.UB	Produc.RC	Produc.STATUS
A	0	50	1.7977E+308	0	B
B	0	50	1.7977E+308	0	B

[1]	Consum_recurs.LB	Consum_recurs.BODY	Consum_recurs.UB	Consum_recurs.DUAL	Consum_recurs.STATUS
fusta	-1.7977E+308	250	300	0.0	B
ma	-1.7977E+308	150	150	125.0	L
plastic	-1.7977E+308	100	100	87.5	L

Problema de mezcla

	Disolvents				
	1	2	3	4	Contingut mescla (ml/l)
Clor (ml/l)	180	120	90	60	≥ 90
Amoníac (ml/l)	3	2	6	5	≤ 4
Cost (€/l)	16	12	10	11	

- Formuleu la modelització matemàtica parametritzada
- Implementeu i resoleu amb OPTMODEL
- Indiqueu els valors de la solució òptima: B^*, x_B^*, N^*, r^* i λ^*

Modelo Optimización (papel)

	Disolvents				
	1	2	3	4	Contingut mescla (ml/l)
Clor (ml/l)	180	120	90	60	≥ 90
Amoníac (ml/l)	3	2	6	5	≤ 4
Cost (€/l)	16	12	10	11	

Variables de decisión: x_j: **proporción** de disolvente j en la mezcla, $j \in \{1, 2, 3, 4\}$

cantidad (en litros) disolvente j en un litro de mezcla

Dominio : $x_j \geq 0$, $j \in \{1, 2, 3, 4\}$

Restricciones:

$$
\begin{array}{rcll}
180\, x_1 + 120\, x_2 + 90\, x_3 + 60\, x_4 & \geq & 90 & \text{(Cloro)} \\
3\, x_1 + 2\, x_2 + 6\, x_3 + 5\, x_4 & \leq & 4 & \text{(Amoniaco)} \\
x_1 + x_2 + x_3 + x_4 & = & 1 & \text{(CANTIDAD DE MEZCLA PRODUCIDA)}
\end{array}
$$

Función Objetivo: $\text{Min } 16\, x_1 + 12\, x_2 + 10\, x_3 + 11\, x_4$

OPTMODEL

	Disolvents				
	1	2	3	4	Contingut mescla (ml/l)
Clor (ml/l)	180	120	90	60	≥ 90
Amoníac (ml/l)	3	2	6	5	≤ 4
Cost (€/l)	16	12	10	11	

Parámetros:

```
number nD=4;
set DISOLVENTS = 1..nD;
set COMPONENTS_MIN = {'Clor'};
set COMPONENTS_MAX = {'Amoniac'};
set COMPONENTS     = COMPONENTS_MIN UNION COMPONENTS_MAX;

number contingut{ COMPONENTS, DISOLVENTS} =
    [  180    120      90       60
         3      2       6        5  ];

number mescla{COMPONENTS} = [90 4];

number cost{DISOLVENTS} = [16 12 10 11];
```

OPTMODEL

	Disolvents				
	1	2	3	4	Contingut mescla (ml/l)
Clor (ml/l)	180	120	90	60	≥ 90
Amoníac (ml/l)	3	2	6	5	≤ 4
Cost (€/l)	16	12	10	11	

Variables: var Proporcio {DISOLVENTS} >= 0;

Restricciones:

constraint Contingut_minim {i in COMPONENTS_MIN}:
sum {j in DISOLVENTS} contingut[i,j]*Proporcio[j] >= mescla[i];

constraint Contingut_maxim {i in COMPONENTS_MAX}:
sum {j in DISOLVENTS} contingut[i,j]*Proporcio[j] <= mescla[i];

constraint Cons_mescla: sum{j in DISOLVENTS} Proporcio[j] = 1;

Función Objetivo: min Total_cost = sum {j in DISOLVENTS} cost[j]*Proporcio[j];

```
proc optmodel;

/* Paràmetres */
number nD=4;
set<number> DISOLVENTS = 1..nD;
set<string> COMPONENTS_MIN = {'Clor'};
set<string> COMPONENTS_MAX = {'Amoniac'};
set<string> COMPONENTS     = COMPONENTS_MIN UNION COMPONENTS_MAX;
number contingut{ COMPONENTS, DISOLVENTS} =
[          180       120       90        60
           3         2         6         5];
number mescla{COMPONENTS} = [90 4];
number cost{DISOLVENTS} = [16 12 10 11];
```

```
/* Model d'optimització */
var Proporcio {DISOLVENTS} >= 0;
min Total_cost = sum {j in DISOLVENTS} cost[j]*Proporcio[j];
con Contingut_minim {i in COMPONENTS_MIN}:
        sum {j in DISOLVENTS} contingut[i,j]*Proporcio[j] >= mescla[i];
con Contingut_maxim {i in COMPONENTS_MAX}:
        sum {j in DISOLVENTS} contingut[i,j]*Proporcio[j] <= mescla[i];
con Cons_mescla: sum{j in DISOLVENTS} Proporcio[j] = 1;

/* Model extens */
expand;

/* Optimització i resultats */
solve;

print Proporcio.lb Proporcio.sol Proporcio.ub Proporcio.rc Proporcio.status;
print Contingut_minim.lb Contingut_minim.body Contingut_minim.ub
Contingut_minim.dual Contingut_minim.status;
print Contingut_maxim.lb Contingut_maxim.body Contingut_maxim.ub
Contingut_maxim.dual Contingut_maxim.status;
```

