SOLUCIONS

SOLUCIÓ EXERCICI 1. Planificació de la producció*.

(EXERCICI 1)

a) Formulació: veure apunts de IIO.
b) Implementació i resolució amb OPTMODEL

```
proc optmodel presolver = 0;

/* Paràmetres */
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];

/* Model d'optimització */

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

/* 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

Solution Summary	
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

c) Informació sobre la solució:

$$\begin{cases} \mathcal{B}^* = & \{A, \quad B, \quad fusta\} \\ x_B^{*\prime} = & [50 \quad 50 \quad 50] \end{cases}, \begin{cases} \mathcal{N}^* = & \{ma, \quad plastic\} \\ r^{*\prime} = & [-125.0 \quad -87.5] \end{cases}$$

Els costos reduïts de les variables de folga coincideixen sempre amb les variables duals o preus ombra de les constriccions de $\leq$ associades (sufix .DUAL) canviats de signe: $r_j^* = -\lambda_j^*, j = ma, plastic$. Aquests costos reduïts son negatius a l'òptim del problema resolt perquè està plantejat com a problema de maximització. Podeu provar a formular-lo com a problema de minimització i veureu com el signe de les variables duals i, conseqüentment, el dels costos reduïts, canvien, tot i que la relació $r_j^* = -\lambda_j^*$ continua sent vàlida.

SOLUCIÓ EXERCICI 2. Problema de la dieta.

(EXERCICI 2)

a) Formulació: veure apunts de IIO.
b) Implementació i resolució amb OPTMODEL

```
proc optmodel presolver = 0;

/* Paràmetres */
set<string> NUTRIENTS = {'V','HC', 'O', 'P'};
set<string> MENJARS   = {'carn', 'peix', 'cereals', 'fruita', 'pa'};
number contingut{ MENJARS, NUTRIENTS} =
[      25   20      10   150
      200   50      10   200
      300   300     10    50
        0   160     50    20
        0   120    100    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 {i in MENJARS} preu[i]*Quantitat[i];
con Aportacio_min {j in NUTRIENTS}:
      sum {i in MENJARS} contingut[i,j]*Quantitat[i] >= apor[j];

/* 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

Solution Summary

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

c) Informació sobre la solució:

$$\begin{cases} \mathcal{B}^* = & \{cereals, & pa, & HC, & O & \} \\ x_B^{*\prime} = & [0.2 & 4.5 & 560 & 352 &] \end{cases}, \begin{cases} \mathcal{N}^* = & \{carn, & fruita, & peix, & P, & V\} \\ r^{*\prime} = & [4.1875 & 1 & 4.5 & 0.025 & 0.0025] \end{cases}$$

En aquest cas els costos reduïts de les variables d'escreix de les contriccions P i V coincideixen amb les variables duals o preus ombra d'aquestes constriccions: $r_j^* = \lambda_j^*, j = P, V$

SOLUCIÓ EXERCICI 3. Problema de mescla.

(EXERCICI 3)

a) Formulació: veure apunts de IIO.

b) Implementació i resolució amb OPTMODEL

```
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 {i in DISOLVENTS} cost[i]*Proporcio[i];
con Contingut_minim {j in COMPONENTS_MIN}:
      sum {i in DISOLVENTS} contingut[j,i]*Proporcio[i] >= mescla[j];
con Contingut_maxim {j in COMPONENTS_MAX}:
      sum {i in DISOLVENTS} contingut[j,i]*Proporcio[i] <= mescla[j];
con Cons_mescla: sum{i in DISOLVENTS} Proporcio[i] = 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;
print Cons_mescla.lb Cons_mescla.body Cons_mescla.ub Cons_mescla.dual
Cons_mescla.status;
```

```
                         The OPTMODEL Procedure

                            Solution Summary

                   Solver                        Dual Simplex
                   Objective Function              Total_cost
                   Solution Status                    Optimal
                   Objective Value                         11
                   Iterations                               2

                   Primal Infeasibility                     0
                   Dual Infeasibility                       0
                   Bound Infeasibility                      0

            Proporcio.    Proporcio.    Proporcio.    Proporcio.    Proporcio.
     [1]           LB           SOL            UB            RC    STATUS
       1            0           0.0    1.79769E308           4.5    L
       2            0           0.5    1.79769E308           0.0    B
```

3	0	0.5	1.79769E308	0.0	B
4	0	0.0	1.79769E308	0.5	L

[1]	Contingut_minim.LB	Contingut_minim.BODY	Contingut_minim.UB	Contingut_minim.DUAL	Contingut_minim.STATUS
Clor	90	105	1.79769E308	0	B

[1]	Contingut_maxim.LB	Contingut_maxim.BODY	Contingut_maxim.UB	Contingut_maxim.DUAL	Contingut_maxim.STATUS
Amoniac	-1.7977E308	4	4	-0.5	L

Cons_mescla.LB	Cons_mescla.BODY	Cons_mescla.UB	Cons_mescla.DUAL	Cons_mescla.STATUS
1	1	1	13	U

c) Informació sobre la solució:

$$\begin{cases} \mathcal{B}^* = & \{2, \quad 3 \quad Clor\} \\ x_B^{*\prime} = & [0.5 \quad 0.5 \quad 15] \end{cases}, \begin{cases} \mathcal{N}^* = & \{1, \quad 4, \quad Amoniac\} \\ r^{*\prime} = & [4.5 \quad 0.5 \quad 0.5] \end{cases} \; (r^*_{Amoniac} = -\lambda^*_{Amoniac})$$

SOLUCIÓ EXERCICI 4. Problema de transport.

(EXERCICI 4)

a) Formulació: veure apunts de IIO.

b) Implementació i resolució amb 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    {j in MERCATS}    :
      sum {i in REFINERIES} Trans[i,j] >= demanda[j];

/* Formulacío estensa */
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;
```

```
                              The OPTMODEL Procedure

                                 Solution Summary

                       Solver                        Dual Simplex
                       Objective Function              Total_cost
                       Solution Status                    Optimal
                       Objective Value                         87
                       Iterations                               7

                       Primal Infeasibility                     0
                       Dual Infeasibility                       0
                       Bound Infeasibility                      0
```

[1]	[2]	Trans.LB	Trans.SOL	Trans.UB	Trans.RC	STATUS
1	1	0	5	1.79769E308	0	B
1	2	0	1	1.79769E308	0	B
1	3	0	0	1.79769E308	3	L
1	4	0	0	1.79769E308	1	L
2	1	0	0	1.79769E308	5	L
2	2	0	2	1.79769E308	0	B

2	3	0	4	1.79769E308	0	B
2	4	0	4	1.79769E308	0	B
3	1	0	0	1.79769E308	7	L
3	2	0	0	1.79769E308	1	L
3	3	0	4	1.79769E308	0	B
3	4	0	0	1.79769E308	1	L

[1]	Produccio_ cons.LB	Produccio_ cons.BODY	Produccio_ cons.UB	Produccio_ cons.DUAL	Produccio_ cons.STATUS
1	-1.7977E308	6	6	0	B
2	-1.7977E308	10	10	-3	L
3	-1.7977E308	4	4	-2	L

[1]	Demanda_ cons.LB	Demanda_ cons. BODY	Demanda_ cons.UB	Demanda_ cons. DUAL	Demanda_ cons. STATUS
1	5	5	1.79769E308	4	U
2	3	3	1.79769E308	7	U
3	8	8	1.79769E308	6	U
4	4	4	1.79769E308	9	U

c) Informació sobre la solució:

$$\begin{cases} \mathcal{B}^* = & \{(1,1), & (1,2), & (2,2), & (2,3), & (2,4), & (3,3), & Prod[1]\} \\ x_B^{*\prime} = & [5 & 1 & 2 & 4 & 5 & 4 & 0] \end{cases},$$

$$\begin{cases} \mathcal{N}^* = & \{(1,3), & (1,4), & (2,1), & (3,1), & (3,2), & (3,4), & Prod[2], & Prod[3], & Dem[1] & Dem[1], & Dem[3], & Dem[4]\} \\ r^{*\prime} = & [3 & 1 & 5 & 7 & 1 & 1 & 3 & 2 & 4 & 7 & 6 & 9] \end{cases}$$

$$\left(r^*_{Prod[j]} = -\lambda^*_{Prod[j]}, r^*_{Dem[j]} = \lambda^*_{Dem[j]}\right)$$

SOLUCIÓ EXERCICI 5. Prodem S.L.

(EXERCICI 5)

Model de producció i demanda:

Fitxer `Prodem.sas`

```
proc optmodel presolver = 0;

/* Paràmetres */
set<str> PRODUCTE = {'A','B', 'C'};
number consum{PRODUCTE} = [ 3 2 1 ];
number disp = 40;
number dem  = 33;
number cost{PRODUCTE} = [ 10 2 3 ];

/* Model d'optimització */
var Produc {PRODUCTE} >= 0;
max Total_benefici =
      sum {i in PRODUCTE} cost[i]*Produc[i];
con Consum_recurs:
      sum {i in PRODUCTE} consum[i]*Produc[i] <= disp;
con Demanda : sum{i in PRODUCTE} Produc[i] >= dem;

/* Model extens */
expand;

/* Optimització i resultats */
solve;

print _var_.name _var_.lb _var_.sol _var_.ub _var_.rc _var_.status;
print _con_.name _con_.lb _con_.body _con_.ub _con_.dual _con_.status;
```

Solució:

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB	_VAR_.RC	_VAR_.STATUS
1	Produc[A]	0	3.5	1.7977E+308	9.9920E-16	B
2	Produc[B]	0	0.0	1.7977E+308	-4.5000E+00	L
3	Produc[C]	0	29.5	1.7977E+308	1.1102E-16	B

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_CON_.UB	_CON_.DUAL	_CON_.STATUS
1	Consum_recurs	-1.7977E308	40	40	3.5	L
2	Demanda	33	33	1.7977E308	-0.5	U

Informació sobre l'òptim:

Variables	x_1	x_2	x_3
Valor a l'òptim x^*	3.5	0.0	29.5
Estat ($\mathcal{B},\mathcal{N}$)	$\mathcal{B}$	$\mathcal{N}$	$\mathcal{B}$
Cost reduït (r_i^*)	0.0	-4.5	0.0
Costos de producció (z^*)	123.5		

Constriccions	***Recurs***	***Demanda***
Valor folga/escreix a l'òptim	0.0	0.0
Estat ($\mathcal{B},\mathcal{N}$)	$\mathcal{N}$	$\mathcal{N}$
Variable dual (λ_j^*)	3.5	-0.5

SOLUCIÓ EXERCICI 6. Coalco.

(EXERCICI 5)

- **Fitxers:** `Coalco.sas`

Paràmetres:		
Nombre de mines	$n^M = 2$	`number nM = 2;`
Nombre de clients	$n^C = 2$	`number nC = 2;`
Components carbó	$\mathcal{C} = \{cendra, sulfur\}$	`set <string> C ={'cendra', 'sulfur'};`
Cost transport mina i →client j (€/Tm)	t_{ij} $i = 1, \dots, n^M$ $j = 1, \dots, n^C$ $t = \begin{bmatrix} 4 & 6 \\ 9 & 6 \end{bmatrix}$	`number t{ 1..nM , 1..nC } = [4 6 9 6];`
Per a cada mina $i = 1, \dots, n^M$		
• Cost producció (€/Tm)	$p_i \ , p = [50 \quad 55]'$	`number p{ 1..nM } = [ 50 55 ];`
• Capacitat mina i (Tm)	$b_i \ , b = [120 \quad 100]'$	`number b{ 1..nM } = [ 120 100];`
• Contingut component $k \in \mathcal{C}$ (Tm/Tm carbó):	$\alpha_{ik}, \alpha = \begin{bmatrix} 0.1 & 0.04 \\ 0.05 & 0.09 \end{bmatrix}$	`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):	$\overline{\alpha}_k, \overline{\alpha} = [0.08 \quad 0.07]$	`number almax{ C } = [ 0.08 0.07];`
Demanda client j (Tm)	$d_j \ , j = 1, \dots, n^C$ $d = [90 \quad 110]'$	`number d{ 1..nC } = [90 110];`

Variables		
Tones a transportar mina i → client j:	$x_{ij} \geq 0$ $i = 1, \dots, n^M$ $j = 1, \dots, n^C$	`var X { 1..nM, 1..nC } >= 0;`

Model de programació lineal

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];`
Capacitat mines:	s.a: $\sum_{j=1}^{n^C} x_{ij} \le b_i$ $i = 1, \dots, n^M$	`con Capacitat { i in 1..nM } : sum{ j in 1..nC } X[i,j] <=b[i];`
Demanda clients:	$\sum_{i=1}^{n^M} x_{ij} \ge d_j$ $j = 1, \dots, n^C$	`con Demanda { j in 1..nC } : sum{ i in 1..nM }X[i,j] >= d[j];`
Continguts màxims:	$\frac{\sum_{i=1}^{n^M}\alpha_{ik}x_{ij}}{\sum_{i=1}^{n^M}x_{ij}} \le \bar{\alpha}_k \rightarrow$ $\rightarrow \sum_{i=1}^{n^M}(\alpha_{ik} - \bar{\alpha}_k)x_{ij} \le 0$ $k \in \mathcal{C}, j = 1, \dots, n^C$ $x_{ij} \ge 0$ $i = 1, \dots, n^M, j = 1, \dots, n^C$	`con Contingut { j in 1..nC, k in C} : sum{ i in 1..nM } (al[i,k]-almax[k])*X[i,j] <= 0;`

Solució òptima

```
solve;
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;
```

```
                         The OPTMODEL Procedure

                            Solution Summary

                   Solver                     Dual Simplex
                   Objective Function           Cost_total
                   Solution Status                 Optimal
                   Objective Value                   11600
                   Iterations                            4

                   Primal Infeasibility                  0
                   Dual Infeasibility                    0
                   Bound Infeasibility                   0
```

[1]	[2]	X.LB	X.SOL	X.RC	X.STATUS
1	1	0	54	0	B
1	2	0	66	0	B
2	1	0	36	0	B
2	2	0	44	0	B

[1]	Capacitat. LB	Capacitat. BODY	Capacitat. UB	Capacitat. DUAL	Capacitat. STATUS
1	-1.7977E308	120	120	0	B
2	-1.7977E308	80	100	0	B

[1]	Demanda. LB	Demanda. BODY	Demanda.UB	Demanda. DUAL	Demanda. STATUS
1	90	90	1.79769E308	58	U
2	110	110	1.79769E308	58	U

[1]	[2]	Contingut. LB	Contingut. BODY	Contingut. UB	Contingut. DUAL	Contingut. STATUS
1	cendra	-1.7977E308	0.0	0	-200	L
1	sulfur	-1.7977E308	-0.9	0	0	B
2	cendra	-1.7977E308	0.0	0	-100	L
2	sulfur	-1.7977E308	-1.1	0	0	B

SOLUCIÓ EXERCICI 7. CSL

(EXERCICI 7)

- **Fitxers:** `CSL.sas`

Paràmetres:		
Nombre de mesos	$m = 5$	`number m = 5;`
Demanda projectes mes i (h)	$d_i, i = 1,2,\ldots,m$ $b = \begin{bmatrix} 3000 \\ 4000 \\ 7500 \\ 10000 \\ 15000 \end{bmatrix}$	`number d{ i in 1..m } = [3000 4000 7500 10000 15000];`

Consultors qualificats:
 • Nre. inicial
 • Hores treball
 • Cost (€)
 • Fracció continuitat
	$n^Q = 35$
	 $h^Q = 160$
	 $c^Q = 1800$
	 $\alpha = 0.95$
		`number nq = 35;`
		 `number hq = 160;`
		 `number cq = 1800;`
		 `number alpha = 0.95;`

Consultors en formació: • Hores supervisió i-èssim mes en formació • Cost (€)	$h^F = [50\ 10]'$ $c^F = 900$	`number hf {1..2} = [50 10];` `number cf = 900;`

Variables		
Nombre de consultors nous contractats el mes i (en primer mes de formació)	$x_i \geq 0$ $i = 0,1,\ldots,m$	`var X { 0..m } >= 0;`
Nombre de consultors en segon mes de formació en el mes i	$y_{i\_} \geq 0$ $i = 0,1,\ldots,m$	`var Y { 0..m } >= 0;`
Nombre de consultors qualificats mes i	$z_i \geq 0$ $i = 0,1,\ldots,m$	`var Z { 0..m } >= 0;`

Model de programació lineal

Cost total nòmina	$\min z = \sum_{i=1}^{m}[c^Q z_i + c^F(x_i + y_i)]$	`min Cost_nomina = sum{ i in 1..m} ( cq*Z[i] + cf*(X[i]+Y[i]) );`
Els consultors qualificats han de fer front als projectes més formació	s.a: $h^Q z_i - h_1^F x_i - h_2^F y_i \geq d_i$ $i = 1,2,\ldots,m$	`con Hores_mes { i in 1..m }: hq*Z[i]-hf[1]*X[i]-hf[2]*Y[i] >= d[i];`
Cada més es disposa del 95% dels consultors qualificats al final del mes anterior	$z_i - \alpha(z_{i-1} + y_{i-1}) \leq 0$ $i = 1,2,\ldots,m$	`con Cons_qualif { i in 1..m }: Z[i] - alpha*(Z[i-1]+Y[i-1]) <= 0;`
Els consultors del mes i en primer mes de formació pasen a segon al mes $i+1$	$y_i - x_{i-1} = 0, i = 1,\ldots,m$	`con Cons_form { i in 1..m }: Y[i] - X[i-1] = 0;`

Valors inicials:	$z_0 = n^Q$
	 $x_0 = y_0 = 0$
		`fix Z[0] = nq;`
		 `fix X[0] = 0;`
		 `fix Y[0] = 0;`

	$x_i, y_i, z_i \geq 0, i = 1,2,\ldots,m$	

Solució òptima

```
solve;
number total_hq{ i in 1..m} = hq*Z[i].sol;
number total_d { i in 1..m} = d[i]+hf[1]*X[i].sol+hf[2]*Y[i].sol;
print X Y Z total_hq total_d;
```

```
                          Solution Summary

               Solver                     Dual Simplex
               Objective Function          Cost_nomina
               Solution Status                 Optimal
               Objective Value            635903.97939
               Iterations                            7

               Primal Infeasibility       9.094947E-13
               Dual Infeasibility                    0
               Bound Infeasibility                   0
```

[1]	X	Y	Z	total_hq	total_d
0	0.0000	0.0000	35.000		
1	31.1564	0.0000	31.620	5059.2	4557.8
2	9.8942	31.1564	30.039	4806.3	4806.3
3	34.0557	9.8942	58.136	9301.7	9301.7
4	0.0000	34.0557	64.628	10340.6	10340.6
5	0.0000	0.0000	93.750	15000.0	15000.0

SOLUCIÓ EXERCICI 8. Hospital del Mar

(EXERCICI 8)

a) **Fitxers:** HMa.sas, HMb.sas, HMDB.sas

Paràmetres:		
Conjunt de mostres	$\mathcal{M} = \{1,2,\ldots,5\}$	`set<string> MOSTRES= {'1', '2', '3', '4', '5'};`
Conjunt de màquines	$Q = \{A,B,C\}$	`set<string> MAQUINES= {'A','B','C'};`
Temps de processat d'un mililitre de la mostra tipus i si s'assigna a la màquina j (minuts)	$t_{ij}, i \in \mathcal{M}, j \in Q$ $T = \begin{bmatrix} 3 & 5 & 2 \\ 4 & 3 & 5 \\ 4 & 5 & 3 \\ 5 & 4 & 3 \\ 3 & 5 & 4 \end{bmatrix}$	`number temps_proc{ MOSTRES, MAQUINES } = [ 3 5 2 4 3 5 4 5 3 5 4 3 3 5 4];`
Volum de mostres a analitzaar del tipus i (ml)	$v_i, i \in \mathcal{M}$ $v = [80\ 75\ 80\ 12\ 60]'$	`number volum_mostra{ MOSTRES } = [80 75 80 12 60];`
Temps diari disponible de les màquines (minuts)	$d_j = 480, j \in Q$	`number temps{ MAQUINES }= [ 480 480 480 ];`

Variables		
Nombre d'unitats de la mostra tipus i assignats a la màquina j	$x_{ij} \geq 0$ $i \in \mathcal{M}, j \in Q$	`var Volum {MOSTRES,MAQUINES} >= 0;`

Model de programació lineal		
Temps total de processat de totes les mostres (minuts)	$\min z = \sum_{i \in \mathcal{M}, j \in Q} t_{ij} x_{ij}$	`min Total_temps = sum {i in MOSTRES, j in MAQUINES} temps_proc[i,j] * Volum[i,j];`
Límit de temps disponible de cada màquina (minuts)	s.a: $\sum_{i \in \mathcal{M}} t_{ij} x_{ij} \leq d_j, j \in Q$	`con Temps_maq {j in MAQUINES}: sum {i in MOSTRES} temps_proc[i,j]*Volum[i,j] <= temps[j];`
Es processa el volum total de cada mostra (ml)	$\sum_{j \in Q} x_{ij} \geq v_i, i \in \mathcal{M}$ $x_{ij} \geq 0, i \in \mathcal{M}, j \in Q$	`con Proces_mostra {i in MOSTRES}: sum {j in MAQUINES} Volum[i,j] >= volum_mostra[i];`

Solució òptima (taula `HM.Maquines`)

```
solve;
print Volum.sol ;
```

```
          Volum.SOL
          A      B      C
1         0      0     80
2         0     75      0
3         0      0     80
4         0      0     12
5        60      0      0
```

b) S'haurien d'afegir els següents paràmetres i constriccions:

Paràmetres:

Fràcció màxima de temps de funcionament:	$\alpha = 0.5$	`number alpha = 0.5;`
Fràcció màxima de volum:	$\beta = 0.4$	`number beta  = 0.4;`

Constriccions:

b.1) Cap mostra més d'una fracció α del temps total de funcionament d'una màquina	$t_{ij}x_{ij} \leq \alpha \sum_{k\in\mathcal{M}} t_{kj}x_{kj}$
	 $i \in \mathcal{M}, j \in \mathcal{Q}$
		`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] <= 0;`

b.2) Cap màquina més d'una fracció β del volum total de les proves:	$\sum_{i\in\mathcal{M}} x_{ij} \leq \beta \sum_{i\in\mathcal{M}} v_i$
	 $j \in \mathcal{Q}$
		`con Limit_b2 {j in MAQUINES}:`
		 `sum{i in MOSTRES}`
		 `Volum[i,j] <=`
		 `beta*sum{i in MOSTRES} volum_mostra[i];`

Comparativa:

```
set CAS = {'a', 'b1', 'b2'};          /* Conjunt de casos a comparar */
string c;
drop Limit_b1; /* Es desactiven cons. Limit_b1 */
drop Limit_b2; /* Es desactiven cons. Limit_b2 */
do c=CAS;
      if c='b1' then restore Limit_b1; /* s'activa Limit_b1 */
      if c='b2' then restore Limit_b2; /* s'activa Limit_b1 */
      solve;
      print Volum.sol ;
end;
```

Resultat:

```
Cas a) Volum.SOL
       A      B      C
1      0      0     80
2      0     75      0
3      0      0     80
4      0      0     12
5     60      0      0
```

```
Cas b.1) Volum.SOL
         A        B        C
1   0.0000   0.0000  80.0000
2  45.0000  30.0000   0.0000
3   0.0000  16.5185  63.4815
4   0.0000   1.8519  10.1481
5  60.0000   0.0000   0.0000
```

```
Cas b.2) Volum.SOL
        A       B       C
1   6.320   0.000  73.680
2  27.988  47.013   0.000
3  12.273  18.608  49.120
4   0.000  12.000   0.000
5  60.000   0.000   0.000
```

SOLUCIÓ EXERCICI 9. Pelletier***.

- **Fitxers:** `Pelletier.sas`

Paràmetres:		
Nombre de mesos	$m = 5$	`number m init 5;`
Espai adicional mes i (en $10^3 m^2$)	$b_i, i = 1,2,\dots,m$ $b = [25\ 10\ 20\ 10\ 5]'$	`number b{ i in 1..m } = [25 10 20 10 5];`
Preu lloguer espai adicional durant mes j mesos (€)	$c_j, j = 1,2,\dots,m$ $c = [300\ 525\ 775\ 850\ 975]'$	`number c{ j in 1..m} = [300 525 775 850 975];`

Variables		
Espai adicional (m^2) llogats al començament mes i durant j mesos.	$x_{ij} \geq 0$ $i = 1,2,\dots,m,$ $j = 1,2,\dots,m-i+1$	`var X { i in 1..m , j in (1..m-i+1)} >= 0;`

Model de programació lineal		
Cost total lloguer	$\min z = \sum_{j=1}^{m} c_j \sum_{i=1}^{m-j+1} x_{ij}$	`min Cost_lloguer = sum{ j in 1..m} c[j]* sum{i in 1..(m-j+1)} x[i,j];`
Espai necessari cada mes	s.a: $\sum_{j=1}^{m-i+1} x_{ij} + \sum_{k=1}^{i-1}\left(\sum_{l=i-k+1}^{m-k+1} x_{kl}\right) \geq b_i$ $i = 1,2,\dots,m$	`con Espai_mes {i in 1..m}: sum{ j in 1..(m-i+1)} x[i,j] + sum{ k in 1..(i-1) } sum{ l in (i-k+1)..(m-k+1)} x[k,l] >= b[i];`
	$x_{ij} \geq 0, i,j = 1,2,\dots,m$	

Solució òptima

Solution Summary	
Solver	LP
Algorithm	Dual Simplex
Objective Function	Cost_lloguer
Solution Status	Optimal
Objective Value	16625
Iterations	8
Primal Infeasibility	0
Dual Infeasibility	0
Bound Infeasibility	0

[1]	[2]	X.LB	X.SOL	X.UB	X.RC	X.STATUS
1	1	0	15	1.7977E+308	0	B
1	2	0	0	1.7977E+308	50	L
1	3	0	0	1.7977E+308	0	B
1	4	0	5	1.7977E+308	0	B
1	5	0	5	1.7977E+308	0	B
2	1	0	0	1.7977E+308	125	L
2	2	0	0	1.7977E+308	50	L
2	3	0	0	1.7977E+308	225	L
2	4	0	0	1.7977E+308	175	L
3	1	0	10	1.7977E+308	0	B
3	2	0	0	1.7977E+308	150	L
3	3	0	0	1.7977E+308	275	L
4	1	0	0	1.7977E+308	225	L
4	2	0	0	1.7977E+308	325	L
5	1	0	0	1.7977E+308	175	L

[1]	Espai_mes.LB	Espai_mes.BODY	Espai_mes.UB	Espai_mes.DUAL	Espai_mes.STATUS
1	25	25	1.7977E+308	300	U
2	10	10	1.7977E+308	175	U
3	20	20	1.7977E+308	300	U
4	10	10	1.7977E+308	75	U
5	5	5	1.7977E+308	125	U

SOLUCIÓ EXERCICI 10. Optirisk

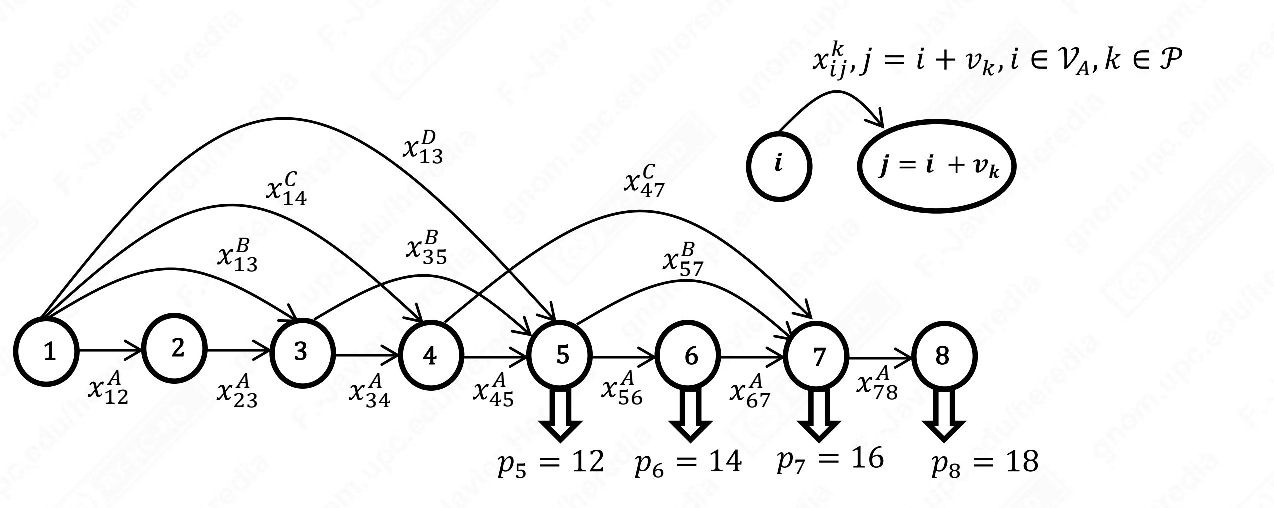
(EXERCICI 10)

- **Fitxers:** `Optirisk.sas`

Paràmetres:		
Nombre de d'anys	$n^A = 8$	`number nA = 8;`
Conjunt anys inversió	$\mathcal{A} = 1,2,\dots,n^A$	`set A = {1..nA};`
Cartera productes financers ("Portfolio")	$\mathcal{P} = \{A, B, C, D\}$	`set P = {'A', 'B', 'C', 'D'};`
Per a cada producte $k \in \mathcal{P}$		
• Termini venciment:	v_k, $v = [1\ 2\ 3\ 4]'$	`number venc{P}  = [1 2 3 4];`
• Anys venciment:	$\mathcal{AV}_k$ $\mathcal{AV}_A = \{1,2,3,4,5,6,7,8\}$ $\mathcal{AV}_B = \{1,3,5,7\}$ $\mathcal{AV}_C = \{1,4,7\}$ $\mathcal{AV}_D = \{1,5\}$	`set A_venc{k in P} = {i in 1..nA by venc[k]};`
• Rendiment (tant per u):	r_k , $r = [0.01\ 0.035\ 0.04\ 0.05]'$	`number rend{P}  = [0.01 0.035 0.04 0.05];`
• Índex de risc:	s_k, $s = [1\ 3\ 6\ 8]$	`number risc{P}  = [1 3 6 8];`
Màxim índex risc inversió:	$\bar{s} = 2.5$	`number risc_max = 2.5`
Pagaments anuals (10^3€)	p_j $, j = 1,2,\dots,n^A$ $p = [0\ 0\ 0\ 12\ 14\ 16\ 18]'$	`number pagaments{2..nA} = [0 0 0 12 14 16 18];`

Variables		
Capital invertit en el producte k a l'inici de l'any i i rescatat al començament de l'any j (10^3€):	$x_{ij}^k \geq 0$ $k \in \mathcal{P}$ $i \in \mathcal{AV}_k$ $j = i + v_k : i + v_k \leq n^A$	`var Capital { k in P , i in A_venc[k], {i+venc[k]} : i+venc[k] <= nA } >=0;`

El problema de flux de caixa correspon en realitat a un problema de fluxos en xarxes. La xarxa associada al cas concret de les dades de l'enunciat seria:



Model de programació lineal		
Capital total invertit	$\min z = \sum_{k \in \mathcal{P}} x^k_{1,1+v_k}$	`min Inversio = sum{k in P}` `Capital[k,1,1+venc[k]];`

Constricció de flux de caixa a començament de cada any (tret del primer) :	s.a:
	$\sum_{k \in \mathcal{P}: j \in \mathcal{AV}_k} (1 + r_k) x^k_{(j-v_k)j} +$
	$- \sum_{\substack{k \in \mathcal{P}: j \in \mathcal{AV}_k, \\ j+v_k \le n^A}} x^k_{(j-v_k)j}$
	$= p_j \,, j = 2, \ldots, n^A$
		`con Flux_caixa {any in 2..nA}:`
		
		`/* Rescat capital principi any i */`
		`sum{k in P : any in A_venc[k]}`
		`(1+rend[k])*Capital[k,any-venc[k],any]`
		
		`/* Inversió capital inici any i  */`
		`- sum{k in P : any in A_venc[k] AND`
		`any+venc[k]<= nA }`
		`   Capital[k,any,any+venc[k]]`
		`= pagaments[any];`

Constricció de limit de risc ponderat:	$\sum_{\substack{k \in \mathcal{P}: j \in \mathcal{AV}_k, \\ j+v_k \le n^A}} (s_k - \bar{s}) x^k_{j(j+v_k)} +$
	$+ \sum_{k \in \mathcal{P}: j \notin \mathcal{AV}_k} \sum_{\substack{i \in \mathcal{AV}_k: \\ i<j, \\ i+v_k>j, \\ i+v_k \le n^A}} (s_k - \bar{s}) x^k_{i(i+v_k)}$
	≤ 0
	$j = 1,2, \ldots, n^A - 1$
		`con Risc_promig {any in 1..nA-1}:`
		
		`/* Inversió al començament any */`
		`sum{ k in P : any in A_venc[k] AND`
		`              any+venc[k] <= nA }`
		`(risc[k]-risc_max)*`
		`Capital[k,any,any+venc[k]]`
		
		`/* Capital no rescatat invertit abans de començament any */`
		`+ sum{ k in P : any not in A_venc[k]}`
		`   sum{ i in A_venc[k] :`
		`         i < any`
		`      AND i+venc[k] > any`
		`      AND i+venc[k] <= nA}`
		`     (risc[k]-risc_max)*`
		`      Capital[k,i,i+venc[k]]`
		`<= 0;`

	$x^k_{ij} \ge 0$ $k \in \mathcal{P}, i \in \mathcal{AV}_k,$ $j = i + v_k : i + v_k \le n^A$	

Comparativa

```
/* COMPARATIVA */

/* Declaració paràmetres auxiliars */
set M = {'arriscat', 'conservador'};/* Conjunt de models a analitzar */
number rescatat{2..nA, M};          /* Capital total rescatat començament any */
number reinvertit{2..nA, M};        /* Capital total reinvertit començament any */
number invertit{A,M};               /* Total invertit durant any */
number risc_sol{A,M};               /* Risc promig any */
number pagaments_total{ M };        /* Total pagaments */
number estalvi{ M };                /* Estalvi */
number inversio_mod{ M };           /* Total inversió */
number Capital_mod{k in P , i in A_venc[k], j in {{i+venc[k]} : i+venc[k] <= nA}, M};

string mod;
do mod=M;
       if mod='arriscat' then
              drop Risc_promig;    /* Es desactiva la constricció */
       else
              restore Risc_promig; /* S'activa la constricció */
       expand;
       solve;

       /* Càlcul de resultats */

       for{k in P , i in A_venc[k], j in {i+venc[k]} : i+venc[k] <= nA}
              Capital_mod[k,i,j,mod] = Capital[k,i,j].sol;

       /* Capital total rescatat començament any */
       for {any in 2..nA}
              rescatat[any,mod] =
                     sum{k in P : any in A_venc[k]}
                            (1+rend[k])*Capital[k, any-venc[k] , any].sol;
       /* Capital total reinvertit començament any */
       for {any in 2..nA}
              reinvertit[any,mod] =
                     sum{k in P : any in A_venc[k] AND any+venc[k]<= nA}
                            Capital[k, any , any+venc[k]].sol;
       /* Total invertit durant any */
       for {any in A}
              invertit[any,mod]  =
                     sum{ k in P : any in A_venc[k] AND any+venc[k] <= nA}
                            Capital[k,any,any+venc[k]].sol
                     + sum{ k in P : any not in A_venc[k]}
                            sum{ i in A_venc[k] : i < any AND i+venc[k] > any
                            AND i+venc[k] <= nA }
                                   Capital[k,i,i+venc[k]].sol;
       /* Risc promig any */
       for{any in A}
              risc_sol[any,mod] = if invertit[any,mod] > 0 then
                     (
                     sum{ k in P : any in A_venc[k] AND any+venc[k] <= nA}
                                   risc[k]*Capital[k,any,any+venc[k]].sol
                     + sum{ k in P : any not in A_venc[k]}
                            sum{ i in A_venc[k] : i < any AND i+venc[k] > any AND
                            i+venc[k] <= nA}
                                   risc[k]*Capital[k,i,i+venc[k]].sol
                     )/invertit[any,mod];

       /* Inversió total i estalvi */
       pagaments_total[mod] = sum{any in 2..nA} pagaments[any];
       estalvi[mod] = pagaments_total[mod] - Inversio.sol;
       inversio_mod[mod] = Inversio.sol;
end;


/* Sortida resultats */

print rescatat reinvertit Flux_caixa.body risc_sol risc_max;
```

```
print pagaments_total Inversio estalvi;
print Capital_mod;
```

```
                         The OPTMODEL Procedure

                            Solution Summary

                  Solver                    Dual Simplex
                  Objective Function            Inversio
                  Solution Status                Optimal
                  Objective Value            55.152832859
                  Iterations                             7

                  Primal Infeasibility      1.065814E-14
                  Dual Infeasibility                     0
                  Bound Infeasibility                    0


              [1]    [2]             rescatat    reinvertit
               2     arriscat           0.000        0.0000
               2     conservador       13.926       13.9261
               3     arriscat          56.560       56.5598
               3     conservador       56.878       56.8777
               4     arriscat           0.000        0.0000
               4     conservador       14.362       14.3616
               5     arriscat          58.539       46.5394
               5     conservador       58.657       46.6566
               6     arriscat          14.000        0.0000
               6     conservador       22.219        8.2191
               7     arriscat          33.822       17.8218
               7     conservador       33.822       17.8218
               8     arriscat          18.000        0.0000
               8     conservador       18.000        0.0000

                                Flux_
                                caixa.
                        [1]      BODY
                         2          0
                         3         -0
                         4          0
                         5         12
                         6         14
                         7         16
                         8         18

                          risc_sol
                      arriscat    conservador
                  1     3.0000         2.5000
                  2     3.0000         2.4963
                  3     3.0000         2.5000
                  4     3.0000         2.4963
                  5     2.4043         2.0570
                  6     3.0000         2.5000
                  7     1.0000         1.0000
                  8     0.0000         0.0000

                          risc_max
                               2.5

                                 pagaments_
                  [1]                 total
                  arriscat               60
                  conservador            60
```

Inversio
55.153

[1]	estalvi
arriscat	5.3528
conservador	4.8472

[1]	[2]	[3]	[4]	Capital_ mod
A	1	2	arriscat	0.0000
A	1	2	conservador	13.7882
A	2	3	arriscat	0.0000
A	2	3	conservador	13.9261
A	3	4	arriscat	0.0000
A	3	4	conservador	14.2194
A	4	5	arriscat	0.0000
A	4	5	conservador	14.3616
A	5	6	arriscat	13.8614
A	5	6	conservador	21.9992
A	6	7	arriscat	0.0000
A	6	7	conservador	8.2191
A	7	8	arriscat	17.8218
A	7	8	conservador	17.8218
B	1	3	arriscat	54.6472
B	1	3	conservador	41.3646
B	3	5	arriscat	56.5598
B	3	5	conservador	42.6583
B	5	7	arriscat	32.6781
B	5	7	conservador	24.6574

SOLUCIÓ EXERCICI 11. Coalco (2).

(EXERCICI 11)

a) Els unics canvis entre el model Coalco i Coalco(2) a) afecten al valor dels parèmetres. Això vol dir que tant el model matemàtic com el codi codi OPTMODEL són els mateixos. Només cal canviar el valor numèric d'alguns paràmetres:

Paràmetres fitxer `Coalco(2)_a.sas`

```
set <string>  C ={'cendra', 'sulfur', 'nitrats'};
number nM = 3;
number nC = 2;
number t{ 1..nM , 1..nC } = [4 6
                             9 6
                             1 2];
number p{ 1..nM } = [ 10 55 80];
number b{ 1..nM } = [ 200 100 80];
number al{ 1..nM , C }  = [ 0.10 0.04 0.01
                            0.05 0.09 0.007
                            0.03 0.02 0.005];
number almax{ C } =       [ 0.07 0.08 0.009];
number d{ 1..nC } = [150 110];
```

Resultats:

```
                          Solution Summary

               Solver                     Dual Simplex
               Objective Function           Cost_total
               Solution Status                 Optimal
               Objective Value            11137.142857
               Iterations                            4

               Primal Infeasibility       1.561251E-17
               Dual Infeasibility                    0
               Bound Infeasibility                   0

        [1]     [2]    X.LB       X.SOL        X.RC    X.STATUS
          1       1       0      85.714     -0.0000    B
          1       2       0      44.000      0.0000    B
          2       1       0       0.000      2.1429    L
          2       2       0      66.000     -0.0000    B
          3       1       0      64.286     -0.0000    B
          3       2       0       0.000      3.0000    L

           Capacitat.     Capacitat.     Capacitat.     Capacitat.     Capacitat.
   [1]             LB           BODY             UB           DUAL     STATUS
     1    -1.7977E308        129.714            200              0     B
     2    -1.7977E308         66.000            100              0     B
     3    -1.7977E308         64.286             80              0     B

              Demanda.     Demanda.                       Demanda.     Demanda.
       [1]          LB         BODY     Demanda.UB            DUAL     STATUS
         1         150          150     1.79769E308         42.714     U
         2         110          110     1.79769E308         43.000     U

                        Contingut.     Contingut.     Contingut.     Contingut.     Contingut.
  [1]     [2]                   LB           BODY             UB           DUAL     STATUS
    1     cendra       -1.7977E308        0.00000              0        -957.14     L
    1     nitrats      -1.7977E308       -0.17143              0           0.00     B
    1     sulfur       -1.7977E308       -7.28571              0           0.00     B
```

```
2    cendra    -1.7977E308    -0.00000    0    -900.00    L
2    nitrats   -1.7977E308    -0.08800    0       0.00    B
2    sulfur    -1.7977E308    -1.10000    0       0.00    B
```

Informació sobre la solució:

- $\mathcal{B}^*: x_{11}, x_{12}, x_{22}, x_{31}$, folgues cons. `Capacitat`, `folgues Contingut[1,nitrats]`, `Contingut[1,sulfur]`, `Contingut[2,nitrats]`.
- $\begin{cases} \mathcal{N}^*: & x_{21} & x_{32} & x_{\text{Demanda[1]}} & \text{Demanda[2]} & \text{Contingut[1, cendra]} & \text{Contingut[2, nitrats]} \\ r^{*\prime} = & [\,2.1429 & 3.0 & 42.714 & 43.0 & 957.14 & 900.0\,] \end{cases}$

b) En aquest cas només cal modificar el paràmetre $\overline{\alpha}$ (**almax**) per fer que depengui tant de la component com del client (fitxer `Coalco(2)_b.sas`):

Modificació dels paràmetres:		
Per a cada client $j = 1 \dots n^C$ contingut màxim component $k \in \mathcal{C}$ carbó mescla (Tm/Tm mescla):	$\overline{\alpha}_{jk}, j = 1 \dots n^C, k \in \mathcal{C}$ $\overline{\alpha} = \begin{bmatrix} 0.08 & 0.05 & 0.01 \\ 0.06 & 0.07 & 0.007 \end{bmatrix}$	`number almax{ 1..nC, C } =[ 0.08 0.05 0.01 0.06 0.07 0.007];`
Modificació del model de programació lineal		
Continguts màxims:	$\sum_{i=1}^{n^M} (\alpha_{ik} - \overline{\alpha}_{jk}) x_{ij} \le 0$ $k \in \mathcal{C}, j = 1, \dots, n^C$	`con Contingut { j in 1..nC, k in C} : sum{ i in 1..nM } (al[i,k]-almax[j,k])*X[i,j] <= 0;`
Solució:		

```
                    Solution Summary
Solver                          Dual Simplex
Objective Function                Cost_total
Solution Status                      Optimal
Objective Value                        11190
Iterations                                 7

Primal Infeasibility            4.422461E-16
Dual Infeasibility                         0
Bound Infeasibility                        0

[1]   [2]   X.LB    X.SOL        X.RC       X.STATUS
1     1     0       96.6667     -7.3238E-15  B
1     2     0       42.2222      9.3675E-16  B
2     1     0       36.6667      3.5009E-15  B
2     2     0        4.4444      0.0000E+00  B
3     1     0       16.6667      1.3787E-14  B
3     2     0       63.3333     -6.2450E-16  B
```

El cost total d'extracció i transport amb el canvi introduït és de $z^* = 11.190$€ mentre que cost sense el canvi era de ≈ 11.137€. Veiem doncs com aquest canvi encareix els costos totals en 53€

SOLUCIÓ EXERCICI 12. Coalco (3)

(EXERCICI 12)

Cal modificar el model matemàtic i la implementació de la següent forma (fitxer `Coalco(3).sas`):

Paràmetres:

Conjunt d'operacions	$\mathcal{O} = \{arranc, càrrega, transport\}$	`set<string> O={'arranc','carrega','transport'};`

Hores necessàries per extreure una tona	$h_{ik}, i = 1,\ldots,n^M, k \in \mathcal{O}$
	 $H = \begin{bmatrix} 10 & 11 & 8 \\ 10 & 13 & 6 \\ 25 & 20 & 16 \end{bmatrix}$
		`number h{1..nM, O} =`
		`[ 10 11  8`
		`  10 13  6`
		`  25 20 16];`

Hores totals disponibles	$h_k^T, k \in \mathcal{O},$ $h^T = [3300 \quad 3600 \quad 2200]$	`number hT{O} = [3300 3600 2200];`

Constriccions:

Hores total operacions	$\sum_{i=1}^{n^M} h_{ik} \sum_{j=1}^{n^C} x_{ij} \leq h_k^T, k \in \mathcal{O}$	`con Operacio{ k in O }:`
		`sum{ i in 1..nM }`
		`h[i,k]*sum{ j in 1..nC }X[i,j] <= hT[k];`

Solució:

```
                         Solution Summary

              Solver                     Dual Simplex
              Objective Function           Cost_total
              Solution Status                 Optimal
              Objective Value                   11210
              Iterations                            5

              Primal Infeasibility                  0
              Dual Infeasibility                    0
              Bound Infeasibility                   0

         [1]      [2]      X.LB     X.SOL     X.RC     X.STATUS
          1        1         0        76        0      B
          1        2         0        44        0      B
          2        1         0        34        0      B
          2        2         0        66        0      B
          3        1         0        40        0      B
          3        2         0         0        6      L

             Capacitat.     Capacitat.     Capacitat.     Capacitat.     Capacitat.
     [1]             LB           BODY             UB           DUAL     STATUS
      1     -1.7977E308            120            200              0     B
      2     -1.7977E308            100            100              0     B
      3     -1.7977E308             40             80              0     B

                Demanda.      Demanda.                    Demanda.     Demanda.
         [1]          LB          BODY    Demanda.UB          DUAL     STATUS
          1          150           150   1.79769E308        45.889     U
          2          110           110   1.79769E308        44.889     U

                Contingut.     Contingut.     Contingut.     Contingut.     Contingut.
```

[1]	[2]	LB	BODY	UB	DUAL	STATUS
1	cendra	-1.7977E308	-0.000	0	-988.89	L
1	nitrats	-1.7977E308	-0.152	0	0.00	B
1	sulfur	-1.7977E308	-5.100	0	0.00	B
2	cendra	-1.7977E308	-0.000	0	-888.89	L
2	nitrats	-1.7977E308	-0.088	0	0.00	B
2	sulfur	-1.7977E308	-1.100	0	0.00	B

[1]	Operacio. Operacio.LB	Operacio. BODY	Operacio. UB	Operacio. DUAL	STATUS
arranc	-1.7977E308	3200	3300	0.00000	B
carrega	-1.7977E308	3420	3600	0.00000	B
transport	-1.7977E308	2200	2200	-0.27778	L

SOLUCIÓ EXERCICI 13. Pelletier i base de dades*.

(EXERCICI 13)

a) El codi modificat és:

Fitxer `Pelletier_DB.sas` (parcial)

```
proc optmodel;

/* Parameter definition */
set <num> MESOS;
number m;
number b{ MESOS };
number c{ MESOS };

/* Read data from datasets */
read data Pelletier into MESOS=[mes] b c;
m = CARD(MESOS);
```

(la resta és equivalent al codi original)

b) El codi que cal incloure al final del codi original per ampliar el conjunt de dades original amb la solució òptima és:

Fitxer `Pelletier_DB.sas` (parcial)

```
/* 1.- A temporary data set is created with the optimal solution */
number sol{MESOS,MESOS} init 0;
for {i in MESOS, j in 1..m-i+1} sol[i,j]=X[i,j].sol;
create data Pelletier_sol
      from [ mes ] = MESOS {j in MESOS} <col("m2_"||j||"m")=sol[mes,j]>;

/* 2.- The optimal solution is incorporated ino the original data set */
proc sort data=Pelletier; by mes; run;
proc sort data=Pelletier_sol; by mes; run;
data Pelletier; merge Pelletier Pelletier_sol; by mes;  run;

/* 3.- The temporary data set is deleted */
proc datasets library=work; delete Pelletier_sol; run;
```

(la resta és equivalent al codi original)

La taula final és:

Obs	mes	b	c	m2_1m	m2_2m	m2_3m	m2_4m	m2_5m
1	1	25	300	15	0	0	5	5
2	2	10	525	0	0	0	0	0
3	3	20	775	10	0	0	0	0
4	4	10	850	0	0	0	0	0
5	5	5	975	0	0	0	0	0

SOLUCIÓ EXERCICI 14. Hospital del Mar i bases de dades.

(EXERCICI 14)

a) **Fitxers:** HMaBD.sas, HMbBD.sas, HMDB.sas

Creació BD amb SAS:

```
LIBNAME HM ".";
data HM.Maquines;
input   maquina $ tp_1 tp_2 tp_3 tp_4 tp_5 temps;
datalines;
             A       3    4    4    5    3   480
             B       5    3    5    4    5   480
             C       2    5    3    3    4   480
;

data HM.Mostres;
input      mostra $  volum_mostra;
datalines;
               1  80
               2   75
               3   80
               4   12
               5   60

;
```

Lectura BD amb SAS:

```
/* Lectura dels paràmetres del model */

/* Mostres */
read data HM.Mostres into MOSTRES=[mostra] volum_mostra;

/* Maquines */
read data HM.Maquines
      into
            MAQUINES=[maquina]
            {mostra in MOSTRES}
<temps_proc[mostra,maquina]=col("tp_"||mostra)>
            temps
;
```

Escriptura de resultat a la BD

```
/* 1.- Es crea la base de dades temporal amb la solució */
create data HM_sol
      from [ maquina ] = MAQUINES
      {mostra in MOSTRES} <col("volum_"||mostra)=Volum[mostra, maquina]>;

/* 2.- Les BD s'ordenen i mesclen */

proc sort data=HM.Maquines; by maquina; run;

proc sort data=HM_sol; by maquina; run;

data HM.Maquines; merge HM.Maquines HM_sol; by maquina; run;

/* 3.- S'elimina la BD temporal */

proc datasets library=work; delete HM_sol; run;
```

Solució òptima (taula `HM.Maquines`**)**

Obs	maquina	volum_1	volum_2	volum_3	volum_4	volum_5
1	A	0	0	0	0	60
2	B	0	75	0	0	0
3	C	80	0	80	12	0

b) **Fitxers:** HMbBD.sas, HMbBD.sas, HMDB.sas

b.1) Cap mostra més del 50% del temps total de funcionament d'una màquina

Solució òptima (taula `HM.Maquines`**)**

Obs	maquina	volum_1	volum_2	volum_3	volum_4	volum_5
1	A	0	45	0.0000	0.0000	60
2	B	0	30	16.5185	1.8519	0
3	C	80	0	63.4815	10.1481	0

b.2) Cap màquina més del 40% de volum total de les proves:

Solució òptima (taula `HM.Maquines`**)**

Obs	maquina	volum_1	volum_2	volum_3	volum_4	volum_5
1	A	6.32	27.9875	12.2725	0	60
2	B	0.00	47.0125	18.6075	12	0
3	C	73.68	0.0000	49.1200	0	0

SOLUCIÓ EXERCICI 15. SBF inicial del problema de transport*.

(EXERCICI 15)

a) En primer lloc, obtenim el conjunt total d'iteracions per identificar l'última iteració de fase I:

Transport_sbfi.sas:

```
proc optmodel;

call streaminit(1123581321);      /* Llavor del generador de nombres
aleatòris */

/* Parametres */
number nREF = 5;                  /* Nre. de refineries */
number nMER = 10;                 /* Nre. de mercats    */
set<number> REFINERIES = 1..nREF; /* Cjt. de refineries */
set<number> MERCATS    = 1..nMER; /* Cjt. de mercats    */

number demanda  { j in MERCATS } = 100*rand('uniform'); /* demanda
generada aleatòriament */
number dem_tot = sum{j in MERCATS} demanda[j];          /* Demanda total
*/
number produccio{ i in REFINERIES } = dem_tot/nREF;     /* Es distribueix
la demanda entre les refineries */
number cost { i in REFINERIES, j in MERCATS } = 10*rand('uniform'); /*
Costos generats aleatòriament */

/* 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   {j in MERCATS}    :
      sum {i in REFINERIES} Trans[i,j] >= demanda[j];

/* Optimització i resultats */
solve with LP / presolver =0 solver = primal printfreq = 1;
```

Log:

```
                       Objective     Entering           Leaving
      Phase Iteration   Value         Variable           Variable
        1          1    521.113734  Trans[1,1]         Produccio_cons[1](S)
        1          2    464.006826  Trans[5,10]        Produccio_cons[5](S)
        1          3    425.003119  Trans[5,5]         Demanda_cons[10] (S)
        1          4    400.146933  Trans[1,2]         Demanda_cons[1]  (S)
        1          5    375.471567  Trans[4,6]         Produccio_cons[4](S)
        1          6    329.948917  Trans[4,2]         Demanda_cons[6]  (S)
        1          7    252.862617  Trans[3,8]         Produccio_cons[3](S)
        1          8    205.884530  Trans[3,4]         Demanda_cons[8]  (S)
        1          9    128.597900  Trans[2,3]         Produccio_cons[2](S)
        1         10    109.977908  Trans[2,7]         Demanda_cons[3]  (S)
        1         11     82.024136  Trans[5,9]         Demanda_cons[5]  (S)
        1         12     69.882844  Trans[2,4]         Demanda_cons[7]  (S)
        1         13     45.026658  Trans[1,10]        Trans[1,2]       (S)
        1         14     29.010185  Trans[3,5]         Demanda_cons[4]  (S)
        1         15      6.022950  Trans[4,5]         Trans[5,5]       (S)
        2         16   2541.014117  Trans[4,10]        Demanda_cons[2]  (S)
        2         17   2474.527711  Trans[3,10]        Trans[4,10]      (S)
        2         18   2227.679927  Trans[1,7]         Trans[2,7]       (S)
        2         19   2156.420174  Trans[1,5]         Trans[1,10]      (S)
```

```
          2          20   2120.381065  Trans[5,5]           Trans[3,5]       (S)
          2          21   1945.576323  Trans[4,3]           Trans[3,4]       (S)
          2          22   1818.866979  Trans[2,8]           Trans[5,10]      (S)
          2          23   1744.153642  Trans[2,1]           Trans[4,5]       (S)
          2          24   1579.340930  Trans[1,6]           Trans[4,6]       (S)
          2          25   1443.554395  Trans[3,2]           Trans[2,3]       (S)
          2          26   1293.992599  Trans[5,8]           Trans[5,5]       (S)
          2          27   1261.701180  Trans[1,8]           Trans[2,8]       (S)
          2          28   1012.200055  Trans[4,1]           Trans[1,1]       (S)
          2          29   1012.200055  Produccio_cons[2](S) Demanda_cons[9]  (S)
          2          30    981.478981  Trans[4,6]           Trans[3,8]       (S)
          2          31    975.810995  Trans[2,10]          Trans[4,2]       (S)
```

Observem que a l'acabament de la iteració 15 encara hi ha una infactibilitat de 6.022950, mentre que a l'acabament de la iteració 16 la solució és factible (primera iteració de fase II) Així doncs, és a l'acabament de la iteració 16 que es troba la primera s.b.f. del nostra problema. Per a obtenir l'última s.b.f. de fase I cal imprimir la solució després d'haver-se realitzat 15 iteracions (maxiter = 15):

Transport_sbfi.sas:

```
/* Optimització */
solve with LP / presolver =0 solver = primal printfreq = 1 maxiter=15;
/* Solució */
print Trans.lb Trans.sol Trans.ub cost 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;
```

Output:

```
                          The OPTMODEL Procedure

                          Resumen de la solución

               Solver                           Primal Simplex
               Objective Function                   Total_cost
               Solution Status            Iteration Limit Reached
               Objective Value                    2527.5359407
               Iterations                                   15

               Primal Infeasibility               6.0229502385
               Dual Infeasibility                            1
               Bound Infeasibility                           0

                                                                         Trans.
  [1]     [2]   Trans.LB    Trans.SOL      Trans.UB       cost    Trans.RC   STATUS
  1       1            0       99.208   1.7977E+308    5.80970   5.8097038   B
  1       2            0        0.000   1.7977E+308    7.04448   8.0444765   L
  1       3            0        0.000   1.7977E+308    8.01985   9.0198462   L
  1       4            0        0.000   1.7977E+308    1.51945   2.5194513   L
  1       5            0        0.000   1.7977E+308    2.69484   3.6948362   L
  1       6            0        0.000   1.7977E+308    2.85087   3.8508732   L
  1       7            0        0.000   1.7977E+308    1.96401   2.9640077   L
  1       8            0        0.000   1.7977E+308    1.59417   2.5941711   L
  1       9            0        0.000   1.7977E+308    7.64935   7.6493535   L
  1      10            0       24.856   1.7977E+308    5.32879   5.3287908   B
  2       1            0        0.000   1.7977E+308    3.07646   2.0764615   L
  2       2            0        0.000   1.7977E+308    4.04414   4.0441421   L
  2       3            0       77.287   1.7977E+308    6.30396   6.3039586   B
  2       4            0       28.158   1.7977E+308    1.03090   1.0309019   B
  2       5            0        0.000   1.7977E+308    8.28660   8.2865960   L
  2       6            0        0.000   1.7977E+308    8.45627   8.4562719   L
```

2	7	0	18.620	1.7977E+308	7.91345	7.9134533	B
2	8	0	0.000	1.7977E+308	3.16523	3.1652302	L
2	9	0	0.000	1.7977E+308	5.91207	4.9120727	L
2	10	0	0.000	1.7977E+308	0.99610	-0.0038971	L
3	1	0	0.000	1.7977E+308	4.20076	3.2007573	L
3	2	0	0.000	1.7977E+308	1.90485	1.9048503	L
3	3	0	0.000	1.7977E+308	4.39794	4.3979374	L
3	4	0	30.962	1.7977E+308	3.85283	3.8528305	B
3	5	0	16.016	1.7977E+308	9.63588	9.6358750	B
3	6	0	0.000	1.7977E+308	7.31212	7.3121199	L
3	7	0	0.000	1.7977E+308	4.83150	4.8315001	L
3	8	0	77.086	1.7977E+308	4.33310	4.3331043	B
3	9	0	0.000	1.7977E+308	1.48524	0.4852409	L
3	10	0	0.000	1.7977E+308	0.84303	-0.1569720	L
4	1	0	0.000	1.7977E+308	1.27357	0.2735708	L
4	2	0	76.402	1.7977E+308	1.91168	1.9116834	B
4	3	0	0.000	1.7977E+308	1.72526	1.7252621	L
4	4	0	0.000	1.7977E+308	9.36719	9.3671885	L
4	5	0	22.987	1.7977E+308	6.80740	6.8074042	B
4	6	0	24.675	1.7977E+308	2.21818	2.2181754	B
4	7	0	0.000	1.7977E+308	8.81424	8.8142373	L
4	8	0	0.000	1.7977E+308	5.91093	5.9109264	L
4	9	0	0.000	1.7977E+308	9.10324	8.1032354	L
4	10	0	0.000	1.7977E+308	9.05340	8.0534008	L
5	1	0	0.000	1.7977E+308	7.40073	7.4007341	L
5	2	0	0.000	1.7977E+308	9.35000	10.3500049	L
5	3	0	0.000	1.7977E+308	9.27484	10.2748410	L
5	4	0	0.000	1.7977E+308	6.65152	7.6515248	L
5	5	0	0.000	1.7977E+308	4.36341	5.3634125	L
5	6	0	0.000	1.7977E+308	9.75805	10.7580458	L
5	7	0	0.000	1.7977E+308	9.71527	10.7152730	L
5	8	0	0.000	1.7977E+308	1.86463	2.8646276	L
5	9	0	91.814	1.7977E+308	0.25834	0.2583350	B
5	10	0	32.251	1.7977E+308	5.16225	5.1622494	B

[1]	Produccio_ cons.LB	Produccio_ cons.BODY	Produccio_ cons.UB	Produccio_ cons.DUAL	Produccio_ cons.STATUS
1	-1.7977E+308	124.06	124.06	-1	L
2	-1.7977E+308	124.06	124.06	0	L
3	-1.7977E+308	124.06	124.06	0	L
4	-1.7977E+308	124.06	124.06	0	L
5	-1.7977E+308	124.06	124.06	-1	L

[1]	Demanda_ cons.LB	Demanda_ cons. BODY	Demanda_ cons.UB	Demanda_ cons. DUAL	Demanda_ cons. STATUS
1	99.208	99.208	1.7977E+308	1	U
2	70.379	76.402	1.7977E+308	0	B
3	77.287	77.287	1.7977E+308	0	U
4	59.119	59.119	1.7977E+308	0	U
5	39.004	39.004	1.7977E+308	0	U
6	24.675	24.675	1.7977E+308	0	U
7	18.620	18.620	1.7977E+308	0	U
8	77.086	77.086	1.7977E+308	0	U
9	**97.837**	**91.814**	**1.7977E+308**	**1**	**B**
10	57.107	57.107	1.7977E+308	1	U

Es pot observar com la única constricció no satisfeta és `Demanda_cons[9]`, amb una infactibilitat de 97.837-91.814=6.023.

b) Si repetim les operacions de l'apartat anterior amb `maxiter=16` i imprimim l'estat de les constriccions observarem que la base a l'acabament de la iteració 15 ja és factible:

Output:

```
                        The OPTMODEL Procedure

                        Resumen de la solución

            Solver                              Primal Simplex
            Objective Function                      Total_cost
            Solution Status             Iteration Limit Reached
            Objective Value                        2541.014117
            Iterations                                      16

            Primal Infeasibility                  2.842171E-14
            Dual Infeasibility                               0
            Bound Infeasibility                              0

         Produccio_    Produccio_    Produccio_    Produccio_    Produccio_
 [1]        cons.LB     cons.BODY       cons.UB     cons.DUAL     cons.STATUS
 1      -1.7977E+308       124.06        124.06             0     L
 2      -1.7977E+308       124.06        124.06             0     L
 3      -1.7977E+308       124.06        124.06             0     L
 4      -1.7977E+308       124.06        124.06             0     L
 5      -1.7977E+308       124.06        124.06             0     L
                          Demanda_                  Demanda_    Demanda_
            Demanda_         cons.      Demanda_       cons.    cons.
      [1]    cons.LB          BODY       cons.UB        DUAL    STATUS
        1     99.208        99.208   1.7977E+308           0    U
        2     70.379        70.379   1.7977E+308           0    U
        3     77.287        77.287   1.7977E+308           0    U
        4     59.119        59.119   1.7977E+308           0    U
        5     39.004        39.004   1.7977E+308           0    U
        6     24.675        24.675   1.7977E+308           0    U
        7     18.620        18.620   1.7977E+308           0    U
        8     77.086        77.086   1.7977E+308           0    U
        9     97.837        97.837   1.7977E+308           0    B
```

SOLUCIÓ EXERCICI 16. Planificació de la producció: taxació-eficiència computacional*.

Amb l'ajut del següent codi:

Producció_rand.sas:

```
proc optmodel presolver = 0 printlevel=2;

call streaminit(DNI);

/* Paràmetres */
set PRODUCTE = 1..10000;
set RECURS   = 1..100;
number consum{ j in RECURS, i in PRODUCTE} = 100*rand('uniform');
number benefici{i in PRODUCTE} = 200*rand('uniform');
number disp{j in RECURS} = sum{i in PRODUCTE} consum[j,i]*10;

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

/* Optimització i resultats */
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 0;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 1;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 2;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 3;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 4;
```

S'obté la taula:

printfreq =	Nre. ITER	temps/iter. (sec.)	Temps execució (sec.)
STEEPESTEDGE (4)	420	0.00469	1.97
HYBRID (0)	932	0.00299	2.79
DEVEX (3)	932	0.00299	2.79
PARTIAL (1)	4913	0.00295	14.51
FULL (2)	4913	0.00298	14.66

S'observa que la opció 4 realitza un nombre molt inferior d'iteracions que la resta de mètodes de forma que, tot i que quasi duplica el temps per iteració, la velocitat de millora de la funció objectiu és tal que compensa el temps dedicat a cada iteració. HYBRID i DEVEX redueixen a quasi 1/5 part el nombre d'iteracions (o el que és el mateix, quintupliquen la millora de la funció objectiu $\Delta z = \theta^* r_q$ per iteració) en comparació a PARTIAL i FULL, amb pràcticament el mateix cost per iteració.

SOLUCIÓ EXERCICI 17. Coalco: estudi taxació-eficiència computacional**.

Coalco_rand.sas:

```
proc optmodel printlevel=2;

/* Llavor del generador de nombres aleatòris  */
call streaminit(1123581321);

/* Definició de paràmetres */
number nCOM = 10;                                    /* Nre de component */
set <number>  C = 1..nCOM;                           /* Components carbó */
number nM = 100;                                     /* Nombre de mines */
number nC = 2000;                                    /* Nre de clients */
number  t{  1..nM  ,  1..nC  }  =  10+10*rand('uniform');   /*  Cost  transport
number p{ 1..nM }       = 30+20*rand('uniform');     /* Cost producció */
number al{ 1..nM , C } = 0.7*rand('uniform');        /* Contingut */
number  almax{  i  in  C  }  =  0.6+0.3*rand('uniform');    /*  Contingut  màxim
component mescla (Tm/Tm mescla)*/
number d{ 1..nC } = 5+10*rand('uniform');            /* Demanda */
number b{ 1..nM } = 200+200*rand('uniform');        /* Producció */

/* Variables de decisió */
var X { 1..nM, 1..nC } >= 0; /* Tones a transportar mina i-> client j */

/* Formulació del model d'optmització*/
min Cost_total = sum{ i in 1..nM , j in 1..nC } (p[i]+t[i,j])*X[i,j];
/* Cost total producción més transport */
con Capacitat { i in 1..nM } : sum{ j in 1..nC } X[i,j] <= b[i];
con Demanda   { j in 1..nC } : sum{ i in 1..nM } X[i,j] >= d[j];
con Contingut { j in 1..nC, k in C} :
    sum{ i in 1..nM } (al[i,k]-almax[k])*X[i,j] <= 0;

/* Optimització */
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 0;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 1;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 2;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 3;
solve with LP / presolver =0 solver = primal printfreq = 0 pricetype = 4;
```

printfreq =	Nre. ITER	temps/iter. (sec.)	Temps execució (sec.)
STEEPESTEDGE (4)	13901	0.004461	62.01
HYBRID (0)	12349	0.004148	51.22
DEVEX (3)	25765	0.002812	72.45
FULL (2)	36171	0.002682	97.02
PARTIAL (1)	36171	0.002687	97.19

SOLUCIÓ EXERCICI 18. Analisi de sensibilitat problema Pelletier**

a) Ordre de les variables de decisió i vector c:

$$\begin{array}{rcccccccccccccccccccccl}
 & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & \\
x = & [& x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} & x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{3,1} & x_{3,2} & x_{3,3} & x_{41} & x_{42} & x_{5,1} & s_1 & s_2 & s_3 & s_4 & s_5 &]^T \\
c = & [& 300 & 525 & 775 & 850 & 975 & 300 & 525 & 775 & 850 & 300 & 525 & 775 & 300 & 525 & 300 & 0 & 0 & 0 & 0 & 0 &]^T
\end{array}$$

b) Ordre de les constriccions i components vector b i matriu A:

$$A = \begin{array}{cccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \\
\end{array}$$

$$A = \left[\begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1
\end{array}\right], b = \left[\begin{array}{c} 25 \\ 10 \\ 20 \\ 10 \\ 5 \end{array}\right]\begin{array}{l} \text{Mes 1} \\ \text{Mes 2} \\ \text{Mes 3} \\ \text{Mes 4} \\ \text{Mes 5} \end{array}$$

c) Fitxer **PelletierLP.sas**:

Fitxer RemingtonDB.sas

