

# **PRÀCTICA D'OPTIMITZACIÓ AMB EL PAQUET MINOS**

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

curs 1994-95

## OPTIMITZACIÓ DEL COST D'UN COET MULTIETAPA.

### 1. Presentació del problema.

Aquest document presenta un exercici de resolució d'un problema real (amb força simplificacions) mitjançant un paquet professional d'optimització no lineal. Passem tot seguit a descriure'l.

Abans de l'arribada de les darreres llançadores espacials (com ara l'Space Shuttle), els astronautes i satèl·lits eren posats en òrbita mitjançant el que podríem anomenar "coets multietapa". Aquests coets estaven formats per diferents mòduls (de massa  $M_i$  tones,  $i = 1, \dots, n$ , essent  $n$  el nombre total d'etapes) i una càpsula final amb la càrrega que havia de ser posada en òrbita (de massa  $P$  tones), tal i com mostra la figura 1. La major part de la massa  $M_i$  del  $n$  mòduls (o etapes) corresponia a grans quantitats de fuel per poder impulsar el coet. La idea d'aquests coets multietapa era que, un cop s'hagués consumit tot el fuel d'un determinat mòdul, aquest seria expulsat i per tant no s'hauria de mantenir com a llast per la resta de mòduls.

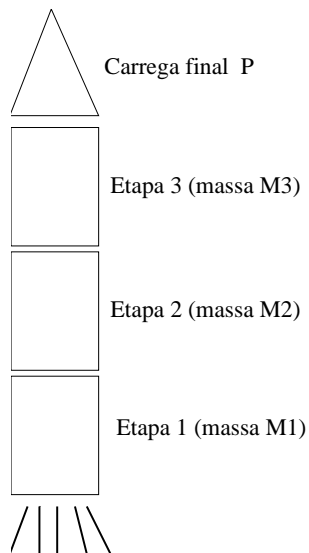


Fig. 1 Esquema d'un coet espacial de tres etapes.

El problema que es plantejava era conèixer el nombre d'etapes i la massa que havia de tenir cada mòdul per tal de poder situar en òrbita el coet, un cop es tenien fixades una sèrie de paràmetres relacionats amb les característiques constructives dels mòduls (les quals més tard seran descrites). El que sí que es verificava era que, com més etapes consideréssim, menor seria la massa total de tots els mòduls necessària per enlairar el coet fins la posició desitjada. Tanmateix, no era viable considerar coets amb moltes etapes pels riscos que suposava tenir una fallida tècnica en alguna de les maniobres de separació d'un mòdul respecte la resta del coet. Es va observar que l'equilibri perfecte entre els dos objectius contraposats anteriors s'assolia considerant coets de 3 etapes. Un cop fixat el nombre d'etapes, calia saber la massa mínima  $M_i$  de cada mòdul que ens garantís l'enlairament del coet. El problema que intentarem solucionar serà una petita variant d'aquest. En comptes de trobar la mínima massa de tots els mòduls mirarem de minimitzar el cost del coet, de forma que puguem assegurar que la càrrega

final de massa  $P$  assolirà una certa velocitat que ens garantirà la seva orbitació dins la posició desitjada. Aquest cost estarà en funció de la massa i de les característiques constructives dels mòduls.

## 2. Modelització del problema.

Un cop sabem quin és l'objectiu del nostre problema, passem a fer la modelització del sistema físic que ens descriu el comportament del coet. A partir d'ara suposarem que les masses venen expressades en tones i totes les velocitats en Km/hora.

En primer lloc considerarem el problema on només disposem d'un coet d'una etapa. En aquest cas tenim tan sols un mòdul de massa  $M_0$  i la càrrega final de massa  $P$ . Així mateix considerarem que la "tara" del nostre mòdul és  $M_v$ . Per "tara" entenem la massa sense considerar el fuel, és a dir

$$M_0 = M_v + fuel \quad (1)$$

Anomenarem factor estructural a la relació:

$$S = \frac{M_v}{M_0} \quad (2)$$

on  $S < 1$  sempre, i ens indica la relació entre la massa del mòdul i la total del mòdul, incloent el fuel. Aquesta és una característica constructiva del mòdul que ens ve fixada. Suposarem també que el nostre coet consumeix fuel a un ritme constant  $-k$ . Llavors la massa total del nostre coet depèn clarament de l'instant  $t$  en que ens trobem, i la seva expressió serà:

$$M(t) = P + M_0 - kt \quad (3)$$

Tenint en compte els conceptes anteriors, l'equació d'un coet en absència de forces exteriors (com ara la gravetat o el fregament) es pot trobar sabent que en un sistema físic es verifica que l'impuls mecànic (força per temps)  $I(t)$  que rep el coet durant un  $dt$  equival a la quantitat de moviment (massa per velocitat)  $p(t)$  dels gasos expulsats durant el mateix  $dt$ . Tindrem que  $I(t) = M(t)a(t)dt$ , essent  $a(t)$  l'acceleració del coet a l'instant  $t$ , i  $p(t) = -cdM(t)$ , essent  $c$  la velocitat (constant) d'expulsió dels gasos i  $dM(t)$  la massa dels gasos expulsats durant el lapse  $dt$ . Així

$$M(t)a(t)dt = -cdM(t)$$

d'on, tenint en compte que  $a(t)$  no és més que  $\frac{dv(t)}{dt}$  essent  $v(t)$  la velocitat del coet a l'instant  $t$ , tenim que:

$$a(t) = \frac{dv}{dt} = \frac{-c}{M(t)} \frac{dM(t)}{dt} \quad (4)$$

Clarament aquí introduïm una simplificació, donat que en enlairar un coet des de la terra sempre hi ha forces de fregament amb l'aire i, òbviament, la força d'atracció gravitatòria. Tanmateix, considerarem el cas ideal anterior, per no complicar l'equació que governa el comportament del coet.

Substituint (3) a (4) i derivant tenim que

$$\frac{dv}{dt} = \frac{ck}{P + M_0 - kt} \quad (5)$$

Integrant (5) obtenim l'equació de la velocitat del coet en funció del temps:

$$v(t) = \int \frac{ck}{P + M_0 - kt} dt = -c \ln(P + M_0 - kt) + B \quad (6)$$

El terme  $B$  es troba en funció de les condicions inicials. Si considerem que a l'instant inicial  $t = 0$  la velocitat inicial del coet és  $v_0$ , llavors:

$$v(0) = -c \ln(P + M_0) + B = v_0 \quad \Rightarrow \quad B = v_0 + c \ln(P + M_0) \quad (7)$$

Llavors l'equació (6) pot ser rescrita com

$$\begin{aligned} v(t) &= -c \ln(P + M_0 - kt) + v_0 + c \ln(P + M_0) \\ &= v_0 - c \ln\left(\frac{P + M_0 - kt}{P + M_0}\right) \\ &= v_0 - c \ln\left(1 - \frac{kt}{P + M_0}\right) \end{aligned} \quad (8)$$

La massa total de fuel del mòdul, donada per l'equació (1), és  $M_0 - M_v$ , i es crema a un ritme constant de  $k$  tal i com ja havíem dit anteriorment. Llavors, tenint en compte la definició (2) del factor estructural  $S$ , el temps  $t_b$  necessari per cremar-se tot el fuel serà

$$t_b = \frac{M_0 - M_v}{k} = \frac{\left(1 - \frac{M_v}{M_0}\right)M_0}{k} = \frac{(1 - S)M_0}{k} \quad (9)$$

Llavors l'increment de velocitat del coet un cop hem exhaurit tot el combustible de l'únic mòdul de que disposem vindrà donat per l'expressió

$$\begin{aligned} \Delta v &= v(t_b) - v_0 \\ &= v_0 - c \ln\left(1 - \frac{kt_b}{P + M_0}\right) - v_0 \\ &= -c \ln\left(1 - \frac{k}{P + M_0} \frac{(1 - S)M_0}{k}\right) \\ &= -c \ln\left(1 - \frac{(1 - S)M_0}{P + M_0}\right) \end{aligned} \quad (10)$$

Aquesta és l'equació a la que volíem arribar, i que haurem d'usar al nostre problema d'optimització resultant. Coneixent la velocitat inicial del coet (normalment sortirem del repòs), amb ella tenim una forma de determinar la velocitat final de la càrrega  $P$  en funció de la massa  $M_0$  i de la velocitat dels gasos  $c$  del mòdul, que serà expulsat un cop es cremi tot el seu combustible.

Tot el dit fins ara corresponia al cas de disposar d'un coet amb una única etapa. I hem vist com obtenir la velocitat final de  $P$  en funció de la massa  $M_0$  i velocitat d'expulsió de gasos  $c$ . En el cas multietapa la velocitat final  $v_f$  de  $P$  estarà format per la suma d'increments de velocitat  $\Delta v_i$  deguts a cada etapa concreta. És a dir,

$$v_f = \sum_{i=1}^n \Delta v_i \quad (11)$$

Cal, llavors, trobar ara una expressió per  $\Delta v_i$ . Per això, considerarem que a la etapa  $i$  la massa del nostre coet és

$$M_i + M_{i+1} + \dots + M_n + P$$

I podem reduir aquest cas al de tenir un coet amb una única etapa, on disposem d'un únic mòdul de massa  $M_0 = M_i$  i una càrrega final de massa  $\hat{P} = M_{i+1} + \dots + M_n + P$ . Llavors si apliquem ara l'equació (10) amb les variables  $M_0$  i  $\hat{P}$  definides anteriorment, i considerant que la velocitat d'expulsió de gasos  $c$  i el factor estructural  $S$  ara es troben associats a la etapa  $i$  (és a dir, que realment tenim  $c_i$  i  $S_i$ ) s'obté:

$$\begin{aligned} \Delta v_i &= -c_i \ln \left( 1 - \frac{(1 - S_i)M_0}{\hat{P} + M_0} \right) \\ &= -c_i \ln \left( 1 - \frac{(1 - S_i)M_i}{M_i + M_{i+1} + \dots + M_n + P} \right) \\ &= -c_i \ln \left( \frac{S_i M_i + M_{i+1} + \dots + M_n + P}{M_i + M_{i+1} + \dots + M_n + P} \right) \\ &= c_i \ln \left( \frac{M_i + M_{i+1} + \dots + M_n + P}{S_i M_i + M_{i+1} + \dots + M_n + P} \right) \end{aligned} \quad (12)$$

### 3. Formulació del problema.

A la secció anterior hem modelitzat el nostre sistema físic i hem obtingut l'equació (12) que ens determina l'increment de velocitat degut a cada etapa del coet. Recordem que el nostre problema original consistia en minimitzar els cost del coet, subjecte a que la càrrega final de massa  $P$  aconseguís una velocitat  $v_f$ , necessària per posar-se en òrbita. Considerarem que el cost del coet és funció lineal de la massa de cada mòdul, i quadràtica respecte la velocitat d'expulsió de gasos. Per tant el cost del coet l'escriurem com:

$$\text{Cost}(M_i, c_i) = \sum_{i=1}^n k_1 M_i + k_2 c_i^2 \quad (13)$$

on  $k_1$  i  $k_2$  són constants conegudes. Si denotem per  $\underline{c}$  i  $\bar{c}$  els límits tècnics inferior i superior de les velocitats d'expulsió de gasos  $c_i$ , llavors el nostre problema d'optimització serà:

$$\begin{aligned} \min \quad & \text{Cost}(M_i, c_i) \\ \text{subj. a} \quad & \sum_{i=1}^n \Delta v_i = v_f \\ & M_i \geq 0 \quad \bar{c} \geq c_i \geq \underline{c} \quad i = 1, \dots, n \end{aligned} \quad (14)$$

Substituint  $\Delta v_i$  pel seu valor definit a (12) el problema (14) pot ser escrit com

$$\begin{aligned} \min \quad & \text{Cost}(M_i, c_i) \\ \text{subj. a} \quad & \sum_{i=1}^n c_i \ln \left( \frac{M_i + M_{i+1} + \dots + M_n + P}{S_i M_i + M_{i+1} + \dots + M_n + P} \right) = v_f \\ & M_i \geq 0 \quad \bar{c}_i \geq c_i \geq \underline{c} \quad i = 1, \dots, n \end{aligned} \quad (15)$$

La restricció no lineal que apareix a (15) és força complexa, i per simplificar el problema el que farem és introduir unes variables auxiliars  $N_i$ ,  $i = 1, \dots, n$ , definides de la següent manera

$$N_i = \frac{M_i + M_{i+1} + \dots + M_n + P}{S_i M_i + M_{i+1} + \dots + M_n + P} \quad i = 1, \dots, n \quad (16)$$

Donat que les masses  $M_i$  són sempre positives, i que els factors estructurals  $S_i$ , tal i com s'han definit a (2), són sempre superiors a 1 tenim que les noves variables  $N_i$  hauran de verificar que  $N_i \geq 1$ . Aquest serà el límit inferior que les imposarem dins el problema d'optimització. Finalment, utilitzant les noves variables  $N_i$  abans definides, i escrivint l'equació (16) d'una forma equivalent, tenim que el nostre problema definitiu a resoldre és:

$$\begin{aligned} \min \quad & \sum_{i=1}^n k_1 M_i + k_2 c_i^2 \\ \text{subj. a} \quad & N_i (S_i M_i + \dots + M_n + P) - (M_i + \dots + M_n + P) = 0 \quad i = 1, \dots, n \\ & \sum_{i=1}^n c_i \ln N_i = v_f \\ & M_i \geq 0 \quad N_i \geq 1 \quad \bar{c}_i \geq c_i \geq \underline{c} \quad i = 1, \dots, n \end{aligned} \quad (17)$$

Es recomana que useu aquest ordre de restriccions i de variables: pel que fa al vector de variables, primer considereu les  $n$  masses  $M_i$ , a continuació les  $n$  variables  $N_i$  i finalment les  $n$  velocitats  $c_i$ . I pel que fa a les restriccions, que primer apareguin les  $n$  restriccions de canvi de variable i finalment la restricció de la velocitat final.

#### 4. Resolució del problema.

Aquesta darrera secció la dividirem en tres subapartats.

1. Dades del problema.
2. Breu introducció al paquet Minos.
3. Presentació de l'exercici.

Tot seguit passem a descriure cadascun dels punts anteriors.

##### 4.1. Dades del problema.

Al llarg de l'explicació que s'ha fet durant la modelització i formulació, han anat apareixent diversos coeficients que cal conèixer per tal de tenir completament determinat el problema. Per tal d'obtenir aquests coeficients haureu de fer des del sistema:

```
RUN DISK$H:[FIB.EIO.ONLC]GENDAT95
```

El programa us demanarà el vostre número d'alumne (aquest número el trobareu a la llista que surtirà al tauló del Departament d'EIO o bé se us donarà a classe). A continuació us crearà al vostre directori un fitxer anomenat *DADESXXX.DAT*, essent *XXX* el vostre número d'alumne. Aquest fitxer té un aspecte com ara:

```
DADES DEL PROBLEMA DEL COET ESPACIAL
ALUMNE No.100
VELOCITAT FINAL Vf (km/h)= 33791.0
MASSA CARREGA FINAL P (Tm)= 1.8610
```

COEFICIENT K1 DE COST= .14000E+03  
 COEFICIENT K2 DE COST= .68400E-04  
 LIMIT INFERIOR DE LA VELOCITAT D'EXPULSIÓ DE GASOS  $\underline{c}_i$  (km/h)= 11723.00  
 LIMIT SUPERIOR DE LA VELOCITAT D'EXPULSIÓ DE GASOS  $\bar{c}_i$  (km/h)= 12723.00  
 FACTOR ESTRUCTURAL  $S_i$ :

Etapa $i$	$S_i$
1	0.280
2	0.170
3	0.200
4	0.180

Es pot observar com aquest fitxer proporciona totes les dades necessàries per executar el problema. En primer lloc, dona la velocitat final  $v_f$  i massa  $P$  de la càrrega final que quedarà orbitant. A continuació ens diu els coeficients lineal  $k_1$  i quadràtic  $k_2$  que intervenen en la funció objectiu. Tot seguit ens mostra els límits de la velocitat d'expulsió de gasos  $\underline{c}$  i  $\bar{c}$ . Finalment ens dona per cada etapa el factor estructural  $S_i$ . El nombre d'etapes és un valor fixat, i valdrà 4 per tots els fitxers de dades. Aquest valor no és realista, donat que en apartats anteriors havíem dit que el nombre d'etapes que s'acostumava a usar era de 3. El fet de considerar 4 etapes es deu a que així el problema d'optimització resultant té una mida "raonable".

#### 4.2. Breu introducció al paquet Minos.

En aquest apartat descriurem el paquet Minos d'optimització, amb el qual s'haurà de resoldre el problema anterior. 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.

#### 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**ncore **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).

#### Problema estandard.

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

$$\begin{aligned}
 \text{min.} \quad & F(x) + c^t x + d^t y \\
 \text{subj.} \quad & l_1 \leq f(x) + A_1 y \leq u_1 \\
 & l_2 \leq A_2 x + A_3 y \leq u_2 \\
 & l \leq \begin{pmatrix} x \\ y \end{pmatrix} \leq u
 \end{aligned} \tag{18}$$

on:

- *Vectors constants:*

$$\begin{aligned} c &\in \mathbb{R}^{n_1} ; d \in \mathbb{R}_2^n \\ u_1, l_1 &\in \mathbb{R}^{m_1} ; u_2, l_2 \in \mathbb{R}_2^m \\ l, u &\in \mathbb{R}^{n_1+n_2} \end{aligned}$$

- *Matrius constants:*

$$\begin{aligned} A_1 &\in (m_1 \times n_2) \\ A_2 &\in (m_2 \times n_1) \\ A_3 &\in (m_2 \times n_2) \end{aligned}$$

- *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.

### 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  $\hat{f}_k(x)$  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:

$$\begin{aligned} \min. \quad & F(x) + c^t x + d^t y - \lambda_k^t (f - \hat{f}_k) + \frac{1}{2} \rho (f - \hat{f}_k)^t (f - \hat{f}_k) \\ \text{subj.} \quad & \hat{f}_k + A_1 y = b_1 \\ & A_2 x + A_3 y = b_2 \\ & l \leq \begin{pmatrix} x \\ y \end{pmatrix} \leq u \end{aligned} \tag{19}$$

on:

- La funció objectiu de (2) s'anomena *Lagrangià augmentat*.
- $\lambda_k$  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  $\rho$ .



**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	}	junttes en un sol fitxer COET.FOR o COET.C
rutina FUNCON		
fitxer SPECS	}	junts, i en aquest ordre, en un sol fitxer COET.DAT
fitxer MPS		

Trobareu unes plantilles d'aquests fitxers al directori:

DISK\$H:[FIB.EIO.ONLC]

Els fitxers s'anomenen COET.DAT i COET.FC. Al fitxer COET.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 COET.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.

### Lectura i escriptura de dades a les rutines *FUNOBJ* i *FUNCON*.<sup>1</sup>

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, COET.DAT).

IPRINT: unitat lògica d'escriptura assignada al fitxer de sortida de resultats (per exemple, COET.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 .DAT 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.

### 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
```

---

<sup>1</sup> 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.

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 paràmetres 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 nombre 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.

*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.

### 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:

```

12345678901123456789021234567890312345678904123456789051234567890612345678907
NAME          nom_problema
ROWS
  ww constricció
COLUMNS
  variable  constr_1  coeficient_1  constr_2  coeficient_2
RHS
  nom_indp  constr_1  terme_independent
RANGES
  nom_rang  constr_1  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:

```

123456789011234567890212
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:

```

1234567890112
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:

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

### 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 declari 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  $i$ -èsima del jacobià, indicant el nom de la constricció no lineal corresponent i un valor numèric fictici.
3. els coeficients no nuls de la columna  $i$ -èsima 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:

```
1234567890112345678902123456789031234567890412345678905123456789061
COLUMNS
...variable..constr1..coeficient1...constr2..coeficient2
```

on els punts indiquen espais en blanc, `variable` és el nom de la variable que estem tractant, `constr1` i `constr2` són noms de constriccions on intervé la variable en qüestió, i `coeficient1` i `coeficient2` 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:

```

123456789011234567890212345678903123456
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:

```

123456789011234567890212345678903123456
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 de ser `rang = u - l`.

*Secció BOUNDS:*

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

```

123456789011234567890212345678903123456
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:

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

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:

```
123456789011234567890212345678903123456
.FX.INITIAL□..variable..valorinicial
```

### Exemple de codificació d'un problema en format MPS

Sigui el problema:

$$\begin{array}{llll}
 \text{min.} & (x_1 & +x_2 & +x_3)^2 \\
 \text{subj.} & x_1^2 & +x_2^2 & = 2 \\
 & & x_2^4 & = 4 \\
 & x_1^4 & & +x_3^4 \geq 0 \\
 & 2x_1 & +4x_2 & -x_3 \leq 1 \\
 & x_1 \geq 0, & x_2 \geq 0, & x_3 \leq 1
 \end{array}$$

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

```
12345678901123456789021234567890312345678904123456789051234567890612345678907
NAME          PROBLEM1
ROWS
E  CONSNOL1
E  CONSNOL2
G  CONSNOL3
L  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  2.0
  TERMINDE   CONSNOL2  4.0
  TERMINDE   CONSNOL3  0.0
  TERMINDE   CONSLIN1  1.0
BOUNDS
LO LIMSIMP   X1          0.
LO LIMSIMP   X2          0.
UP LIMSIMP   X3          1.
ENDDATA
```

**Muntatge i execució.**

Un cop s'ha fet el fitxer COET.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) COET.OBJ l'ordre que haurem d'executar per fer el muntatge serà:

```
LINK COET.OBJ,USER$LIB:[MINOS53]MINOS53/LIB
```

Si per contra hem escrit les dos rutines *FUNOBJ* i *FUNCON* en c i les tenim al fitxer, per exemple, COET.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,COET.OBJ,USER$LIB:[MINOS53]MINOS53/LIB
```

Després d'executar el programa (si utilitzeu el programa principal que us suministrem) obtindreu un fitxer anomenat COET.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 .DAT, com ara:

```
XXXX WARNING - NO LINEAR OBJECTIVE SELECTED
```

Aquest missatge, en concret, si el trobeu us indica que esteu resolent un problema amb funció objectiu no lineal. 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 jacobiana 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 jacobiana 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.

### 4.3. Presentació de l'exercici.

Es demana solucionar el problema presentat, amb les dades particulars de cada alumne, usant el paquet Minos. Per fer-ho haureu d'escriure els apartats SPECS i MPS, i programar la *FUNOBJ* i *FUNCON*.

**Pel que fa al jacobià els alumnes que entreguin la pràctica al juny han de presentar la rutina FUNCON amb el jacobià dens, mentre que els alumnes que entreguin la pràctica al setembre han de presentar la rutina FUNCON amb el jacobià espars.**

### Normes generals.

L'informe d'aquesta pràctica haurà de ser lliurat en fulls, tots de la mateixa mida (excepte els que continguin output del Minos, donat que és a 132 columnes), grapats i amb una primera pàgina on hi constin les següents dades:

- Nom de la pràctica i assignatura.
- Data d'entrega.
- Nom de l'alumne.
- Número d'alumne que té assignat.
- USERNAME d'un compte on l'alumne pugui rebre missatges.

No s'acceptaran informes que no respectin aquestes normes. En particular, no s'acceptarà cap informe amb llistats sense grapar ni retallar a la mida de la resta de fulls (plegueu els fulls de 132 columnes).

### Contingut.

L'informe ha de contenir:

1. Llista del total de variables que utilitzeu, indicant quines heu considerat com no lineals i quines com lineals. Indicació de quines constriccions heu considerat com no lineals i quines com lineals.
2. Llistat de les rutines *FUNOBJ* i *FUNCON* programades.
3. Llistat del fitxer .DAT que hagueu escrit.
4. Un cop hagueu executat el programa, heu de presentar:
  - Les parts del fitxer .LIS que continguin:
    - La comprovació dels elements del jacobiana obtinguda amb VERIFY DERIVATIVES.
    - La comprovació dels elements del gradient obtinguda amb VERIFY DERIVATIVES.
    - L'estat final de l'optimització, a l'apartat EXIT-OPTIMAL SOLUTION FOUND.
    - El valor de les constriccions a l'òptim, a l'apartat SECTION 1 - ROWS.
    - El valor de les variables a l'òptim, a l'apartat SECTION 2 - COLUMNS.
5. Responen la següent qüestió:

A la formulació del problema (secció 3) es van introduir les variables  $N_i$  per realitzar un canvi de variable i simplificar l'estructura de les restriccions. Per la pròpia definició de  $N_i$  a l'equació (16) es podia veure que sempre es verificaria que  $N_i \geq 1$ , i aquest és el límit inferior que vam introduir a la nostra formulació (17). Tanmateix, no caldria haver introduït aquests límits explícitament a (17), i podríem haver deixat les variables  $N_i$  lliures i el propi procés d'optimització a l'òptim ja garantiria que  $N_i \geq 1$ , donat que s'han de verificar les equacions de canvi de variable. Això és cert, en teoria. A la pràctica, però, hi ha una petit problema. Feu la prova d'eliminar a l'apartat BOUNDS de la definició MPS del vostre problema el límit inferior 1 de les variables  $N_i$ , i torneu a executar el vostre programa. Què succeeix? Podeu intentar raonar per què? Per ajudar-vos a contestar la pregunta, us diré que en eliminar el límit inferior 1 d'aquestes variables, automàticament Minos les considera amb fita inferior igual a 0 (és a dir  $N_i \geq 0$ ) i en començar el procés d'optimització totes les variables (les  $M_i$ , les  $N_i$  i les  $c_i$ ) es troben inicialitzades a la seva fita inferior (és a dir, el nostre punt inicial d'iteració serà ara  $X_0 = (\vec{0}_n, \vec{0}_n, \vec{c}_n)^t$ ).