The OPTMODEL Procedure

```
Var Proporcio[1] >= 0
Var Proporcio[2] >= 0
Var Proporcio[3] >= 0
Var Proporcio[4] >= 0
Minimize Total_cost=16*Proporcio[1] + 12*Proporcio[2] + 10*Proporcio[3] + 11*Proporcio[4]
Restricción Contingut_minim[Clor]: 180*Proporcio[1] + 120*Proporcio[2] + 90*Proporcio[3] + 60*Proporcio[4] >= 90
Restricción Contingut_maxim[Amoniac]: 3*Proporcio[1] + 2*Proporcio[2] + 6*Proporcio[3] + 5*Proporcio[4] <= 4
Restricción Cons_mescla: Proporcio[1] + Proporcio[2] + Proporcio[3] + Proporcio[4] = 1
```

Page Break

The OPTMODEL Procedure

Resumen del problema	
Objective Sense	Minimization
Objective Function	Total_cost
Objective Type	Linear
Number of Variables	4
Bounded Above	0
Bounded Below	4
Bounded Below and Above	0
Free	0
Fixed	0
Number of Constraints	3
Linear LE (<=)	1
Linear EQ (=)	1
Linear GE (>=)	1
Linear Range	0
Coeficientes de restricción	12

Información de rendimiento	
Execution Mode	On Client
Number of Threads	1

The OPTMODEL Procedure

Resumen de la solución	
Solver	LP
Algorithm	Dual Simplex
Objective Function	Total_cost
Solution Status	Optimal
Objective Value	11
Iterations	6
Primal Infeasibility	0
Dual Infeasibility	0
Bound Infeasibility	0

[1]	Proporcio.LB	Proporcio.SOL	Proporcio.UB	Proporcio.RC	Proporcio.STATUS
1	0	0.0	1.7977E+308	4.5	L
2	0	0.5	1.7977E+308	0.0	B
3	0	0.5	1.7977E+308	0.0	B
4	0	0.0	1.7977E+308	0.5	L

[1]	Contingut_minim.LB	Contingut_minim.BODY	Contingut_minim.UB	Contingut_minim.DUAL	Contingut_minim.STATUS
Clor	90	105	1.7977E+308	0	B

[1]	Contingut_maxim.LB	Contingut_maxim.BODY	Contingut_maxim.UB	Contingut_maxim.DUAL	Contingut_maxim.STATUS
Amoniac	-1.7977E+308	4	4	-0.5	L

Problema de la dieta

	Contingut per kg de menjar					
	Vitamines (ui)	Hidrats de Carboni (u.i.)	Oligoelements (u.i.)	Proteines (u.i.)	Preu (€/kg)	$\boldsymbol{x}^*$ (kg)
Carn	25	20	10	150	8	
Peix	200	50	10	200	10	
Cereals	300	300	10	50	2	
Fruita	-	160	50	20	1.5	
Pa	-	120	100	20	0.5	
Aportació minima diaria	60u.i./dia	40u.i./dia	100u.i./dia	100 u.i./dia		

- Formuleu la modelització matemàtica parametritzada
- Implementeu i resoleu amb OPTMODEL
- Indiqueu els valors de la solució òptima: B^*, xB^*, N^*, r^* i λ^*

Modelo Optimización (papel)

	Contingut per kg de menjar				
	Vitamines (ui)	Hidrats de Carboni (u.i.)	Oligoelements (u.i.)	Proteines (u.i.)	Preu (€/kg)
Carn	25	20	10	150	8
Peix	200	50	10	200	10
Cereals	300	300	10	50	2
Fruita	-	160	50	20	1.5
Pa	-	120	100	20	0.5
Aportació dia	60u.i./dia	40u.i./dia	100u.i./dia	100 u.i./dia	

Variables de decisión: x_j: Kg de alimento j consumida al día, $j \in \{1, 2, 3, 4, 5\}$

Dominio : $x_j \geq 0$, $j \in \{1, 2, 3, 4, 5\}$

Restricciones:

$$25 x_1 + 200 x_2 + 300 x_3 \geq 60 \quad \text{(Vitaminas)}$$
$$20 x_1 + 50 x_2 + 300 x_3 + 160 x_4 + 120 x_5 \geq 40 \quad \text{(Hidratos Carbono)}$$
$$10 x_1 + 10 x_2 + 10 x_3 + 50 x_4 + 100 x_5 \geq 100 \quad \text{(Oligoelementos)}$$
$$150 x_1 + 200 x_2 + 50 x_3 + 20 x_4 + 20 x_5 \geq 100 \quad \text{(Proteinas)}$$

$$\sum_{j \in Alimentos} a_{ij} x_j \geq b_i \quad i \in Nutrientes$$

Función Objetivo: Min $8 x_1 + 10 x_2 + 2 x_3 + 1.\ x_4 + 0.5 x_3$ $\qquad$ $\text{Min} \sum_{j \in Alimentos} c_j x_j$

OPTMODEL

		Carn	Peix	Cereals	Fruita	Pa	Aportació dia
Contingut per kg de menjar	Vitamines (u.i.)	25	200	300	-	-	60u.i./dia
	Hidrats Carboni (u.i.)	20	50	300	160	120	40u.i./dia
	Oligoelements (u.i.)	10	10	10	50	100	100u.i./dia
	Proteines (u.i.)	150	200	50	20	20	100 u.i./dia
	Preu (€/kg)	8	10	2	1.5	0.5	

