Un algorithme général de recherche des horizons de planification (problème mono-produit, discret et à coûts concaves)

A. Bensoussan, Jean-Marie Proth, M. Queyranne

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 406

UN ALGORITHME GÉNÉRAL DE RECHERCHE DES HORIZONS DE PLANIFICATION

(PROBLÈME MONO-PRODUIT, DISCRET ET À COUTS CONCAVES)

Alain BENSOUSSAN
Jean-Marie PROTH
Maurice QUEYRANNE

Mai 1985

UN ALGORITHME GENERAL DE RECHERCHE DES HORIZONS DE PLANIFICATION

(Problème mono-produit, discret et à coûts concaves)

BENSOUSSAN Alain *

PROTH Jean-Marie *

QUEYRANNE Maurice **

* INRIA

** Université de British Columbia

ABSTRACT :

This paper gives a general algorithm which leads to a planning horizon (if it exists) in the case of a mono-product and discrete production problem with concave costs, knowing the forecast horizon.

The only assumption is that the cost functions are piecewise linear. It does not restrict the generality of these results on a practical point of view.

RESUME :

Ce papier donne un algorithme général qui conduit à un horizon de planification (si il existe) dans le cas d'un problème de production discret, mono-produit et à coûts concaves. L'horizon de prévision est donné.

La seule hypothèse est que les fonctions de coût sont linéaires par morceaux. Cela ne réduit en rien la généralité de ces résultats d'un point de vue pratique.

I - INTRODUCTION

Nous nous intéressons au problème mono-produit, discret, à coûts concaves et non décroissants. Ce problème est bien connu lorsqu'il est à horizon fini.

Nous rappellerons d'abord ses principales propriétés.

Nous aborderons ensuite le problème de la recherche d'un horizon de planification connaisant un horizon de prévision. On appelle horizon de prévision un point K dans le temps en deça duquel les demandes sont connues. Un horizon de planification N est un second point dans le temps, antérieur au précédent ($N \leq K$) et tel que tout problème à horizon $M \geq K$ admet un contrôle optimal qui s'obtient en prolongeant de manière adéquate le contrôle optimal du problème à horizon N, quelles que soient les demandes entre K et M.

Nous montrons d'abord qu'il suffit de se limiter à l'étude des problèmes à horizon K+1 pour trouver un horizon de planification inférieur ou égal à l'horizon de prévision K.

Nous indiquons ensuite comment rechercher un horizon de planification connaissant un horizon de prévision. La seule difficulté rencontrée au cours de cette démarche peut être surmontée si on suppose les fonctions de coûts linéaires par morceaux. C'est ce que nous faisons. Nous aboutissons ainsi à l'algorithme annoncé.

Nous appliquons enfin cet algorithme à un exemple simple.

II - DEFINITIONS ET NOTATIONS

L'origine des temps est 0.

$\xi_i, i = 1,2,3,\ldots,$ est la demande positive ou nulle qui apparaît à l'instant i. Elle est connue pour $i = 1,2,\ldots,K$.

K est appelé <u>horizon de prévision.</u>

Nous connaissons également le niveau de stock $y_0 \geq 0$ à l'instant 0. Nous pouvons, sans restreindre la généralité de ce qui suit, prendre $y_0 = 0$.

Pour $i = 1,2,3,\ldots,$ y_i désigne le niveau des stock sur la $(i-1)^{\text{ème}}$ période et v_i est la production au cours de la $i^{\text{ème}}$ période.

Les équations d'état s'écrivent donc :

$$y_i = y_{i-1} + v_i - \xi_i, \; i = 1,2,3,\ldots \qquad (1)$$

Nous appelons :

$V = \{v_i\} \; i = 1,2,3,\ldots$ <u>contrôle.</u>

et

$Y = \{y_i\} \; i = 0,1,2,\ldots$ <u>ensemble des états associés au contrôle V.</u>

Nous imposons les contraintes suivantes :

$$\left.\begin{array}{l} y_i \geqq 0 \\ \text{et} \\ v_i \geqq 0 \end{array}\right\} \quad \text{pour } i = 1,2,3,\ldots \qquad (2)$$

Ces contraintes signifient respectivement que les ruptures de stock ne sont pas admises et que les réapprovisionnements sont positifs ou nuls.

Un contrôle qui vérifie les contraintes (2) est dit <u>admissible</u>.

Nous connaissons également les coûts associés au problème :

<u>pour $i = 1,2,3,\ldots$:</u>

a. f_i est la fonction qui donne le coût de stockage sur la $i^{\text{ème}}$ période. Elle est définie sur $\mathbb{R}^+$ et prend ses valeurs dans $\mathbb{R}^+$.
Elle est non décroissante et concave.

b. c_i est la fonction donnant le coût de production sur la $i^{\text{ème}}$ période. Elle est également définie sur $\mathbb{R}^+$, à valeurs dans $\mathbb{R}^+$, concave et non décroissante.