```
data PelletierLP;
  input _row_ $ x11 x12 x13 x14 x15 x21 x22 x23 x24 x31 x32 x33 x41 x42 x51 _type_ $ _rhs_;
      datalines;
        Cost   300 525 775 850 975 300 525 775 850 300 525 775 300 525 300      MIN   .
    Espai_m1     1   1   1   1   1   0   0   0   0   0   0   0   0   0   0       GE  25
    Espai_m2     0   1   1   1   1   1   1   1   1   0   0   0   0   0   0       GE  10
    Espai_m3     0   0   1   1   1   0   1   1   1   1   0   0   0   0   0       GE  20
    Espai_m4     0   0   0   1   1   0   0   1   1   0   1   1   1   1   0       GE  10
    Espai_m5     0   0   0   0   1   0   0   0   1   0   0   1   0   1   1       GE   5
     Non_neg     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 lowerbd   .
;
run;
proc print  data=PelletierLP;
run;
proc lp data=PelletierLP rangeprice rangerhs;
run;
```

Solució:

Solution Summary	
Terminated Successfully	
Objective Value	16625
Phase 1 Iterations	3
Phase 2 Iterations	4
Phase 3 Iterations	0
Integer Iterations	0
Integer Solutions	0
Initial Basic Feasible Variables	7
Time Used (seconds)	0
Number of Inversions	3
Epsilon	1E-8
Infinity	1.797693E308
Maximum Phase 1 Iterations	100
Maximum Phase 2 Iterations	100
Maximum Phase 3 Iterations	99999999
Maximum Integer Iterations	100
Time Limit (seconds)	120

Variable Summary

Col	Variable Name	Status	Type	Price	Activity	Reduced Cost
1	x11	BASIC	LOWERBD	300	15	0
2	x12		LOWERBD	525	0	225
3	x13		LOWERBD	775	0	175
4	x14	BASIC	LOWERBD	850	5	0
5	x15	BASIC	LOWERBD	975	5	0
6	x21		LOWERBD	300	0	300
7	x22		LOWERBD	525	0	225
8	x23		LOWERBD	775	0	225
9	x24		LOWERBD	850	0	175
10	x31	BASIC	LOWERBD	300	10	0
11	x32		LOWERBD	525	0	275
12	x33		LOWERBD	775	0	400
13	x41		LOWERBD	300	0	50
14	x42		LOWERBD	525	0	150
15	x51		LOWERBD	300	0	175
16	Espai_m1		SURPLUS	0	0	300
17	Espai_m2	DEGEN	SURPLUS	0	0	0
18	Espai_m3		SURPLUS	0	0	300
19	Espai_m4		SURPLUS	0	0	250
20	Espai_m5		SURPLUS	0	0	125

Constraint Summary

Row	Constraint Name	Type	S/S Col	Rhs	Activity	Dual Activity
1	Cost	OBJECTVE	.	0	16625	.
2	Espai_m1	GE	16	25	25	300
3	Espai_m2	GE	17	10	10	0
4	Espai_m3	GE	18	20	20	300
5	Espai_m4	GE	19	10	10	250
6	Espai_m5	GE	20	5	5	125

Intervals d'estabilitat:

RHS Range Analysis

Row	Minimum Phi Rhs	Minimum Phi Leaving	Minimum Phi Objective	Maximum Phi Rhs	Maximum Phi Leaving	Maximum Phi Objective
Espai_m1	10	x11	12125	INFINITY	.	.
Espai_m2	-INFINITY	.	.	10	Espai_m2	16625
Espai_m3	10	x31	13625	INFINITY	.	.
Espai_m4	10	Espai_m2	16625	20	x31	19125
Espai_m5	0	x15	16000	10	x14	17250

Price Range Analysis

Col	Variable Name	Minimum Phi Price	Minimum Phi Entering	Minimum Phi Objective	Maximum Phi Price	Maximum Phi Entering	Maximum Phi Objective
1	x11	250	x41	15875	475	x13	19250
2	x12	300	x12	16625	INFINITY	.	16625
3	x13	600	x13	16625	INFINITY	.	16625
4	x14	675	x51	15750	900	x41	16875
5	x15	850	Espai_m5	16000	1125	x42	17375
6	x21	0	x21	16625	INFINITY	.	16625
7	x22	300	x22	16625	INFINITY	.	16625
8	x23	550	x23	16625	INFINITY	.	16625
9	x24	675	x24	16625	INFINITY	.	16625
10	x31	250	x41	16125	475	x13	18375
11	x32	250	x32	16625	INFINITY	.	16625
12	x33	375	x33	16625	INFINITY	.	16625
13	x41	250	x41	16625	INFINITY	.	16625
14	x42	375	x42	16625	INFINITY	.	16625
15	x51	125	x51	16625	INFINITY	.	16625
16	Espai_m1	-300	Espai_m1	16625	INFINITY	.	16625
17	Espai_m2	-175	x13	16625	50	x41	16625
18	Espai_m3	-300	Espai_m3	16625	INFINITY	.	16625
19	Espai_m4	-250	Espai_m4	16625	INFINITY	.	16625
20	Espai_m5	-125	Espai_m5	16625	INFINITY	.	16625

d) Sabem quines són les variables bàsiques a partir de la següent taula **Variable Summary**, $\mathcal{B}^* = \{1,4,5,10,17\}$, corresponent a les variables $x_{1,1}$, $x_{1,4}$, $x_{1,5}$, $x_{3,1}$ i s_2. La matriu A correspon al nostre problema es pot consultar a la finestra **SAS output**:

Obs	_row_	x11	x12	x13	x14	x15	x21	x22	x23	x24	x31	x32	x33	x41	x42	x51	_type_	_rhs_
1	Cost	300	525	775	850	975	300	525	775	850	300	525	775	300	525	300	MIN	.
2	Espai_m1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	GE	25
3	Espai_m2	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	GE	10
4	Espai_m3	0	0	1	1	1	0	1	1	1	1	0	0	0	0	0	GE	20
5	Espai_m4	0	0	0	1	1	0	0	1	1	0	1	1	1	1	0	GE	10
6	Espai_m5	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1	GE	5
7	Non_neg	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	lowerbd	.

La matriu bàsica B està formada per les columnes 1 ($x_{1,1}$), 4 ($x_{1,4}$), 5 ($x_{1,5}$) i 10 ($x_{3,1}$) d'aquesta matriu, conjuntament amb la columna associada a la variable d'escreix de la segona constricció s_2 ($A_{17} = [0 \quad -1 \quad 0 \quad 0 \quad 0]^T$):

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, B^{-1} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix}$$

e) Amb la matriu bàsica obtinguda a l'apartat anterior, comproveu que els valors de Φ_{b_1}, Φ_{c_2} i Φ_{c_4} de les taules **RHS** and **Price Range Analysis** coincideixen amb les obtingudes aplicant les expressions derivades a classe.

- $\boldsymbol{\Phi_{b_1}}$:per a calcular aquest interval necessitem conèixer

$$b = \begin{bmatrix} 25 \\ 10 \\ 20 \\ 10 \\ 5 \end{bmatrix}, x_B = \begin{bmatrix} 15 \\ 5 \\ 5 \\ 10 \\ 0 \end{bmatrix}, \gamma_1 = B^{-1} \cdot e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Fem els càlculs:

$$\begin{cases} \phi_{b_1}^{min} = \max\limits_{k=1,\dots,5} \left\{ b_1 - \dfrac{x_{B(k)}}{\gamma_{k1}} : \gamma_{k1} > 0 \right\} = b_1 - \dfrac{x_{B(1)}}{\gamma_{11}} = 25 - \dfrac{15}{1} = 10 \\ \phi_{b_1}^{max} = \min\limits_{k=1,\dots,m} \left\{ b_1 - \dfrac{x_{B(k)}}{\gamma_{k1}} : \gamma_{k1} < 0 \right\} = +\infty \text{ (no existeix fita superior)} \end{cases}$$

$$\boxed{\Phi_{b_1} = [\phi_{b_1}^{min}, \phi_{b_1}^{max}] = [10, +\infty[}$$

Observem com el valor coincideix amb el trobat per **PROC LP**:

RHS Range Analysis						
	Minimum Phi			Maximum Phi		
Row	Rhs	Leaving	Objective	Rhs	Leaving	Objective
Espai_m1	10	x11	12125	INFINITY	.	.

- $\boldsymbol{\Phi_{c_2}}$: com que x_2 és variable no bàsica l'expressió de l'interval d'estabilitat és:

$$\Phi_{c_i} = [\phi_{c_i}^{min}, \phi_{c_i}^{max}] = [c_i - r_i, +\infty[$$

amb $c_2 = 525$ i $r_2 = 225$, és a dir, $\boxed{\Phi_{c_2} = [300, +\infty[}$, que també coincideix amb el valor calculat per **PROC LP**:

Price Range Analysis							
		Minimum Phi			Maximum Phi		
Col	Variable Name	Price	Entering	Objective	Price	Entering	Objective
1	x11	250	x41	15875	475	x13	19250
2	x12	300	x12	16625	INFINITY	.	16625

- $\boldsymbol{\Phi_{c_4}}$: en aquest x_4 és la segona variable bàsica ($p = 2$). La matriu V és calcula fent

$$V = B^{-1}A_N$$
$$= \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & -1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}.$$

Llavors $v_2 = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad -1 \quad 0 \quad 0 \quad -1 \quad 1]$. El vector de costos reduïts (columna **Reduced Cost** de la taula **Variable Summary**) és:

$$r' = [225 \quad 175 \quad 300 \quad 225 \quad 225 \quad 175 \quad 275 \quad 400 \quad 50 \quad 150 \quad 175 \quad 300 \quad 300 \quad 250 \quad 125].$$

Ara ja podem calcular l'interval d'estabilitat $\Phi_{c_4} = [\Phi_{c_4}^{min}, \Phi_{c_4}^{max}]$

$$\begin{cases} \phi_{c_4}^{min} = \max_{j\in\mathcal{N}^*}\left\{c_4 + \dfrac{r_j}{v_{2j}} : v_{2j} < 0\right\} = 850 + \max\left\{\dfrac{175}{-1}, \dfrac{250}{-1}\right\} = 675 \\ \phi_{c_4}^{max} = \min_{j\in\mathcal{N}^*}\left\{c_4 + \dfrac{r_j}{v_{2j}} : v_{2j} > 0\right\} = 850 + \min\left\{\dfrac{225}{1}, \dfrac{275}{1}, \dfrac{50}{1}, \dfrac{125}{1}\right\} = 900 \end{cases} \rightarrow$$

$$\rightarrow \boxed{\Phi_{c_4} = [\Phi_{c_4}^{min}, \Phi_{c_4}^{max}] = [675,900]}$$

que coincideix de nou amb el valor proporcionat per **PROC LP**:

Price Range Analysis							
		Minimum Phi			Maximum Phi		
Col	Variable Name	Price	Entering	Objective	Price	Entering	Objective
1	x11	250	x41	15875	475	x13	19250
2	x12	300	x12	16625	INFINITY	.	16625
3	x13	600	x13	16625	INFINITY	.	16625
4	x14	675	x51	15750	900	x41	16875

f) Els intervals d'estabilitat que mostra **PROC LP** són:

RHS Range Analysis						
	Minimum Phi			Maximum Phi		
Row	Rhs	Leaving	Objective	Rhs	Leaving	Objective
Espai_m1	10	x11	12125	INFINITY	.	.
Espai_m2	-INFINITY	.	.	10	Espai_m2	16625
Espai_m3	10	x31	13625	INFINITY	.	.
Espai_m4	10	Espai_m2	16625	20	x31	19125
Espai_m5	0	x15	16000	10	x14	17250

Observem que el valor dels nous elements $\phi_{b_1} = 40$ i $\phi_{b_5} = 0$ estan dins dels seus respectius intervals d'estabilitat. Així doncs, la base òptima del problema modificat $(P)_{\phi_b}$ coincideix amb la del problema original, és a dir $\mathcal{B}^*_{\phi_b} = \mathcal{B}^* = \{1,4,5,10,17\}$. Per tal de calcular el nou valor de $x^*_{\phi_b}$ i $z^*_{\phi_b}$ apliquem les expressions del símplex:

$$x^*_{\phi_b} = \begin{bmatrix} x_{1,1} \\ x_{1,4} \\ x_{1,5} \\ x_{3,1} \\ x_{s_2} \end{bmatrix} = B^{-1}_{\phi_b}\phi_b = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 40 \\ 10 \\ 20 \\ 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 30 \\ 10 \\ 0 \\ 10 \\ 0 \end{bmatrix}, z^*_{\phi_b} = c'_B x^*_{\phi_b} = 20500$$

Podem calcular també el nou valor de la funció objectiu fent us dels preus ombra λ^* (columna **`Dual Activity`** de la taula **`Constraints Summary`**):

$$z^*_{\phi_b} = z^* + \lambda^*_1 \cdot \Delta b_1 + \lambda^*_5 \cdot \Delta b_5 = 16625 + 300 \cdot 15 + 125 \cdot (-5) = 20500$$

Si ara introduïm el nou valor del vector de termes independents ϕ_b al fitxer **`PelletierLP.sas`** i calculem el bnou òptim obtenim:

Variable Summary						
Col	Variable Name	Status	Type	Price	Activity	Reduced Cost
1	x11	BASIC	LOWERBD	300	30	0
2	x12		LOWERBD	525	0	225
3	x13		LOWERBD	775	0	175
4	x14	BASIC	LOWERBD	850	10	0
5	x15		LOWERBD	975	0	125
6	x21		LOWERBD	300	0	300
7	x22		LOWERBD	525	0	225
8	x23		LOWERBD	775	0	225
9	x24		LOWERBD	850	0	300
10	x31	BASIC	LOWERBD	300	10	0
11	x32		LOWERBD	525	0	275
12	x33		LOWERBD	775	0	525
13	x41		LOWERBD	300	0	50
14	x42		LOWERBD	525	0	275
15	x51		LOWERBD	300	0	300
16	Espai_m1		SURPLUS	0	0	300
17	Espai_m2	DEGEN	SURPLUS	0	0	0
18	Espai_m3		SURPLUS	0	0	300
19	Espai_m4		SURPLUS	0	0	250
20	Espai_m5	DEGEN	SURPLUS	0	0	0

RHS Range Analysis						
	Minimum Phi			Maximum Phi		
Row	Rhs	Leaving	Objective	Rhs	Leaving	Objective
Espai_m1	10	x11	11500	INFINITY	.	.
Espai_m2	-INFINITY	.	.	10	Espai_m2	20500
Espai_m3	10	x31	17500	INFINITY	.	.
Espai_m4	10	Espai_m2	20500	20	x31	23000
Espai_m5	-INFINITY	.	.	0	Espai_m5	20500

Observem que, malgrat que el valor de la funció objectiu i variables és el predit, s'ha produït un canvi de base obtenint-se la nova SBF òptima $\mathcal{B}^*_{\phi_b} = \{1,4,10,17,20\}$, malgrat que les coordenades del punt extrem solució són les mateixes.

$$x^*_{\phi_b} = [30 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 10 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Això no repressenta cap contradicció, doncs sabem que, quan fem coincidir el valor de d'alguns elements de ϕ_{b_j} amb el d'un dels extrems dels seus intervals d'estabilitat (aquí, $\phi_{b_5} = \phi^{min}_{b_5} = 0$) la nova base és degenerada primal. Això vol dir que el canvi ϕ_b situa la nova solució òptima sobre el punt extrem $x^*_{\phi_b}$ que té més d'una SBF associada. Finalment, fixem-nos que en l'intercanvi de VB que s'ha produït ha sortit la variable 5 $x_{1,5}$, tal com indica la taula **`RHS Range Analysis`**.

SOLUCIÓ EXERCICI 19. Remington Manufacturing: problemes de càrrega fixa*.

(EXERCICI 19)

- **Model matemàtic:**

Paràmetres:		
Conjunt de productes	$\mathcal{P}$	`set <num> PRODUCTS;`
Conjunt de processos	$\mathcal{O}$	`set <str> OPERATIONS;`

Per a cada producte $i \in \mathcal{P}$:
• Benefici unitari (€)
• Costos fixos (€)
	
	b_i
	f_i
		
		`number incomes{PRODUCTS};`
		`number fix_cost{PRODUCTS};`

Hores disponibles operació $j \in \mathcal{O}$ (h).	h_j	`number  h_avail{OPERATIONS};`
Hores consumides d'operació $j \in \mathcal{O}$ per unitat de producció producte $i \in \mathcal{P}$ (h).	a_{ji}	`number  h_req{OPERATIONS, PRODUCTS};`

Variables		
Quantitat producte $i \in \mathcal{P}$	$x_i \in \mathbb{Z}^{\geq 0}$	`Var Amount{PRODUCTS} >=0 integer;`
Fabricació producte $i \in \mathcal{P}$ (1 es fabrica, 0 no es fabrica).	$y_i \in \{0,1\}$	`Var Make{PRODUCTS} binary;`

Model de programació lineal entera		
Maximització del benefici total:	$\max_{x,y\in\mathbb{Z}} z = \sum_{i\in\mathcal{P}} (b_i x_i + f_i y_i)$	`max Total_profit = sum{ i in PRODUCTS} (incomes[i]*Amount[i] - fix_cost[i]*Make[i]);`
Disponibilitat hores operació:	s.a: $\sum_{i\in\mathcal{P}} a_{ji}\, x_i \leq h_j \quad j \in \mathcal{O}$	`con Limit_hours {j in OPERATIONS}: sum{ i in PRODUCTS} h_req[j,i]*Amount[i] <= h_avail[j];`
Acoblament $x_i - y_i$:	$x_i \leq M_i y_i \quad i \in \mathcal{P}$	`con Coupling { i in PRODUCTS}: Amount[i] <= (min{j in OPERATIONS}(h_avail[j]/h_req[j,i])) *Make[i];`
	$x_i \geq 0, y_i \in \{0,1\} \quad i \in \mathcal{P}$	

- **Codi SAS/OR:**

Fitxer `RemingtonDB.sas`

```
LIBNAME Reming ".";
data Reming.Product;
input   product incomes fix_cost;
datalines;
               1      48     1000
               2      55      800
               3      50      900
;

data Reming.Operation;
input      operation $ h_avail h_req_p1 h_req_p2 h_req_p3;
datalines;
               Mach      600         2         3         6
               Grin      300         6         3         4
               Asse      400         5         6         2
;
```

Fitxer `Remington_DB.sas`

```
proc optmodel presolver = 0;

/* Parameters of the model */

set <num> PRODUCTS;
number incomes{PRODUCTS};
number fix_cost{PRODUCTS};

set <str> OPERATIONS;
number  h_avail{OPERATIONS};
number  h_req{OPERATIONS, PRODUCTS};

/* Products */
read data Reming.Product into PRODUCTS=[product] incomes fix_cost;

/* Processes */
read data Reming.Operation
      into
            OPERATIONS=[operation]
            h_avail {i in PRODUCTS} <h_req[operation,i]=col("h_req_p"||i)>
;

var Amount{PRODUCTS} >=0 integer;
var   Make{PRODUCTS} >=0 binary;

max Total_profit = sum{ i in PRODUCTS} (incomes[i]*Amount[i] - fix_cost[i]*Make[i]);

con Limit_hours {j in OPERATIONS}:
      sum{ i in PRODUCTS} h_req[j,i]*Amount[i] <= h_avail[j];

con Coupling { i in PRODUCTS}:
      Amount[i] <= (min{j in OPERATIONS}(h_avail[j]/h_req[j,i]))*Make[i];

/* Extended model */
expand;

/* Optimize and output */
solve with MILP / presolver = 0;
number total_incomes = sum{ i in PRODUCTS} incomes[i]*Amount[i].sol;
number total_costs   = sum{ i in PRODUCTS} fix_cost[i]*Make[i].sol;
print Total_profit total_incomes total_costs;
print _var_.name _var_.lb _var_.sol _var_.ub;
print _con_.name _con_.lb _con_.body _con_.ub;
```

```
/* Storing the solution to a DB */
 create data Reming.Sol
      from [ product ] = PRODUCTS      amount = Amount make=Make;
run;
/* Merging the optimal solution with the parameters datasets */

/* Workers by shift */
proc sort data=Reming.Product; by product; run;
proc sort data=Reming.Sol; by product; run;
data Reming.Product; merge Reming.Product Reming.Sol; by product; run;
proc datasets library=Reming; delete Sol; run;

/* Printing the extended dataset */
proc print; run;
```

- **Solució optimal mostrada a la finestra SAS Output:**

Solution Summary	
Solver	MILP
Algorithm	Branch and Cut
Objective Function	Total_profit
Solution Status	Optimal
Objective Value	2980
Iterations	44
Best Bound	2980
Nodes	5
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	0
Bound Infeasibility	0
Integer Infeasibility	1.421085E-14

Total_profit	total_incomes	total_costs
2980	4680	1700

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB
1	Amount[1]	0	0	1.7977E308
2	Amount[2]	0	56	1.7977E308
3	Amount[3]	0	32	1.7977E308
4	Make[1]	0	0	1
5	Make[2]	0	1	1
6	Make[3]	0	1	1

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_COI
1	Limit_hours[Mach]	-1.7977E+308	360.000	
2	Limit_hours[Grin]	-1.7977E+308	296.000	
3	Limit_hours[Asse]	-1.7977E+308	400.000	
4	Coupling[1]	-1.7977E+308	0.000	
5	Coupling[2]	-1.7977E+308	-10.667	
6	Coupling[3]	-1.7977E+308	-43.000	

- **Dataset ampliat amb la solució òptima:**

Obs	product	incomes	fix_cost	amount	make
1	1	48	1000	0	0
2	2	55	800	56	1
3	3	50	900	32	1

SOLUCIÓ EXERCICI 20. Air-Express: problemes de planificació de plantilles.**

(EXERCICI 20)

- **Model matemàtic:**

Paràmetres:		
Conjunt de dies	$\mathcal{D}$	`set <str> DAYS;`
Conjunt de torns	$\mathcal{S}$	`set <num> SHIFTS;`

Per a cada torn $i \in \mathcal{D}$:
• Dies de descans
• Salari treballador (€)
	
	$\mathcal{H}_i \subset \mathcal{D}$
	c_i
		
		`string holy{SHIFTS, 1..2};`
		`number lab_costs{SHIFTS};`

Demanda treballadors dia $j \in \mathcal{D}$	$d_j \;, j \in \mathcal{D}$	`number workforce{DAYS};`

Variables		
Nre. de treballadors a contractar torn i	$x_i \in \mathbb{Z}^{\geq 0}, i \in \mathcal{S}$	`var Workers{SHIFTS} >=0 integer;`

Model de programació lineal		
Cost total nòmina mes fix:	$\min_{x\in\mathbb{Z}} z = \sum_{i\in\mathcal{S}} c_i x_i$	`minimize Tot_labcost= sum{ i in SHIFTS} lab_costs[i]*Workers[i];`
Demanda diària:	s.a: $\sum_{i:j\notin\mathcal{H}_i} x_i \geq b_j \quad j \in \mathcal{D}$	`con Workforce_con{j in DAYS}: sum{ i in SHIFTS : and{k in 1..2} j NE holy[i,k] } Workers[i] >= workforce[j];`
	$x_i \geq 0, y_i \in \{0,1\} \quad i \in \mathcal{D}$	

- **Codi SAS/OR:**

Fitxer `AirexpressDB.sas`

```
LIBNAME Airexp ".";
data Airexp.Days;
input   day $ workforce ;
datalines;
        Sun     18
        Mon     27
        Tue     22
        Wed     26
        Thu     25
        Fri     21
        Sat     19
;

data Airexp.Shifts;
input      shift    lab_costs holy1 $ holy2 $;
datalines;
               1     680        Sun       Mon
```

```
                 2     705         Mon      Tue
                 3     705         Tue      Wed
                 4     705         Wed      Thu
                 5     705         Thu      Fri
                 6     680         Fri      Sat
                 7     655         Sat      Sun
;
```

Fitxer `Airexpress_DB.sas`

```
proc optmodel;

/* Parameters of the model */

set <str> DAYS;
set <num> SHIFTS;
string holy{SHIFTS, 1..2};
number lab_costs{SHIFTS};
number workforce{DAYS};

/* Gathering data parameters from datasets */

/* DAYS */
read data Airexp.Days into DAYS=[day] workforce;

/* SHIFTS */
read data Airexp.Shifts
       into
              SHIFTS=[shift]
              lab_costs {i in 1..2} <holy[shift,i]=col("holy"||i)>
;

var Workers{SHIFTS} >=0 integer;

minimize Tot_labcost= sum{ i in SHIFTS} lab_costs[i]*Workers[i];

con Workforce_con{j in DAYS}:
sum{ i in SHIFTS : and{k in 1..2} j NE holy[i,k]} Workers[i] >= workforce[j];

/* Show the model */
expand;

/* Optimize and output */
solve with MILP / presolver = 0 printfreq=1 allcuts=0; /*  cutgomory=-1   */
print Tot_labcost;
print _var_.name _var_.lb _var_.sol _var_.ub;
print _con_.name _con_.lb _con_.body _con_.ub;

/* Storing the solution to a DB */

 create data Airexp.Sol
       from [ shift ] = SHIFTS
       workers = Workers;
create data Airexp.workforce
       from [ day ] = DAYS
       workers = Workforce_con.body;
run;

/* Merging the optimal solution with the parameters datasets */

/* Workers by shift */
proc sort data=Airexp.Shifts; by shift; run;
proc sort data=Airexp.Sol; by shift; run;
data Airexp.Shifts; merge Airexp.Shifts Airexp.Sol; by shift; run;
proc datasets library=Airexp; delete Sol; run;

/* Workers by day */
```

```
proc sort data=Airexp.Days; by day; run;
proc sort data=Airexp.Workforce; by day; run;
data Airexp.Days; merge Airexp.Days Airexp.Workforce; by day; run;
proc datasets library=Airexp; delete Workforce; run;

/* Printing the extended dataset */
proc print; run;
```

- **Solució optimal mostrada a la finestra SAS Output:**

Solution Summary	
Solver	MILP
Algorithm	Branch and Cut
Objective Function	Tot_labcost
Solution Status	Optimal
Objective Value	22540
Iterations	239
Best Bound	22540
Nodes	47
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	0
Bound Infeasibility	0
Integer Infeasibility	0

Tot_labcost
22540

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB
1	Workers[1]	0	4	1.7977E+308
2	Workers[2]	0	2	1.7977E+308
3	Workers[3]	0	6	1.7977E+308
4	Workers[4]	0	1	1.7977E+308
5	Workers[5]	0	6	1.7977E+308
6	Workers[6]	0	3	1.7977E+308
7	Workers[7]	0	11	1.7977E+308

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_CON_.UB
1	Workforce_con[Fri]	21	24	1.7977E+308
2	Workforce_con[Mon]	27	27	1.7977E+308
3	Workforce_con[Sat]	19	19	1.7977E+308
4	Workforce_con[Sun]	18	18	1.7977E+308
5	Workforce_con[Thu]	25	26	1.7977E+308
6	Workforce_con[Tue]	22	25	1.7977E+308
7	Workforce_con[Wed]	26	26	1.7977E+308

- **Dataset ampliat amb la solució òptima:**

Obs	day	workforce	workers
1	Fri	21	24
2	Mon	27	27
3	Sat	19	19
4	Sun	18	18
5	Thu	25	26
6	Tue	22	25
7	Wed	26	26

SOLUCIÓ EXERCICI 21. CRT-Technologies: problemes de selecció de projectes**.

(EXERCICI 21)

a) Model bàsic:

- **Model matemàtic:**

Paràmetres:		
Conjunt d'anys.	$\mathcal{Y}$	`set <num> YEAR;`
Conjunt de projectes.	$\mathcal{P}$	`set <num> PROJECTS;`
NPV projecte $i \in \mathcal{P}$ ($k€$).	n_i	`number npv{PROJECTS};`
Pressupost any $j \in \mathcal{Y}$ ($k€$).	b_j	`number budget{YEARS};`
Inversió necessària projecte $i \in \mathcal{P}$ any $j \in \mathcal{Y}$ ($k€$).	a_{ji}	`number investment{PROJECTS,YEARS};`

Variables		
Decisió de seleccionar (1) o descartar (0) el projecte $i \in \mathcal{P}$.	$y_i \in \{0,1\}$	`var Selected{PROJECTS} binary;`

Model de programació lineal		
Es maximitza el NPV total de la inversió:	$\min_{x\in\mathbb{Z}} z = \sum_{i\in\mathcal{P}} n_i y_i$	`maximize Total_npv = sum{ p in PROJECTS} npv[p]*Selected[p];`
Pressupost anual:	s.a: $\sum_{i\in\mathcal{P}} a_{ji} y_i \le b_j \quad j \in \mathcal{Y}$	`con Limit_budget {y in YEARS}: sum{ p in PROJECTS} investment[p,y]*Selected[p] <= budget[y];`
	$y_i \in \{0,1\} \quad i \in \mathcal{Y}$	

- **Codi SAS/OR:**

Fitxer `CRTDB.sas`

```
LIBNAME CRT ".";
data CRT.Years;
input   years budget ;
datalines;
          1     250
          2      75
          3      50
          4      50
          5      50
;
data CRT.Projects;
```

```
input      project  npv inv_y1 inv_y2 inv_y3 inv_y4 inv_y5;
datalines;
                 1  141     75     25     20     12     20
                 2  187     90     35      0      0     30
                 3  121     60     15     15     15     15
                 4   83     30     20     10      5      5
                 5  265    100     25     20     20     20
                 6  127     50     20     10     30     40
;
```

Fitxer `CRT_DB.sas`

```
proc optmodel presolver = 0;

/* Parameters of the model */

set <num> YEARS;
set <num> PROJECTS;

number npv{PROJECTS};
number budget{YEARS};
number investment{PROJECTS,YEARS};

/* Years */
read data CRT.Years into YEARS=[years] budget;

/* Projects */
read data CRT.Projects
       into
              PROJECTS=[project]
              npv {i in YEARS} <investment[project,i]=col("inv_y"||i)>
;

var Selected{PROJECTS} binary;

maximize Total_npv = sum{ p in PROJECTS} npv[p]*Selected[p];

con Limit_budget {y in YEARS}:
       sum{ p in PROJECTS} investment[p,y]*Selected[p] <= budget[y];

/* Show the model */
expand;

/* Optimize and output */
solve with MILP / presolver = 0;
print Total_npv;
print _var_.name _var_.lb _var_.sol _var_.ub;
print _con_.name _con_.lb _con_.body _con_.ub;

/* Storing the solution to a DB */
create data CRT.Sol from [ project ] = PROJECTS      sel = Selected;

/* Merging the optimal solution with the parameters datasets */
proc sort data=CRT.Sol; by project; run;
proc sort data=CRT.Projects; by project; run;
data CRT.Projects; merge CRT.Projects CRT.Sol; by project; run;
proc datasets library=CRT; delete Sol; run;

/* Printing the extended dataset */
proc print; run;
```

- **Solució optimal mostrada a la finestra SAS Output:**

Solution Summary	
Solver	MILP
Algorithm	Branch and Cut
Objective Function	Total_npv
Solution Status	Optimal
Objective Value	489
Iterations	25
Best Bound	489
Nodes	1
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	0
Bound Infeasibility	0
Integer Infeasibility	0

Total_npv
489

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB
1	Selected[1]	0	1	1
2	Selected[2]	0	0	1
3	Selected[3]	0	0	1
4	Selected[4]	0	1	1
5	Selected[5]	0	1	1
6	Selected[6]	0	0	1

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_CON_.UB
1	Limit_budget[1]	-1.7977E+308	205	250
2	Limit_budget[2]	-1.7977E+308	70	75
3	Limit_budget[3]	-1.7977E+308	50	50
4	Limit_budget[4]	-1.7977E+308	37	50
5	Limit_budget[5]	-1.7977E+308	45	50

- **Dataset ampliat amb la solució òptima:**

Obs	project	npv	inv_y1	inv_y2	inv_y3	inv_y4	inv_y5	sel
1	1	141	75	25	20	12	20	1
2	2	187	90	35	0	0	30	0
3	3	121	60	15	15	15	15	0
4	4	83	30	20	10	5	5	1
5	5	265	100	25	20	20	20	1
6	6	127	50	20	10	30	40	0

b) CRT, model ampliat:

- **Model matemàtic:** els elements nous del model són:

Paràmetres:		
Incompatibilitat entre projectes $i - j$ (1 incompatibles, 0 compatibles).	d_{ji}	`number incomp{PROJECTS, PROJECTS};`

Model de programació lineal: s'afegeix la constricció		
Incompatibilitats:	$y_i + y_j \leq 1 \quad (i,j) \vert d_{ij} = 1$	`con Incomp_con { i in PROJECTS, j in PROJECTS : incomp[i,j]=1}: Selected[i]+Selected[j] <= 1;`

- **Codi SAS/OR:**

Afegim a la llibreria `CRT` la informació sobre incompatibilitat entre projectes:

Fitxer `CRTDB_Incomp.sas`

```
data CRT.Incomp;
input        proj   ip1  ip2  ip3  ip4  ip5  ip6;
datalines;
                1      .    1    0    1    0    0
                2      .    .    1    0    0    0
                3      .    .    .    0    0    0
                4      .    .    .    .    0    0
                5      .    .    .    .    .    1
                6      .    .    .    .    .    .
;
```

Fitxer `CRT_DB_b.sas`

```
proc optmodel presolver = 0;

/* Parameters of the model */

set <num> YEAR;
set <num> PROJECTS;

number npv{PROJECTS};
number budget{YEAR};
number investment{PROJECTS,YEAR};
number incomp{PROJECTS, PROJECTS};

/* Years */
read data CRT.Years into YEAR=[years] budget;

/* Projects */
read data CRT.Projects
       into
              PROJECTS=[project]
              npv
        {i in YEAR} <investment[project,i]=col("inv_y"||i)>
;

/* Incompatibilities among projects */
number proj;
read data CRT.Incomp
       into [proj] {p IN PROJECTS} <incomp[proj, p]=col("ip"||p)>
```

```
;

print budget npv investment;
print incomp;

var Selected{PROJECTS} binary;

maximize Total_npv = sum{ p in PROJECTS} npv[p]*Selected[p];

con Limit_budget {y in YEAR}:
      sum{ p in PROJECTS} investment[p,y]*Selected[p] <= budget[y];

con Incomp_con {i in PROJECTS}:
      Selected[i] + sum{ j in PROJECTS : j>i} incomp[i,j]*Selected[j] <= 1;

/* Show the model */
expand;

/* Optimize and output */
solve with MILP / presolver = 0;
print Total_npv;
print _var_.name _var_.lb _var_.sol _var_.ub;
print _con_.name _con_.lb _con_.body _con_.ub;

/* Storing the solution to a DB */
create data CRT.Sol from [ project ] = PROJECTS     sel = Selected;

/* Merging the optimal solution with the parameters datasets */
proc sort data=CRT.Sol; by project; run;
proc sort data=CRT.Projects; by project; run;
data CRT.Projects; merge CRT.Projects CRT.Sol; by project; run;
proc datasets library=CRT; delete Sol; run;