Parámetros:

```
set NUTRIENTS = {'V','HC', 'O', 'P'};
set MENJARS   = {'carn', 'peix', 'cereals', 'fruita', 'pa'};

number   contingut{NUTRIENTS,  MENJARS} =
         [       25      200      300       0       0
                 20       50      300     160     120
                 10       10       10      50     100
                150      200       50      20      20
         ];

number apor{NUTRIENTS} = [60 40 100 100];

number preu{MENJARS}   = [8 10 2 1.5 0.5];
```

OPTMODEL

		Carn	Peix	Cereals	Fruita	Pa	Aportació dia
Contingut per kg de menjar	Vitamines (u.i.)	25	200	300	-	-	60u.i./dia
	Hidrats Carboni (u.i.)	20	50	300	160	120	40u.i./dia
	Oligoelements (u.i.)	10	10	10	50	100	100u.i./dia
	Proteines (u.i.)	150	200	50	20	20	100 u.i./dia
	Preu (€/kg)	8	10	2	1.5	0.5	

Variables: var Quantitat {MENJARS} >= 0;

Restricciones:

```
con Aportacio_min {i in NUTRIENTS}:
        sum {j in MENJARS} contingut[i,j]*Quantitat[j] >= apor[i];
```

Función Objetivo: min Total_cost = sum {j in MENJARS} preu[j]*Quantitat[j];

```
/* Paràmetres */
set NUTRIENTS = {'V','HC', 'O', 'P'};
set MENJARS   = {'carn', 'peix', 'cereals', 'fruita', 'pa'};
number contingut{NUTRIENTS, MENJARS} =
        [        25     200     300       0       0
                 20      50     300     160     120
                 10      10      10      50     100
                150     200      50      20      20
        ];
number apor{NUTRIENTS} = [60 40 100 100];
number preu{MENJARS}   = [8 10 2 1.5 0.5];

/* Model d'optimització */
 var Quantitat {MENJARS} >= 0;
min Total_cost = sum {j in MENJARS} preu[j]*Quantitat[j];
con Aportacio_min {i in NUTRIENTS}:
        sum {j in MENJARS} contingut[i,j]*Quantitat[j] >= apor[i];
```

```
/* Model extens */
expand;

/* Optimització i resultats */
solve;

print Quantitat.lb Quantitat.sol Quantitat.ub Quantitat.rc Quantitat.status;
print Aportacio_min.lb Aportacio_min.body Aportacio_min.ub
Aportacio_min.dual Aportacio_min.status;
```

The OPTMODEL Procedure

```
Var Quantitat[carn] >= 0
Var Quantitat[peix] >= 0
Var Quantitat[cereals] >= 0
Var Quantitat[fruita] >= 0
Var Quantitat[pa] >= 0
Minimize Total_cost=8*Quantitat[carn] + 10*Quantitat[peix] + 2*Quantitat[cereals] + 1.5*Quantitat[fruita] + 0.5*Quantitat[pa]
Restricción Aportacio_min[V]: 25*Quantitat[carn] + 200*Quantitat[peix] + 300*Quantitat[cereals] >= 60
Restricción Aportacio_min[HC]: 20*Quantitat[carn] + 50*Quantitat[peix] + 300*Quantitat[cereals] + 160*Quantitat[fruita] + 120*
Quantitat[pa] >= 40
Restricción Aportacio_min[O]: 10*Quantitat[carn] + 10*Quantitat[peix] + 10*Quantitat[cereals] + 50*Quantitat[fruita] + 100*
Quantitat[pa] >= 100
Restricción Aportacio_min[P]: 150*Quantitat[carn] + 200*Quantitat[peix] + 50*Quantitat[cereals] + 20*Quantitat[fruita] + 20*
Quantitat[pa] >= 100
```

The OPTMODEL Procedure

Resumen del problema	
Objective Sense	Minimization
Objective Function	Total_cost
Objective Type	Linear
Number of Variables	5
Bounded Above	0
Bounded Below	5
Bounded Below and Above	0
Free	0
Fixed	0
Number of Constraints	4
Linear LE (<=)	0
Linear EQ (=)	0
Linear GE (>=)	4
Linear Range	0
Coeficientes de restricción	18

Información de rendimiento	
Execution Mode	On Client
Number of Threads	1

Page Break

The OPTMODEL Procedure

Resumen de la solución	
Solver	LP
Algorithm	Dual Simplex
Objective Function	Total_cost
Solution Status	Optimal
Objective Value	2.65
Iterations	6
Primal Infeasibility	0
Dual Infeasibility	0
Bound Infeasibility	0

[1]	Quantitat.LB	Quantitat.SOL	Quantitat.UB	Quantitat.RC	Quantitat.STATUS
carn	0	0.0	1.7977E+308	4.1875	L
cereals	0	0.2	1.7977E+308	0.0000	B
fruita	0	0.0	1.7977E+308	1.0000	L
pa	0	4.5	1.7977E+308	-0.0000	B
peix	0	0.0	1.7977E+308	4.5000	L

[1]	Aportacio_min.LB	Aportacio_min.BODY	Aportacio_min.UB	Aportacio_min.DUAL	Aportacio_min.STATUS
HC	40	600	1.7977E+308	0.0000	B
O	100	452	1.7977E+308	0.0000	B
P	100	100	1.7977E+308	0.0250	U
V	60	60	1.7977E+308	0.0025	U