Soit, sur un horizon N donné, un contrôle admissible :

$$V = \{v_1, v_2, \ldots, v_N\}$$

et la suite des états correspondants :

$$Y = \{y_0 = 0, y_1, \ldots, y_N\}$$

Le coût associé à ce contrôle s'écrit :

$$C(V) = \sum_{i=1}^{N} \{c_i(v_i) + f_i(y_i)\} \qquad (3)$$

Un contrôle admissible V* est optimal si :

$$C(V^*) = \underset{V\in D}{\mathrm{Min}}\ C(V)$$

où D est l'ensemble des contrôles admissibles.

Nous savons qu'il existe toujours un contrôle optimal

$V^* = \{v_1^*, v_2^*, \ldots, v_N^*\}$ tel que $y_N^* = 0$ ($Y^* = \{y_0 = 0, y_1^*, \ldots y_N^*\}$

est la suite des niveaux du stock correspondant à V*).

Dans la suite, nous ne considérons que les contrôles optimaux ayant cette propriété, c'est à dire les contrôles optimaux qui conduisent à un niveau de stock nul à l'horizon choisi. Lorsque plusieurs contrôles optimaux répondent à cette condition, nous en choisirons un quelconque. Nous parlerons simplement du contrôle optimal associé au problème.

Nous savons également qu'il existe un sous-ensemble fini de D qui contient le contrôle optimal. C'est l'ensemble des contrôles $V=\{v_1, v_2, \ldots, v_N\}$ qui admettent comme suite de niveaux de stocks associée $Y = \{y_0=0, y_1, y_2, \ldots, y_N\}$ et qui vérifient :

$$\text{et} \left\{ \begin{array}{l} v_i = 0 \text{ si } y_{i-1} \geq \xi_i,\ i = 1,2,\ldots,N \\ v_i = \sum_{k=i}^{p} \xi_k - y_{i-1} \text{ avec } p \in \{i, i+1, \ldots, N\} \text{ si } y_{i-1} < \xi_i \end{array} \right\} \quad i=1,2,\ldots,N \qquad (4)$$

avec peut-être une exception pour le premier réapprovisionnement non nul lorsque le problème est non stationnaire,c'est à dire lorsque l'un au moins des coûts de stockage ou de production dépend du temps.

Dans ce cas en effet, si :

$$y_{i-1} = y_0 - \sum_{k=1}^{i-1} \xi_k \geq \xi_i ,$$

alors :

$$\left\{ \begin{array}{l} \text{soit } v_i = 0 \\ \text{soit } v_i = \sum_{k=i}^{p} \xi_k - y_{i-1} \end{array} \right.$$

où $p \in \{p^*, p^*+1, \ldots N\}$

p* est le plus petit entier qui vérifie :

$$\sum_{k=i}^{p^*} \xi_i - y_{i-1} \geq 0$$

Dans le cas particulier où $y_0=0$, cas auquel nous nous intéressons ici, la propriété (4) et son exception se réduisent à :

$$\text{et :}\quad \left.\begin{array}{l} y_{i-1}\,.\,v_i = 0 \text{ pour } i = 0,1,\ldots,N \\ v_i \in \{\sum_{k=i}^{r} \xi_k\}_{r=i-1,i,\ldots,M} \end{array}\right\} \qquad (5)$$

Nous retiendrons encore les définitions suivantes :

N est horizon de planification si :

> quels que soient $M \geq N$ et $\xi_{N+1}, \xi_{N+2},\ldots, \xi_M$, demandes positives ou nulles, alors le contrôle optimal du problème à horizon M s'obtient en prolongeant de manière adéquate le contrôle optimal du problème à horizon N (rappelons que $\xi_1, \xi_2,\ldots,\xi_N$ sont connus). La condition est triviale lorsque M = N.

Soit encore $K \geq N$. On suppose connus $\xi_1, \xi_2,\ldots, \xi_K$.

Alors N est horizon de planification connaissant l'horizon de prévision K si :

> quels que soient $M \geq K$ et $\xi_{K+1}, \xi_{K+2},\ldots, \xi_M$, demandes positives ou nulles, alors le contrôle optimal du problème à horizon M s'obtient en prolongeant de manière adéquate le contrôle optimal du problème à horizon N (rappelons que $\xi_1, \xi_2,\ldots,\xi_K$ sont connus). La condition est triviale lorsque M = K.

On remarque que les affirmations "N est horizon de planification" et "N est horizon de plannification connaissant l'horizon de prévision N" sont équivalentes. La notion d'horizon de planification n'est donc qu'un cas particulier de la notion d'horizon de planification connaissant un horizon de prévision. Dans toute la suite, nous ne nous attacherons plus qu'à cette dernière définition.

III - CONDITIONS D'EXISTENCE D'UN HORIZON DE PLANIFICATION

Le premier résultat nous donne une condition sur le niveau du stock au cours de la période qui précède l'horizon de planification pour le contrôle optimal de tout problème à horizon supérieur ou égal à l'horizon de prévision.

RESULTAT I

N est horizon de planification connaissant l'horizon de prévision K ($N \leq K$) si et seulement si il existe, pour tout problème à horizon $M \geq K$, un contrôle optimal V^M tel que $y_N^M = 0$ (y_N^M est le niveau du stock sur la Nème période si l'on applique V^M).

DEMONSTRATION

a. La condition est nécessaire car le contrôle optimal du problème à horizon N a un stock final nul et, par définition, le contrôle optimal du problème à horizon M s'obtient en prolongeant de manière adéquate le contrôle optimal du problème à horizon N.

b. La condition est suffisante car si $y_N=0$ et si le contrôle est optimal pour le problème à horizon M, alors sa restriction à l'horizon N est optimale.
Le contrôle optimal du problème à horizon M est donc bien obtenu en prolongeant de manière adéquate le contrôle optimal du problème à horizon N.

Ceci achève la démonstration. □

Le second résultat montre que la recherche d'un horizon de planification connaissant l'horizon de prévision K peut se limiter à l'étude des problèmes à horizon K+1.

RESULTAT II

N est horizon de planification connaissant l'horizon de prévision K si et seulement si, quelle que soit la demande $x \geq 0$ à l'horizon K+1, le contrôle optimal du problème à l'horizon K+1 s'obtient en prolongeant de manière adéquate le contrôle optimal du problème à horizon N.

DEMONSTRATION

a. La condition est nécessaire. Pour s'en convaincre, il suffit de faire M=K+1 dans la définition de l'horizon de planification connaissant l'horizon de prévision.

b. Nous montrons maintenant que la condition est suffisante. On notera C(N1,N2,X1,X2) le coût optimal associé au problème débutant à l'instant N1 avec un stock X1 et se terminant à l'instant N2 avec une demande X2 (il est équivalent de dire que le stock final est X2 et la demande en N2 nulle).

Nous utiliserons encore la notation suivante :

$$\sigma^{p,q} = \begin{cases} \sum_{i=p}^{q} \xi_i & \text{si } q \geq p \\ 0 & \text{sinon} \end{cases} \qquad (6)$$

Alors pour tout $M > K$ et quelles que soient les demandes $\xi_{K+1}, \xi_{K+2}, \ldots, \xi_M$ positives :

$$C(0,M,0,\xi_M) = C(0,K+1,0,\sigma^{K+1,r}) + C(K+1,M,\sigma^{K+1,r},0) \qquad (7)$$

avec $r \in \{K+1, K+2, \ldots, M\}$

Le choix de $\sigma^{K+1,r}$ est la conséquence de (5).
Le premier terme du second membre de (7) est le coût optimal du problème à horizon K+1 avec une demande $x = \sigma^{K+1,r}$ à l'instant K+1 et x peut prendre toutes les valeurs de $\mathbb{R}^+$, car les demandes $\xi_{K+1}, \ldots, \xi_M$ sont quelconques.

Ce qui achève la démonstration. □

IV - RECHERCHE DE L'HORIZON DE PLANIFICATION

K désigne toujours l'horizon de prévision.

Avec les notations que nous venons d'introduire, le coût optimal s'écrit, lorsque le problème est à horizon K+1 avec une demande x à l'instant K+1 :

$$C(0,K+1,0,x) = \underset{r \in (0,1,\ldots,K)}{\mathrm{Min}} \{C(0,r,0,0) + c_{r+1}(\sigma^{r+1,K} + x) + \sum_{s=r+2}^{K+1} f_s(\sigma^{s,K} + x)\} \qquad (8)$$

Nous désignons par $R^*(x)$ l'ensemble des entiers r qui réalisent l'optimum lorsque la demande est x à l'instant K+1.

Les deux résultats qui suivent sont à la base de l'algorithme que nous proposons plus loin.

RESULTAT III

S'il existe $x \in \mathbb{R}^+$ tel que $R^*(x) = \{0\}$, alors le problème n'admet pas d'horizon de planification connaissant l'horizon de prévision K.

DEMONSTRATION

Le résultat précédent signifie que, pour tout $x \in \mathbb{R}^+$ tel que $R^*(x) = \{0\}$, il est optimal de produire l'ensemble de la demande au cours de la seule première période. A l'optimum, le stock n'est donc jamais nul sur [1,K].
Le problème n'admet donc pas d'horizon de planification connaissant l'horizon de prévision K (voir résultat I).

Supposons maintenant que $R^*(x) \neq \{0\}$ quel que soit $x \in R^+$.

Nous allons montrer le résultat suivant :

RESULTAT IV

Soit K l'horizon de prévision et $N \in \{2,3,\ldots,K\}$. Supposons que, pour tout $x \in \mathbb{R}^+$, il existe $r^x \in R^*(x)$ tel que :

1. $r^x \geq N$

2. Le contrôle optimal V^x du problème à horizon K+1, avec demande x à l'instant K+1, vérifie $y^x_{N-1} = y^x_{r^x-1} = 0$, où y^x_{N-1} et $y^x_{r^x-1}$ sont les niveaux du stock associés à V^x sur les périodes N-1 et r^x-1 respectivement. Alors N est horizon de planification connaissant l'horizon de prévision K.

DEMONSTRATION

La restriction de V^x à l'horizon N est optimale pour le problème à horizon N, sans quoi V^x ne serait pas optimal pour le problème à horizon K+1. V^x s'obtient donc bien en prolongeant de manière adéquate le contrôle optimal du problème à horizon N. N est donc bien horizon de planification (voir le résultat II). □

Nous voyons qu'il est possible de trouver une solution de notre problème, ou de décider qu'il n'y en a pas, à partir du moment où l'on sait trouver les ensembles $R^*(x)$, $x \in \mathbb{R}^+$, et qu'ils sont en nombre fini. Cela est possible lorsque les fonctions de coût sont linéaires par morceaux.

Nous étudions ce cas qui reste très général d'un point de vue pratique.

V - CAS DES FONCTIONS DE COUT LINEAIRES PAR MORCEAUX

Pour tout r=1,2,...,K,K+1 nous connaissons les réels :

$$\begin{cases} 0 \leq a_r^1 < a_r^2 < \ldots < a_r^{H(r)} \\ b_r^1 > b_r^2 > \ldots > b_r^{H(r)} \geq 0 \end{cases}$$

et

$$\begin{cases} 0 \leq g_r^1 < g_r^2 < \ldots < g_r^{L(r)} \\ h_r^1 > h_r^2 > \ldots > h_r^{L(r)} \geq 0 \end{cases}$$

qui servent à définir :

1. <u>Le coût de production</u> sur la $r^{\text{ème}}$ période :

$$c_r(v)=\underset{j\in\{1,\dots,H(r)\}}{\operatorname{Min}}\{ a_r^j \chi_{v>0}+b_r^j v\} \text{ quel que soit } v\in R^+ \tag{9}$$

2. <u>Le coût de stockage</u> sur la $r^{\text{ème}}$ période :

$$f_r(y)=\underset{j\in\{1,\dots,L(r)\}}{\operatorname{Min}}\{ g_r^j \chi_{y>0}+h_r^j y\} \quad \text{quel que soit } y\in R^+ \tag{10}$$

Alors la relation (8) devient :

$$\begin{aligned} C(0,K+1,0,x)=\underset{r\in(0,1,\dots,K)}{\operatorname{Min}} \Bigg\{ & C(0,r,0,0) \\ & +\underset{j\in\{1,\dots,H(r+1)\}}{\operatorname{Min}} [a_{r+1}^j \chi_{(\sigma^{r+1,K}+x)>0} + b_{r+1}^j(\sigma^{r+1,K}+x)] \\ & +\sum_{s=r+2}^{K+1}\left(\underset{j\in\{1,\dots,L(s)\}}{\operatorname{Min}} [g_s^i \chi_{(\sigma^{s,K}+x)>0} + h_s^j(\sigma^{s,K}+x)]\right)\Bigg\} \end{aligned} \tag{11}$$

Nous distinguons les cas $x=0$ et $x>0$

a. Lorsque $x=0$, nous sommes ramenés au problème à horizon K et $R^*(x)=\{K\}$

b. Lorsque $x>0$, la relation (11) peut s'écrire :

$$\begin{aligned} C(0,K+1,0,x)=\underset{r\in(0,1,\dots,K)}{\operatorname{Min}} \Bigg\{ & C(0,r,0,0) \\ & +\underset{j\in\{1,\dots,H(r+1)\}}{\operatorname{Min}} [a_{r+1}^j + b_{r+1}^j\sigma^{r+1,K} + b_{r+1}^j x] \\ & +\sum_{s=r+2}^{K=1}\left(\underset{j\in\{1,\dots,L(s)\}}{\operatorname{Min}} [g_s^j + h_s^j\sigma^{s,K} + h_s^j x]\right)\Bigg\} \end{aligned} \tag{12}$$

Partant de la relation (12), nous bâtissons l'algorithme suivant.

Il a été écrit en supposant que :

- les demandes ont été rangées au préalable dans un fichier "Demandes",
- les coefficients des coûts de lancement (ou de production) ont été rangés au préalable dans le fichier "Lancement", à raison d'un enregistrement par période dans le cas non stationnaire. Dans le cas stationnaire, seul le premier enregistrement est occupé. Le fichier est à accès direct.

- les coefficients des coûts de stockage ont été rangés au préalable dans le fichier "Stockage", à raison d'un enregistrement par période dans le cas non-stationnaire. Dans le cas stationnaire, seul le premier enregistrement est occupé. Le fichier est à accès direct.

1. Lecture des demandes d(i),i=1,...,K dans le fichier "Demandes"
2. Faire jz = 0
3. <u>Pour r = 0 à K faire</u> :
   - 3.1 cc = 0 si r = 0 et cc = C(0 , r, 0 , 0) sinon
   - 3.2 Lecture, dans le $(r+1)^{\text{ème}}$ enregistrement du fichier "Lancement", de : $m,\left(a(j),b(j),j=1,m\right)$
   - 3.3 Calculer :
     sig 1 = $\sigma^{r+1,K}$ (nul si r = K)
   - 3.4 <u>Pour j = 1 à m, faire</u> :
     - 3.4.1 bb(j)=b(j) + a(j)* sig 1 + cc
     - 3.4.2 aa(j) = a(j)
   - 3.5 Faire mm 1 = m
   - 3.6 si r = k, aller en 4.3
   - 3.7 Si r $\neq$ k, faire :
     - 3.7.1 <u>Pour s = r+2 à K+1</u>
       - 3.7.1.1 Lecture, dans le $s^{\text{ème}}$ enregistrement du fichier "Stockage", de m, $\left(a(j),b(j),j=1,m\right)$
       - 3.7.1.2 Calculer
         sig 2 = $\sigma^{s,K}$ (nul si s = K+1)
       - 3.7.1.3 <u>Pour j = 1 à m, faire</u> :
         - 3.7.1.3.1 uu(j) = a(j)
         - 3.7.1.3.2 vv(j) = a(j)* sig 2 + b(j)
       - 3.7.1.4 Faire mm2 = m
       - 3.7.1.5 Faire kk = 0
       - 3.7.1.6 <u>Pour k1 = 1,mm1</u>
         <u>Pour k2 = 1,mm2</u>
         - 3.7.1.6.1 Faire : kk = kk+1
         - 3.7.1.6.2 Faire : pp(kk) = aa(k1) + uu(k2)
         - 3.7.1.6.3 Faire : qq(kk) = bb(k1) + vv(k2)
       - 3.7.1.7 Classement des éléments des tableaux pp et qq dans l'ordre décroissant des éléments de pp

3.7.1.8 Supression des couples pp(kk) et qq(kk) tels qu'il existe kkl tel que :

$$pp(kk) \geqq pp(kkl)$$
$$qq(kk) \geqq qq(kkl)$$

Soit mm 1 le nombre de couples restants.

3.7.1.9 <u>Pour j = 1 à mml faire</u> :

3.7.1.9.1 aa(j) = pp(j)

3.7.1.9.2 bb(j) = qq(j)

3.8 <u>Pour j = 1 à mml faire</u> :

3.8.1 jz = jz+1

3.8.2 iw (jz) = r

3.8.3 ss(jz) = aa(j)

3.8.4 tt(jz) = bb(j)

3.9 Suppression des triplets iw.(k),ss(k) et tt(k) tel qu'il existe kl vérifiant :

$$ss(kl) \leqq ss(k) \quad \text{et} \quad tt(kl) \leqq tt(k)$$

Désignons encore par jz le nombre d'éléments de ces tableaux.

4. Recherche de toutes les solutions optimales des problèmes dont l'horizon est un élément de iw(k),k=1,...jz.
5. S'il existe un point dans le temps qui est un instant de réapprovisionnement strictement positif pour au moins une solution optimale de chacun des problèmes désignés en 4., alors ce point est horizon de planification.

VI - UN EXEMPLE NUMERIQUE

Considérons le problème stationnaire suivant :

- horizon de prévision : 15
- stock initial : 0
- demande : d(i)=1 pour i=1,2,...,15
- coût de production : $c(v) = 4v + 5X_{v>0}$
- coût de stockage : $f(y) = v + 0.5X_{y>0}$

Les horizons des problèmes dont il faut chercher la solution optimale sont ici : $r_1 = 12$, $r_2 = 13$, $r_3 = 14$, $r_4 = 15$

Ce sont les valeurs de r qui réalisent le minimum de (12) lorsque x varie de 0 à + ∞. Ce sont aussi les valeurs prises par les éléments du tableau iw de l'algorithme ci-dessus.

Les différentes solutions optimales de chacun de ces problèmes sont données par le tableau :

PERIODES	H. 12	HORIZON 13				HORIZON 14					H. 15
	V_1^{12}	V_1^{13}	V_2^{13}	V_3^{13}	V_4^{13}	V_1^{14}	V_2^{14}	V_3^{14}	V_4^{14}	V_5^{14}	V_1^{15}
1	3	3	3	3	4	2	3	3	3	3	3
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	3	0	0	0	0	0
4	3	3	3	4	0	0	2	3	3	3	3
5	0	0	0	0	3	0	0	0	0	0	0
6	0	0	0	0	0	3	3	0	0	0	0
7	3	3	4	0	0	0	0	2	3	3	3
8	0	0	0	3	3	0	0	0	0	0	0
9	0	0	0	0	0	3	3	3	0	0	0
10	3	4	0	0	0	0	0	0	2	3	3
11	0	0	3	3	3	0	0	0	0	0	0
12	0	0	0	0	0	3	3	3	3	0	0
13		0	0	0	0	0	0	0	0	2	3
14						0	0	0	0	0	0
15											0

En considérant :

V_1^{12} et (V_1^{13} ou V_2^{13} ou V_3^{13}) et (V_2^{14} ou V_3^{14} ou V_4^{14} ou V_5^{15}) et V_1^{15},

on voit que le début de la période 3 est horizon de plannification.

CONCLUSION

Les développements qui précèdent résolvent le problème de la recherche d'un horizon de planification connaissant un horizon de prévision.
La seule condition restrictive est la nécessité d'utiliser des fonctions de coût linéaires par morceaux. Cette condition n'est restricitive qu'en apparence : dans la pratique, les fonctions de coût sont en effet presque toujours données point par point.

BIBLIOGRAPHIE

[1] A. BENSOUSSAN, M. CROUHY, J.M. PROTH
"Mathematical Theory of Production Management", Advanced Serie in Management, North Holland Publishing, 1983.

[2] L.A. JOHNSON and MONTGOMERY
"Opération Research in Production Planning", Scheduling and Inventory Control, Wiley, New-York,1974.

LE LOGICIEL

Ce logiciel est actuellement implanté sur MULTICS.

On donne trois programme :

1. le programme NCHAR, dont le rôle est de stocker les données du problème sur disque ,
2. le programme SINGH qui :
   - 2.1. lit les données sur disque,
   - 2.2. traite le problème en faisant appel au sous-programme LRET,
   - 2.3. déduit des solution optimales des problèmes à horizons finis obtenus, l'horizon de planification le plus court (s'il existe)
3. le sous-programme LRET qui est la phase rétrograde de la recherche d'une solution optimale d'un problème à horizon fini lorsqu'on utilise les équations de la programmation dynamique de type rétrograde,
4. le sous-programme CL qui calcule les coûts.

NCHAR

nchar.fortran 04/19/85 1534.5 hfe Fri

```
c              * * * * * * * * * * * * * * * * * * * *
c              *       DATA      LOADING             *
c              * * * * * * * * * * * * * * * * * * * *
c n:horizon of  the problem.
c eps: allows to test zero.
c icod:0 if the problem is stationary
c       1 if not stationary.
c d:vector of demands.
c y0:initial stock level.
c a and b:coefficients of slope and ordinate at the origin for the cost functi
c ons of the periods given.
c m:number of different determinations for the period given.
      character*2 aa
      dimension a(20),b(20),d(100)

      open(50,form="formatted")
      open(51,form="formatted")
      open(52,access="direct",form="formatted",recl=570)
      open(53,access="direct",form="formatted",recl=570)
      write(0,1)
1     format(2x,"Horizon of the problem?")
      ind=5
      read(0,2)n
2     format(v)
      write(0,3)
3     format(2x,"Initial stock?")
      read(0,2)y0
      write(0,4)
4     format(2x,"Value of eps used to test zero?")
      read(0,2)eps
      write(0,7)
7     format(2x,"Stationnary problem?")
      read(0,25)aa
25    format(a2)
      icod=0
      if(aa.eq."no")icod=1
      write(50,5)n,y0,eps,icod,ind
5     format(i4,2e14.7,i1,i2)
      write(0,6)
6     format(2x,"Constant demand?")
      read(0,25)aa
      nn=1
      if(aa.eq."no")nn=n
      do 8 i=1,nn
      write(0,9)i
9     format(2x,"Demand during the period",i4,"?")
8     read(0,2)d(i)
      if(nn.eq.n)go to 10
      do 11 i=2,n
```

```
11      d(i)=d(1)
10      write(51,12)(d(i),i=1,n)
12      format(100e14.7)
        nn=1
        if(icod.eq.1)nn=n
        write(0,40)
40      format(2x,"Input of the production cost functions")
        do 13 i=1,nn
        write(0,14)i
14      format(2x,"Number of different determinations for the period",i4,"?")
        read(0,2)m
        write(0,15)
15      format(4x,"Gives the (coefficient of slope,ordinate at the origin)")
        j=1
19      read(0,2)a(j),b(j)
        if(j.eq.1)go to 16
        if((a(j).le.a(j-1)).and.(b(j).ge.b(j-1)))go to 16
        write(0,17)
17      format(2x,"The coefficient of slopes must be given in the decreasing orde
EOP
\cr")
        write(0,18)
18      format(2x,"and the ordinates at the origin in the increasing order")
        go to 19
16      j=j+1
        if(j.le.m)go to 19
        write(52,20,rec=i,err=30)m,(a(k),b(k),k=1,m)
20      format(i2,10e14.7)
13      continue
        write(0,50)
50      format(2x,"Input of the storage cost function") .
        do 53 i=1,nn
        write(0,14)i
        read(0,2)m
        write(0,15)
        j=1
59      read(0,2)a(j),b(j)
        if(j.eq.1)go to 56
        if((a(j).le.a(j-1)).and.(b(j).ge.b(j-1)))go to 56
        write(0,17)
        write(0,18)
        go to 59
56      j=j+1
        if(j.le.m)go to 59
        write(53,20,rec=i,err=32)m,(a(k),b(k),k=1,m)
53      continue
        go to 100
30      write(0,31)
        go to 100
32      write(0,33)
33      format(2x,"Error in writing file 53")
31      format(2x,"Error in writing the file 52")
        close(50)
        close(51)
        close(52)
        close(53)
100     stop
        end
```

SINGH

singh.fortran 04/19/85 1534.5 hfe Fri

```
c Computations of the optimal solutions.
        dimension d(100),y(101),v(100,5),mm(100),w(100),u(100)
        dimension itt(100),tt(100),iuu(100)
        open(50,form="formatted")
        open(51,form="formatted")
        open(52,access="direct",form="formatted",recl=570)
        open(53,access="direct",form="formatted",recl=570)
        open(71,form="formatted")
        read(50,5)n,y0,eps,icod,ind
5       format(i4,2e14.7,i1,i2)
        read(51,12)(d(i),i=1,n)
12      format(100e14.7)
        call lret(d,n,y0,y,v,eps,icod,ind,mm,cc)
        write(71,1)cc
1       format(2x,"Optimal cost",e14.7)
        do 2 i=1,n
2       itt(i)=1
        kk=0
50      x=y0
        y(1)=y0
        do 10 i=1,n
10      iuu(i)=0
        do 3 i=1,n
        if(abs(y(i)-x).lt.eps)go to 4
        y(i+1)=y(i)-d(i)
        tt(i)=0
        go to 6
4       y(i+1)=y(i)-d(i)+v(i,itt(i))
EOP
        tt(i)=v(i,itt(i))
        iuu(i)=1
6       x=x-d(i)
        if(x.lt.0) x=0
3       continue
        kk=kk+1
        write(71,16)kk
16      format(2x,"Optimal solution number:",i4,"stock level")
        do 7 i=1,n
7       write(71,8)i,y(i),tt(i)
8       format(3x,i4,x,e14.7,3x,e14.7)
        do 20 i=n,1,-1
        itt(i)=itt(i)+1
        if(itt(i).le.mm(i))go to 55
        itt(i)=1
        if(i.eq.1) go to 56
        go to 20
55      if(iuu(i).eq.1) go to 50
        itt(i)=1
20      continue
56      write(0,4445)kk
4445    format("    NUMBER OF OPTIMAL SOLUTIONS:",i6)
        close(50)
        close(51)
        close(52)
        close(53)
        close(71)
        stop
        end
```

LRET

lret.fortran 04/19/85 1535.9 hfe Fri

```
      subroutine lret(d,n,y0,y,v,eps,icod,ind,mm,cc)
      dimension d(100),y(101),v(100,5),mm(100),w(100),u(100)
      dimension a(30),b(30)
c * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c          THE FINITE HORIZON PROBLEM-DISCRETE AND DETERMINISTIC CASE         *
c * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
c                 * * * * * * * * * * * * * * * * *
c                 *    The     Backward      Step *
c                 * * * * * * * * * * * * * * * * *
c The whole program is written over 100 periods.
c d:Demands.
c n:Number of periods.
c y0:Initial stock.
c y: Stock levels.
c v:Optimal replenishments.
c eps:Allows to test 0.
c icod:=0,if the problem is stationary
c       =1,if the problem is non-stationary
c ind:Maximum number of optimal replenishments.
c mm:Gives the number of optimal controls permitted.
      dt=0
      do 1 i=1,n
1     dt=dt+d(i)
      if(y0.lt.dt)go to 2
      do 3 i=1,n
      mm(i)=1
3     v(i,1)=0
      cc=0
      y(1)=y0
      do 4 i=1,n
      ij=i
      if(icod.eq.0) ij=1
      read(52,220,rec=ij,err=35)m,(a(k),b(k),k=1,m)
      xx1=cl(a,b,m,i,0.,eps)
      read(53,220,rec=ij,err=36)m,(a(k),b(k),k=1,m)
      xx2=cl(a,b,m,i,y(i),eps)
EOP
      cc=cc+xx1+xx2
4     y(i+1)=y(i)-d(i)
      go to 100
2     y(n+1)=0
      z=y0
      s=d(n)
      w(1)=0
      do 5 i=1,n-1
      z=z-d(i)
5     s=s+d(i)
      kk=1
      do 6 i=n,1,-1
      ij=i
      if(icod.eq.0)ij=1
      m5=1
      if(y0.gt.s)go to 7
      if(z.lt.0)go to 8
      yy=z
      go to 9
8     yy=0
```

```
9       r=-yy
        jj=0
        ii=i
        go to 10
7       v(i,m5)=0
        read(52,220,rec=ij,err=35)m,(a(k),b(k),k=1,m)
        xx1=cl(a,b,m,i,0.,eps)
        x=xx1+w(1)
        yy=z
        if(icod.eq.0)go to 11
        r=-y0
        jj=1
        ii=1
10      do 12 j=ii,n
        r=r+d(j)
        if(r.le.eps)go to 12
        jj=jj+1
        if(jj.gt.kk)go to 11
        read(52,220,rec=ij,err=35)m,(a(k),b(k),k=1,m)
        xx1=cl(a,b,m,i,r,eps)
        xx=xx1+w(jj)
        if(jj.gt.1)go to 13
14      x=xx
        m5=1
15      v(i,m5)=r
        go to 12
13      if(xx.lt.x-eps)go to 14
        if(abs(xx-x).ge.eps)go to 12
        m5=m5+1
        if(m5.le.ind)go to 15
        write(0,16)i
16      format(5x,"Maximum number of optimal controls passed for the period:",i4)
        m5=m5-1
12      continue
11      read(53,220,rec=ij,err=36)m,(a(k),b(k),k=1,m)
        xx2=cl(a,b,m,i,yy,eps)
        u(1)=xx2+x
        mm(i)=m5
        if(i.eq.1)go to 20
        ii=1
        r=0
        do 21 j=i,n
        r=r+d(j)
        if(r.le.yy+eps)go to 21
        ii=ii+1
        jj=ii-1
        if(y0.gt.s-eps)jj=ii
        if(jj.gt.kk)go to 22
        read(52,220,rec=ij,err=35)m,(a(k),b(k),k=1,m)
        xx1=cl(a,b,m,i,0.,eps)
        read(53,220,rec=ij,err=36)m,(a(k),b(k),k=1,m)
        xx2=cl(a,b,m,i,r,eps)
EOP
        u(ii)=xx1+xx2+w(jj)
21      continue
        ii=ii+1
22      kk=ii-1
```

```
      x=u(ii-1)+eps
      do 23 j=ii-1,1,-1
      if(u(j).ge.x)go to 24
      kk=j
      x=u(j)-eps
24    w(j)=u(j)
23    continue
      z=z+d(i-1)
      s=s-d(i)
6     continue
20    cc=u(1)
220   format(i2,10e14.7)
      go to 100
30    format(i2,10e14.7)
37    format(4x,"Error in reading the file 52",i4)
35    write(0,37)ij
      go to 100
36    write(0,38)ij
38    format(4x,"Error in reading the file 53",i4)
100   return
      end
```

CL

cl.fortran 04/19/85 1538.9 hfe Fri

```
      function cl(a,b,m,i,x,eps)
c ************************************************************************
c *      COMPUTATION OF THE PRODUCTION AND INVENTORY COSTS.              *
c ************************************************************************
      dimension a(20),b(20)
c a:Matrix of coefficients of slopes.
c b:Matrix of ordinates at the origin.
c m:Number of different determinations for the period considered(20 is maximum)
c x:Quantity considered.
c i:Period considered.
c eps:Allows to test zero.
      cl=0
      if(x.lt.eps)go to 100
      c=a(1)*x+b(1)
      if(m.eq.1)go to 100
      do 3 j=2,m
      c1=a(j)*x+b(j)
      if(c1.lt.c)c=c1
3     continue
100   cl=c
      return
      end

r 15:38 0.529 55
```

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique