Programació Lineal i Entera

Tema 1: Introducció a la PL

Introducció a la programació lineal

- **Objectius tema 1.**
- **Teoria: conceptes bàsics (setmana 1).**
  - Definició de problema de programació lineal.
  - Resolució gràfica de problemes de PL.
  - Forma estàndard de problemes de PL.
  - Solucions bàsiques factibles.
- **Laboratori: formulació i resolució computacional de problemes (PL) (setmana 2).**
  - **Problema de planificació de la producció:**
  - **Problema de la dieta.**
  - **Problema de transport.**
  - **Exercici de laboratori 1.**

Planificació de la producció: SteelCat

- Una empresa de manufactura d'acer SteelCat ha de programar la producció setmanal d'un taller de laminat que transforma planxes d'acer en tres tipus de peces, **bandes**, **espirals** i **plaques**, d'acord amb les següents dades:

SteelCat	Benefici unit.	Comandes setmanals	Capacitat mercat	Capacitat de la fase d'escalfat	Capacitat del fase de laminat
Unitats	(€/Tm)	(Tm)	(Tm)	(Tm/h)	(Tm/h)
Bandes	25	1.000	6.000	200	200
Espirals	30	500	4.000	200	140
Plaques	29	750	3.500	200	160
Hores setmanals disponibles				35	40

Problema SteelCat: elements del model

Donats els **PARÀMETRES:**

n: nombre de productes

m: nombre de recursos diferents

c_i: benefici unitari producte i $\quad i = 1,\ldots,n$

u_i: quantitat màxima a produir producte i $\quad i = 1,\ldots,n$

a_{ij}: quantitat de recurs j consumit pel producte j $\quad j = 1,\ldots,m$

b_j: quantitat màxima disponible del recurs j $\quad j = 1,\ldots,m$

Es vol trobar el valor de les **VARIABLES:**

x_i: quantitat de producte i a produir $\quad i = 1,\ldots,n$

Amb l'objectiu de maximitzar la **FUNCIÓ OBJECTIU:**

$$z = \sum_{i=1}^{n} c_i x_i = c'x \quad \text{(beneficis totals)}$$

Sense violar les **CONSTRICCIONS:**

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j \quad \text{(el consum del recurs } j \text{ no supera la seva disponibilitat)}$$

$$0 \le x_i \le u_i \quad \text{(fites a les variables)}$$

Problema SteelCat: formulació genèrica

Formulació algebràica:

$$\begin{cases} \max \quad z = \sum_{i=1}^{n} c_i x_i = c'x & & (1) & \textbf{funció objectiu} \\ \text{s.a.:} \quad \sum_{i=1}^{n} a_{ij} x_i \le b_j & j = 1,\ldots,m & (2) & \textbf{constriccions} \\ \quad 0 \le x_i \le u_i & i = 1,\ldots,n & (3) & \textbf{domini} \end{cases}$$

Formulació algebràica amb conjunts:

P: conjunt de productes. R: conjunt de recursos.

$$\begin{cases} \max \quad z = \sum_{i \in P} c_i x_i = c'x & & (1) & \textbf{funció objectiu} \\ \text{s.a.:} \quad \sum_{i \in P} a_{ij} x_i \le b_j & j \in R & (2) & \textbf{constriccions} \\ \quad 0 \le x_i \le u_i & i \in P & (3) & \textbf{domini} \end{cases}$$

Problema SteelCat: dades

PARÀMETRES:

