

# INTRODUCCIÓ A LINGO

Jordi Castro  
Universitat Politècnica de Catalunya  
Departament d'Estadística i Investigació Operativa.  
Secció Informàtica.

Lingo és un paquet comercial de modelització de problemes. Això vol dir que amb ell es poden formular, en un llenguatge més o menys senzill, una gran varietat de models de programació matemàtica (tant lineal, com no lineal, com entera), i obtenir directament la solució del mateix. Internament el que fa Lingo quan li entrem un model és analitzar-lo, decidir quin tipus de problema és (lineal, no lineal, etc.) i cridar a una rutina interna d'optimització específica per aquest tipus de problemes. La versió que teniu disponible de Lingo corre sota Windows 3.xx i Windows95. Podeu obtenir una versió d'estudiant gratuïta a l'adreça:

<http://www.lindo.com/download.html>

Aquí només esmentarem algunes de les característiques del llenguatge de modelització de Lingo. Per tenir més detalls es pot consultar el manual de referència del paquet (adreceu-vos al vostre professor), o bé es pot donar una ullada al "help" (força clar) que té incorporat.

La forma de treballar amb Lingo es pot resumir breument de la següent manera. Quan entreu a Lingo us apareixerà una finestra on podreu editar el vostre model. Aquest convé que el salveu en un fitxer. Això ho podeu fer amb l'opció "Save" del menú "File" de la barra superior (a través d'aquesta barra tenim accés a totes les possibilitats que Lingo ens ofereix). Un cop introduït el model, podem mirar de solucionar-lo amb l'opció "Solve" del menú "Lingo". Si el model que heu entrat es correcte, Lingo mirarà de solucionar-lo. Si hi ha algun error de sintaxi, us ho dirà. Si tot ha anat bé, una nova finestra us mostrarà la solució obtinguda. A l'igual que abans amb el model, podeu salvar aquesta solució en un altre fitxer (per tal de poder incorporar-la a l'informe). Adoneu-vos que, en la solució, Lingo ens mostra el valor òptim de cada variable, i el valor de les restriccions en aquest punt. La primera restricció acostuma a ser el valor de la funció objectiu.

Ara ens falta el més important, que és saber com formular un problema amb el llenguatge de Lingo. Aquí només mostrarem el seu funcionament a través d'uns pocs exemples (fer una descripció detallada implicaria reproduir el manual de Lingo, cosa que no tindria sentit). Aquells que necessiteu o tingueu interès en saber més coses, consulteu el manual de referència o el "help" interactiu de Lingo. Passem a formular aquests exemples amb el llenguatge de Lingo.

## Exemple 1.

Volem solucionar el següent problema de programació lineal:

$$\begin{aligned} \min \quad & x_1 + x_2 + x_3 \\ \text{s.t.} \quad & \\ & 2x_1 - 3x_2 \geq 3 \\ & -3x_1 + 3x_2 \leq 5 \\ & 1 \leq x_1 \leq 3 \quad x_2 \geq 0 \quad x_3 \text{ lliure} \end{aligned}$$

Aquest problema el podríem formular de la següent manera usant Lingo:

---

!tot el que comença amb admiració es un comentari;

```

!totes les sentències han d'acabar amb un punt i coma, comentaris inclosos;

!això es la funció objectiu;
!si ocupés més d'una línia no caldria;
!afegir punt i coma a cada una d'elles;
min= x1+x2+x3;

!ara escrivim les restriccions;
2*x1-3*x2 >= 3;
-3*x1+3*x2 <= 5;

!i finalment les fites a les variables;
!per defecte totes les variables tenen una fita inferior igual a 0;
!la funció @bnd ens permet definir límits superiors i inferiors;
@bnd(1,x1,3);
!la funció @free ens permet definir variables lliures;
@free(x3);

```

---

### Exemple 2.

Volem solucionar el següent problema de programació no lineal:

$$\begin{aligned}
 \max \quad & e^{x_1} + \ln x_2 + x_3^2 \\
 \text{s.t.} \quad & \\
 & \frac{2x_1}{3x_2} \geq 3 \\
 & x_1/x_2 \leq 5 \\
 & x_1 \geq 3
 \end{aligned}$$

Això podria ser formulat com:

```

!funció objectiu;
!aquí es mostren les funcions @exp i @log. Lingo té moltes;
!més funcions matemàtiques;
max= @exp(x1)+@log(x2)+x3^2;
!restriccions;
(2*x1)/(3*x2) >= 3;
x1/x2 <= 5;
!fites a les variables;
@bnd(3,x1,1.0e+20);
!tambe podríem haver fet x1>=3.0, però en aquest cas consideraria;
!això com una nova restricció, i no com un límit de variables;

