Aplicacions de la Programació No Lineal.

Pràctiques.

Jordi Castro, F. Javier Heredia
Departament d'Estadística i Investigació Operativa
Secció d'Informàtica
UPC

Índex

1 Pràctiques sense restriccions amb Matlab.

Presentem en aquest capítol l'enunciat de les cinc pràctiques a realitzar amb ajut de Matlab.

EXERCICI 1. : MÈTODE DE NELDER I MEAD.

OBJECTIUS : familiaritzar-se amb el detall del mecanisme d'iteració del mètode de Nelder i Mead. Observar la dependència del comportament del mètode amb els paràmetres t, α, β i γ. Comprendre el comportament global del mètode.

GUIÓ :

1.1.– Primer, heu de crear un fitxer `ini.m` que contingui com a punt inicial l'origen de coordenades:

```
x0=[ 0 0 0 0 ]'
```

1.2.– El valor per defecte dels paràmetres que controlen el mètode de Nelder i Mead és $t = 1$, $\alpha = 1$, $\gamma = 2$, $\beta = 0.5$. Executeu el mètode de Nelder i Mead amb aquest paràmetres per defecte i amb dues variacions (per exemple, amb $t = 2$ i $t = 0.5$, $\gamma = 4$ i $\gamma = 1.5$, etc). Construïu una taula comparativa amb els valors del nombre d'iteracions necessaris per a satisfer el criteri de convergència, nombre d'avaluacions de la funció objectiu i valor de la funció objectiu a l'òptim. Comenteu els resultats.

1.3.– Torneu a col.locar el valor per defecte dels paràmetres. Un cop resolt el problema, sortiu de l'aplicació ONLSR i carregueu el fitxer `*.mat` (fent `load out.mat`, si heu triat el nom per defecte). Volem ara resseguir l'aplicació del mètode al llarg de dues iteracions. Busqueu, mirant el vector `vtpaso`, dues iteracions consecutives *que no siguin dues reflexions*. Un cop identificades, heu d'extreure les dades corresponents a aquestes iteracions de la matriu `mat`. Imaginem que hem seleccionat les iteracions 7 i la 8. Si fem:

```
> [Xpol7,Fpol7]=XFNeMe(7,mat) [Enter]
> [Xpol8,Fpol8]=XFNeMe(8,mat) [Enter]
```

les primeres $n+1$ files de la matriu `Xpol7` i del vector `Fpol7` contindran, respectivament,

les components i valors de la funció objectiu dels vèrtexs del políedre a la iteració 7, ordenats per valors creixents de $f(x)$. Les posicions $n+2$ i $n+3$ de `Xpol7` i `Fpol7` corresponen, respectivament, al centroide x^7_{n+2} i al punt reflexat x^7_{n+3}. Amb aquesta informació feu el següent:

a) Calculeu les coordenades del centroide x^k_{n+2} i punt reflexat x^k_{n+3}, així com $f(x^k_{n+2})$ i $f(x^k_{n+3})$, i comproveu que coincideixen amb les que proporciona `XFNeMe(k,ma)`.

b) Realitzeu, amb l'ajut de MATLAB, dues iteracions del mètode de Nelder i Mead, comprovant que els resultats que obteniu (actualització del vèrtexs del políedre) coincideixen amb els emmagatzemats a la matriu `mat`.

1.4.– Observeu si es produeix el fet que en un conjunt d'iteracions successives el valor de la funció objectiu sobre el millor vèrtex no millora. Si és així expliqueu a què es deu aquest fet.

INFORME :

L'informe d'aquest exercici ha de contenir:

1.- *E*l llistat de sortida de la resolució amb els paràmetres per defecte.

2.- *L*a taula comparativa *comentada* demanada a l'apartat 1.2.

3.- *E*l detall dels càlculs realitzats en les dues iteracions de l'apartat 1.3.

4.- *L*a part del llistat de sortida on es vegi l'efecte indicat a l'apartat 1.4 i l'explicació d'aquest fet.

EXERCICI 2. : Exploració Lineal.

OBJECTIUS : familiaritzar-se amb el mecanisme bàsic d'una rutina d'exploració lineal. Comprendre com els ajustos quadràtics i cúbics són usats per obtenir els successius punts d'avaluació. Comprendre les condicions d'Armijo-Goldstein, usades per detectar la factibilitat o no dels punts generats. Comprovar que el punt solució de l'exploració lineal verifica les condicions d'Armijo-Goldstein.

GUIÓ :

2.1.– Primer heu de crear un fitxer `ini.m` que contingui com a punt inicial x_0 l'origen de coordenades, i com direcció de descens p la de menys el gradient de la funció que esteu considerant (en aquest cas, la funció de Wood amb els paràmetres particulars per a cadascú de vosaltres):

```
x0=[ 0 0 0 0 ]';
p= -gwood(x0)';
```

Nota: Procureu no definir els paràmetres a, b, c i d de la funció de Wood com a `a`, `b`, `c`, i `d` dins `ini.m`, fent un "`global a b c d`" després. Això us donarà problemes durant

l'execució dels programes de ONLSR donat que internament usen també unes variables anomenades a, b, c o d. El fet d'haver-les fet globals provocaria que perdéssiu els valors que originalment les hi havíeu assignades.

2.2.– Usar ONLSR per fer una exploració lineal des de x_0 en la direcció p, fins trobar un punt solució pel que fa a l'exploració lineal. Adoneu-vos de com el primer punt es calcula mitjançant un ajust quadràtic, i el segon per un ajust cúbic. Considerant aquestes dos primeres iteracions de l'algorisme heu de:

a) Realitzar l'ajust quadràtic inicial a ma. Comproveu com el resultat obtingut coincideix amb el proporcionat per ONLSR. Comproveu també la condició primera d'Armijo-Goldstein, i observeu com obteniu els mateixos resultats que ONLSR.

b) Realitzar l'ajust cúbic següent a ma. De nou, comproveu que obteniu el mateix punt que ONLSR. També heu de verificar la condició primera d'Armijo-Goldstein, observant com el resultat del test coincideix amb el proporcionat per ONLSR.

2.3.– Comproveu analíticament com el punt solució proporcionat per ONLSR verifica les dues condicions d'Armijo-Goldstein.

2.4.– Amb l'ajuda del programa `agplot.m`, comproveu gràficament com el punt solució de l'exploració lineal proporcionat per ONLSR verifica les dues condicions d'Armijo-Goldstein.

INFORME :

L'informe d'aquest exercici ha de contenir:

1.- El llistat de sortida de l'exploració lineal proporcionat per ONLSR, on es resaltin les dades corresponents als dos primers ajustos (quadràtic i cúbic respectivament).

2.- El detall dels càlculs realitzats en els dos punts de l'apartat 2.2, tant pel que fa a la comprovació del punt obtingut com de la verificació o no de la primera condició d'Armijo-Goldstein.

3.- El detall dels càlculs realitzats a l'apartat 2.3 per comprovar que les dues condicions d'Armijo-Goldstein es verifiquen al punt solució.

4.- La gràfica obtinguda a l'apartat 2.4, resaltant la situació del punt solució.

EXERCICI 3. : Gradient Conjugat.

OBJECTIUS : familiaritzar-se amb l'algorisme del gradient conjugat per a funcions no lineals (variant de Fletcher-Reeves). Adonar-se de que l'algorisme d'exploració lineal és usat per solucionar un subproblema a cada iteració de l'algorisme del gradient conjugat no lineal.

GUIÓ :

3.1.– Primer heu de crear un fitxer `ini.m` que contingui com a punt inicial x_0 el vector

$(-3, 1, -3, 1)$:

```
x0=[ -3 1 -3 1 ]';
```

Nota: Treballarem amb la funció de Wood. Procureu no definir els paràmetres a, b, c i d de la funció de Wood com a `a`, `b`, `c`, i `d` dins `ini.m`, fent un "`global a b c d`" després. Això us donarà problemes durant l'execució dels programes de ONLSR donat que internament usen també unes variables anomenades a, b, c o d. El fet d'haver-les fet globals provocaria que perdéssiu els valors que originalment les hi havíeu assignades.

3.2.– Usar ONLSR per minimitzar la funció de Wood amb l'algorisme del gradient conjugat. Si l'algorisme no convergeix perquè ha assolit el màxim d'iteracions, amplieu aquest paràmetre. Si no convergeix per problemes amb l'exploració lineal, modifiqueu el punt inicial.

3.3.– Repetiu el punt 3.2 per a diferents valors del paràmetre β_{AG2} de la segona condició d'Armijo-Goldstein, usant sempre el mateix punt inicial x_0. En particular, useu els 9 valors $0.1, 0.2, \ldots, 0.8, 0.9$. Realitzeu una taula on, per a cada valor de β_{AG2} aparegui el nombre d'iteracions requerit pel mètode del gradient conjugat per obtenir el punt òptim, el nombre d'avaluacions de la funció objectiu, i el valor de la funció objectiu al punt òptim.

3.4.– Usar ONLSR per minimitzar la funció de Wood amb l'algorisme del gradient, usant com punt x_0 el mateix de l'apartat 3.2. Observeu com el mètode del gradient, si convergeix, requereix més iteracions que el del gradient conjugat.

3.5.– Escolliu dues iteracions consecutives. En la primera d'elles la direcció de moviment s'ha d'haver calculat com $d_k = -\nabla f(x_k) = -g_k$. En la segona la direcció de moviment s'ha d'haver actualitzat usant la fórmula habitual $d_{k+1} = -g_{k+1} + \beta_k d_k$. Realitzeu les dues iteracions manualment, usant com longituds de pas α_k i α_{k+1} les indicades per ONLSR, i comproveu com les direccions i punts obtinguts coincideixen amb els proporcionats per l'aplicació.

3.6.– Escolliu una de les dues iteracions del punt 3.5, i comproveu analíticament com el punt retornat per l'exploració lineal satisfà les dues condicions d'Armijo-Goldstein.

INFORME :

L'informe d'aquest exercici ha de contenir:

1.- *E*l llistat de sortida de l'evolució de l'algorisme del gradient conjugat obtingut al punt 3.2.

2.- *L*a taula generada al punt 3.3.

3.- *E*l llistat de sortida de l'evolució de l'algorisme del gradient obtingut al punt 3.4.

4.- *E*l detall dels càlculs realitzats als punt 3.5.

5.- *E*l detall dels càlculs realitzats a l'apartat 3.6.

EXERCICI 4. : Mètode de Newton.

OBJECTIUS : familiaritzar-se amb el mètode de Newton per a funcions no lineals. Entendre un dels mètodes de Newton modificats (variant de Luenberger). Entendre el paper que juga la definició de la Hessiana en el càlcul de la direcció de moviment. Adonar-se de que l'algorisme d'exploració lineal és usat per solucionar un subproblema a cada iteració del mètode de Newton.

GUIÓ :

4.1.– Primer heu de crear un fitxer `ini.m` que contingui com a punt inicial x_0 el vector $(0, 100, 0, 100)$:

```
x0=[0 100 0 100]';
```

Nota: Treballarem amb la funció de Wood. Procureu no definir els paràmetres a, b, c i d de la funció de Wood com a `a`, `b`, `c`, i `d` dins `ini.m`, fent un "`global a b c d`" després. Això us donarà problemes durant l'execució dels programes de ONLSR donat que internament usen també unes variables anomenades a, b, c o d. El fet d'haver-les fet globals provocaria que perdéssiu els valors que originalment les hi havíeu assignades.