Problema de transport

c_{ij} $(10^6€/Hm^3)$	Mercats				
Refineries	1	2	3	4	Producció refineria (Hm^3)
1	4	7	9	10	6
2	6	4	3	6	10
3	9	6	4	8	4
Demanda (Hm^3)	5	3	8	4	

- Formuleu la modelització matemàtica parametritzada
- Implementeu i resoleu amb OPTMODEL
- Indiqueu els valors de la solució òptima: B^*, xB^*, N^*, r^* i λ^*

Modelo Optimización (papel)

c_{ij} $(10^6€/Hm^3)$	Mercats				
Refineries	1	2	3	4	Producció refineria (Hm^3)
1	4	7	9	10	6
2	6	4	3	6	10
3	9	6	4	8	4
Demanda (Hm^3)	5	3	8	4	

Variables de decisión: x_{ij}: cantidad de producto (Hm^3) enviado desde refinería i a mercado j, $i \in \{1, 2, 3\}$, $j \in \{1, 2, 3, 4\}$

Dominio : $x_{ij} \geq 0$, $i \in \{1, 2, 3\}$, $j \in \{1, 2, 3, 4\}$

Restricciones: $x_{i1} + x_{i2} + x_{i3} + x_{i4} \leq b_i$ (Refinería i) $\sum_{j=1}^{n} x_{ij} \leq b_i$

$x_{1j} + x_{2j} + x_{3j} \geq d_j$ (Mercado j) $\sum_{i=1}^{m} x_{ij} \geq d_j$

Función Objetivo: Min $\sum_{i \in Refinerias} \sum_{j \in Mercados} c_{ij} x_{ij}$

OPTMODEL

c_{ij} $(10^6€/Hm^3)$	Mercats				
Refineries	1	2	3	4	Producció refineria (Hm^3)
1	4	7	9	10	6
2	6	4	3	6	10
3	9	6	4	8	4
Demanda (Hm^3)	5	3	8	4	

Parámetros:

```
set<number> REFINERIES = 1..3;
set<number> MERCATS    = 1..4;

number produccio{ REFINERIES } = [ 6 10 4 ];
number demanda  { MERCATS }    = [5 3 8 4 ];
number cost     { REFINERIES, MERCATS } =
        [ 4     7     9     10
          6     4     3     6
          9     6     4     8];
```

OPTMODEL

c_{ij} $(10^6€/Hm^3)$	Mercats				
Refineries	1	2	3	4	Producció refineria (Hm^3)
1	4	7	9	10	6
2	6	4	3	6	10
3	9	6	4	8	4
Demanda (Hm^3)	5	3	8	4	

Variables: var Trans { REFINERIES, MERCATS } >= 0;

Restricciones:

```
con Produccio_cons {i in REFINERIES}:
        sum {j in MERCATS}   Trans[i,j] <= produccio[i];

con Demanda_cons   {i in MERCATS}   :
        sum {j in REFINERIES} Trans[i,j] >= demanda[i];
```

Función Objetivo: min Total_cost = sum {i in REFINERIES, j in MERCATS} cost[i,j] * Trans[i,j];

OPTMODEL

```
proc optmodel;

/* Parametres */
set<number> REFINERIES = 1..3;
set<number> MERCATS    = 1..4;

number produccio{ REFINERIES } = [ 6 10 4 ];
number demanda  { MERCATS }    = [5 3 8 4 ];
number cost     { REFINERIES, MERCATS } =
[4      7      9      10
 6      4      3      6
 9      6      4      8];

/* Optimization model */
var Trans { REFINERIES, MERCATS } >= 0;
min Total_cost = sum {i in REFINERIES, j in MERCATS} cost[i,j] * Trans[i,j];
con Produccio_cons {i in REFINERIES}:
        sum {j in MERCATS}   Trans[i,j] <= produccio[i];
con Demanda_cons   {i in MERCATS}   :
        sum {j in REFINERIES} Trans[i,j] >= demanda[i];
```

```
/* Model extens */
expand;

/* Optimització i resultats */
solve;

print Trans.lb Trans.sol Trans.ub Trans.rc Trans.status;

print Produccio_cons.lb Produccio_cons.body Produccio_cons.ub
Produccio_cons.dual Produccio_cons.status;

print Demanda_cons.lb Demanda_cons.body Demanda_cons.ub
Demanda_cons.dual Demanda_cons.status;
```

Hospital del mar

- 5 tipus diferents de mostres fluids.
- Cada màquina pot ser usada per a analitzar qualsevol tipus de mostra
- el temps (minuts) que triga cadascuna depèn del tipus de mostra
- Cada màquina es pot usar un màxim de 8h al dia

	Màquina			
Temps de processat (minuts/ml)	A	B	C	Volum (ml)
Mostra 1	3	5	2	80
Mostra 2	4	3	5	75
Mostra 3	4	5	3	80
Mostra 4	5	4	3	12
Mostra 5	3	5	4	60

Formuleu el problema de PL que permet trobar com distribuir de forma òptima les mostres entre les màquines i resoleu-lo amb OPTMODEL.