```

---

### Exemple 3.

Als exemples anteriors hem usat Lingo més com un paquet per solucionar problemes que per modelitzar-los. De fet als exemples anteriors no hem explotat prou les possibilitats de Lingo. Veurem ara com modelitzar un problema de programació entera, concretament el problema de la motxilla. En aquest problema, disposem d'una sèrie d'objectes que volem col·locar dins una motxilla. Tenim 5 objectes, i sabem que afegir l'objecte  $i$  ens suposa un benefici de  $g_i$  unitats, i ocupa un volum de  $v_i$  litres. El volum total de la motxilla és de  $V$  litres. Hem de formular el problema de forma que carreguem la motxilla maximitzant el nostre benefici. La formulació matemàtica d'aquest problema és:

$$\begin{aligned} \max \quad & \sum_{i=1}^5 g_i x_i \\ \text{s.t.} \quad & \\ & \sum_{i=1}^5 v_i x_i \leq V \\ & x_i \in \{0, 1\} \quad i = 1, \dots, 5 \end{aligned}$$

Usant Lingo podríem fer:

---

```
!amb la sentència ‘‘model’’ indiquem que creem un nou model;
!acabem el model amb la darrera sentència ‘‘end’’;
!model: no acaba amb punt i coma;
model:
!la sentència ‘‘sets’’ serveix per crear conjunts de dades;
!en aquest cas creem un conjunt de 5 objectes;
!cada objecte tindrà 3 atributs: volum, benefici i x;
!(x és la variable a optimitzar que valdrà 1 o 0);
!sets: no acaba amb punt i coma;
  sets:
    objectes/1..5/:x,volum,benefici;
  endsets

!la sentència data serveix per assignar valors a variables;
!assignarem per als 5 objectes el benefici i volum que ocupa, i el;
!volum total de la motxilla;
!data: no acaba amb punt i coma;
  data:
    V= 54;
    benefici= 10 11 9 13 12;
    volum= 17 18 15 21 20;
  enddata

!funcio objectiu;
!aquí indiquem amb la funció @sum que sumi el producte de x per benefici;
```

```

!per a tots els objectes;
  max= @sum(objectes: x*benefici);

!restriccions;
!aquí indiquem que la suma per a tots els objectes de x pel volum ha de;
!ser menor que el volum total de la motxi_l.la;
  @sum(objectes:x*volum) <= V;

!límits de les variables;
!finalment indiquem amb la funció @bin que totes les variables x de totes;
!els objectes són binàries (només poden prendre els valors 0 o 1);
!la funció @for ens permet tractar tots els elements del conjunt de dades;
!objectes;
  @for(objectes:@bin(x));
end

```

---

#### Exemple 4.

Acabarem modelitzant amb el llenguatge de Lingo el problema de la dieta. Disposem de 3 aliments A, B i C per confeccionar una dieta, cadascun dels quals té una certa proporció de dos nutrients N1 i N2. Aquestes proporcions ens vénen donades segons la taula següent:

	N1	N2
A	35%	10%
B	45%	20%
C	10%	30%

Sabem que cada kg. dels aliments A, B i C costa 100, 110 i 85 pts. respectivament. Volem obtenir quines quantitats d'aquests aliments ens calen per obtenir la dieta més barata possible que aportí més de 40 i 50 kg dels nutrients N1 i N2. La formulació d'aquest problema quedaria com:

$$\begin{aligned}
 \min \quad & 100x_A + 110x_B + 85x_C \\
 \text{s.t.} \quad & \\
 & 0.35x_A + 0.45x_B + 0.10x_C \geq 40 \\
 & 0.10x_A + 0.20x_B + 0.30x_C \geq 50 \\
 & x_A \geq 0 \quad x_B \geq 0 \quad x_C \geq 0
 \end{aligned}$$

Utilitzant el llenguatge de modelització de Lingo podríem haver fet:

---

```

model:
  sets:
  ! definim el conjunt de 3 productes A,B,C amb atributs preu i quantitat;
  ! de producte (la variable a optimitzar);
    productes /A,B,C/: quant_p,preu;

```

```

! definim el conjunt d'ingredients, amb la quantitat requerida;
    ingredients /N1,N2/: quant_i;
! definim la matriu que dona el percentatge d'ingredient per producte;
! les matrius es defineixen com un producte cartesià de conjunts;
    composicio(productes,ingredients): percent;
endsets
! assignem valors;
data:
    preu= 100 110 85;
    quant_i= 40 50;
    percent= 0.35 0.10
             0.45 0.20
             0.10 0.30;

enddata
! funcio objectiu;
! suma de la quantitat pel preu per a tots els productes;
min= @sum(productes: quant_p*preu);
! restriccions;
! una restricció per a cada ingredient, amb terme;
! de la dreta igual a quant_i(i);
@for(ingredients(i):
! i el terme de l'esquerra de cada restricció suma per a tots;
! els aliments el producte de la seva quantitat pel percentatge;
! de l'ingredient considerat;
    @sum(productes(j): percent(j,i)*quant_p(j) ) >= quant_i(i));
end

```

---