$n = 3$: nombre de **PRODUCTES** $= \{$**bandes, espirals, plaques**$\}$

$m = 2$: nombre de **FASES** $= \{$**escalfat, laminat**$\}$

c_i: **benefici** unitari producte i $\quad c = [25 \quad 30 \quad 29]'$

u_i: **cap**acitat **mercat** $\quad u = [6000 \quad 4000 \quad 3500]'$

l_i: **comanda** $\quad l = [6000 \quad 4000 \quad 3500]'$

t_{ij}: **taxa** de producció del producte i a la fase j $(a_{ij} = 1/t_{ij})$

t_{ij}	bandes	espirals	plaques
escalfat	200	200	200
laminat	200	140	160

b_j: **límit temps** de processat de cada fase j $\quad b = [35 \quad 40]'$

Problema SteelCat: formulació específica

$$
\begin{cases}
\max & z = \sum_{i=1}^{3} c_i x_i = 25x_1 + 30x_2 + 29x_3 \\
\text{s.a.:} & \sum_{i=1}^{3} a_{i1} x_i \le b_1 \rightarrow \frac{1}{200}x_1 + \frac{1}{200}x_2 + \frac{1}{200}x_3 \le 35 \\
& \sum_{i=1}^{3} a_{i2} x_i \le b_2 \rightarrow \frac{1}{200}x_1 + \frac{1}{140}x_2 + \frac{1}{160}x_3 \le 40 \\
& [1000 \quad 500 \quad 750] \le x \le [6000 \quad 4000 \quad 3500]
\end{cases}
$$

benefici total

Temps total escalfat

Temps total laminat

comanda i mercat

- **Implementació/resolució amb AMPL/CPLEX:**
  - **Fitxer *.mod:** declaració de **conjunts**, **paràmetres** i **model**.
  - **Fitxer *.dat:** assignació de **valors** als conjunts i paràmetres.

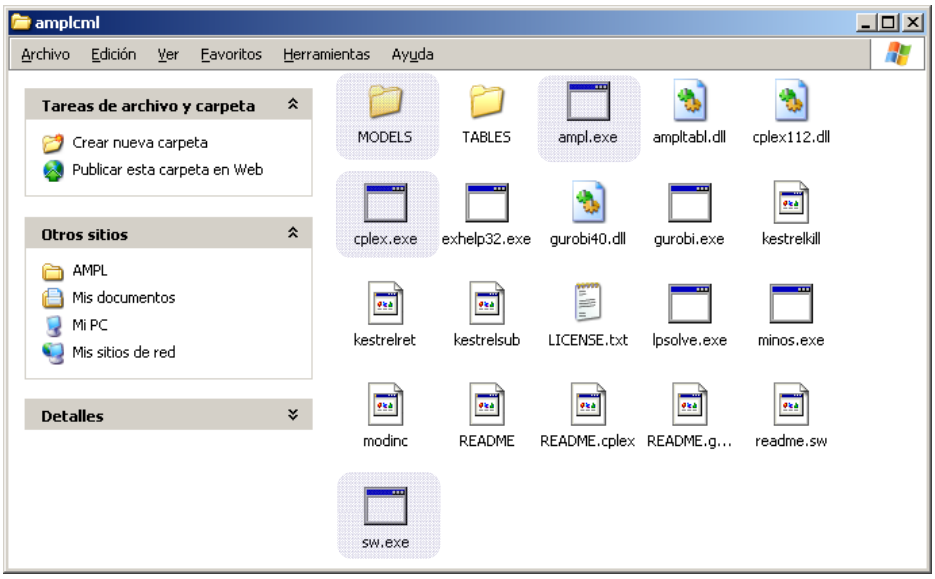
AMPL/CPLEX

- **AMPL (http://www.ampl.com/):**
  - Llenguatge de modelització per a Programació Matemàtica: permet definir problemes d'optimització en una notació similar a la matemàtica.
  - Permet usar els més importants programes d'optimització (CPLEX, MINOS, LANCELOT, SNOPT,…) sota una interfície comuna.
- **CPLEX (http://www.ilog.com/products/cplex/):**
  - Llibreria de rutines de referència d'optimització per a problemes de **programació lineal i quadràtica**, **lineal entera**, **quadràtica entera** i **fluxos en xarxes**.