Hospital del mar: Modelo Optimización (papel)

	Màquina			
Temps de processat (minuts/ml)	A	B	C	Volum (ml)
Mostra 1	3	5	2	80
Mostra 2	4	3	5	75
Mostra 3	4	5	3	80
Mostra 4	5	4	3	12
Mostra 5	3	5	4	60

v_i: volumen de la muestra i

Variables de decisión: x_{ij}: cantidad (volumen en ml) de muestra i asignada a máquina j, $i \in \{1, 2, 3, 4, 5\}$, $j \in \{A, B, C\}$

Dominio : $x_{ij} \geq 0$, $i \in \mathcal{M}=\{1, 2, 3, 4, 5\}$, $j \in \mathcal{Q}=\{A, B, C\}$

Restricciones:

$$\sum_{j \in \mathcal{Q}} t_{ij} x_{ij} \geq v_i, \; i \in \mathcal{M} \qquad \text{(Muestra } i\text{)}$$

$$\sum_{i \in \mathcal{M}} t_{ij} x_{ij} \leq d_j, \; j \in \mathcal{Q} \qquad \text{(Máquina } j\text{)}$$

d_j: tiempo disponible máquina j (8 horas)

Función Objetivo:

$$\text{Min} \sum_{j \in \mathcal{M}} \sum_{j \in \mathcal{Q}} t_{ij} x_{ij}$$

OPTMODEL

proc optmodel;

/* Parametres */

```
set<string> MOSTRES= {'1', '2', '3', '4', '5'};
set<string> MAQUINES= {'A','B','C'};
number temps_proc{ MOSTRES, MAQUINES } = [ 3 5 2
   4 3 5
   4 5 3
   5 4 3
   3 5 4];
number volum_mostra{ MOSTRES } =  [80 75 80 12 60];
number temps{ MAQUINES }= [ 480 480 480 ];
```

/* Variables*/

```
var Volum {MOSTRES,MAQUINES} >= 0;
```

/* Restricciones*/

```
min Total_temps =
sum {i in MOSTRES, j in MAQUINES} temps_proc[i,j] * Volum[i,j];

con Temps_maq {j in MAQUINES}: sum {i in MOSTRES} temps_proc[i,j]*Volum[i,j] <= temps[j];

con Proces_mostra {i in MOSTRES}: sum {j in MAQUINES}Volum[i,j] >= volum_mostra[i];
```

/* Show the model */

expand;

/* Optimize and output */

solve;

```
print Volum.lb Volum.sol Volum.ub Volum.rc Volum.status;
print Temps_maq.lb Temps_maq.body Temps_maq.ub Temps_maq.dual Temps_maq.status;
print Proces_mostra.lb Proces_mostra.body Proces_mostra.ub Proces_mostra.dual
Proces_mostra.status;
```


```
print Volum.lb Volum.sol Volum.ub Volum.rc Volum.status;
print Temps_maq.lb Temps_maq.body Temps_maq.ub Temps_maq.dual Temps_maq.status;
print Proces_mostra.lb Proces_mostra.body Proces_mostra.ub Proces_mostra.dual
Proces_mostra.status;
```

```
Var Volum['1',A] >= 0
Var Volum['2',A] >= 0
Var Volum['3',A] >= 0
Var Volum['4',A] >= 0
Var Volum['5',A] >= 0
Var Volum['1',B] >= 0
Var Volum['2',B] >= 0
Var Volum['3',B] >= 0
Var Volum['4',B] >= 0
Var Volum['5',B] >= 0
Var Volum['1',C] >= 0
Var Volum['2',C] >= 0
Var Volum['3',C] >= 0
Var Volum['4',C] >= 0
Var Volum['5',C] >= 0
Minimize Total_temps=3*Volum['1',A] + 4*Volum['2',A] + 4*Volum['3',A] + 5*Volum['4',A] + 3*Volum['5',A] + 5*Volum['1',B] + 3*
Volum['2',B] + 5*Volum['3',B] + 4*Volum['4',B] + 5*Volum['5',B] + 2*Volum['1',C] + 5*Volum['2',C] + 3*Volum['3',C] + 3*Volum['4',C]
+ 4*Volum['5',C]
Restricción Temps_maq[A]: 3*Volum['1',A] + 4*Volum['2',A] + 4*Volum['3',A] + 5*Volum['4',A] + 3*Volum['5',A] <= 480
Restricción Temps_maq[B]: 5*Volum['1',B] + 3*Volum['2',B] + 5*Volum['3',B] + 4*Volum['4',B] + 5*Volum['5',B] <= 480
Restricción Temps_maq[C]: 2*Volum['1',C] + 5*Volum['2',C] + 3*Volum['3',C] + 3*Volum['4',C] + 4*Volum['5',C] <= 480
Restricción Proces_mostra['1']: Volum['1',A] + Volum['1',B] + Volum['1',C] >= 80
Restricción Proces_mostra['2']: Volum['2',A] + Volum['2',B] + Volum['2',C] >= 75
Restricción Proces_mostra['3']: Volum['3',A] + Volum['3',B] + Volum['3',C] >= 80
Restricción Proces_mostra['4']: Volum['4',A] + Volum['4',B] + Volum['4',C] >= 12
Restricción Proces_mostra['5']: Volum['5',A] + Volum['5',B] + Volum['5',C] >= 60
```

Page Break

The OPTMODEL Procedure

Resumen del problema	
Objective Sense	Minimization
Objective Function	Total_temps
Objective Type	Linear
Number of Variables	15
Bounded Above	0
Bounded Below	15
Bounded Below and Above	0
Free	0
Fixed	0
Number of Constraints	8
Linear LE (<=)	3
Linear EQ (=)	0
Linear GE (>=)	5
Linear Range	0
Coeficientes de restricción	30

Información de rendimiento	
Execution Mode	On Client
Number of Threads	1

[1]	[2]	Volum.LB	Volum.SOL	Volum.UB	Volum.RC	Volum.STATUS
1	A	0	0	1.7977E+308	1	L
1	B	0	0	1.7977E+308	3	L
1	C	0	80	1.7977E+308	0	B
2	A	0	0	1.7977E+308	1	L
2	B	0	75	1.7977E+308	0	B
2	C	0	0	1.7977E+308	2	L
3	A	0	0	1.7977E+308	1	L
3	B	0	0	1.7977E+308	2	L
3	C	0	80	1.7977E+308	0	B
4	A	0	0	1.7977E+308	2	L
4	B	0	0	1.7977E+308	1	L
4	C	0	12	1.7977E+308	0	B
5	A	0	60	1.7977E+308	0	B
5	B	0	0	1.7977E+308	2	L
5	C	0	0	1.7977E+308	1	L

[1]	Temps_maq.LB	Temps_maq.BODY	Temps_maq.UB	Temps_maq.DUAL	Temps_maq.STATUS
A	-1.7977E+308	180	480	0	B
B	-1.7977E+308	225	480	0	B
C	-1.7977E+308	436	480	0	B

[1]	Proces_mostra.LB	Proces_mostra.BODY	Proces_mostra.UB	Proces_mostra.DUAL	Proces_mostra.STATUS
1	80	80	1.7977E+308	2	U
2	75	75	1.7977E+308	3	U
3	80	80	1.7977E+308	3	U
4	12	12	1.7977E+308	3	U
5	60	60	1.7977E+308	3	U

Hospital del mar: Limitacions addicionals

- Cap mostra pot ocupar més del 50% del temps total de funcionament d'una màquina.

$$t_{ij}x_{ij} \leq \alpha \sum_{k \in \mathcal{M}} t_{kj}x_{kj} \qquad i \in \mathcal{M},\ j \in \mathcal{Q}$$

Temps de la mostra $i \in \mathcal{M}$ a la máquina $j \in \mathcal{Q}$

Temps total de funcionament de la màquina $j \in \mathcal{Q}$

en aquest exemple α= 0.5

- Cap màquina pot realitzar més del 40% de volum total de les proves

$$\sum_{i \in \mathcal{M}} x_{ij} \leq \beta \sum_{i \in \mathcal{M}} v_i \qquad j \in \mathcal{Q}$$

volum de les proves en la máquina $j \in \mathcal{Q}$

Volum total de les proves

en aquest exemple β= 0.4

OPTMODEL

Nuevos Parametros :

```
number alpha = 0.5;
number beta  = 0.4;
```

Nuevas Restricciones:

```
con Limit_b1 { i in MOSTRES, j in MAQUINES}:
        temps_proc[i,j]*Volum[i,j] <= alpha*sum{k in MOSTRES}   temps_proc[k,j]*Volum[k,j];

con Limit_b2 {j in MAQUINES}:
        sum{i in MOSTRES} Volum[i,j] <= beta*sum{i in MOSTRES} volum_mostra[i];
```

[1]	[2]	Volum.LB	Volum.SOL	Volum.UB	Volum.RC	Volum.STATUS
1	A	0	6.320	1.7977E+308	0.0000	B
1	B	0	0.000	1.7977E+308	0.9375	L
1	C	0	73.680	1.7977E+308	0.0000	B
2	A	0	27.988	1.7977E+308	0.0000	B
2	B	0	47.012	1.7977E+308	0.0000	B
2	C	0	0.000	1.7977E+308	1.9750	L
3	A	0	12.272	1.7977E+308	0.0000	B
3	B	0	18.608	1.7977E+308	0.0000	B
3	C	0	49.120	1.7977E+308	-0.0000	B
4	A	0	0.000	1.7977E+308	1.6875	L
4	B	0	12.000	1.7977E+308	0.0000	B
4	C	0	0.000	1.7977E+308	0.7500	L
5	A	0	60.000	1.7977E+308	0.0000	B
5	B	0	0.000	1.7977E+308	0.5625	L
5	C	0	0.000	1.7977E+308	1.5500	L

[1]	Temps_maq.LB	Temps_maq.BODY	Temps_maq.UB	Temps_maq.DUAL	Temps_maq.STATUS
A	-1.7977E+308	360.00	480	0	B
B	-1.7977E+308	282.07	480	0	B
C	-1.7977E+308	294.72	480	0	B

1]	Proces_mostra.LB	Proces_mostra.BODY	Proces_mostra.UB	Proces_mostra.DUAL	Proces_mostra.STATUS
	80	80	1.7977E+308	2.8125	U
	75	75	1.7977E+308	3.7500	U
	80	80	1.7977E+308	3.7500	U
	12	12	1.7977E+308	3.0000	U
	60	60	1.7977E+308	3.1875	U

[1]	[2]	Limit_b1.LB	Limit_b1.BODY	Limit_b1.UB	Limit_b1.DUAL	Limit_b1.STATUS
1	A	-1.7977E+308	-161.040	0	0.000	B
1	B	-1.7977E+308	-141.038	0	0.000	B
1	C	-1.7977E+308	0.000	0	-0.025	L
2	A	-1.7977E+308	-68.050	0	0.000	B
2	B	-1.7977E+308	0.000	0	-0.500	L
2	C	-1.7977E+308	-147.360	0	0.000	B
3	A	-1.7977E+308	-130.910	0	0.000	B
3	B	-1.7977E+308	-48.000	0	0.000	B
3	C	-1.7977E+308	0.000	0	0.000	B
4	A	-1.7977E+308	-180.000	0	0.000	B
4	B	-1.7977E+308	-93.037	0	0.000	B
4	C	-1.7977E+308	-147.360	0	0.000	B
5	A	-1.7977E+308	0.000	0	-0.125	L
5	B	-1.7977E+308	-141.038	0	0.000	B
5	C	-1.7977E+308	-147.360	0	0.000	B

[1]	Limit_b2.LB	Limit_b2.BODY	Limit_b2.UB	Limit_b2.DUAL	Limit_b2.STATUS
A	-1.7977E+308	106.58	122.8	0.0000	B
B	-1.7977E+308	77.62	122.8	0.0000	B
C	-1.7977E+308	122.80	122.8	-0.7875	L

Coalco

Cada client pot rebre carbó d'una única mina o de totes dues, mesclant, en aquest últim cas, els dos tipus de carbó rebut. En tot cas, la composició del carbó rebut, ja sigui d'una única mina o per mescla de totes dues, no pot contenir més d'un 8% de cendres i d'un 7% de sulfur.

Cost de transport (€/Tm)	Client 1	Client 2	Cost de producció (€/Tm)	Capacitat (Tm)	Contingut en cendra (Tm/Tm carbó)	Contingut en sulfur (Tm/Tm carbó)
Mina 1	4	6	50	120	0.1	0.04
Mina 2	9	6	55	100	0.05	0.09
Demanda (Tm)	90	110				

Coalco: Modelo Optimización (papel)

Cost de transport (€/Tm)	Client 1	Client 2	Cost de producció (€/Tm)	Capacitat (Tm)	Contingut en cendra (Tm/Tm carbó)	Contingut en sulfur (Tm/Tm carbó)
Mina 1	4	6	50	120	0.1	0.04
Mina 2	9	6	55	100	0.05	0.09
Demanda (Tm)	90	110				

Variables de decisión: x_{ij}: cantidad (en Tm) transportada desde mina i a cliente j,

$i \in \{1, 2\}$, $j \in \{1,2\}$

Dominio : $x_{ij} \geq 0$, i∈ {1, 2, 3, 4, 5}, j∈ $\mathcal{Q}$={A, B, C}

Restricciones:

$$\sum_{j\in\{1,2\}} x_{ij} \geq b_i,\ i \in \{1,\ 2\} \quad \text{(Capacidad mina } i\text{)}$$

$$\sum_{i\in\{1,2\}} x_{ij} \leq d_j,\ j\in \{1,2\} \quad \text{(Demanda cliente } j\text{)}$$

$$\sum_{i=\{1,2\}} \boxed{\alpha_{ik}} x_{ij} \leq \overline{\alpha_k} \sum_{i\in\{1,2\}} x_{ij} \ \text{(contenido de elemento k)}$$

Función Objetivo:

$$\text{Min} \sum_{j\in\mathcal{M}} \sum_{j\in\mathcal{Q}} (p_i + t_{ij}) x_{ij}$$

Coalco: OPTMODEL

Cost de transport (€/Tm)	Client 1	Client 2	Cost de producció (€/Tm)	Capacitat (Tm)	Contingut en cendra (Tm/Tm carbó)	Contingut en sulfur (Tm/Tm carbó)
Mina 1	4	6	50	120	0.1	0.04
Mina 2	9	6	55	100	0.05	0.09
Demanda (Tm)	90	110				

Parámetros:

Nombre de mines	number nM = 2;
Nombre de clients	number nC = 2;
Components carbó	set <string> C ={'cendra', 'sulfur'};
Cost transport mina i →client j (€/Tm): t_{ij}	number t{ 1..nM , 1..nC } = [4 6 9 6];
Per a cada mina $i = 1, \dots, n^M$	
• Cost producció (€/Tm): p_i	number p{ 1..nM } = [50 55];
• Capacitat mina i (Tm): b_i	number b{ 1..nM } = [120 100];
• Contingut component $k \in \mathcal{C}$ (Tm/Tm carbó): α_{ik}	number al{ 1..nM , C } = [0.10 0.04 0.05 0.09];
Contingut màxim component $k \in \mathcal{C}$ carbó mescla (Tm/Tm mescla): $\overline{\alpha}_k$	number almax{ C } = [0.08 0.07];
Demanda client j (Tm): d_j	number d{ 1..nC } = [90 110];

Coalco: OPTMODEL

Cost de transport (€/Tm)	Client 1	Client 2	Cost de producció (€/Tm)	Capacitat (Tm)	Contingut en cendra (Tm/Tm carbó)	Contingut en sulfur (Tm/Tm carbó)
Mina 1	4	6	50	120	0.1	0.04
Mina 2	9	6	55	100	0.05	0.09
Demanda (Tm)	90	110				

Variables:

Tones a transportar mina $i \rightarrow$ client j:	var X { 1..nM, 1..nC } >= 0;

Restriccions:

Capacitat mines:	$\sum_{j=1}^{n^C} x_{ij} \leq b_i$ $i = 1, \ldots, n^M$	con Capacitat { i in 1..nM } : sum{ j in 1..nC } X[i,j] <=b[i];
Demanda clients:	$\sum_{i=1}^{n^M} x_{ij} \geq d_j$ $j = 1, \ldots, n^C$	con Demanda { j in 1..nC } : sum{ i in 1..nM }X[i,j] >= d[j];
Continguts màxims:	$\sum_{i=1}^{n^M} \alpha_{ik} x_{ij} \leq \bar{\alpha}_k \sum_{i=1}^{n^M} x_{ij}$ $k \in \mathcal{C}, j = 1, \ldots, n^C$	con Contingut { j in 1..nC, k in C} : sum{ i in 1..nM } al[i,k)*X[i,j] <= almax[k]*sum{ i in 1..nM } X[i,j] ;

Coalco: OPTMODEL

Cost de transport (€/Tm)	Client 1	Client 2	Cost de producció (€/Tm)	Capacitat (Tm)	Contingut en cendra (Tm/Tm carbó)	Contingut en sulfur (Tm/Tm carbó)
Mina 1	4	6	50	120	0.1	0.04
Mina 2	9	6	55	100	0.05	0.09
Demanda (Tm)	90	110				

Función objetivo:

Cost total producción més transport:	$\min z = \sum_{i=1}^{n^M} \sum_{j=1}^{n^C} (p_i + t_{ij}) x_{ij}$	min Cost_total = sum{ i in 1..nM , j in 1..nC } (p[i]+t[i,j])*X[i,j];

/* Model extens */

```
expand;
```

/* Optimització */

```
solve;
```

/* Sortida resultats */

```
print X.lb X.sol X.rc X.status;
print Capacitat.lb Capacitat.body Capacitat.ub Capacitat.dual Capacitat.status;
print Demanda.lb Demanda.body Demanda.ub Demanda.dual Demanda.status;
print Contingut.lb Contingut.body Contingut.ub Contingut.dual Contingut.status;
```

/* Model extens */

```
Var X[1,1] >= 0
Var X[1,2] >= 0
Var X[2,1] >= 0
Var X[2,2] >= 0
Minimize Cost_total=54*X[1,1] + 56*X[1,2] + 64*X[2,1] + 61*X[2,2]
Restricción Capacitat[1]: X[1,1] + X[1,2] <= 120
Restricción Capacitat[2]: X[2,1] + X[2,2] <= 100
Restricción Demanda[1]: X[1,1] + X[2,1] >= 90
Restricción Demanda[2]: X[1,2] + X[2,2] >= 110
Restricción Contingut[1,cendra]: 0.02*X[1,1] - 0.03*X[2,1] <= 0
Restricción Contingut[1,sulfur]: - 0.03*X[1,1] + 0.02*X[2,1] <= 0
Restricción Contingut[2,cendra]: 0.02*X[1,2] - 0.03*X[2,2] <= 0
Restricción Contingut[2,sulfur]: - 0.03*X[1,2] + 0.02*X[2,2] <= 0
```

Page Break

The OPTMODEL Procedure

Resumen del problema	
Objective Sense	Minimization
Objective Function	Cost_total
Objective Type	Linear
Number of Variables	4
Bounded Above	0
Bounded Below	4
Bounded Below and Above	0
Free	0
Fixed	0
Number of Constraints	8
Linear LE (<=)	6
Linear EQ (=)	0
Linear GE (>=)	2
Linear Range	0
Coeficientes de restricción	16

Información de rendimiento	
Execution Mode	On Client
Number of Threads	1

Page Break

The OPTMODEL Procedure

Resumen de la solución	
Solver	LP
Algorithm	Dual Simplex
Objective Function	Cost_total
Solution Status	Optimal
Objective Value	11600
Iterations	7
Primal Infeasibility	0
Dual Infeasibility	0
Bound Infeasibility	0

[1]	[2]	X.LB	X.SOL	X.RC	X.STATUS
1	1	0	54	7.7716E-16	B
1	2	0	66	3.8858E-16	B
2	1	0	36	2.2204E-16	B
2	2	0	44	1.1102E-16	B

[1]	Capacitat.LB	Capacitat.BODY	Capacitat.UB	Capacitat.DUAL	Capacitat.STATUS
1	-1.7977E+308	120	120	0	B
2	-1.7977E+308	80	100	0	B

[1]	Demanda.LB	Demanda.BODY	Demanda.UB	Demanda.DUAL	Demanda.STATUS
1	90	90	1.7977E+308	58	U
2	110	110	1.7977E+308	58	U

[1]	[2]	Contingut.LB	Contingut.BODY	Contingut.UB	Contingut.DUAL	Contingut.STATUS
1	cendra	-1.7977E+308	0.0	0	-200	L
1	sulfur	-1.7977E+308	-0.9	0	0	B
2	cendra	-1.7977E+308	-0.0	0	-100	L
2	sulfur	-1.7977E+308	-1.1	0	0	B

Page Break

Obs	name	fullname	prodcode	prodmod	installed	licensed
1	BASE	Base	prodnum0	sasmeans	9.03 TS1M2	1
2	STAT	SAS/STAT	stat	sasglm	9.03 TS1M2	1
3	GRAPH	SAS/GRAPH	graph	sasgchar	9.03 TS1M2	1
4	GIS	SAS/GIS	prodnum28	sasgis		0
5	ETS	SAS/ETS	ets	sasarima	9.03 TS1M2	1
6	QC	SAS/QC	qc	sascapab	9.03 TS1M2	1
7	CONNECT	SAS/CONNECT	prodnum15	saszrlnk	9.03 TS1M2	1
8	ACCPC	SAS/ACCESS PC Files	prodnum208	sasdbf	9.03 TS1M2	1