/* Printing the extended dataset */
proc print; run;
```

- **Solució optimal mostrada a la finestra SAS Output:**

Solution Summary	
Solver	MILP
Algorithm	Branch and Cut
Objective Function	Total_npv
Solution Status	Optimal
Objective Value	469.00000022
Iterations	14
Best Bound	469.00000022
Nodes	1
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	0
Bound Infeasibility	0
Integer Infeasibility	9.0000005E-9

Total_npv
469

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB
1	Selected[1]	0	5.0E-09	1
2	Selected[2]	0	4.0E-09	1
3	Selected[3]	0	1.0E+00	1
4	Selected[4]	0	1.0E+00	1
5	Selected[5]	0	1.0E+00	1
6	Selected[6]	0	0.0E+00	1

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_CON_.UB
1	Limit_budget[1]	-1.7977E+308	190	250
2	Limit_budget[2]	-1.7977E+308	60	75
3	Limit_budget[3]	-1.7977E+308	45	50
4	Limit_budget[4]	-1.7977E+308	40	50
5	Limit_budget[5]	-1.7977E+308	40	50
6	Incomp_con[1]	-1.7977E+308	1	1
7	Incomp_con[2]	-1.7977E+308	1	1
8	Incomp_con[3]	-1.7977E+308	1	1
9	Incomp_con[4]	-1.7977E+308	1	1
10	Incomp_con[5]	-1.7977E+308	1	1
11	Incomp_con[6]	-1.7977E+308	0	1

- **Dataset ampliat amb la solució òptima:**

Obs	project	npv	inv_y1	inv_y2	inv_y3	inv_y4	inv_y5	sel
1	1	141	75	25	20	12	20	0.00000
2	2	187	90	35	0	0	30	0.00000
3	3	121	60	15	15	15	15	1.00000
4	4	83	30	20	10	5	5	1.00000
5	5	265	100	25	20	20	20	1.00000
6	6	127	50	20	10	30	40	0.00000

SOLUCIÓ EXERCICI 22. Airexpress (2): planificació de plantilles amb cost fix *.

(EXERCICI 22)

- **Model matemàtic:**

Paràmetres:		
Conjunt de dies	$\mathcal{D}$	`set <str> DAYS;`
Conjunt de torns	$\mathcal{S}$	`set <num> SHIFTS;`
Per a cada torn $i \in \mathcal{D}$:		
• Dies de descans	$\mathcal{H}_i \subset \mathcal{D}$	`string holy{SHIFTS, 1..2};`
• Salari treballador (€)	c_i	`number lab_costs{SHIFTS};`
• Costos fixos torn (€):	k_i	`number fix_costs{SHIFTS};`
• Fita superior x_i:	$M_i = \max_{j \in \mathcal{D} \setminus \mathcal{H}_i} b_j$	`number M{i in SHIFTS} = max{ j in DAYS : and{k in 1..2} j NE holy[i,k] } workforce[j];`
Demanda treballadors dia $j \in \mathcal{D}$	d_j	`number workforce{DAYS};`

Variables		
Nre. de treballadors a contractar torn $i \in \mathcal{S}$.	$x_i \geq 0$	`var Workers{SHIFTS} >=0 integer;`
Contractació torn $i \in \mathcal{S}$ (1 es contracta, 0 no es contracta)	$y_i \in \{0,1\}$	`var  Select{SHIFTS} binary;`

Model de programació lineal		
Cost total nòmina més fix:	$\min_{x,y} z = \sum_{i \in \mathcal{S}} (c_i x_i + k_i y_i)$	`minimize tot_cost= sum{ i in SHIFTS} (lab_costs[i]*Workers[i] + fix_costs[i]*Select[i]);`
Demanda diaria:	s.a: $\sum_{i: j \notin \mathcal{H}_i} x_i \geq b_j \quad j \in \mathcal{D}$	`con Workforce_con{j in DAYS}: sum{ i in SHIFTS : and{k in 1..2} j NE holy[i,k] } Workers[i] >= workforce[j];`
Acoblament $x_i - y_i$:	$x_i \leq M_i y_i \quad i \in \mathcal{D}$	`con Coupling_con{i in SHIFTS}: Workers[i]<= M[i]*Select[i];`
	$x_i \geq 0, y_i \in \{0,1\} \quad i \in \mathcal{D}$	

- **Codi SAS/OR:**

Fitxer `Airexpress(2)DB.sas`

```
LIBNAME Airexp ".";
data Airexp.Days;
input   day $ workforce ;
datalines;
        Sun     18
        Mon     27
        Tue     22
        Wed     26
        Thu     25
        Fri     21
        Sat     19
;

data Airexp.Shifts;
input      shift  lab_costs fix_costs holy1 $ holy2 $;
datalines;
              1        680      1000      Sun      Mon
              2        705       950      Mon      Tue
              3        705       950      Tue      Wed
              4        705       950      Wed      Thu
              5        705       950      Thu      Fri
              6        680      1000      Fri      Sat
              7        655      1000      Sat      Sun
;
```

Fitxer `Airexpress(2).sas`

```
proc optmodel;

/* Parameters of the model */
set <str> DAYS;                /* Set of days */
set <num> SHIFTS;              /* Set of shifts */
string holy{SHIFTS, 1..2};     /* Pair of non-working days per shift */
number lab_costs{SHIFTS};      /* Individual monthly labour costs (€) */
number fix_costs{SHIFTS};      /* Monthly fix shit management costs (€) */
number workforce{DAYS};        /* Daily workforce requirement */
number M{i in SHIFTS}          /* Coupling x-y contraint constant s.t. x_i <= M_i */
      = max{j in DAYS : and{k in 1..2} j NE holy[i,k]}workforce[j];

/* Parameters reading */

/* DAYS */
read data Airexp.Days into DAYS=[day] workforce;

/* SHIFTS */
read data Airexp.Shifts
      into
            SHIFTS=[shift]
            lab_costs fix_costs {i in 1..2} <holy[shift,i]=col("holy"||i)>
;

/* Decision variables */
var Workers{SHIFTS} >=0 integer;  /* Number of workers per shift */
var Select{SHIFTS} binary;        /* Selection of shift i */

/* Objective function: to minimize labour and fix costs */
minimize tot_cost= sum{ i in SHIFTS} (lab_costs[i]*Workers[i] +
fix_costs[i]*Select[i]);

/* Subject to daily workforce requirements. */
con Workforce_con{j in DAYS}:
      sum{ i in SHIFTS : and{k in 1..2} j NE holy[i,k]} Workers[i] >= workforce[j];

/* x-y coupling constraints */
con Coupling_con{i in SHIFTS}:
      Workers[i]<= M[i]*Select[i];
```

```
/* Extended model */
expand;

/* Optimize and output */
solve;

/* Display */
number cost_lab = sum{ i in SHIFTS} lab_costs[i]*Workers[i].sol;
number cost_fix = sum{ i in SHIFTS} fix_costs[i]*Select[i].sol;
print tot_cost cost_lab cost_fix;
print _var_.name _var_.lb _var_.sol _var_.ub;
print _con_.name _con_.lb _con_.body _con_.ub;

/* Solution writing to data sets */
 create data Airexp.Sol
       from [ shift ] = SHIFTS
       workers = Workers select=Select;
create data Airexp.workforce
       from [ day ] = DAYS
       workers = Workforce_con.body; run;

/* Merging the optimal solution with the parameters datasets */

/* Writing optimal shifts's related information to data sets. */
proc sort data=Airexp.Shifts; by shift; run;
proc sort data=Airexp.Sol; by shift; run;
data Airexp.Shifts; merge Airexp.Shifts Airexp.Sol; by shift; run;
proc datasets library=Airexp; delete Sol; run;

/* Writing optimal worker's related information to data sets. */
proc sort data=Airexp.Days; by day; run;
proc sort data=Airexp.Workforce; by day; run;
data Airexp.Days; merge Airexp.Days Airexp.Workforce; by day; run;
proc datasets library=Airexp; delete Workforce; run;

/* Printing the extended dataset */
proc print; run;
```

- **Solució optimal mostrada a la finestra SAS Output:**

Solution Summary	
Solver	MILP
Algorithm	Branch and Cut
Objective Function	tot_cost
Solution Status	Optimal
Objective Value	26415
Iterations	146
Best Bound	26415
Nodes	9
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	0
Bound Infeasibility	0
Integer Infeasibility	0

tot_cost	cost_lab	cost_fix
26415	22565	3850

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB
1	Workers[1]	0	0	1.7977E308
2	Workers[2]	0	4	1.7977E308
3	Workers[3]	0	7	1.7977E308
4	Workers[4]	0	0	1.7977E308
5	Workers[5]	0	8	1.7977E308
6	Workers[6]	0	0	1.7977E308
7	Workers[7]	0	14	1.7977E308
8	Select[1]	0	0	1
9	Select[2]	0	1	1
10	Select[3]	0	1	1
11	Select[4]	0	0	1
12	Select[5]	0	1	1
13	Select[6]	0	0	1
14	Select[7]	0	1	1

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_CON_.UB
1	Workforce_con[Fri]	21	25	1.7977E308
2	Workforce_con[Mon]	27	29	1.7977E308
3	Workforce_con[Sat]	19	19	1.7977E308
4	Workforce_con[Sun]	18	19	1.7977E308
5	Workforce_con[Thu]	25	25	1.7977E308
6	Workforce_con[Tue]	22	22	1.7977E308
7	Workforce_con[Wed]	26	26	1.7977E308
8	Coupling_con[1]	-1.7977E308	0	0
9	Coupling_con[2]	-1.7977E308	-22	0
10	Coupling_con[3]	-1.7977E308	-20	0
11	Coupling_con[4]	-1.7977E308	0	0
12	Coupling_con[5]	-1.7977E308	-19	0
13	Coupling_con[6]	-1.7977E308	0	0
14	Coupling_con[7]	-1.7977E308	-13	0

- **Solució optima al conjunt de dades:**

Obs	day	workforce	workers
1	Fri	21	25
2	Mon	27	29
3	Sat	19	19
4	Sun	18	19
5	Thu	25	25
6	Tue	22	22
7	Wed	26	26

SOLUCIÓ EXERCICI 23. Coalco (4): transport i mescla amb costos fixos*.

(EXERCICI 23)

a) Elements modificats/afegits a la formulació `Coalco(2)` (fitxer `Coalco(4).sas`):

Paràmetres:		
Nombre mínim de mines en funcionament	$n^M_{min} = 2$	`number nM_min = 2;`
Per a cada mina $i = 1, \dots, n^M$ • Costos fixos de tancament [equivalent €/mes] • Costos fixos de funcionament [€/mes]	$c^T_i, c^T = \begin{bmatrix}1000\\1000\\2000\end{bmatrix}$ $c^F_i, c^F = \begin{bmatrix}4000\\3000\\1000\end{bmatrix}$	`number ct{ 1..nM } = [1000 1000 2000];` `number cf{ 1..nM } = [4000 3000 1000];`

Variables		
Variable de decisió tancament mina i ($y_i = 0 \Rightarrow$ tanca, $y_i = 1 \Rightarrow$no tanca:	$y_i \in \{0,1\}$ $i = 1, \dots, n^M$	`var Y { 1..nM } binary;`

Model de programació lineal:		
Cost total producció 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]`
més costos de tancament	$+ \sum_{i=1}^{n^M} c^T_i (1 - y_i)$	`+ sum{ i in 1..nM } ct[i]*(1-Y[i])`
més costos de funcionament	$+ \sum_{i=1}^{n^M} c^F_i y_i$	`+ sum{ i in 1..nM } cf[i]*Y[i];`
Constricció sindical:	s.a: $\sum_{i=1}^{n^M} y_i \geq n^M_{min}$	`con Sindical:` `sum{ i in 1..nM } Y[i] >= nM_min;`
Acoblament $x - y$ (b_i fa el paper de M_i)	$\sum_{j=1}^{n^C} x_{ij} \leq b_i y_i$ $i = 1, \dots, n^M$	`con Acoblament{ i in 1..nM}:` `sum{j in 1..nC} X[i,j] <= b[i]*Y[i];`

b) Solució amb i sense constricció sindical:

	x^*amb costos de funcionament i tancament (Tm)			**x^*amb costos de funcionament + tancament i constricció sindical (Tm)**		
Mina	**1**	**2**	**3**	**1**	**2**	**3**
Client 1	0 Tm	0 Tm	40 Tm	22.857 Tm	0 Tm	17.143 Tm

Client 2	0 Tm	0 Tm	30 Tm	17.143 Tm	0 Tm	12.857 Tm
Costos totals (€/mes)	87.000'00			90.037'142		

SOLUCIÓ EXERCICI 24. Prodem S.L. (2) .**

(EXERCICI 24)

a) Model de selecció de procés de manufactura (Fitxer **`Prodem(2).sas`**).

Paràmetres:		
Conjunt de productes	$\mathcal{P} = \{A, B, C\}$	`set<str> PRODUCTE = {'A','B','C'};`
Conjunt de processos de manufactura	$\mathcal{M} = \{1,2\}$	`set<num> PROCES = 1..2;`
Per a cada procés $p \in \mathcal{M}$ i producte $i \in \mathcal{P}$: • Consum de recurs [Tm/unitat] • Costos de producció [u.m./unitat]	$a_{pi}, a = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}$ $c_{pi}, c = \begin{bmatrix} 10 & 2 & 3 \\ 20 & 3 & 3 \end{bmatrix}$	`number consum{PROCES, PRODUCTE} = [ 3.0 2.0 1.0 1.5 1.0 0.5];` `number cost{PROCES,PRODUCTE} = [ 10 2 3 15 2 4];`
Disponibilitat recurs [Tm]:	$b = 40$	`number disp = 40;`
Demanda [Tm]:	$d = 40$	`number dem = 33;`

Variables		
Per a cada procés $p \in \mathcal{M}$: • Quantitat fabricada producte $i \in \mathcal{P}$ [Tm]: • Selecció procés manufactura:	$x_{pi} \geq 0$ $y_p = \begin{cases} 1 & \text{es selecciona} \\ 0 & \text{no es selecciona} \end{cases}$	`var Produc {PROCES,PRODUCTE} >= 0;` `var Activa {PROCES} binary;`

Model de programació lineal entera:		
Cost total producció [u.m.]:	$\min z = \sum_{p \in \mathcal{M}} \sum_{i \in \mathcal{P}} c_{pi} x_{pi}$	`max Total_benefici = sum {p in PROCES, i in PRODUCTE} cost[p,i]*Produc[p,i];`
Disponibilitat recurs:	s.a: $\sum_{p \in \mathcal{M}} \sum_{i \in \mathcal{P}} a_{pi} x_{pi} \leq b$	`con Consum_recurs: sum {p in PROCES, i in PRODUCTE} consum[p,i]*Produc[p,i] <= disp;`
Satisfacció demanda:	$\sum_{p \in \mathcal{M}} \sum_{i \in \mathcal{P}} x_{pi} \geq b$	`con Demanda : sum{p in PROCES, i in PRODUCTE} Produc[p,i] >= dem;`
Incompatibilitat processos:	$\sum_{p \in \mathcal{M}} y_p = 1$	`con Incompatibilitat: sum{p in PROCES} Activa{p] = 1;`
Acoblament $x - y$ (b fa el paper de M_i)	$\sum_{i \in \mathcal{P}} x_{pi} \leq b y_p \quad p \in \mathcal{M}$	`con Acoblament {p in PROCES}: sum{i in PRODUCTE} Produc[p,i] <= disp*Activa[p];`

Solució:

Solution Summary	
Solver	MILP
Algorithm	Branch and Cut
Objective Function	Total_cost
Solution Status	Optimal
Objective Value	66
Iterations	23
Best Bound	66
Nodes	1
Relative Gap	0
Absolute Gap	0
Primal Infeasibility	2.538509E-11
Bound Infeasibility	5.385089E-12
Integer Infeasibility	7.692544E-13

[1]	_VAR_.NAME	_VAR_.LB	_VAR_.SOL	_VAR_.UB
1	Produc[1,A]	0	0	1.7977E+308
2	Produc[1,B]	0	-0	1.7977E+308
3	Produc[1,C]	0	0	1.7977E+308
4	Produc[2,A]	0	0	1.7977E+308
5	Produc[2,B]	0	33	1.7977E+308
6	Produc[2,C]	0	0	1.7977E+308
7	Activa[1]	0	-0	1.0000E+00
8	Activa[2]	0	1	1.0000E+00

[1]	_CON_.NAME	_CON_.LB	_CON_.BODY	_CON_.UB
1	Consum_recurs	-1.7977E+308	33	40
2	Demanda	3.30000E+01	33	1.7977E308
3	Incompatibilitat	1.00000E+00	1	1
4	Acoblament[1]	-1.7977E+308	0	0
5	Acoblament[2]	-1.7977E+308	-7	0