Instal·lació de AMPL/CPLEX

- **Descarregueu `amplcml.zip` de la plana web de AMPL:**

  http://ampl.com/NEW/TABLES/amplcml.zip



Implementació: fitxer `MODELS/steel4.mod`

Paràmetres model

```
# Conjunts
set PROD;   # productes
set STAGE;  # fases
# Paràmetres
param profit {PROD};          # benefici unitari (eur./Tm)
param market {PROD} >= 0;     # capacitat mercat (Tm)
param commit {PROD} >= 0;     # comandes (Tm)
param rate {PROD,FASE} > 0;   # Cap. de proc. de cada fase (Tm/h)
param avail {FASE} >= 0;      # hores set de proc. de cada fase
```

Formulació matemàtica

```
# Variables
var Make {p in PROD} >= commit[p], <= market[p];  # Tm produides

# Model
maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];

subject to Time {s in STAGE}:
   sum {p in PROD} (1/rate[p,s]) * Make[p] <= avail[s];
        # a cada fase: el nre. Total d'hores usades per tots
        # els productes no pot excedir el total d'hores disponibles
```

Dades: fitxer MODELS/steel4.dat

Paràmetres model

```
data;

set PROD  := bands coils plate;
set STAGE := reheat roll;

param rate:  reheat  roll :=
  bands        200   200
  coils        200   140
  plate        200   160 ;

param:    profit  commit  market :=
  bands     25      1000    6000
  coils     30       500    4000
  plate     29       750    3500 ;

param avail :=  reheat 35     roll  40 ;
```


Resolució amb AMPL/CPLEX (1/2)

sw: finished ampl = 0

File Edit Help

```
sw: ampl
ampl: model MODELS/steel4.mod;
ampl: model MODELS/steel4.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 11.2.0: optimal solution; objective 190071.4286
2 dual simplex iterations (0 in phase I)
ampl: display Make, Time;
:          Make      Time     :=
bands    3357.14      .
coils     500         .
plate    3142.86      .
reheat      .       1800
roll        .       3200
;

ampl: quit
sw:
```

F.O.

Valor de les variables

?

Resolució amb AMPL/CPLEX (2/2)

```
sw: finished ampl = 0
File  Edit  Help
sw: ampl
ampl: model MODELS/steel4.mod;
ampl: data MODELS/steel4.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 11.2.0: optimal solution; objective 190071.4286
2 dual simplex iterations (0 in phase I)
ampl:display Make.lb, Make.val, Make.ub, Make.rc;
:      Make.lb   Make.val Make.ub      Make.rc        :=
bands    1000     3357.14   6000     -1.77636e-15
coils     500      500      4000     -1.85714
plate     750     3142.86   3500     -3.55271e-15
;

ampl: display Time.lb, Time.body, Time.ub, Time.slack, Time.dual;
:          Time.lb  Time.body Time.ub Time.slack Time.dual    :=
reheat    -Infinity     35        35        0        1800
roll      -Infinity     40        40        0        3200
;

ampl: quit
sw: |
```

- Sabríeu trobar apartir d'aquesta taula, quina és la solució bàsica òptima?

$$x^{*T} = \left[x_B^{*T} \;\middle|\; x_N^{*T} \right]$$

Problema de la dieta: exemple

Llaunes de plats pre-cuinats	Cost Unitari	Vitamina A	Vitamina B1	Vitamina B2	Vitamina C
Unitats	$/llauna	% del mínim diari			
Vedella	3.19	60	20	10	15
Pollastre	2.59	8	0	20	20
Peix	2.29	8	10	15	10
Pernil	2.89	40	40	35	10
Macarrons	1.89	15	35	15	15
Pastis de carn	1.99	70	30	15	15
Spaghetti	1.99	25	50	25	15
Gall d'indi	2.49	60	20	15	10

(*AMPL A Modeling Language for Mathematical Programming,* cap. 2)

Problema de la dieta: elements del model

Donats els **PARÀMETRES:**

n: nombre de menjars

m: nombre de nutrients

c_i: cost unitari menjars i $\quad i = 1, \ldots, n$

l_i, u_i: quantitat mínima i màxima menjar i $\quad i = 1, \ldots, n$

a_{ij}: quantitat de nutrient j continguda en una unit. menjar i $\quad j = 1, \ldots, m$

$\underline{b}_j, \overline{b}_j$: quantitat mínima i màxima nutrient j a la dieta $\quad j = 1, \ldots, m$

Es vol trobar el valor de les **VARIABLES:**

x_i: quantitat de menjar i a la dieta $\quad i = 1, \ldots, n$

Amb l'objectiu de minimitzar la **FUNCIÓ OBJECTIU:**

$$z = \sum_{i=1}^{n} c_i x_i = c'x \quad \text{(cost total dieta)}$$

Sense violar les **CONSTRICCIONS:**

$$\underline{b}_j \sum_{i=1}^{n} a_{ij} x_i \le \overline{b}_j \quad \text{(la dieta aporta la quantita mínima de nutrients)}$$

$$l_i \le x_i \le u_i \quad \text{(fites a les variables)}$$

Problema de la dieta: formulació i implementació

Formulació algebràica amb conjunts:

$$\begin{cases} \min \quad z = \sum_{i \in P} c_i x_i = c'x & (1) \quad \textbf{funció objectiu} \\ \text{s.a.:} \quad \underline{b}_j \le \sum_{i \in P} a_{ij} x_i \le \overline{b}_j \quad j \in R & (2) \quad \textbf{constriccions} \\ \qquad l_i \le x_i \le u_i \quad i \in P & (3) \quad \textbf{domini} \end{cases}$$

P: conjunt de menjars.

R: conjunt de nutrients.

Fitxer `diet.mod`

```
set NUTR;
set FOOD;

param cost {FOOD} > 0;
param f_min {FOOD} >= 0;
param f_max {j in FOOD} >= f_min[j];

param n_min {NUTR} >= 0;
param n_max {i in NUTR} >= n_min[i];

param amt {NUTR,FOOD} >= 0;

var Buy {j in FOOD} >= f_min[j], <= f_max[j];

minimize Total_Cost:  sum {j in FOOD} cost[j] * Buy[j];

subject to Diet {i in NUTR}:
   n_min[i] <= sum {j in FOOD} amt[i,j] * Buy[j] <= n_max[i];
```

Problema de la dieta: dades

Fitxer `diet.dat`

```
data;

set NUTR := A B1 B2 C ;
set FOOD := BEEF CHK FISH HAM
   MCH MTL

param:   cost  f_min  f_max :=
  BEEF   3.19    0     100
  CHK    2.59    0     100
  FISH   2.29    0     100
  HAM    2.89    0     100
  MCH    1.89    0     100
  MTL    1.99    0     100
  SPG    1.99    0     100
  TUR    2.49    0     100 ;

param:   n_min  n_max :=
   A      700   10000
   C      700   10000
   B1     700   10000
   B2     700   10000 ;
```

```
param amt (tr):
           A    C   B1   B2 :=
   BEEF   60   20   10   15
   CHK     8    0   20   20
   FISH    8   10   15   10
   HAM    40   40   35   10
   MCH    15   35   15   15
   MTL    70   30   15   15
   SPG    25   50   25   15
   TUR    60   20   15   10 ;
```


Problema de la dieta: solució òptima

	Nre. De llaunes setmanal	Cost reduït		Vitamina A	Vitamina B1	Vitamina B2	Vitamina C
Vedella			% setmanal				
Pollastre			Preu ombra				
Peix							
Pernil							
Macarrons							
Pastis de carn							
Spaghetti							
Gall d'indi							

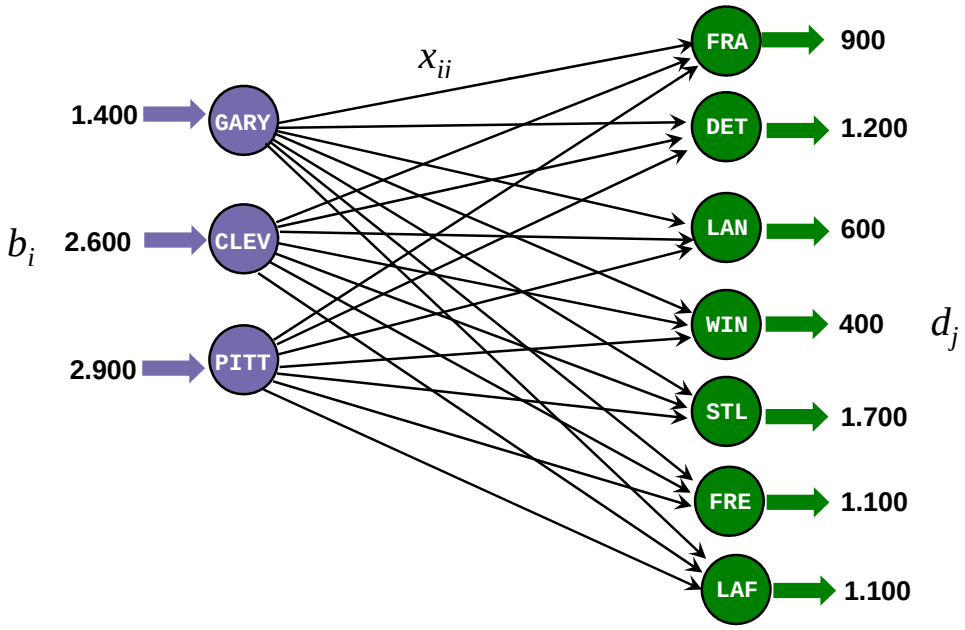
Ompliu la taula amb els valors proporcionats per AMPL/CPLEX