4.2.– Minimitzeu la funció de Wood amb l'ajut de ONLSR a partir del punt x_0 anterior usant el mètode del gradient, del gradient conjugat, de Newton, i de Newton modificat en la variant de Luenberger. Escolliu un altre punt qualsevol a l'atzar x'_0. Torneu a minimitzar la funció a partir de x'_0 usant els mateixos algorismes que abans. Tabuleu els resultats obtinguts per a cada punt i amb cada algorisme, indicant el valor final de la funció objectiu, la norma del gradient assolida, i el nombre d'iteracions realitzats. Comenteu els resultats obtinguts, especialment en tot allò referent amb la velocitat de convergència, tot indicant qualsevol anomalia obtinguda en les execucions.

4.3.– Minimitzeu la funció de Wood amb el mètode de Newton modificat en la variant de Luenberger amb diferents valors del paràmetre ϵ (proveu $\epsilon = 0.1, 0.2, 0.3, \ldots, 0.9, 1.0$). Tabuleu els resultats obtinguts, indicant en cada cas el valor final de la funció objectiu, la norma del gradient assolida, i el nombre d'iteracions realitzats. Comenteu els resultats obtinguts.

4.4.– Escolliu una iteració on la Hessiana sigui indefinida (per ex., la primera o alguna de les primeres iteracions). Realitzeu una iteració manualment (amb l'ajut de Matlab), i comproveu que els resultats obtinguts coincideixen amb els proporcionats per ONLSR. (Només heu de calcular la direcció, sense fer l'exploració lineal; useu el valor de α_k proporcionat per ONLSR.)

4.5.– Comproveu que, tot i que la Hessiana a x_0 es indefinida, la direcció que obtindríeu usant el mètode de Newton (sense cap modificació) proporciona una direcció de descens. (Feu els càlculs necessaris per fer aquesta comprovació.) Contradiu això la teoria sobre el mètode de Newton? Per què convé aleshores introduir els mètodes de Newton modificats?

INFORME :

L'informe d'aquest exercici ha de contenir:

1.- *E*ls llistats de sortida de l'evolució del mètode de Newton i Newton modificat en la variant de Luenberger obtinguts al punt 4.2.

2.- *L*a taula generada al punt 4.2, amb els comentaris realitzats.

3.- *L*a taula generada al punt 4.3, amb els comentaris realitzats.

4.- *E*l detall dels càlculs realitzats al punt 4.4.

5.- *E*l detall dels càlculs realitzats a l'apartat 4.5, i la resposta a la qüestió plantejada.

EXERCICI 5. : Mètodes quasi-Newton.

OBJECTIUS : familiaritzar-se amb els mètode quasi-Newton DFP i BFGS. Entendre sota quines condicions pot garantir-se que la nova actualització serà definida positiva. Adonar-se de que l'actualització BFGS, en general, proporciona millors resultats que la DFP. Adonar-se de que l'algorisme d'exploració lineal és usat per solucionar un subproblema a cada iteració del mètode de Newton.

GUIÓ :

5.1.– Primer heu de crear un fitxer `ini.m` que contingui com a punt inicial x_0 el vector $(0, 0, 0, 0)$:

```
x0=[0 0 0 0]';
```

Nota: Treballarem amb la funció de Wood. Procureu no definir els paràmetres a, b, c i d de la funció de Wood com a `a`, `b`, `c`, i `d` dins `ini.m`, fent un "`global a b c d`" després. Això us donarà problemes durant l'execució dels programes de ONLSR donat que internament usen també unes variables anomenades a, b, c o d. El fet d'haver-les fet globals provocaria que perdéssiu els valors que originalment les hi havíeu assignades.

5.2.– Minimitzeu la funció de Wood amb l'ajut de ONLSR a partir del punt x_0 anterior usant el mètode del gradient, del gradient conjugat, de Newton, de Newton modificat en la variant de Luenberger, quasi-Newton amb actualització DFP, i quasi-Newton amb actualització BFGS. Escolliu un altre punt qualsevol a l'atzar x'_0. Torneu a minimitzar la funció a partir de x'_0 usant els mateixos algorismes que abans. Tabuleu els resultats obtinguts per a cada punt i amb cada algorisme, indicant la norma del gradient assolida i el nombre d'iteracions realitzades. Comenteu els resultats obtinguts.

5.3.– Minimitzeu la funció de Wood amb els mètodes quasi-Newton DFP i BFGS considerant 5 punts inicials diferents (un d'ells que sigui el punt x_0 original). Tabuleu els resultats obtinguts, indicant la norma del gradient assolida, i el nombre d'iteracions realitzades. Compareu el comportament dels mètodes DFP i BFGS.

5.4.– Realitzeu manualment amb l'ajut de Matlab la primera iteració dels mètodes DFP i BFGS. Calculeu S_1 i H_1, i les respectives direccions d_1. Tingueu present que a ONLSR H_0 i S_0 s'inicialitzen com $|f(x_0)| \cdot I$.

a) Comproveu que S_1 i H_1 són definides positives de dues formes: i) calculat els valors propis (comanda Matlab `eig()`); ii) observant que $p_0^T y_0 > 0$ tant al mètode DFP com al BFGS.

b) Comproveu que d_1 és una direcció de descens, per al mètode DFP i BFGS. Digueu si calia fer aquesta comprovació, a la vista del resultat del punt a) anterior.

c) Proveu que tots els punts que proporciona ONLSR satisfan la propietat $p_k^T y_k > 0$, degut a que verifiquen la 2a condició d'Armijo-Goldstein (és a dir, heu de demostrar que si l'exploració lineal satisfà la condició AG2 aleshores es garanteix $p_k^T y_k > 0$).

INFORME :

L'informe d'aquest exercici ha de contenir:

1.- *E*ls llistats de sortida de l'evolució dels mètodes DFP i BFGS partint del punt inicial x_0 obtinguts a l'apartat 5.2.

2.- La taula generada al punt 5.2, amb els comentaris realitzats.

3.- La taula generada al punt 5.3, amb els comentaris realitzats.

4.- El detall dels càlculs realitzats al punt 5.4 corresponents a la primera iteració dels mètodes DFP i BFGS.

5.- Els càlculs i comentaris realitzats per als apartats 5.4 a) y 5.4 b).

6.- La petita demostració de l'apartat 5.4 c).

2 Introducció a LINGO.

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 problemes 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, i es troba instal·lada als PC's de les aules informàtiques de la F.M.E.

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'opcio "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:

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

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;
```

```
!aixo com una nova restriccio, 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 motxil·la. En aquest problema, disposem d'una sèrie d'objectes que volem col·locar dins una motxil·la. 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 motxil·la és de V litres. Hem de formular el problema de forma que carreguem la motxil·la maximitzant el nostre benefici. La formulació matemàtica d'aquest problema és:

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

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 motxil·la;
```

```
!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;
!aqui indiquem amb la funció @sum que sumi el producte de x per benefici;
!per a tots els objectes;
  max= @sum(objectes:  x*benefici);

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

!límits de les variables;
!finalment indiquem amb la funcio @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 aporti 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 restriccio 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
```

3 Aprenentatge d'una xarxa neuronal.

Les xarxes neuronals són un tipus de funcions, o eines matemàtiques, que durant els darrers anys han gaudit d'una forta acceptació per solucionar una gran diversitat de problemes. Dues de les raons que expliquen aquesta acceptació han estat, per una banda, la seva facilitat d'ús, i en segon lloc, el fet que permeten solucionar, de forma acceptable, certs tipus de problemes on altres tècniques, o bé requeririen una gran quantitat de temps, o bé no són capaces de trobar una solució el suficientment acurada. Sempre que es vol usar una xarxa neuronal, el primer que cal fer és ajustar una sèrie de paràmetres (anomenats pesos) que determinaran la seva forma de funcionament. L'objectiu de la pràctica serà, doncs, el de trobar aquests paràmetres (calibració de la xarxa) usant tècniques d'optimització. En primer lloc es descriurà de forma simple què és una xarxa neuronal, i a continuació s'indicarà el problema d'optimització que caldrà solucinar, amb l'ajut del paquet AMPL.

3.1 Funcionament d'una xarxa neuronal.

Una xarxa neuronal no és més que una funció matemàtica $F : \mathbb{R}^{n_e} \to \mathbb{R}^{n_s}$ que rep un vector d'entrada $x_e \in \mathbb{R}^{n_e}$ i genera un vector de sortida $x_s \in \mathbb{R}^{n_s}$. El vector d'entrada correspon a una sèrie de valors coneguts per nosaltres. La xarxa neuronal és avaluada amb aquests valors, i ens proporciona un determinat resultat (que pot estar compost per un o varis valors, en funció de la dimensió n_s dels vectors de sortida x_s). Per entendre com funciona una xarxa neuronal, considerarem una de molt simple, tal i com es mostra a la Figura 3.1. La xarxa neuronal es troba formada per un conjunt de nodes (anomenats "neurones" en l'argot de les xarxes neuronals), els quals es transmeten informació en el sentit indicat pels arcs. Aquests nodes es disposen per capes. Al cas concret mostrat a la Figura 3.1, tenim un total de set nodes disposats en tres capes. A la xarxa que estem considerant, la capa inferior (formada per quatre nodes) rep quatre valors d'entrada (els $x_i,\ i = 1, \ldots, 4$), mentre que la capa de sortida, formada només per una neurona, ens proporciona el resultat de la xarxa. En aquest cas, doncs, tenim que $n_e = 4$ i $n_s = 1$.

El funcionament de la xarxa és el següent. Cada node i rep un "input" I_i, i el transforma en un "output" O_i. Aquesta transformació es realitza mitjançant una determinada funció $f(x)$. Entre les més usades tenim la funció tangent hiperbòlica $tgh(x) = (e^x - e^{-x})/(e^x + e^{-x})$, i la funció sigmoidal

$$f(x) = \frac{1}{1 + e^{-x}}$$

la qual té un aspecte com el que es mostra a la Figura 3.2. Podem observar com la funció sigmoidal sempre torna un valor entre 0 i 1 (la tangent hiperbòlica torna valors entre -1 i 1). En aquesta pràctica usarem la funció sigmoidal. Per tant, els valors O_i de sortida d'una neurona

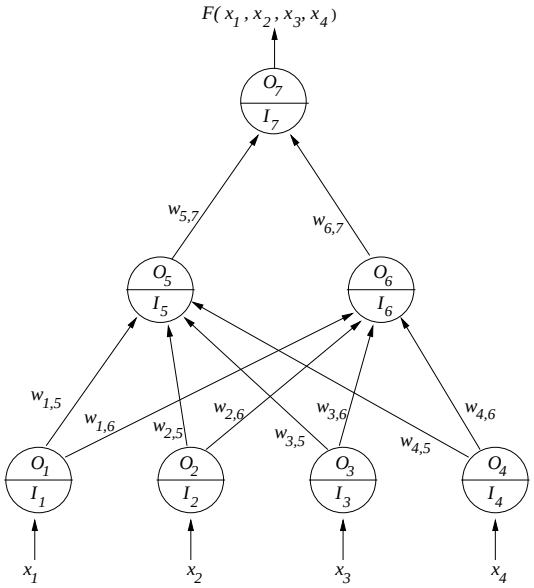

Figura 3.1 : Esquema d'una xarxa neuronal simple.

seran calculats de forma $O_i = f(I_i)$, on f serà la funció sigmoidal. L'"output" O_7 de la darrera neurona ens proporciona el resultat de la xarxa neuronal.

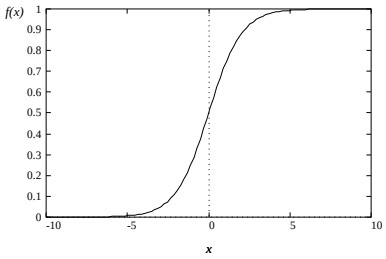

Figura 3.2 : Gràfica de la funció sigmoidal $f(x) = 1/(1 + e^{-x})$.

Ara cal veure com calcular els "inputs" I_i de cada neurona i. Aquests es calcularan sumant els "outputs" de les neurones j que envien informació a la neurona i, multiplicant-los abans pels valors $w_{j,i}$, que són mostrats a la Figura 3.1. Aquests valors $w_{j,i}$ s'anomenen "pesos", i serveixen per ponderar la informació que arriba a cada neurona. Per exemple, al node 5 de la Figura 3.1, l'"input" I_5 que li arriba es calcularia com:

$$I_5 = w_{1,5}O_1 + w_{2,5}O_2 + w_{3,5}O_3 + w_{4,5}O_4$$

Anàlogament es faria per a la resta de nodes. Procedint d'aquesta forma, proporcionant uns

valors x_1, x_2, x_3 i x_4 determinats a la xarxa neuronal, el resultat que obtindríem seria:

$$\begin{aligned} F(x_1,x_2,x_3,x_4) = O_7 = f(I_7) = f(w_{5,7}O_5 + w_{6,7}O_6) = f(w_{5,7}f(I_5) + w_{6,7}f(I_6)) = \\ = f(w_{5,7}f(\sum_{i=1}^{4} w_{i,5}O_i) + w_{6,7}f(\sum_{i=1}^{4} w_{i,6}O_i)) = \\ = f(w_{5,7}f(\sum_{i=1}^{4} w_{i,5}f(x_i)) + w_{6,7}f(\sum_{i=1}^{4} w_{i,6}f(x_i))) \end{aligned} \tag{3.1}$$

essent $f()$ la funció sigmoidal, com abans hem indicat. En aquest cas tan senzill hem pogut escriure de forma detallada la funció que representa la xarxa neuronal mitjançant l'equació (3.1) . Si tinguéssim una quantitat més elevada de nodes i capes, això ja no seria possible, pels múltiples nivells de recurrència que apareixerien. Tanmateix, per realitzar la pràctica n'hi ha prou amb aquesta xarxa neuronal tan simple.

Abans de continuar, val la pena indicar que, a les xarxes neuronals usades realment a la pràctica, a part dels pesos w mostrats a la Figura 3.1, cada node té un pes addicional que sempre es suma directament a l'"input" de la xarxa (és a dir, sense multiplicar-lo per cap "output" d'un node d'una capa inferior). Aquests pesos especials s'anomenen "bias" a l'argot de les xarxes neuronals. A la pràctica, per tal de reduir el nombre de variables del problema plantejat, no considerarem aquests pesos addicionals.

En funció del vist fins ara, podem veure com el comportament de la xarxa neuronal ve regit precisament pels valors dels pesos w. Depenen dels valors concrets de w, la xarxa produirà uns o altres resultats. Ara queda la qüestió de: com obtenir els pesos w escaients?. Doncs calen dues coses: en primer lloc, unes dades per tal de poder fer "aprendre" a la xarxa el comportament que ha de tenir (veurem això més clar una mica més endavant), i en segon lloc, un mètode (numèric) per poder fer aquest "aprenentatge" (el concepte d'"aprenentatge" també forma part de l'argot de les xarxes neuronals). Un cop hem calibrat la xarxa (hem ajustat els valors w), aquesta ja està llesta per, a partir d'un vector x_e de dades d'entrada, donar una determinada resposta.

3.2 Obtenció dels pesos w.

Tal i com abans s'ha indicat, ens queda veure com determinar els pesos w. Hi ha diverses formes. En aquest treball, però, usarem una basada en un problema de mínims quadrats no lineals (problema de minimització sense restriccions), on les variables a optimitzar seran els pesos w. Per obtenir aquests pesos cal disposar de p vectors de dades d'entrada $x_e \in \mathbb{R}^{n_e}$, i p vectors de dades de sortida $x_s \in \mathbb{R}^{n_s}$, que han de correspondre amb els valors associats a les dades d'entrada. La idea és que s'ajustin els pesos w de la xarxa de forma que $F(x_{e_i}; w) \approx x_{s_i},\ i = 1, \ldots, p$, on $F(x; w)$ representa la resposta de la xarxa quan la seva entrada és el vector x, i té uns pesos w. Podríem dir que els p vectors $x_{e_i}, x_{s_i},\ i = 1, \ldots, p$ són una mostra per a que la xarxa "aprengui" quina resposta ha de donar en funció d'una determinada entrada.

En funció del dit abans, una bona forma d'ajustar els w serà plantejar un problema de mínims quadrats no lineals, on es minimitzi la distància entre $F(x_{e_i}; w)$ i x_{s_i}. El problema que plantejaríem seria:

$$\min_w \sum_{i=1}^{p} ||F(x_{e_i}; w) - x_{s_i}||^2 \tag{3.2}$$

En el cas més simple, on la sortida de la xarxa té un únic valor, tenim que $n_s = 1$ (aquesta és la situació que considerarem a la pràctica), i el problema a solucionar en aquest cas pot ser rescrit com:

$$\min_w \sum_{i=1}^{p} (F(x_{e_i}; w) - x_{s_i})^2 \tag{3.3}$$

3.3 Camps d'aplicació de les xarxes neuronals.

Les xarxes neuronals són usades en diferents camps. Un d'ells és el de la previsió. En aquest cas, les dades d'entrada podrien correspondre, a n dades d'una determinada sèrie temporal (com, per exemple, les hores de sol mensuals d'un país, temperatura d'un determinat compost a cada hora, nombre de neixements anuals d'una regió, consum diari d'energia elèctrica —en Kwh— d'una determinada ciutat, etc.), i el resultat de la xarxa neuronal seria una estimació sobre la dada corresponent al següent interval.

Un altre dels camps d'aplicació és del reconeixement de formes i patrons. Per exemple, considerem que podem classificar una sèrie d'objectes (o situacions) en funció de les seves característiques, i disposem d'un determinat nombre de classes. Llavors, donat un objecte, el que faríem és mesurar les seves característiques, i aquestes mesures serien el vector d'entrada a la xarxa neuronal. La sortida de la xarxa hauria de ser un valor que ens digués a quina classe pertany l'objecte, en funció de les dades d'entrada. Per posar un exemple més concret, podem pensar que les mesures corresponen a certs valors de proves clíniques fetes a un pacient, i la sortida de la xarxa (alimentada amb les dades anteriors) ens indicaria quin tipus de malaltia té el pacient en funció de les dades (hem "classificat" el pacient, doncs). Un dels avantatges (o desavantatges, segons com es miri) de les xarxes neuronals, quan són usades per classificar, és que no requereixen d'un coneixement previ sobre les dades mesurades, o les classes existents. És a dir, al cas de la classificació d'un pacient segons les dades clíniques, això vol dir que la xarxa neuronal no sap si la primera mesura correspon a la pressió sanguínia del pacient, o a la seva freqüència cardíaca, per exemple. Això fa que sigui còmode usar xarxes neuronals: només cal donar-li unes dades per fer l'aprenentatge, sense pensar en què volen dir aquestes dades; la xarxa no requereix d'aquesta informació addicional. Per altra banda, té l'inconvenient de que es difícil poder explicar la classificació que ha fet en funció de les dades entrades, el qual seria molt útil en alguns casos (com, per exemple, al cas anterior de les malalties, on podríem adonar-nos de certes patologies). En aquesta pràctica, tal i com veurem tot seguit, treballarem amb un cas molt simple de reconeixement de patrons.

3.4 Problema considerat a la pràctica.

Com abans s'ha indicat, a la pràctica considerarem un cas molt simple de reconeixement de patrons. El vector x_e de dades d'entrada tindrà quatre components $x_i,\ i = 1, \ldots, 4$. La sortida serà un únic valor (és a dir, $n_s = 1$), que podem anomenar y (és a dir, $x_s = y \in \mathbb{R}$. Normalment, quan s'usa una xarxa neuronal, no es coneix la relació entre les dades d'entrada i la de sortida (si es conegués, no ens caldria usar la xarxa). En aquest cas, però, si que coneixerem la relació entre l'entrada i la sortida, permetent, doncs, comprovar a posteriori si la xarxa ha fet un bon ajust dels pesos w, tot provant el seu funcionament amb vectors d'entrada qualsevols. Aquesta

relació entre l'entrada i la sortida serà:

$$y = \begin{cases} 1 & \text{si } \sum_{i=1}^{4} x_i > 2 \\ 0 & \text{si } \sum_{i=1}^{4} x_i \leq 2 \end{cases} \tag{3.4}$$

És a dir, si els quatre valors d'entrada sumen més de 2 la sortida serà un 1, i serà un 0 altrament. Cal indicar que tots els valors $x_i,\ i = 1, \ldots, 4$ d'entrada a la xarxa han d'estar entre 0 i 1. Aquest és un requeriment de caràcter pràctic de les xarxes neuronals, per poder garantir el seu bon funcionament.

Per poder realitzar l'"aprenentatge" de la xarxa neuronal, tal i com abans s'ha comentat, cal disposar d'un total de p vectors d'entrada amb la seva corresponent sortida. Per tal de generar aquests valors teniu a la vostra disposició un generador automàtic de dades segons la relació (3.4) . Aquest es troba al Cluster format per les màquines Alpha chooyu.fib.upc.es i meiga.fib.upc.es de la F.I.B. , al qual tots vosaltres teniu accés. Per obtenir les dades del vostre problema cal que, des del vostre directori, executeu primer l'ordre:

```
genxndat := $DIR$EIO:[ONLC]genxndat
```

i tot seguit ja podeu generar fitxers de dades amb la comanda

```
genxndat fitxer p llavor
```

El primer dels tres paràmetres, `fitxer`, serà el nom del fitxer (del vostre directori) on les dades seran escrites. El segon paràmetre, `p`, representa el nombre de dades d'entrada i sortida que seran escrites al fitxer. En teoria, com major sigui p, més bo serà l'aprenentage. Podeu usar un valor de $p = 50$, el qual no generarà un conjunt de dades molt gran, però sí suficient per garantir un aprenentatge acceptable. El darrer paràmetre és una llavor (entera) per generar les dades de forma aleatòria. Podeu introduir, per exemple, el número del vostre DNI. Així, si feu `genxndat dades.dat 6 654321`, obtindríeu un fitxer `dades.dat` similar a:

```
0.222 0.828 0.061 0.505   0.0
0.097 0.676 0.121 0.742   0.0
0.859 0.735 0.848 0.530   1.0
0.565 0.150 0.625 0.139   0.0
0.067 0.428 0.329 0.808   0.0
0.832 0.776 0.538 0.210   1.0
```

Cada fila correspon a unes dades d'entrada/sortida determinades (tenim $p = 6$). Dins de cada fila, les quatre primeres columnes corresponen als valors d'entrada $x_i,\ i = 1, \ldots, 4$, i la darrera columna indica el valor de sortida y. Es pot observar com les dades preserven la relació indicada a (3.4) .

La xarxa que considerarem serà com la de la Figura 3.1: una xarxa de tres capes i set nodes (quatre a la primera capa, dos a la segona i un node a la darrera capa). El problema de minimització que plantejarem serà el presentat a (3.3) , usant l'expressió de la xarxa neuronal indicada a (3.1) . Les variables del problema de mínims quadrats seran els pesos w. Observant la Figura 3.1, podem veure com el nombre de variables a optimitzar és de 10. El problema de minimització sense restriccions resultant haurà de ser solucionat amb el paquet AMPL.

Un cop realitzat l'"aprenentatge" de la xarxa neuronal amb l'ajut del paquet AMPL, convé que verifiqueu el bon funcionament de la mateixa introudint-li vectors de dades aleatoris, i comprovant que el resultat proporcionat per la xarxa està d'acord amb la regla (3.4) . Podeu

veure que, si hi ha alguns vectors d'entrada on la solució proporcionada no és correcta, aquest corresponen a casos on $\sum_{i=1}^{4} x_i \approx 2$.

3.5 Detalls importants a tenir en compte!.

Hi ha dues coses que cal tenir presents en intentar solucionar aquest problema amb el paquet AMPL. La primera d'elles és que els pesos w no tenen cap restricció sobre el seu signe (tant poden ser negatius com positius), i per tant són variables lliures. És convenient indicar-li a AMPL aquesta circumstància, per evitar que, per defecte, les consideri del tipus $w \geq 0$.

El segon punt a tenir en compte fa referència al punt inicial d'iteració. AMPL (com la majoria de paquets d'optimització) realitza un procés iteratiu, i a cada iteració troba un nou punt w^k. El punt inicial w^0, si no s'indica el contrari, és inicialitzat pel paquet a un valor determinat. El problema plantejat en aquesta pràctica (que ve donat per l'expressió (3.3)), és força no lineal, i té molts mínims locals (això és degut a la funció sigmoidal $f(x)$ usada). Si deixem que AMPL inicialitzi el punt w^0, fàcilment podem anar a parar a un mínim local on l'"aprenentatge" de la xarxa sigui nul. Per evitar aquest inconvenient, sovint s'acostuma a inicialitzar el punt inicial amb valors aleatoris per evitar caure sempre dins el mateix mínim local poc satisfactori. Al nostre cas concret, podeu mirar d'inicialitzar el punt inicial de forma que $w^0_{i,j} = 1 \ \forall_{i,j}$, excepte la component $w_{6,7}$, la qual serà inicialitzada com $w^0_{6,7} = -1$. Si aquest punt inicial no proporcionés un "bon" mínim local, mireu d'introduir d'altres de forma aleatòria. Podeu observar si el mínim local obtingut és un "bon mínim" amb el valor final de la funció objectiu: aquest ha de ser proper a 0 si l'"aprenentatge" ha estat satisfactori (un valor de 0 significa que hi ha pocs errors a l'ajust per mínims quadrats). Per indicar el punt inicial d'iteració a AMPL podem usar les ordres

```
INIT:
   variables a inicialitzar
ENDINIT
```

3.6 Presentació de la pràctica.

L'informe que haureu de presentar ha de tenir els següents apartats:

- Un llistat del fitxer amb la modelització del problema en el llenguatge de AMPL.
- Un llistat del resultat que AMPL us ha proporcionat, indicant el valor de funció objectiu i pesos w que heu obtingut.
- Els resultats que heu obtingut avaluant la xarxa amb uns quants vectors d'entrada aleatoris, comprovant el bon (o mal) funcionament de la xarxa per classificar les entrades.
- Qualsevol observació o comentari que considereu escaient, així com qualsevol problema que us hagi aparegut durant la realització de la pràctica.

4 Ajust per parts d'una corba cota-volum.

4.1 Presentació del problema

El problema plantejat a aquesta pràctica consisteix en la realització de l'ajust per parts de la corba cota-volum de l'embassament d'una central de producció d'energia hidroelèctrica. S'enten per corba cota-volum la corba que ajusta un conjunt de n mesures (v_i, c_i). Aquestes mesures ens indiquen quína és l'alçada c_i (*cota*) a la que es troba la superfície de l'aigua quan l'embassament conté un cert volum v_i. L'objectiu és obtenir la funció que millor ajusta la corba cota-volum, definint aquesta funció a partir de la unió de tres polinomis de tercer grau. Els coeficients d'aquests polinomis hauran de garantir que la funció resultant sigui continua, diferenciable i monòtona creixent. Es veurà que aquest ajust dona lloc a un problema d'optimització no lineal amb constriccions lineals i no lineals.

Els coeficients obtinguts en aquest ajust són necessaris en l'optimització de l'ús d'aigues emmagatzemades a les conques hidrogràfiques dels rius, i són ampliament emprats en problemes de gestió de la producció i distribució d'energia elèctrica, planificació de la distribució d'aigua per a rec, subministrament d'aigua potable, etc.

4.2 Formulació del problema.

4.2.1 Funció objectiu i variables

L'ajust *per parts* d'una corba a un conjunt de punts consisteix en fer una divisió del conjunt total de punts en subconjunts de punts, agrupats segons les seves abscisses, i ajustar una corba diferent per cada subconjunt de punts. La forma especial de les corbes cota-volum fa adequada l'aplicació d'un ajust d'aquestes característiques.

Es proposa fer un ajust per parts d'una corba cota-volum de tercer grau dividint l'eix d'abscisses en tres intervals. Consideri's que es disposa d'un conjunt de Nm parells de mesures cota-volum (v_i, c_i) i que els dos punts d'unió dels tres intervals coincideixen amb els valors v_i de dos dels Nm punts mesurats. Anomenarem i_1 i i_2 als índexos d'aquests dos valors. Es pretén ajustar un polinomi de tercer grau a cadascun dels tres conjunts de punts:

$$\begin{cases} (v_i, c_i) & i \leq i_1 \\ (v_i, c_i) & i_1 < i \leq i_2 \\ (v_i, c_i) & i_2 < i \end{cases} \tag{4.1}$$

A cada valor de volum mesurat v_i li correspon un valor estimat de cota c_i que ve donat per

l'expressió:

$$c(v,x) = \begin{cases} c_1(v,x) = c_{0_1} + c_{1_1}v + c_{2_1}v^2 + c_{3_1}v^3 & \text{si} \quad v \leq v_{i_1} \\ c_2(v,x) = c_{0_2} + c_{1_2}v + c_{2_2}v^2 + c_{3_2}v^3 & \text{si } v_{i_1} < v \leq v_{i_2} \\ c_3(v,x) = c_{0_3} + c_{1_3}v + c_{2_3}v^2 + c_{3_3}v^3 & \text{si } v_{i_2} < v \end{cases} \tag{4.2}$$

El vector de variables $x \in \Re^{12}$ estarà ara format pels dotze coeficients de les tres corbes:

$$x = [\, c_{0_1} \quad c_{1_1} \quad \ldots \quad c_{3_3} \,]' \in \Re^{12} \tag{4.3}$$

El vector $r \in \Re^{Nm}$ de residus entre valors de cota mesurats c_i i estimats segons la funció (4.2) serà una funció $r : \Re^{12} \to \Re^{Nm}$ que vindrà donada per:

$$r_i(x) = c_i - c(v_i, x) \quad ; \quad i = 1, \ldots, Nm \tag{4.4}$$

Prenent com a criteri de millor ajust aquell que fa mínima la norma al quadrat del vector de residus $r(x)$ es pot plantejar la funció objectiu d'un problema de mínims quadrats:

$$\min_{x \in \Re^{12}} \frac{1}{2}\|r(x)\|_2^2 \tag{4.5}$$

4.2.2 Constriccions.

L'ajust de corbes cota-volum s'acostuma a fer sota la imposició de dos tipus de constriccions: constriccions de *distància màxima*, constriccions de *pendent mínima* i constriccions de *continuïtat de* $c(v,\cdot)$ *i* $c'(v,\cdot)$.

Constriccions de distància màxima

Imposen un valor màxim ϵ a la discrepància entre el valor mesurat i ajustat de les cotes, és a dir, són restriccions no lineals del tipus:

$$\left(c_i - c(v_i, x)\right)^2 \leq \epsilon^2 \quad ; \quad i = 1, \ldots, Nm \tag{4.6}$$

Constriccions de pendent mínima.

Per a un valor donat dels coeficients x, imposen una fita inferior δ al valor de la derivada de la funció $c(v,x)$ a cada punt mesurat v_i, donant lloc a les constriccions lineals:

$$\frac{\mathrm{d}}{\mathrm{d}v} c(v,x)\Big|_{v_i} \geq \delta \quad ; \quad i = 1, \ldots, Nm \tag{4.7}$$

Constriccions de continuïtat de $c(v,\cdot)$ *i* $c'(v,\cdot)$

La funció cota-volum $c(v)$ forma part de la funció objectiu de gran part dels problemes de gestió de conques hidrogràfiques. Per aquesta raó convé que la funció $c(v,x)$ i la seva primera derivada respecte del volum d'aigua siguin contínues. Aquestes condicions es satisfan implícitament per a qualsevol punt de l'eix d'abscisses a excepció dels punts de contacte v_{i_1} i v_{i_2}. En aquests dos punts hem d'imposar explícitament les condicions de continuïtat mitjançant les següents quatre constriccions lineals:

$$c_1(v_{i_1}, x) \;=\; c_2(v_{i_1}, x) \tag{4.8}$$

$$\frac{\mathrm{d}}{\mathrm{d}v}c_1(v,x)\Big|_{v_{i_1}} = \frac{\mathrm{d}}{\mathrm{d}v}c_2(v,x)\Big|_{v_{i_1}} \tag{4.9}$$

$$c_2(v_{i_2},x) = c_3(v_{i_2},x) \tag{4.10}$$

$$\frac{\mathrm{d}}{\mathrm{d}v}c_2(v,x)\Big|_{v_{i_2}} = \frac{\mathrm{d}}{\mathrm{d}v}c_3(v,x)\Big|_{v_{i_2}} \tag{4.11}$$

El gràfic de la figura Figura 4.1 correspon un hipotètic ajust cota-volum per parts.

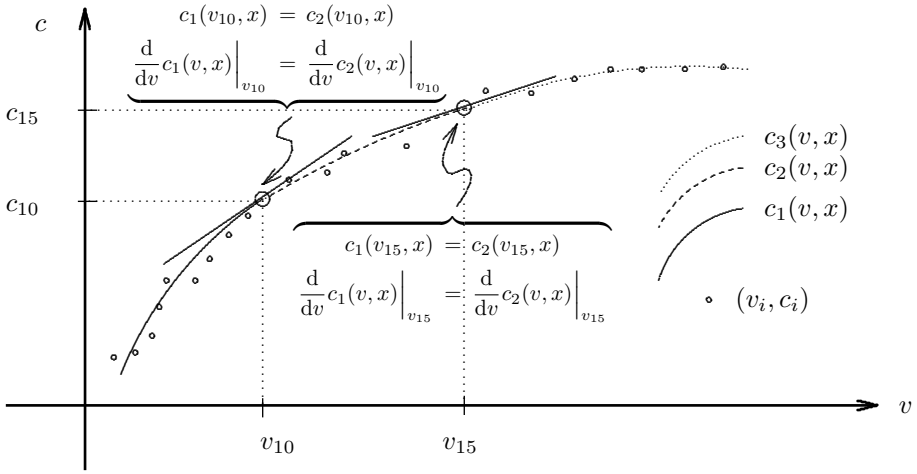

Figura 4.1 : Representació d'una corba cota-volum per parts. Els punts de contacte són $i_1 = 10$ i $i_2 = 15$ i el nombre de mesures és $Nm = 22$.

4.3 Formulació final.

El problema a resoldre és la minimització de la funció objectiu (4.5) subjecte a les constriccions (4.6) , (4.7) , (4.8) , (4.9) , (4.10) i (4.11) :

$$\min_{x\in\Re^{12}} \quad \frac{1}{2}\|r(x)\|_2^2 \tag{4.5}$$

$$\text{subj. a.:} \quad (c_i - c(v_i,x))^2 \le \epsilon^2 \quad ; \quad i = 1,\ldots,Nm \tag{4.6}$$

$$\frac{\mathrm{d}}{\mathrm{d}v}c(v,x)\Big|_{v_i} \ge \delta \quad ; \quad i = 1,\ldots,Nm \tag{4.7}$$

$$c_1(v_{i_1},x) = c_2(v_{i_1},x) \tag{4.8}$$

$$\frac{\mathrm{d}}{\mathrm{d}v}c_1(v,x)\Big|_{v_{i_1}} = \frac{\mathrm{d}}{\mathrm{d}v}c_2(v,x)\Big|_{v_{i_1}} \tag{4.9}$$

$$c_2(v_{i_2},x) = c_3(v_{i_2},x) \tag{4.10}$$

$$\frac{\mathrm{d}}{\mathrm{d}v}c_2(v,x)\Big|_{v_{i_2}} = \frac{\mathrm{d}}{\mathrm{d}v}c_3(v,x)\Big|_{v_{i_2}} \tag{4.11}$$

on c_i, v_i, i_1, i_2, ϵ i δ són paràmetres del problema.

4.4 Dades necessàries per a l'execució del problema.

Les dades associades a cada alumne estan contingudes a un fitxer anomenat `cvpp`*num*`.dat` similar al que mostra la figura Figura 4.2, on *num* correspon al número de identificació de l'alumne.

```
================================
PROBLEMA    :  num
NRE. MESURES:  15
  VOLUM (Hm3)      COTA (m)
-------------- --------------
0.51074968E-01 0.11300567E+04
0.34134443E+00 0.11320585E+04
0.12233030E+01 0.11353524E+04
0.14408906E+01 0.11361249E+04
0.20076070E+01 0.11380442E+04
0.26921036E+01 0.11402516E+04
0.37018112E+01 0.11433363E+04
0.41925184E+01 0.11447713E+04
0.48169097E+01 0.11465457E+04
0.54923961E+01 0.11484040E+04
0.64047942E+01 0.11508210E+04
0.73221260E+01 0.11531631E+04
0.83627568E+01 0.11557261E+04
0.85774402E+01 0.11562424E+04
0.95835685E+01 0.11586109E+04
================================
```

Figura 4.2 : Exemple de fitxer de dades `cvpp`*num*`.dat`

Les mesures cota-volum que contenen aquest fitxers han estat generades a partir de mesures reals de tres embassaments diferents. Els volums es donen en hectòmetres cúbics i la cota en metres.

Per definir completament el problema a minimitzar cal especificar, a part del conjunt de mesures (v_i, c_i), els valors dels paràmetres δ, ϵ, i_1 i i_2. El valor mínim de la derivada respecte del volum es considerarà nul ($\delta = 0$). La resta de paràmetres els haurà de fixar l'alumne. No cal dir que interessa tenir un valor de ϵ tant petit com sigui possible, essent $\epsilon = 1m$ un valor raonable. Els valors de i_1 i i_2 s'hauran de seleccionar a partir de la inspecció de la representació gràfica dels punts (v_i, c_i).

4.5 Informe de la pràctica.

L'informe d'aquesta pràctica ha de contenir els següents apartats:

1.- Llistat dels fitxers AMPL creats.

2.- El desenvolupament detallat de les constriccions (4.7) , (4.9) i (4.9) .

3.- La sortida de AMPL mostrant la solució obtinguda, així com la representació gràfica dels punts (v_i, c_i) i de la funció cota-volum trobada.

4.- Introduïnts els mínims canvis necessàris, feu dues noves execucions del problema, la primera amb una funció cota-volum de segon ordre (quadràtica) i la segona amb una funció cota-volum de primer ordre (lineal). Indiqueu:

4.1.- El procediment seguit per a aconseguir formular les funcions cota-volum quadràtica i lineal.

4.2.- Els llistats de la informació a l'òptim de les dues execucions.

4.3.- Una taula amb els valors a l'òptim de la funció objectiu i de la màxima distància punt-corba per a les tres funcions cota-volum (de tercer grau, quadràtica i lineal).

4.4.- Una anàlisi comparativa dels resultats obtinguts amb les tres execucions, prenent com a base les dades de la anterior taula.

5 Càlcul de la posició d'equilibri d'una cadena.

5.1 Presentació del problema.

Tenim una cadena formada per n baules lineals de dos tipus: *rígides* i *elàstiques*. Cadascuna té una llargària l_i, on l'índex i indica la posició de la baula dins la cadena, començant per l'esquerra.

N'hi ha tres baules elàstiques, amb índexs $i = e_1, e_2, e_3$. Es definirà el conjunt d'indexs de les baules elàstiques $\mathcal{E} = \{e_1, e_2, e_3\}$. Els coeficients d'elasticitat seran, respectivament k_1, k_2 i k_3. Per aquestes baules, la longitud l_i correspon a la l'estat en repós (*longitud en repós*). Quan la cadena arribi a la seva posició d'equilibri, les baules elàstiques s'hauran estirat fins a la seva *longitud en equilibri*), que anomenarem $\hat{l}_i$. Considerarem, a més, que aquestes baules es poden estirar com a màxim fins a una longitud $\bar{l}_i$. Totes les baules estan fabricades amb el mateix material, de densitat lineal $\lambda = (1/9.8)$Kgr/m. Així doncs, la massa de les baules és proporcional a la seva longitud en repós, amb constant de proporcionalitat λ.

La cadena es penja pels seus extrems, separats una distància horitzontal L_x y vertical L_y. L'objectiu de la pràctica és trobar la forma exacta de la cadena penjada de la forma indicada.

5.2 Formulació del Problema.

5.2.1 Variables.

Considerem que la baula i-éssima, degut a la seva orientació dins la cadena, augmenta la longitud d'aquesta en una quantitat x_i, horitzontalment, i y_i verticalment:

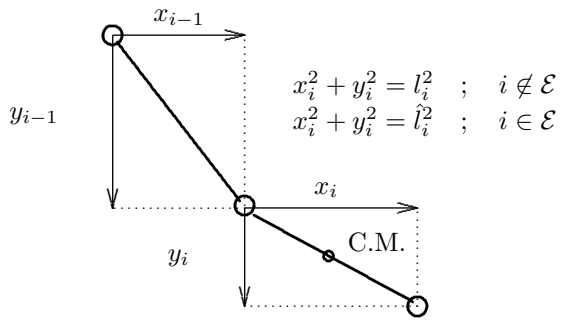

Figura 5.1 : Expansió de la longitud de la cadena

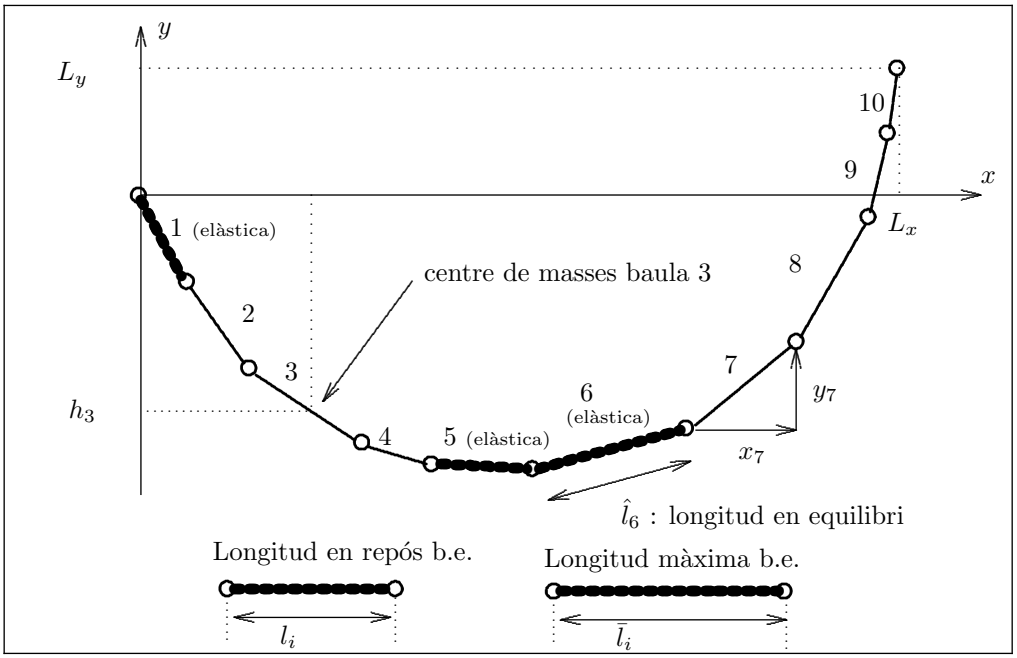

Figura 5.2 : Esquema d'una cadena amb deu baules en posició d'equilibri. En aquest cas $\mathcal{E} = \{1, 5, 6\}$.

Les x_i són sempre positives, però les y_i poden ser positives, si apunten cap a dalt, o negatives, si apunten cap a baix.

Coneixem les l_i de totes les baules rígides, raó per la qual només ens cal conèixer o x_i o y_i per a saber l'orientació de la baula dins la cadena. Agafarem com a variables del problema les y_i, $\forall i \in \mathcal{E}$. De les baules elàstiques no coneixem la seva longitud en equilibri, la qual cosa implica considerar com a variables del problema les x_i i y_i $\forall i \in \mathcal{E}$. Així doncs, les variables del nostre problema seràn:

$$y_i \quad ; \quad i = 1, \ldots, n \tag{5.1}$$

$$x_i \quad ; \quad \forall i \in \mathcal{E} \tag{5.2}$$

5.2.2 Funció Objectiu.

El sistema format per la cadena penjada es trobarà en equilibri quan la seva energia potencial total sigui mínima. L'energia potencial total serà la suma de la energia potencial gravitatória U_G i de l'energia potencial elàstica U_E:

$$U_T = U_G + U_E \tag{5.3}$$

5.2.2.1 Energia potencial gravitatória.

Si fixem com a punt de referència, amb energia potencial nul·la, l'altura del punt d'on es penja la primera baula, i la massa de cada baula concentrada en el seu punt mig (centre de

masses C.M.), l'energia potencial de la baula i-éssima és:

$$U_{G_i} = m_i g h_i \tag{5.4}$$

on h_i és la distància vertical que separa el centre de masses de la baula i-éssima del punt de referència (amb valor negatiu, si està per sota). $g = 9.8 m/s^2$ és l'acceleració de la gravetat prop de la superfície de la terra (que es pot considerar constant). Podem expressar les h_i en funció de les y_i a través de l'expressió:

$$h_i = y_1 + y_2 + \ldots + \frac{1}{2} y_i = \frac{1}{2} y_i + \sum_{j=1}^{i-1} y_j \tag{5.5}$$

aquesta igualtat permet expressar U_{G_i} en funció de les variables y_i. L'energia potencial gravitatoria total serà la suma de totes les U_{G_i}:

$$U_G = \sum_{i=1}^{n} U_{G_i} \tag{5.6}$$

5.2.2.2 Energia potencial elàstica.

La força feta per una molla com a resposta a una variació Δl (amb signe) de la seva longitud respecte de la seva posició en repós és:

$$F_E = -k \cdot \Delta l \tag{5.7}$$

Podem calcular l'energia potencial d'una molla sotmesa a un estirament Δl amb la fórmula:

$$U_E(\Delta l) = -\int_0^{\Delta l} -k \cdot x \cdot dx = k \int_0^{\Delta l} x \cdot dx = \frac{1}{2} \cdot k \cdot \left[x^2\right]_0^{\Delta l} = \frac{1}{2} \cdot k \cdot \Delta l^2 \tag{5.8}$$

Per a cada una de les baules elàstiques $\mathcal{E}$ tindrem:

$$U_{E_i} = \frac{1}{2} \cdot k \cdot \Delta l_i^2 \tag{5.9}$$

on Δli és l'increment de longitud respecte de la longitud en repós de la baula elàstica quan la cadena es troba en equilibri:

$$\Delta l_i = \hat{l}_i - l_i \quad ; \quad \forall i \in \mathcal{E} \tag{5.10}$$

Tant la longitud en repós l_i com el coeficient d'elasticidad k_i són dades de l'enunciat. $\hat{l}_i$ és la longitud de la baula i quan la cadena està en equilibri. Pot ser expressada en funció de les x_i i y_i:

$$\hat{l}_i = \sqrt{x_i^2 + y_i^2} \tag{5.11}$$

de forma que:

$$\Delta l_i = \sqrt{x_i^2 + y_i^2} - l_i \tag{5.12}$$

substituint (5.12) a (5.9) obtindrem l'expressió de l'energia potencial elàstica de la baula i amb $i \in \mathcal{E}$ en funció de les dades del problema i les variables. L'energia potencial elàstica total

serà, evidentment, la suma de U_{E_i} per $i \in \mathcal{E}$:

$$U_E = \sum_{i \in \mathcal{E}} U_{E_i} \tag{5.13}$$

5.2.3 Constriccions

5.2.3.1 Sumatori de les y_i.

La suma dels valors de les variables y_i ha de ser igual a la distància vertical entre els dos punts de suspensió (hem de tenir en compte el conveni de signes: negatiu cap a baix, positiu cap a dalt):

$$\sum_{i=1}^{n} y_i = L_y \tag{5.14}$$

5.2.3.2 Sumatori de les x_i:

La suma dels valors de les variables x_i ha de ser igual a la distància horitzontal entre els dos punts de suspensió:

$$\sum_{i=1}^{n} x_i = L_x \tag{5.15}$$

Les variables del nostre problema són les y_i, $i = 1, \ldots, n$ i les x_i, $i \in \mathcal{E}$. La constricció (5.15) es pot s'expressa en funció d'aquestes variables com a:

$$\sum_{i \in \mathcal{E}} x_i + \sum_{i \notin \mathcal{E}} \sqrt{l_i^2 - y_i^2} = L_x \tag{5.16}$$

5.2.3.3 Longitud màxima i mínima de les baules elàstiques.

Com a dade del problema s'impossa a cada baula elàstica una longitud màxima $\bar{l}_i$. Per altre banda, degut a com es penja la cadena, aquestes baules no es contrauran (per què?), la qual cosa vol dir que la seva longitud mínima serà l_i:

$$l_i \leq \sqrt{x_i^2 + y_i^2} \leq \bar{l}_i \quad ; \quad \forall i \in \mathcal{E} \tag{5.17}$$

5.2.4 Fites a les variables.

S'han d'imposar fites a les longituds en que la cadena és expandida, vertical i horitzontalment, per cada baula:

$$-l_i \leq y_i \leq l_i \quad ; \quad \forall i \notin \mathcal{E} \tag{5.18}$$

$$0 \leq x_i \quad ; \quad \forall i \in \mathcal{E} \tag{5.19}$$

S'ha de fer notar que les fites $-\bar{l}_i \leq y_i \leq \bar{l}_i$ i $x_i \leq \bar{l}_i$ per a les baules elàstiques $i \in \mathcal{E}$ ja estan incloses a les constriccions (5.17) .

5.2.5 Formulació final.

L'expressió final del problema d'optimització a resoldre és:

$$\min_{\substack{x_i, i \in \mathcal{E} \\ y_i, \forall i}} \quad U_G + U_E = \sum_{i=1}^{n} U_{G_i} + \sum_{i \in \mathcal{E}} U_{E_i} \qquad (5.6)\ ,(5.13)$$

Subj.a :

$$\sum_{i=1}^{n} y_i = L_y \qquad (5.14)$$

$$\sum_{i \in \mathcal{E}} x_i + \sum_{i \notin \mathcal{E}} \sqrt{l_i^2 - y_i^2} = L_x \qquad (5.16)$$

$$l_i \leq \sqrt{x_i^2 + y_i^2} \leq \bar{l}_i \quad ; \quad \forall i \in \mathcal{E} \qquad (5.17)$$

$$-l_i \leq y_i \leq l_i \quad ; \quad \forall i \notin \mathcal{E} \qquad (5.18)$$

$$0 \leq x_i \quad ; \quad \forall i \in \mathcal{E} \qquad (5.19)$$

on n, $\mathcal{E}$, L_x, L_y, l_i, $\bar{l}_i$, i els valors dels coeficients d'elasticitat k_i que intervenen a les expressions de U_{E_i} són dades conegudes.

5.3 Dades necessàries per a l'execució del problema.

Les dades associades a cada alumne es poden generar amb el programa *cadegprob.exe* deixat al directori `DIR$EIO:[ONLC]` del cluster VAX de la FIB. Aquest programa només necessita el número d'identificació de l'alumne *num*, i crearà, en el directori on s'executi, un fitxer anomenat `cadena`*num*`.dat` similar al que mostra la figura Figura 5.3. Si la baula més alta és l'última, la situació en que ens trobem correspon a la representada a la Figura 5.2. Si la baula més alta és la primera, llavors, prenent com a criteri arbitrari que la primera baula es penja sempre de l'origen de coordenades, el valor de L_y s'haurà de considerar negatiu.

Les dades del problemes estan expressades en unitats del Sistema Internacional: longituds en metres, masses en quilograms, forces en Newtons, etc. D'aquesta forma el valor de l'energia total del sistema calculada a la funció objectiu del nostre problema estarà expressada en Joules.

5.4 Informe de la práctica.

L'informe d'aquesta pràctica ha de contenir els següents apartats:

1.- Llistat dels fitxers AMPL creats.

2.- La sortida de AMPL mostrant la solució obtinguda.

3.- Un dibuix de la cadena, a escala, en la seva posició d'equilibri, destacant les baules elàstiques. Heu d'indicar, per cada baula, les seves x_i i y_i òptimes i, per a les elàstiques, la seva longitud a l'equilibri.

4.- Respongueu a les següents preguntes:

4.1.- Considereu que es vol disminuir el màxim posible l'energia potencial total de la cadena, i que l'única posibilitat és, o bé modificar el valor de L_x o bé modificar el valor de L_y

```
================================================================================
PROBLEMA : CALCUL DE LA POSICIO D'EQUILIBRI D'UNA CADENA
DADES    :  num
================================================================================
- DISTANCIA HORITZONTAL ENTRE ELS EXTREMS :  Lx = 20.0 m
- DISTANCIA   VERTICAL  ENTRE ELS EXTREMS :  Ly =  1.2 m
- BAULA MES ALTA ..................... :  PRIMERA
- NOMBRE DE BAULES ................... :  n = 11
- LONGITUD CADENA .................... :  38.0 m
- BAULES  ELASTIQUES :
       * POSICIO                        :  e1 =  4     e2 =  8     e3 =  9
       * COEFICIENTS D'ELASTICITAT (N/m) :  K1 =  6.8   K2 = 10.5   K3 = 13.6
       * LONGITUD MAXIMA           (m)   :  LM1=  9.6   LM2=  4.4   LM3=  2.9
- LONGITUD BAULES (m) :
l 1 = 3.2 ; l 2 = 1.6 ; l 3 = 5.4 ; l 4 = 4.9 ; l 5 = 1.9 ; l 6 = 5.6 ;
l 7 = 1.9 ; l 8 = 3.3 ; l 9 = 2.6 ; l10 = 3.9 ; l11 = 3.7 ; l
================================================================================
```

Figura 5.3 : Exemple de fitxer de dades `cadena`*num*`.dat`

en $\pm$1mm. Sense reoptimitzar el model, indiqueu, raonadament, quina seria la millor opció.

4.2.- Sense reoptimitzar el model, calculeu, aproximadament, quina variació provocaria en l'energia potencial total una modificació de +1mm en el valor de L_y. Comproveu la resposta reoptimitzant el model amb el valor de L_y modificat.

4.3.- A partir de la informació que proporciona AMPL, és possible saber si el punt solució obtingut és un mínim local del nostre problema? Per què? En cas de resposta negativa, indiqueu el procediment que caldria seguir per tal de comprovar si es tracta d'un mínim local.

6 Breu introducció al paquet Minos.

En aquest capítol descriurem el paquet Minos d'optimització, amb el qual s'haurà de resoldre un problema de petita dimensió. Val a dir que, per la petita mida d'aquest problema, altres paquets més "amicables" podrien ser usats. Tanmateix, el fet de que Minos sigui actualment un dels millors paquets comercials (per no dir el millor), fa que sigui convenient haver treballat amb ell i conèixer-lo mínimament.

6.1 Dades generals.

Minos és un sistema informàtic escrit en Fortran dissenyat per resoldre problemes d'optimització de grans dimensions (problemes lineals i no lineals, tant pel que fa a la funció objectiu com a les constriccions). El nom és un acrònim i significa **M**odular **I**n-core **N**onlinear **O**ptimization **S**ystem. Els seus autors són Bruce A. Murtagh i Michel A. Saunders (Systems Optimization Laboratory, Department of Operations Research, Stanford University, California).

6.2 Problema estandard.

El problema estandard amb que Minos treballa té l'expressió:

$$\begin{array}{ll} min. & F(x) + c^t x + d^t y \\ subj. & l_1 \leq f(x) + A_1 y \leq u_1 \\ & l_2 \leq A_2 x + A_3 y \leq u_2 \\ & l \leq \begin{pmatrix} x \\ y \end{pmatrix} \leq u \end{array} \tag{1}$$

on:

- *Vectors constants:*
  $c \in \mathbb{R}^{n_1}$; $d \in \mathbb{R}_2^n$
  $u_1, l_1 \in \mathbb{R}^{m_1}$; $u_2, l_2 \in \mathbb{R}_2^m$
  $l, u \in \mathbb{R}^{n_1+n_2}$
- *Matrius constants:*
  $A_1 \in (m_1 \times n_2)$
  $A_2 \in (m_2 \times n_1)$
  $A_3 \in (m_2 \times n_2)$
- *Variables i funcions:*
  $F(x)$: funció escalar de variable vectorial.

$f(x)$: funció vectorial de variable vectorial. $f(x) = \{f(x)_i\},\ i = 1 \dots m_1$.
$x \in \mathbb{R}^{n_1}$: variables no lineals.
$y \in \mathbb{R}^{n_2}$: variables lineals.
$l_1 \leq f(x) + A_1 y \leq u_1$: constriccions no lineals.
$l_2 \leq A_2 x + A_3 y \leq u_2$: constriccions lineals.

6.3 Mètode de treball de Minos.

Per a resoldre un problema amb constriccions d'igualtat no lineals Minos efectua una sèrie d'iteracions majors (MAJOR ITERATIONS). Dins de cada iteració major es resol un subproblema amb constriccions lineals (MINOR ITERATIONS). Aquest subproblema està format per les constriccions lineals i fites del problema original i per una linealització de les constriccions no lineals.

Aquest procés de linealització consisteix en substituir la funció vectorial $f(x)$ de (1) per una aproximació de primer ordre $\hat{f}(x)$ fent servir el jacobià de les constriccions no lineals en el punt x_k (denotarem el jacobià amb J_k):

$$\hat{f}(x, x_k) = f(x_k) + J_k(x - x_k) \iff \hat{f}_k(x) = f_k + J_k(x - x_k)$$

El subproblema resolt a cada iteració major k és:

$$\begin{array}{ll} min. & F(x) + c^t x + d^t y - \lambda_k^t (f - \hat{f}_k) + \frac{1}{2}\rho (f - \hat{f}_k)^t (f - \hat{f}_k) \\ subj. & \hat{f}_k + A_1 y = b_1 \\ & A_2 x + A_3 y = b_2 \\ & l \leq \begin{pmatrix} x \\ y \end{pmatrix} \leq u \end{array} \qquad (2)$$

on:

• La funció objectiu de (2) s'anomena *Lagrangià augmentat*.

• λ_k és una estimació al punt x_k dels multiplicadors de Lagrange de les constriccions no lineals.

• $\frac{1}{2}\rho(f - \hat{f}_k)^t(f - \hat{f}_k)$ és el que es coneix com a *funció de penalització quadràtica*, amb paràmetre de penalització ρ.

6.4 Rutines i fitxers d'usuari.

La informació que Minos necessita per a resoldre el problema se li ha de subministrar mitjançant dues rutines i dos fitxers amb informació (dos fitxers que es poden convertir en un que contingui la informació dels dos anteriors):

rutina FUNOBJ
rutina FUNCON } juntes en un sol fitxer PROVA.FOR o PROVA.C

fitxer SPECS
fitxer MPS } junts, i en aquest ordre, en un sol fitxer PROVA.DAT

Trobareu unes plantilles d'aquests fitxers al directori:

DIR$EIO:[ONLC]

Els fitxers s'anomenen TEMPLATE.DAT i TEMPLATE.FC. Al fitxer TEMPLATE.FC trobareu un possible main pel vostre programa, així com la capçalera de les rutines *FUNOBJ* i *FUNCON* amb la declaració de variables. El main és recomanable que estigui escrit en Fortran (podeu mantenir el que hi ha o canviar-lo). Això ens ajudarà a l'hora de gestionar els fitxers d'entrada i sortida de dades, donat que Minos està escrit en Fortran. Les funcions *FUNOBJ* i *FUNCON* les podeu escriure en c o en Fortran. Al fitxer TEMPLATE.FC hi ha la declaració de variables pels dos llenguatges.

Els paràmetres particulars de cada rutina són:

FUNOBJ:

Funció: Codifica la funció objectiu i el seu gradient.

Paràmetres:

Entrada:

N: nombre de variables no lineals

X: vector de dimensió N que conté el valor de les variables a cada passa. L'ordre en què estan emmagatzemades les variables ha de coincidir amb el declarat a l'apartat COLUMNS del fitxer MPS.

Sortida:

F: valor de la funció objectiu corresponent a la X actual.

G: vector de dimensió N per emmagatzemar el gradient de F (és a dir $G(i) = \frac{\partial F}{\partial x_i}$).

FUNCON:

Funció: Codifica les constriccions no lineals i els seus gradients (jacobià).

Paràmetres:

Entrada:

N: nombre de variables no lineals.

M: nombre de constriccions no lineals (només s'usa si el jacobià es codifica de forma densa).

NJAC: nombre d'elements no nuls del jacobià (només s'usa si el jacobià es codifica de forma esparsa).

X: vector de dimensió N que conté el valor de les variables no lineals a cada iteració.

Sortida:

F: vector de dimensió M la component i del qual correspon al valor de la constricció no lineal número i pels valors de les variables no lineals de la iteració actual (emmagatzemades a X).

G: si el jacobià actual s'emmagatzema dens, és la matriu $(M \times N)$ que correspon al jacobià de F. Si el jacobià s'emmagatzema espars, és el vector de dimensió *NJAC* que conté els elements no nuls del jacobià en el mateix ordre que l'indicat a l'apartat COLUMNS del fitxer MPS.

6.5 Lectura i escriptura de dades a les rutines *FUNOBJ* i *FUNCON.*

<1>

Minos té declarada un zona COMMON anomenada M1FILE amb les variables IREAD, IPRINT, ISUMM. Ens interessa el contingut de les dues primeres:

IREAD: unitat lògica de lectura assignada al fitxer d'entrada de dades (per exemple, PROVA.DAT).
IPRINT: unitat lògica d'escriptura assignada al fitxer de sortida de resultats (per exemple, PROVA.LIS).

És a dir, si des de les rutines *FUNOBJ* i *FUNCON* accediu a la zona COMMON M1FILE afegint al vostre codi:

COMMON /M1FILE/ IREAD,IPRINT,ISUMM

podreu llegir dades afegides del fitxer d'entrada .MINOS i afegir informació al fitxer de sortida .LIS. Això darrer pot ser útil, per exemple, per escriure a la darrera iteració el valor de les variables a l'òptim amb totes les xifres significatives que vulgueu, cosa que MINOS no fa.

6.6 Apartat SPECS.

L'apartat SPECS (o fitxer si està separat de l'altre apartat anomenat MPS) defineix els diferents paràmetres sobre el funcionament del paquet Minos i sobre les característiques del problema. El format d'entrada de dades és lliure, i l'aspecte general de l'apartat SPECS és:

BEGIN
⋮
PARAULA_CLAU_1 [PARAULA_CLAU_2] [VALOR_NUMÈRIC]
⋮
END

De la primera paraula clau només són significatius els tres primers caràcters; de la segona (si n'hi ha segona) només són significatius els 4 primers caràcters. Vegem a continuació una part del paràmetes que poden ser indicat a l'apartat SPECS.

Paràmetres que depenen de les dades del problema:

COLUMNS k: amb k indiquem el nombre sobreestimat de columnes de la matriu de constriccions (nombre sobreestimat de variables).
ROWS k: amb k denotem el nombre sobreestimat de files de la matriu de constriccions (nombre sobreestimat de constriccions lineals i no lineals).
ELEMENTS k: k és el nombree sobreestimat d'elements no nuls a les matrius A_1, A_2, A_3 i al jacobià.
NONLINEAR CONSTR. k : on k és el nombre de constriccions no lineals.
NONLINEAR VARIABLES k : on k és el nombre de variables no lineals.

<1> El dit en aquest apartat, pel que fa a les zones COMMON, només té sentit si les rutines *FUNOBJ* i *FUNCON* estan programades en Fortran.

Paràmetres que no depenen de les dades del problema:

JACOBIAN SPARSE: indica que el jacobià s'emmagatzemarà de forma esparsa, és a dir, només es guardaran els elements no nuls del jacobià. Es diu que una matriu és esparsa si té una gran quantitat d'elements nuls. Si el jacobià és espars resulta convenient triar aquesta opció. Si es volgués emmagatzemar dens no caldria especificar res, donat que aquesta és l'opció per defecte.

DERIVATIVE LEVEL k: controla el càlcul del gradient de la funció objectiu i del jacobià de les constriccions:

- k=1: Minos calcula el jacobià i el gradient s'ha de codificar a la *FUNOBJ*.
- k=2: Minos calcula el gradient i el jacobià s'ha de codificar a la *FUNCON*.
- k=3: S'ha de codificar gradient i jacobià. Aquesta és l'opció amb que haureu de resoldre el problema.

VERIFY: Provoca la comprovació per diferències finites de tots els elements del gradient i del jacobià calculats per les rutines *FUNOBJ* i *FUNCON*.

LOG FREQUENCY k: controla la freqüència amb la que s'escriu informació al fitxer de sortida. S'imprimirà una línia d'informació per cada k iteracions menors.

6.7 Apartat MPS.

Especifica els noms de les constriccions i variables, indica com intervé cada variable dins cada constricció, i defineix els termes independents de les constriccions i els límits de les variables. Aquest format no és propi de Minos; és un format estandard d'especificació de problemes usat per diversos paquets d'optimització.

El format d'entrada no és lliure i cada paraula clau ha d'estar entre unes columnes determinades al fitxer. L'aspecte general de l'apartat MPS és:

```
        1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901234567890

NAME          nom␣problema

ROWS
 ww constricció

COLUMNS
    variable  constr␣1  coeficient␣1   constr␣2  coeficient␣2

RHS
    nom␣indp  constr␣␣  terme␣independent

RANGES
    nom␣rang  constr␣␣  valor␣del␣rang

BOUNDS
 zz nom␣boun  variable  valor␣del␣límit

ENDDATA
```

Al MPS anterior en majúscula apareixen les paraules clau i en minúscula les dades que varien d'un problema a l'altre (i subratllats hi ha el nombre de caràcters màxim que pot ocupar cada nom). Els camps que s'han marcat com `ww, nom␣indp, nom␣rang, zz` i `nom␣boun` indiquen el tipus de constricció (`ww`), el nom donat al conjunt de termes independents (`nom␣indp`) , el nom donat al conjunt de rangs (`nom␣rang`) , el tipus de límit de la variable (`zz`) i el nom donat al conjunt de límits (`nom␣boun`). Descriurem a continuació cadascuna de les seccions del MPS.

Secció NAME:

S'usa per donar un nom al problema que s'està codificant. El format que hem d'utilitzar és:

```
          1         2
12345678901234567890123
NAME.........problema
```

on `problema` és el nom que donem al problema i els punts indiquen espais en blanc.

Secció ROWS:

Declara el nom i tipus de les constriccions (i de la part lineal de la funció objectiu si n'hi ha). S'han de posar **primer les no lineals** i a continuació les lineals. El format que hem d'utilitzar és:

```
          1
1234567890123
ROWS
.ww.constric
```

on els punts indiquen espais en blanc, `constric` és el nom de la constricció i `ww` ens indica el tipus de constricció que pot ser:

$$\texttt{ww} = \begin{cases} E & : \; = \\ G & : \; \geq \\ L & : \; \leq \\ N & : \; \text{funció objectiu o constricció lliure} \end{cases}$$

Secció COLUMNS:

Aquesta secció del MPS serveix per:

1. Declarar els noms de les variables.
2. Donar valors als coeficients amb els que intervenen les variables dins de cada constricció lineal.
3. Si el jacobià s'emmagatzema espars, indica quina és la posició dels elements no nuls del jacobià. En aquest cas, el valor numèric indicat no té importància (pot ser zero, per exemple).

Si hi ha variables lineals i no lineals, **primer s'han de declarar les no lineals**. Fins que no s'han introduït tots els coeficients que afecten a una variable no es pot començar amb una nova variable. L'ordre en que es declarin les variables no lineals ha de coincidir amb l'ordre usat

al vector X de les funcions *FUNOBJ* i *FUNCON*. Així , començant amb $X(1)$ i continuant fins a $X(N)$ haurem de declarar:

1. el nom triat per a la variable $X(i)$.
2. si el jacobià s'emmagatzema en forma esparsa, els elements no nuls de la columna i-èssima del jacobià, indicant el nom de la constricció no lineal corresponent i un valor numèric fictici.
3. els coeficients no nuls de la columna i-èssima de la matriu de constriccions lineals, indicant el nom de la constricció lineal i el valor numèric del coeficient.

El format d'escriptura d'aquesta secció és:

```
                1         2         3         4         5         6
123456789012345678901234567890123456789012345678901234567890123456789012345
COLUMNS

....variable..constr␣1..coeficient␣1...constr␣2..coeficient␣2
```

on els punts indiquen espais en blanc, `variable` és el nom de la variable que estem tractant, `constr␣1` i `constr␣2` són noms de constriccions on intervé la variable en qüestió, i `coeficient␣1` i `coeficient␣2` indiquen els valors amb que la variable que tractem afecta a cada constricció (si la constricció és no lineal es pot posar qualsevol valor). Cal tenir en compte que en aquest apartat han d'aparèixer els noms de totes les variables, lineals i no lineals, fins i tot si no tenen cap coeficient associat.

Secció RHS:

Declara els termes independents de totes les constriccions (lineals i no lineals). Poden anar en qualsevol ordre. El format d'escriptura és:

```
                1         2         3
12345678901234567890123456789012345678
RHS

....nom␣indp..constr␣i..terme␣indept
```

on els punts indiquen espais en blanc, `nom␣indp` indica el nom que donem al conjunt de termes RHS, `constr␣i` indica el nom de la constricció que tractem i `terme␣indept` representa el valor del terme independent. El nom `nom␣indp` és arbitrari però ha de ser el mateix per a totes les components d'un mateix vector de termes independents, i només serveix per a donar nom a aquest vector.

Secció RANGES:

S'usa per definir constriccions del tipus $l \leq f_i \leq u$. El format d'escriptura d'aquesta secció és:

```
                1         2         3
12345678901234567890123456789012345678
RANGES

....nom␣rang..constr␣i..terme␣ranges
```

on els punts indiquen espais en blanc, `nom␣rang` indica el nom que donem al conjunt de rangs,

`constr␣i` indica el nom de la constricció que tractem i `terme␣ranges` representa el valor del rang. El nom `nom␣rang` té la mateixa funció que el nom `nom␣indp`. Si a l'apartat RHS s'ha definit $F(x) \leq u$ i volem tenir $l \leq F(x) \leq u$ el valor del rang ha der ser $rang = u - l$.

Secció BOUNDS:

Declara les fites de les variables. El seu format és:

```
         1         2         3
12345678901234567890123456789012345678
BOUNDS
 .zz.nom␣boun..variable..terme␣bounds
```

on els punts indiquen espais en blanc, `nom␣boun` indica el nom que donem al conjunt de fites, `variable` indica el nom de la variable que tractem i `terme␣bounds` representa el valor de la fita. El nom `nom␣boun` té la mateixa funció que als dos apartats anteriors. El camp `zz` ens indica el tipus de fita i pot prendre els valors:

$$\texttt{zz} = \begin{cases} LO & : \leq \\ UP & : \geq \\ FR & : \text{variable lliure} \\ FX & : = \end{cases}$$

Per defecte es considera que totes les variables es troben afitades inferiorment per 0.

Existeix la possibilitat de fixar dins d'aquest apartat el punt inicial a partir del qual Minos començarà la cerca del punt inicial factible. Per defecte Minos inicialitza les variables a zero o a la fita més propera a zero, la qual cosa pot provocar problemes en certes constriccions no lineals. Per fixar el valor inicial d'una variable cal incloure a l'apartat BOUNDS la següent línia:

```
         1         2         3
12345678901234567890123456789012345678
 .FX.INITIAL␣..variable..valorinicial
```

6.8 Exemple de codificació d'un problema en format MPS.

Sigui el problema:

$$\begin{array}{lccccc} min. & x_1^2 & +x_2^2 & +x_3^2 & & \\ subj. & x_1^2 & +x_2^2 & & & \leq 12 \\ & & x_2^4 & & & \geq 4 \\ & x_1^4 & & +x_3^4 & & \geq 0 \\ & 2x_1 & +4x_2 & -x_3 & & = 10 \\ & x_1 \geq 0, & x_2 \geq 0, & 1 \geq x_3 \geq 0 & & \end{array}$$

El fitxer MPS associat a aquest problema amb emmagatzemament espars del jacobià seria:

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890
```

```
NAME          PROVA
ROWS
 L  CONSNOL1
 L  CONSNOL2
 G  CONSNOL3
 E  CONSLIN1
COLUMNS
    X1        CONSNOL1  0.0
    X1        CONSNOL3  0.0
    X1        CONSLIN1  2.0
    X2        CONSNOL1  0.0
    X2        CONSNOL2  0.0
    X2        CONSLIN1  4.0
    X3        CONSNOL3  0.0
    X3        CONSLIN1  -1.0
RHS
    TERMINDE  CONSNOL1  12.0
    TERMINDE  CONSNOL2  4.0
    TERMINDE  CONSNOL3  0.0
    TERMINDE  CONSLIN1  10.0
BOUNDS
*les dos linies seguents
*no caldrien
 LO LIMSIMP   X1        0.
 LO LIMSIMP   X2        0.
 UP LIMSIMP   X3        1.
ENDDATA
```

6.9 Muntatge i execucio.

Un cop s'ha fet el fitxer PROVA.DAT amb l'apartat SPECS i MPS, tenim codificades la *FUNOBJ* i *FUNCON* (bé en Fortran, bé en c) i un programa principal que faci la crida a Minos, cal, abans de res, compilar-ho tot (el programa principal i el fitxer on es troben la *FUNOBJ* i *FUNCON*). Si suposem que s'ha escrit tot en Fortran i es troba dins d'un fitxer anomenat (un cop hem compilat) PROVA.OBJ l'ordre que haurem d'executar per fer el muntatge serà:

LINK PROVA.OBJ,USER$LIB:[MINOS53.AXP]MINOS53/LIB

Si per contra hem escrit les dos rutines *FUNOBJ* i *FUNCON* en c i les tenim al fitxer, per exemple, PROVA.C, i el programa principal que crida a Minos (escrit en Fortran) es troba al fitxer anomenat, per exemple, MAIN.FOR, per fer el muntatge (un cop compilat tot) farem:

LINK MAIN.OBJ,PROVA.OBJ,USER$LIB:[MINOS53.AXP]MINOS53/LIB

Després d'executar el programa (si utilitzeu el programa principal que us suministrem) obtindreu un fitxer anomenat PROVA.LIS amb el resultat de l'execució, i un fitxer anomenat FOR009.DAT. Aquest darrer no té cap resultat interessant, i us heu de centrar en el .LIS. Aquest fitxer .LIS té diferents apartats amb paràmetres i estadístiques de l'execució. Fixeu-vos en l'apartat MPS FILE, on poden sortir missatges d'error i avisos relatius a la lectura del fitxer .MINOS, com ara:

XXXX WARNING - NO LINEAR OBJECTIVE SELECTED

Aquest missatge, en concret, us l'heu de trobar i us indica que esteu resolent un problema amb funció objectiu no lineal, com és el cas. Altres missatges d'error o warnings no us haurien de sortir. Observeu també l'apartat ITERATIONS. A la columna SINF,OBJECTIVE apareix la suma d'infactibilitats quan encara no s'ha arribat a un punt factible. A partir d'aquest moment conté el valor de la funció objectiu. A la columna ITN podeu veure les iteracions que necessita per trobar un punt factible i per trobar l'òptim.

Minos utilitza com a espai de treball el vector Z(NWCORE). El valor de la dimensió de Z(.), NWCORE, es troba declarada al programa principal. Per verificar si aquesta dimensió és suficient heu de comparar el valor del REASONABLE WORKSPACE LIMITS amb el de ACTUAL WORKSPACE LIMITS, que podreu trobar abans de l'apartat MPS FILE. Si el primer valor fos major que el segon, caldrà augmentar el valor de NWCORE.

Abans de donar un resultat per bo heu de comprovar que les subrutines *FUNOBJ* i *FUNCON* estan proporcionant valors correctes del gradient de la funció objectiu i del jacobià de les constriccions. Una forma aconsellable de procedir és incloure inicialment al fitxer SPECS la clau VERIFY LEVEL 3 o VERIFY. Amb això estareu forçant a Minos a fer una comprovació component a component del gradient i del jacobià al punt inicial d'iteració. Els resultats d'aquestes comprovacions les trobareu als apartats VERIFICATION OF CONSTRAINTS GRADIENTS RETURNED BY SUBROUTINE FUNCON i VERIFICATION OF OBJECTIVE FUNCTION GRADIENTS RETURNED BY SUBROUTINE FUNOBJ.

Si tot ha anat bé, ha d'aparèixer el missatge:

EXIT – OPTIMAL SOLUTION FOUND

o algun altre missatge de EXIT si hi ha hagut problemes. Fixeu-vos també en el valor de la variable interna NSTATE. Aquesta variable l'empra Minos per donar informació a l'usuari de l'estat de l'optimització quan efectua una crida a les rutines *FUNOBJ* i *FUNCON*. El significat dels diferents valors de NSTATE és:

- NSTATE = 0: s'efectua una crida normal a les rutines.
- NSTATE = 1: Minos crida per primera vegada a les rutines.
- NSTATE = 2: Minos crida per darrera vegada a les rutines, havent-se assolit l'òptim.
- NSTATE > 2: Minos crida per darrera vegada a les rutines, sense haver assolit l'òptim. En aquest cas, els diferents valors de la variable NSTATE indiquen l'error que s'ha produït.

Podeu observar el seu valor a l'òptim al final del fitxer .LIS, i si tot ha anat bé ha de valer NSTATE=2.

Finalment a l'apartat SECTION 1 - ROWS podreu observar el valor final de les constriccions a la columna ACTIVITY. En aquesta mateixa columna de l'apartat SECTION 2 - COLUMNS apareix el valor de cada variable a l'òptim si aquest s'ha assolit.

6.10 Exercici.

Amb l'ajut de Minos, solucioneu el problema

$$
\begin{aligned}
\min \quad & x_1^2 + \sqrt{x_2} + (x_3 + x_4)^2 \\
\text{subjecte a} \quad & \\
& x_1^2 + x_2^2 + x_3^2 + x_4^2 \geq 10 \\
& \sqrt{x_1} \cdot x_2 - x_4^3 \leq 0 \\
& 2x_1 + x_2 - 3x_3 - x_4 = 20 \\
& x_1 \geq 5 \\
& 4 \geq x_2 \geq 0 \\
& x_3 \geq 0 \\
& x_4 \geq 0
\end{aligned}
$$

Heu d'entregar:

a) La resposta a la següent qüestió: Tal i com teniu el problema definit, segurament en executar el programa aquest us abortarà. Això és degut a que en la `funobj` s'està avaluant el vector gradient en un punt on $x_2 = 0$. Quin problema veieu en fer aquesta avaluació? Per solucionar aquest problema, podeu introduir un límit inferior fictici a x_2, de forma que els límits de la variable siguin $4 \geq x_2 \geq 0.01$.

b) Considerant la Jacobiana de les restriccions de forma densa:
   1. Fitxer `.for` amb la funció objectiu i restriccions que heu programat.
   2. Fitxer `.dat` amb els apartats SPECS i MPS.
   3. Fitxer `.lis` de sortida, resaltant el valor de les variables a l'òptim, i els valors de les folgues per a les restriccions.

c) Considerant la Jacobiana de les restriccions de forma esparsa:
   1. Fitxer `.for` amb la funció objectiu i restriccions que heu programat.
   2. Fitxer `.dat` amb els apartats SPECS i MPS.
   3. Fitxer `.lis` de sortida.