Problema de Transport

- La producció setmanal dels tres tallers de laminat de l'empresa SteelCat ha de ser transportada a set factories d'automòbils per l'empresa de logística TransCat, d'acord amb les següents dades:

TransCat (costos en €/Tm)		Tallers laminat			***Demanda*** (Tm)
		Gary	Cleveland	Pittsburg	
Factories	Framingham	39	27	24	900
	Detroit	14	9	14	1.200
	Lansing	11	12	17	600
	Windsor	14	9	13	400
	St. Louis	16	26	28	1.700
	Fremont	82	95	99	1.100
	Lafayette	8	17	20	1.000
Producció (Tm)		1.400	2.600	2.900	

Problema de Transport: xarxa associada



Problema de transport: exemple TransCat

PARÀMETRES:

O: Conjunt de centres origen, $O = \{\textbf{GARY, CLEV, PITT}\}$

D: nombre de centres destí, $D = \{\textbf{FRA, DET,...,LAF}\}$

c_{ij}: **cost** unitari transport $i \to j$

b_i: producció de cada taller $i \in O$ $\quad b = [1400 \quad 2600 \quad 2900]'$

d_j: **demand**a de cada factoria $j \in D$ $\quad d = [900 \quad \cdots \quad 1000]'$

VARIABLES:

x_{ij}: Tm **trans**portades de $i \to j$ $\quad i \in O;\ j \in D$

FUNCIÓ OBJECTIU:

$$\min z = \sum_{i \in O} \sum_{j \in D} c_{ij} x_{ij} \quad \text{(cost total transport)}$$

CONSTRICCIONS:

$$\sum_{j \in D} x_{ij} = b_i \quad i \in O \quad \text{(es distribueix tota la producció)}$$

$$\sum_{i \in O} x_{ij} = d_j \quad j \in D \quad \text{(es satisfà tota la demanda)}$$

$$x_{ij} \geq 0 \quad \forall i \in O;\ j \in D \quad \text{(fites a les variables)}$$

Problema TransCat: formulació i implementació

$$\begin{cases} \min & z = \sum_{i \in O} \sum_{j \in D} c_{ij} x_{ij} & \textbf{benefici total} \\ \text{s.a.:} & \sum_{j \in D} x_{ij} = b_i \quad i \in O & \textbf{producció} \\ & \sum_{i \in O} x_{ij} = d_j \quad j \in D & \textbf{demanda} \\ & x_{ij} \geq 0 \quad i \in O;\ j \in D & \end{cases}$$

Fitxer `transp.mod`

```
set ORIG;   # origins
set DEST;   # destinations

param supply {ORIG} >= 0;   # amounts available at origins
param demand {DEST} >= 0;   # amounts required at destinations

   check: sum {i in ORIG} supply[i] = sum {j in DEST}
    demand[j];

param cost {ORIG,DEST} >= 0;   # shipment costs per unit
var Trans {ORIG,DEST} >= 0;    # units to be shipped

minimize total_cost:
   sum {i in ORIG, j in DEST} cost[i,j] * Trans[i,j];

subject to Supply {i in ORIG}:
   sum {j in DEST} Trans[i,j] = supply[i];

subject to Demand {j in DEST}:
   sum {i in ORIG} Trans[i,j] = demand[j];
```

Problema TransCat: dades

```
param: ORIG:  supply :=  # defines set "ORIG" and param "supply"
        GARY   1400
        CLEV   2600
        PITT   2900 ;

param: DEST:  demand :=  # defines "DEST" and "demand"
        FRA     900
        DET    1200
        LAN     600
        WIN     400
        STL    1700
        FRE    1100
        LAF    1000 ;

param cost:
         FRA  DET  LAN  WIN  STL  FRE  LAF :=
   GARY   39   14   11   14   16   82    8
   CLEV   27    9   12    9   26   95   17
   PITT   24   14   17   13   28   99   20 ;
```

Fitxer `transp.dat`

Resolució amb AMPL/CPLEX (1/2)

```
ampl: model transp.mod; data transp.dat;
ampl: option solver cplex;
ampl: solve;
CPLEX 10.1.0: optimal solution; objective 196200
12 dual simplex iterations (0 in phase I)
ampl: display Trans.lb, Trans.val, Trans.ub, Trans.rc;
:        Trans.lb Trans.val   Trans.ub Trans.rc    :=
CLEV DET     0      1200     Infinity      0
CLEV FRA     0         0     Infinity      6
CLEV FRE     0         0     Infinity      4
CLEV LAF     0       400     Infinity      0
CLEV LAN     0       600     Infinity      0
CLEV STL     0         0     Infinity      1
CLEV WIN     0       400     Infinity      0
GARY DET     0         0     Infinity     14
GARY FRA     0         0     Infinity     27
GARY FRE     0      1100     Infinity      0
GARY LAF     0       300     Infinity      0
GARY LAN     0         0     Infinity      8
GARY STL     0         0     Infinity      0
GARY WIN     0         0     Infinity     14
PITT DET     0         0     Infinity      2
PITT FRA     0       900     Infinity      0
PITT FRE     0         0     Infinity      5
PITT LAF     0       300     Infinity      0
PITT LAN     0         0     Infinity      2
PITT STL     0      1700     Infinity      0
PITT WIN     0         0     Infinity      1
;
```

Resolució amb AMPL/CPLEX (2/2)

```
ampl: display Supply.lb, Supply.body, Supply.ub, Supply.slack, Supply.dual;
:    Supply.lb Supply.body Supply.ub Supply.slack Supply.dual    :=
CLEV    2600       2600       2600         0           17
GARY    1400       1400       1400         0            8
PITT    2900       2900       2900         0           20
;
```

```
ampl: display Demand.lb, Demand.body, Demand.ub, Demand.slack, Demand.dual;
:   Demand.lb Demand.body Demand.ub Demand.slack Demand.dual    :=
DET    1200       1200       1200         0           -8
FRA     900        900        900         0            4
FRE    1100       1100       1100         0           74
LAF    1000       1000       1000         0            0
LAN     600        600        600         0           -5
STL    1700       1700       1700         0            8
WIN     400        400        400         0           -8
;
```


Laboratori 1 (1/2)

- Resolgueu amb l'ajut d'AMPL/CPLEX la següent variant del problema de la dieta vist a teoria

Llaunes de plats pre-cuinats	Contingut en sodi (mg)	Contingut en Calories (cal)	Nre. Llaunes mínim	Nre. Llaunes màxim.
Vedella	938	295	2	10
Pollastre	2180	770	2	10
Peix	945	440	2	10
Pernil	278	430	2	10
Macarrons	1182	315	2	10
Pastis de carn	896	400	2	10
Spaghetti	1329	370	2	10
Gall d'indi	1397	450	2	10
Requeriments setmanal	≤ 50.000 mg Na	[16.000, 24.000] cal		

Laboratori 1 (2/2)

- Ompliu la següent taula amb la solució obtinguda per AMPL/CPLEX i entregueu-la en finalitzar la classe

	Nre. de llaunes setmanal	Cost reduït		A	B1	B2	C	Na	Cal
Vedella			% setmanal						
Pollastre			Preu ombra						
Peix									
Pernil									
Macarrons									
Pastis de carn									
Spaghetti									
Gall d'indi									

Nom alumne: