

5376  
1977  
S  
N° d'ordre 656

55376  
1977  
5

# THÈSE

présentée à

l'Université des Sciences et Techniques  
de LILLE I

pour obtenir le titre de

## DOCTEUR DE SPÉCIALITÉ

(Traitement de l'Information)

par

Remi BRISSON

# LA DÉCOMPOSITION BARYCENTRIQUE EN PROGRAMMATION MATHÉMATIQUE

ESSAIS NUMÉRIQUES



Thèse soutenue le 16 septembre 1977, devant la commission d'examen

MM. C. BREZINSKI	Président
J.C. FIOROT	Examineur
P. HUARD	Rapporteur

Ce mémoire est commun aux thèses de D. PACHOLCZYK et R. BRISSON

DOYENS HONORAIRES de l'Ancienne Faculté des Sciences

MM. R. DEFRETIN, H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES des Anciennes Facultés de Droit  
et Sciences Economiques, des Sciences et des Lettres

M. ARNOULT, Mme BEAUJEU, MM. BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, CORSIN, DEHEUVEL  
DEHORS, DION, FAUVEL, FLEURY, P. GERMAIN, HEIM DE BALSAC, HOCQUETTE, KAMPE DE FERIET,  
KOUGANOFF, LAMOTTE, LASSERRE, LELONG, Mme LELONG, MM. LHOMME, LIEBAERT, MARTINOT-LAGARDE  
MAZET, MICHEL, NORMANT, PEREZ, ROIG, ROSEAU, ROUBINE, ROUELLE, SAVART, WATERLOT, WIEMAN,  
ZAMANSKI.

PRESIDENTS HONORAIRES DE L'UNIVERSITE  
DES SCIENCES ET TECHNIQUES DE LILLE

MM. R. DEFRETIN, M. PARREAU.

PRESIDENT DE L'UNIVERSITE  
DES SCIENCES ET TECHNIQUES DE LILLE

M. J. LOMBARD.

PROFESSEURS TITULAIRES

M.	BACCHUS Pierre	Astronomie
M.	BEAUFILS Jean-Pierre	Chimie Physique
M.	BECART Maurice	Physique Atomique et Moléculaire
M.	BILLARD Jean	Physique du Solide
M.	BIAYS Pierre	Géographie
M.	BONNEIAN Pierre	Chimie Appliquée
M.	BONNOT Ernest	Biologie Végétale
M.	BONTE Antoine	Géologie Appliquée
M.	BOUGHON Pierre	Algèbre
M.	BOURIQUET Robert	Biologie Végétale
M.	CELET Paul	Géologie Générale
M.	CONSTANT Eugène	Electronique
M.	DECUYPER Marcel	Géométrie
M.	DELATRE Charles	Géologie Générale
M.	DELHAYE Michel	Chimie Physique
M.	DERCOURT Michel	Géologie Générale
M.	DURCHON Maurice	Biologie Expérimentale
M.	FAURE Robert	Mécanique
M.	FOURET René	Physique du Solide
M.	GABILLARD Robert	Electronique
M.	GLACET Charles	Chimie Organique
M.	GONTIER Gérard	Mécanique
M.	GRUSON Laurent	Algèbre
M.	GUILLAUME Jean	Microbiologie
M.	HEUBEL Joseph	Chimie Minérale
M.	LABLACHE-COMBIER Alain	Chimie Organique
M.	LANSTRAUX Guy	Physique Atomique et Moléculaire
M.	LAVEINE Jean-Pierre	Paléontologie
M.	LEBRUN André	Electronique
M.	LEHMANN Daniel	Géométrie

Mme	LENOBLE Jacqueline	Physique Atomique et Moléculaire
M.	LINDER Robert	Biologie et Physiologie Végétales
M.	LOMBARO Jacques	Sociologie
M.	LOUCHEUX Claude	Chimie Physique
M.	LUCQUIN Michel	Chimie Physique
M.	MAILLET Pierre	Sciences Economiques
M.	MONTARIOL Frédéric	Chimie Appliquée
M.	MONTREUIL Jean	Biochimie
M.	PARREAU Michel	Analyse
M.	POUZET Pierre	Analyse Numérique
M.	PROUVOST Jean	Minéralogie
M.	SALMER Georges	Electronique
M.	SCHILTZ René	Physique Atomique et Moléculaire
Mme	SCHWARTZ Marie-Hélène	Géométrie
M.	SEGUIER Guy	Electrotechnique
M.	TILLIEU Jacques	Physique Théorique
M.	TRIDOT Gabriel	Chimie Appliquée
M.	VIDAL Pierre	Automatique
M.	VIVIER Emile	Biologie Cellulaire
M.	WERTHEIMER Raymond	Physique Atomique et Moléculaire
M.	ZEYTOUNIAN Radyadour	Mécanique

#### PROFESSEURS SANS CHAIRE

M.	BELLET Jean	Physique Atomique et Moléculaire
M.	BODARD Marcel	Biologie Végétale
M.	BOILLET Pierre	Physique Atomique et Moléculaire
M.	BOILLY Bénoni	Biologie Animale
M.	BRIDOUX Michel	Chimie Physique
M.	CAPURON Alfred	Biologie Animale
M.	CORTOIS Jean	Physique Nucléaire et Corpusculaire
M.	DEBOURSE Jean-Pierre	Gestion des entreprises
M.	DEPREZ Gilbert	Physique Théorique
M.	DEVRAINNE Pierre	Chimie Minérale
M.	GOUDMAND Pierre	Chimie Physique
M.	GUILBAULT Pierre	Physiologie Animale
M.	LACOSTE Louis	Biologie Végétale
Mme	LEHMANN Josiane	Analyse
M.	LENTACKER Firmin	Géographie
M.	LOUAGE Francis	Electronique
Mlle	MARQUET Simone	Probabilités
M.	MIGEON Michel	Chimie Physique
M.	MONTTEL Marc	Physique du Solide
M.	PANET Marius	Electrotechnique
M.	RACZY Ladislas	Electronique
M.	ROUSSEAU Jean-Paul	Physiologie Animale
M.	SLIWA Henri	Chimie Organique

#### MAITRES DE CONFERENCES (et chargés d'Enseignement)

M.	ADAM Michel	Sciences Economiques
M.	ANTOINE Philippe	Analyse
M.	BART André	Biologie Animale
M.	BEGUIN Paul	Mécanique
M.	BKOUCHE Rudolphe	Algèbre
M.	BONNELLE Jean-Pierre	Chimie
M.	BONNEMAIN Jean-Louis	Biologie Végétale
M.	BOSCO Denis	Probabilités
M.	BREZINSKI Claude	Analyse Numérique
M.	BRUYELLE Pierre	Géographie

M. CARREZ Christian	Informatique
M. CORDONNIER Vincent	Informatique
M. COQUERY Jean-Marie	Psycho-Physiologie
Mlle DACHARRY Monique	Géographie
M. DEBENEST Jean	Sciences Economiques
M. DEBRABANT Pierre	Géologie Appliquée
M. DE PARIS Jean-Claude	Mathématiques
M. DHAINAUT André	Biologie Animale
M. DELAUNAY Jean-Claude	Sciences Economiques
M. DERIEUX Jean-Claude	Microbiologie
M. DOUKHAN Jean-Claude	Physique du Solide
M. DUBOIS Henri	Physique
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
Mme EVRARD Micheline	Chimie Appliquée
M. FONTAINE Jacques-Marie	Electronique
M. FOURNET Bernard	Biochimie
M. FORELICH Daniel	Chimie Physique
M. GAMBLIN André	Géographie
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GUILLAUME Henri	Sciences Economiques
M. HECTOR Joseph	Géométrie
M. JACOB Gérard	Informatique
M. JOURNEL Gérard	Physique Atomique et Moléculaire
Mlle KOSMAN Yvette	Géométrie
M. KREMBEL Jean	Biochimie
M. LAURENT François	Automatique
Mlle LEGRAND Denise	Algèbre
Mlle LEGRAND Solange	Algèbre
M. LEROY Jean-Marie	Chimie Appliquée
M. LEROY Yves	Electronique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique Théorique
M. LOUCHET Pierre	Sciences de l'Education
M. MACKE Bruno	Physique
M. MAHIEU Jean-Marie	Physique Atomique et Moléculaire
Mme N'GUYEN VAN CHI Régine	Géographie
M. MAIZIERES Christian	Automatique
M. MALAUSSENA Jean-Louis	Sciences Economiques
M. MESSELYN Jean	Physique Atomique et Moléculaire
M. MONTUELLE Bernard	Biologie Appliquée
M. NICOLE Jacques	Chimie Appliquée
M. PAQUET Jacques	Géologie Générale
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie Physique
M. PERROT Pierre	Chimie Appliquée
M. PERTUZON Emile	Physiologie Animale
M. PONSOLLE Louis	Chimie Physique
M. POVY Lucien	Automatique
M. RICHARD Alain	Biologie
M. ROGALSKI Marc	Analyse
M. ROY Jean-Claude	Psycho-Physiologie
M. SIMON Michel	Sociologie
M. SOMME Jean	Géographie
Mlle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. STEEN Jean-Pierre	Informatique

M. THERY Pierre  
M. TOULOTTE Jean-Marc  
M. TREANTON Jean-René  
M. VANDORPE Bernard  
M. VILLETTE Michel  
M. WALLART Francis  
M. WERNIER Georges  
M. WATERLOT Michel  
Mme ZINN-JUSTIN Nicole

Electronique  
Automatique  
Sociologie  
Chimie Minérale  
Mécanique  
Chimie  
Informatique  
Géologie Générale  
Algèbre

Je tiens à exprimer ma profonde gratitude à Monsieur le Professeur BREZINSKI pour l'honneur qu'il me fait de présider le jury.

Je remercie Monsieur FIOROT qui a bien voulu s'intéresser à ce travail et a accepté de le juger.

Que Monsieur le Professeur HUARD, qui m'a donné l'idée de ce travail et m'a permis de le mener à bien par ses précieux conseils et suggestions, veuille bien trouver ici l'expression de ma plus vive reconnaissance.

Je tiens à remercier Monsieur le Professeur DAVID, ainsi que Monsieur GRAND et Monsieur JARLAUD, qui n'ont cessé de me témoigner un amical intérêt et dont la compréhension a facilité la réalisation des essais numériques.

J'adresse enfin mes remerciements pour la qualité de la frappe et la réalisation matérielle à Madame PACHOLCZYK et à Monsieur et Madame DEBOCK dont j'ai pu apprécier la compétence.

TABLE DES MATIERES

XXXXXXXXXXXXXXXXXXXX

# CONSIDERATIONS THEORIQUES

.....

## 0 - INTRODUCTION

1

## I - RAPPELS

I - 1 - Notations	4
I - 2 - Programme mathématique	5
I - 3 - Conditions d'optimalité en programmation mathématique	6
I - 4 - Décomposition barycentrique d'un polyèdre	10
I - 5 - Méthode simpliciale	12

## II - DECOMPOSITION BARYCENTRIQUE - CAS LINEAIRE

II - 1 - Introduction	16
II - 2 - Algorithme de DANTZIG et WOLFE	16
II - 3 - Méthode duale	24
II - 4 - Méthode primale-duale	27

## III - DECOMPOSITION BARYCENTRIQUE - CAS NON LINEAIRE

III - 1 - Introduction	30
III - 2 - Cas quadratique	30
III - 3 - Cas convexe : méthode des colonnes	34
III - 4 - Cas convexe : méthode du barycentre	41
III - 5 - Conclusion	46



IV - ALGORITHME MIXTE

IV - 1 - <i>Sous-optimisation unidimensionnelle</i>	49
IV - 2 - <i>Algorithme de FRANK et WOLFE</i>	51
IV - 3 - <i>Algorithme mixte</i>	53

V - ETUDE D'UN CAS PARTICULIER

V - 1 - <i>Etude d'un cas particulier</i>	64
V - 2 - <i>Paramétrisation du problème P</i>	73
V - 3 - <i>Conclusion</i>	77

VI - DETERMINATION DE SOLUTIONS APPROCHEES

VI - 1 - <i>Paramétrisation</i>	80
VI - 2 - <i>Sous-optimisation unidimensionnelle</i>	80
VI - 3 - <i>Algorithme mixte: résolution approchée du programme auxiliaire</i>	82
VI - 4 - <i>Conséquences</i>	83
VI - 5 - <i>Algorithme mixte: résolution approchée du programme auxiliaire dans un cas particulier</i>	83

## E S S A I S   N U M E R I Q U E S

.....

VII - ALGORITHME MIXTE - METHODE DES COLONNES

VII - 1 - <i>Considérations numériques</i>	89
VII - 2 - <i>Etude comparative</i>	100

## VIII - PARAMETRISATION

VIII - 1 - <i>Introduction</i>	111
VIII - 2 - <i>Evaluation de la taille des problèmes à résoudre</i>	112
VIII - 3 - <i>Paramétrisation</i>	115
VIII - 4 - <i>Variantes de la paramétrisation</i>	117
VIII - 5 - <i>Conclusion</i>	127
<u>ANNEXE 1</u>	128
<u>ANNEXE 2</u>	133
<u>ANNEXE 3</u>	137
<u>BIBLIOGRAPHIE</u>	164

CONSIDERATIONS THEORIQUES  
\*\*\*\*\*

INTRODUCTION  
\*\*\*\*\*

Un programme mathématique de grande taille peut être décomposé en plusieurs sous-programmes de taille plus réduite et de préférence, plus faciles à résoudre. DANTZIG et WOLFE [9] ont réalisé cette décomposition en utilisant la notion de barycentre dans un ensemble convexe. L'introduction de ce principe en programmation mathématique a donné naissance à toute une famille d'algorithmes s'appliquant aux problèmes de type linéaire et de façon plus générale de type convexe. Nous rappellerons aux chapitres II et III les plus importants d'entre eux.

Nous nous intéressons dans ce travail, aux méthodes de décomposition du type "barycentrique". Signalons toutefois qu'il existe d'autres modes de décomposition, n'utilisant pas la notion de barycentre. On peut citer, en particulier, la méthode "des centres" de HUARD [19], la méthode de UZAWA [28], ainsi que celles de BENDERS [4] et ROSEN [26].

En introduisant le principe de décomposition barycentrique dans l'algorithme de FRANK et WOLFE [13], nous définirons au chapitre IV l'algorithme mixte. Des essais numériques ont été réalisés et sa convergence a été comparée à celle de la méthode des colonnes de DANTZIG [8]. Ces résultats figurent au chapitre VII.

Au chapitre V, nous étudierons un cas particulier de programme convexe. Nous verrons que les méthodes de décomposition barycentrique ne peuvent être appliquées directement au problème posé. Un algorithme de paramétrisation sera alors défini, rendant possible leur utilisation. Afin d'en améliorer la convergence numérique, des variantes seront définies. Elles figurent au chapitre VIII ainsi que les résultats numériques correspondants.

Numériquement, il est important de pouvoir résoudre de façon approchée les sous-programmes. Cette étude fera l'objet du chapitre VI.

CHAPITRE I

RAPPELS  
\*\*\*\*\*

I - 1 - NOTATIONS.

$| I |$  Nombre d'éléments d'un ensemble fini d'indices I.

$\mathbb{N}$  Ensemble des entiers naturels.

$\mathbb{R}$  Ensemble des nombres réels.

$\mathbb{R}^n$  Espace vectoriel euclidien de dimension n.

$x \cdot y$  Produit scalaire de  $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ .

$\|x\|$  Norme euclidienne de x,  $\|x\| = \sqrt{x \cdot x}$ .

- Tout vecteur  $x \in \mathbb{R}^n$  est identifié à une matrice colonne dont les éléments sont les composantes de x sur la base canonique de  $\mathbb{R}^n$ .

- Toute forme linéaire  $f \in (\mathbb{R}^n)^*$  [ dual de  $\mathbb{R}^n$  ] est identifiée à la matrice ligne dont les éléments sont les composantes de f par rapport à la base duale de la base canonique de  $\mathbb{R}^n$ .

- A tout  $f \in \mathcal{L}(\mathbb{R}^n, \mathbb{R}^m)$  est associée une matrice M indiquée par  $L \times J$ , avec  $|L| = m$  et  $|J| = n$ .

$M_i$  Ligne i de la matrice M.

$M^j$  Colonne j de la matrice M.

$M_i^j$  Élément en position (i, j) de la matrice M.

$M_I$  Sous-matrice de M comprenant les lignes  $M_i$  avec  $i \in I \subset L$ .

$M^K$  Sous-matrice de M comprenant les colonnes  $M^j$  avec  $j \in K \subset J$ .

$\overline{M}$  Transposée de la matrice M.

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable en  $x^*$ .

$\nabla f(x^*)$  vecteur ligne représentant la valeur du gradient de f calculée au point  $x^*$ .

- Soit  $a : \mathbb{R}^n \rightarrow \mathbb{R}^m$  différentiable en  $x^*$ .

$\nabla a(x^*)$  Matrice jacobienne de a en  $x^*$ .

C'est une matrice de dimension (m, n) de terme général :

$$[\nabla a(x^*)]_i^j = \frac{\partial a_i}{\partial x_j}(x^*)$$

- Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable en  $x^*$ .

$H(x^*)$  Hessien de  $f$  en  $x^*$ . Matrice carrée symétrique de terme

$$\text{général : } \frac{\partial^2 f}{\partial x_i \partial x_j} (x^*) .$$

$T(A, x)$  Cône tangent à  $A$  en  $x$ . (§ I - 3 - 2)

$\Gamma(K)$  Cône polaire ( négatif ) de  $K$ . (§ I - 3 - 3)

### I - 2 - PROGRAMME MATHÉMATIQUE.

- Soit  $A \subset E$  et  $f : E \rightarrow \mathbb{R}$ , le problème suivant :

trouver  $\hat{x}$  tel que  $f(\hat{x}) = \text{Max} \{ f(x) \mid x \in A \}$  est appelé programme mathématique.

On le note symboliquement :

$$\left[ \begin{array}{l} \text{Max } f(x) \\ x \in A \end{array} \right.$$

Dans l'étude qui va suivre on prendra  $E = \mathbb{R}^n$ .

- En général  $A$  est défini par un système de relations :

$$A = \{ x \in \mathbb{R}^n \mid a_i(x) \geq 0, i=1, m \}$$

avec  $a_i : \mathbb{R}^n \rightarrow \mathbb{R}$ .

En introduisant  $a : \mathbb{R}^n \rightarrow \mathbb{R}^m$  de composantes  $a_i, i=1, m$

$$\text{on a : } A = \{ x \in \mathbb{R}^n \mid a(x) \geq 0 \} .$$

Le programme s'écrit alors :

$$\left[ \begin{array}{l} \text{Max } f(x) \\ a(x) \geq 0 \end{array} \right.$$



- Vocabulaire utilisé :

- $f$  Fonction économique.  
 $a_i(x) \geq 0$  Contrainte  $i$ .  
 $x \mid x \in A$  Solution réalisable  
 $A$  Ensemble des solutions réalisables.  
 $\bar{x}$  Solution optimale.  
 $f(\bar{x})$  Valeur du programme mathématique.

- Remarques :

- 1 -  $a_i(x) = 0 \iff \{ a_i(x) \geq 0 \text{ et } -a_i(x) \geq 0 \}$   
- 2 -  $\text{Min} \{ f(x) \mid x \in A \} \iff \text{Max} \{ -f(x) \mid x \in A \}$

I - 3 - CONDITIONS D'OPTIMALITE EN PROGRAMMATION MATHÉMATIQUE.

I - 3 - 1 - Maximum local - Maximum global.

- 1 - On dit que  $x \in A \subset \mathbb{R}^n$  maximise (globalement)  $f$  sur  $A$  si et seulement si :

$$\forall y \in A, \quad f(x) \geq f(y)$$

- 2 - On dit que  $x \in A \subset \mathbb{R}^n$  maximise localement  $f$  sur  $A$  si et seulement s'il existe un voisinage  $V$  de  $x$  tel que  $x$  maximise  $f$  sur  $A \cap V$ .

I - 3 - 2 - Cône tangent.

- 1 - Soient  $A \subset \mathbb{R}^n$  et  $x \in \bar{A}$ . On dit que  $y \in \mathbb{R}^n$  est un vecteur tangent à  $A$  en  $x$  si et seulement s'il existe une suite infinie  $(\lambda_k, x^k) \in \mathbb{R} \times \mathbb{R}^n$  définie par :

$\{ (\lambda_k, x^k) \mid \lambda_k \geq 0, x^k \in A, k \in \mathbb{N} \}$

telle que :

$$- \lim_{k \rightarrow \infty} x^k = x$$

$$- \lim_{k \rightarrow \infty} \lambda_k (x^k - x) = y$$

- 2 - On appelle cône tangent à A en x, l'ensemble des vecteurs tangents à A en x.

On le note  $T(A, x)$ .

### I - 3 - 3 - Cône polaire.

Soit  $K$  un cône de  $\mathbb{R}^n$ , on appelle cône polaire (négatif) de  $K$ , l'ensemble  $\Gamma(K)$  défini par :

$$\Gamma(K) = \{ y \in \mathbb{R}^n \mid \forall x \in K, y \cdot x \leq 0 \}$$

### I - 3 - 4 - Fonction concave.

- 1 - Soient  $C \subset \mathbb{R}^n$  un ensemble convexe et  $f : C \rightarrow \mathbb{R}^m$

$f$  est dite concave si et seulement si :

$\forall (x, y) \in C \times C, \forall \lambda \in [0, 1] :$

$$f[\lambda x + (1 - \lambda)y] \geq \lambda f(x) + (1 - \lambda)f(y)$$

$f$  est dite convexe si  $(-f)$  est concave.

- 2 - Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  concave.

Si  $f$  est différentiable en  $x \in \mathbb{R}^n$  alors :

$$\forall x \in \mathbb{R}^n, f(x) - f(x^*) \leq \nabla f(x^*) \cdot (x - x^*)$$

- 3 - Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois continûment différentiable.

En désignant par  $H(x)$  la matrice des dérivées secondes de  $f$  en  $x$ , on a :

$$f \text{ concave} \iff \forall x \in \mathbb{R}^n, H(x) \text{ semi-définie négative.}$$

I - 3 - 5 - Condition générale d'optimalité.

Soient  $A \subset \mathbb{R}^n$  et  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  différentiable en  $\overset{*}{x} \in A$ .

Soit  $P = \overline{T(A, \overset{*}{x})}$  l'enveloppe convexe fermée de  $T(A, \overset{*}{x})$ .

Dans ces conditions :

- 1 -  $\overset{*}{x}$  maximise localement  $f$  sur  $A$

$$\longrightarrow \forall f(\overset{*}{x}) \in \Gamma(P) = \Gamma(T(A, \overset{*}{x}))$$

- 2 - Si de plus : -  $A$  convexe et

-  $f$  concave alors

$$\forall f(\overset{*}{x}) \in \Gamma(P) \longrightarrow \overset{*}{x} \text{ maximise } f \text{ sur } A.$$

I - 3 - 6 - Conditions de KUHN et TUCKER [ 22 ].

Considérons le programme suivant :

$$\left[ \begin{array}{l} \text{Max } f(x) \\ a(x) \geq 0 \end{array} \right.$$

où  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  et  $a : \mathbb{R}^n \rightarrow \mathbb{R}^m$ .

Posons :

$$A = \{ x \in \mathbb{R}^n \mid a(x) \geq 0 \}$$

$f$  et  $a$  étant différentiables en  $\overset{*}{x} \in A$ , les conditions suivantes :

$$\exists u \in \mathbb{R}^m \text{ tel que : } \left[ \begin{array}{l} - 1 - u \geq 0 \\ - 2 - \nabla f(\overset{*}{x}) + u \cdot \nabla a(\overset{*}{x}) = 0 \\ - 3 - u \cdot a(\overset{*}{x}) = 0 \end{array} \right.$$

sont appelées conditions de KUHN et TUCKER.

I - 3 - 6 - 1 - Conditions nécessaires d'optimalité.

f et a étant supposées différentiables en  $\bar{x} \in A$ .

Posons :

$$E = \{ l \mid a_l(\bar{x}) = 0 \}$$

$$K = \{ y \in \mathbb{R}^n \mid \forall a_l(\bar{x}) \cdot y \geq 0, l \in E \}$$

si  $K = \overline{T(A, \bar{x})}$  , alors :

$\bar{x}$  maximise localement f sur A  $\iff$   $\bar{x}$  vérifie les conditions de KUHN et TUCKER.

I - 3 - 6 - 2 - Conditions suffisantes d'optimalité.

Si  $\left[ \begin{array}{l} - f \text{ et } a \text{ sont différentiables en } \bar{x} \in A, \\ - A \text{ est convexe et} \\ - f \text{ est concave} \end{array} \right.$

alors

$\bar{x}$  vérifie les conditions de KUHN et TUCKER  $\implies$   $\bar{x}$  maximise f sur A.

I - 3 - 6 - 3 - Cas linéaire.

Dans le cas linéaire suivant :

$$P \quad \left[ \begin{array}{l} \text{Max } f \cdot x \\ A \cdot x = a \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

Soit  $\bar{x}$  une solution réalisable de P .

Les C.N.S. d'optimalité s'écrivent :

$\exists (u, v)$  tel que :

$$u \cdot A + v \cdot B + f \leq 0$$

$$(u \cdot A + v \cdot B + f) \cdot \hat{x} = 0$$

I - 3 - 6 - 4 - Cas des programmes convexes

Considérons le programme mathématique :

$$P \quad \begin{cases} \text{Max } f(x) \\ a(x) \geq 0 \\ b(x) \geq 0 \end{cases}$$

Si -  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est concave et différentiable

-  $a : \mathbb{R}^n \rightarrow \mathbb{R}^m$  est concave et différentiable

-  $b : \mathbb{R}^n \rightarrow \mathbb{R}^p$  est affine

Si de plus il existe  $\hat{x}$  tel que :  $a(\hat{x}) > 0$  et  $b(\hat{x}) \geq 0$

Alors :  $\hat{x}$  est solution optimale de P  $\Leftrightarrow$

$\hat{x}$  vérifie les conditions de KUHN et TUCKER.

I - 4 - DECOMPOSITION BARYCENTRIQUE D'UN POLYEDRE [20]

1 - 4 - 1 - Barycentre

- Définition

Soit  $K = \{ 1, 2, \dots, k \}$  un ensemble fini d'indices et

$\{ X^j \mid j \in K \}$  un ensemble de points de  $\mathbb{R}^n$ .

$x$  est appelé barycentre des points  $\{ X^j, j \in K \}$  si et seulement si

$\exists \lambda = (\lambda_1, \dots, \lambda_k) \in \mathbb{R}^k$  avec  $\lambda_j \geq 0$  et  $\sum_{j \in K} \lambda_j = 1$  tel que :

$$x = \sum_{j \in K} x^j \cdot \lambda_j$$

- Propriété

A convexe  $\Leftrightarrow A = \{ x \mid x \text{ barycentre de points de } A \}$

I - 4 - 2 - Polyèdre

- Définition

On appelle polyèdre P de  $\mathbb{R}^n$  toute intersection d'une famille finie de demi-espaces fermés.

On peut donc dire que:

$$\exists B, \exists b \text{ tels que } : P = \{ x \in \mathbb{R}^n \mid B \cdot x \geq b \}.$$

- Remarque

$$B \cdot x = b \Leftrightarrow \{ Bx \geq b \text{ et } -Bx \geq -b \}$$

1 - 4 - 3 - Décomposition barycentrique d'un polyèdre

- Considérons le polyèdre P défini par :

$$P = \{ x \in \mathbb{R}^n \mid B \cdot x = b, x \geq 0 \}$$

- Désignons par :

$\{ X^j, j \in G_0 \}$  l'ensemble des points extrêmes de P

$\{ X^j, j \in G_1 \}$  l'ensemble des directions d'infinitude extrêmes de P

De façon équivalente on a :

1 - Pour tout point x de l'enveloppe convexe des points extrêmes :

$$\exists (\lambda_j \geq 0, j \in G_0) \text{ tels que } \sum_{j \in G_0} \lambda_j = 1 \text{ et } x = \sum_{j \in G_0} X^j \cdot \lambda_j$$

et

2- Pour tout point  $x$  du cône asymptotique :

$$\exists (\lambda_j \geq 0, j \in G_1) \text{ tels que } x = \sum_{j \in G_1} x^j \cdot \lambda_j$$

Comme  $P$  est la somme de l'enveloppe convexe de ses points extrêmes et de son cône asymptotique, en posant :  $G = G_0 + G_1$ , on a donc :

$$x \in P \iff \left[ \begin{array}{l} \exists [ (\lambda_j \geq 0, j \in G) \text{ tels que } \sum_{j \in G_0} \lambda_j = 1 ] \\ \text{et} \\ x = \sum_{j \in G} x^j \cdot \lambda_j \end{array} \right]$$

On appellera générateur extrême de  $P$  tout  $x^j, j \in G$ .

Désignons par :

- $X$  la matrice dont la colonne  $j$  est  $x^j, j \in G$
- $\delta$  le vecteur ligne de composantes  $\delta^j, j \in G$  tel que :

$$\delta^j = \begin{cases} 1 & \text{si } j \in G_0 \\ 0 & \text{si } j \in G_1 \end{cases}$$

- $\lambda$  le vecteur colonne de composantes  $\lambda_j, j \in G$

Alors :

$$x \in P \iff \exists \lambda \text{ tel que } \begin{cases} x = X \cdot \lambda \\ \delta \cdot \lambda = 1 \\ \lambda \geq 0 \end{cases}$$

I - 5 - METHODE SIMPLICIALE [ 8 ], [ 18 ]

Soit à résoudre le programme linéaire suivant :

$$\begin{cases} \text{Max } f \cdot x \\ A \cdot x = a \\ x \geq 0 \end{cases}$$

où A est indicée par L x J

Notations - Définitions

$$I \text{ est une base} \iff \begin{cases} I \subset J \\ |I| = |L| \\ A^I \text{ est inversible} \end{cases}$$

- I étant une base :

$T(I) = (A^I)^{-1} \cdot A$  tableau simplicial

$t(I) = (A^I)^{-1} \cdot a$  second membre du tableau simplicial

$d(I) = f - f^I \cdot T(I)$  vecteur critère de candidature

$$x(I) \text{ tel que } \begin{cases} x(I)_I = t(I) \\ x(I)_{\bar{I}} = 0 \end{cases} \quad \text{solution de base}$$

si de plus  $t(I) \geq 0$  alors  $x(I)$  est une solution de base réalisable et I est une base réalisable.

Théorème

Si pour une base I réalisable on a  $d(I) \leq 0$  alors  $x(I)$  est solution optimale

Principe de la Méthode simpliciale

La méthode simpliciale définit une suite de valeurs  $f \cdot x(I)$  monotone non décroissante, en se déplaçant d'une base I à une base voisine  $I' = I - r + s$  r et s étant définis comme suit :

$d(I) \leq 0$  alors on choisit  $s \in \bar{I}$  tel que  $d^s(I) > 0$



( classiquement  $s$  correspond au  $d^s$  maximum positif)

L'indice  $r$  quittant la base est alors sélectionné de la façon suivante :

$$r : \frac{t_r(I)}{T_r^s(I)} = \text{Min} \left\{ \frac{t_i(I)}{T_i^s(I)} \mid T_i^s(I) > 0, i \in I \right\}$$

On peut noter que,  $t(I)$  et  $s$  étant connus, on utilise uniquement la colonne  $T^s(I)$  du tableau simplicial.

On peut remarquer qu'un changement de base de la méthode simpliciale nécessite de pouvoir connaître à chaque étape :

- $s$  correspondant à un  $d^s > 0$ ,
- la colonne correspondante  $T^s(I)$  du tableau simplicial.

La nouvelle matrice inverse de base et la nouvelle solution de base sont alors calculables. La méthode de décomposition barycentrique de DANTZIG et WOLFE [9] repose sur cette idée.

CHAPITRE II

DECOMPOSITION BARYCENTRIQUE  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CAS LINEAIRE  
XXXXXXXXXXXXXXXXXXXX

## II - 1 - INTRODUCTION

La résolution des programmes linéaires de grande taille a nécessité l'élaboration d'algorithmes spéciaux. L'idée directrice en est, de décomposer le problème initial en plusieurs sous-problèmes de taille plus réduite et, de préférence, plus faciles à résoudre.

Une famille d'algorithmes de décomposition utilise la notion de barycentre dans un ensemble convexe. Dans le cas des programmes linéaires, on effectue une décomposition barycentrique d'une partie du domaine comme on l'a déjà indiqué au § I - 4.

DANTZIG et WOLFE [10], les premiers, ont élaboré un algorithme reposant sur la méthode simpliciale. ABADIE et WILLIAMS [1] utilisent par contre la méthode duale-simpliciale. La méthode primale-duale est à la base de l'algorithme proposé par BALAS [2]. Signalons encore WILLIAMS [32], qui a résolu par décomposition barycentrique le problème du transport simple. Citons enfin les méthodes définies par OBEL [24] et BEN-ISRAEL et ROBERS [3].

Dans ce chapitre, nous développerons la méthode primale (§ II - 2) et nous donnerons les grandes lignes des méthodes duale (§ II - 3) et primale-duale (§ II - 4). Ces rappels nous permettront de mettre en évidence le principe de décomposition et d'en montrer l'intérêt pour le traitement de certains problèmes.

## II - 2 - ALGORITHME DE DANTZIG ET WOLFE [9] [10]

### II - 2 - 1 - Problème posé

Considérons un programme linéaire donné sous la forme suivante:

$$P_1 \left[ \begin{array}{l} \text{Max } f \cdot x \\ A \cdot x = a \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

où

A est une matrice indicée par  $L \times J$

B " " " " "  $L' \times J$

a est un vecteur colonne indicé par L

b " " " " " "  $L'$

x " " " " " " J

f est un vecteur ligne indicé par J

Posons  $B = \{ x \mid B \cdot x = b, x \geq 0 \}$

En appliquant les résultats du §I-4 au polyèdre  $B$ ,  $P_1$  est alors équivalent au programme linéaire suivant :

$$P_2 \left[ \begin{array}{l} \text{Max } f \cdot X \cdot \lambda \\ A \cdot X \cdot \lambda = a \\ \delta \cdot \lambda = 1 \\ \lambda \geq 0 \end{array} \right.$$

où X et  $\delta$  sont supposés connus.

A une solution optimale  $\hat{\lambda}$  de  $P_2$  correspond une solution optimale

$\hat{x} = X \cdot \hat{\lambda}$  de  $P_1$

- Remarques :

1 - On réalise un gain en "lignes". En effet, alors que  $P_1$  comprenait  $|L| + |L'|$  contraintes,  $P_2$  n'en contient que  $|L| + 1$ .

2 - Toutefois, une difficulté apparaît a priori pour la résolution de  $P_2$  ;

X et  $\delta$  ne sont pas explicitement connus.

II - 2 - 2 - Principe.

On se propose de résoudre  $P_2$  par la méthode simpliciale. Soit I une base réalisable de  $P_2$ . On peut calculer la matrice de base et la solution de base correspondante. Les variables duales associées sont alors déterminées par la formule suivante :

$$w(I) = [u(I), u^0(I)] = f \cdot X^I \cdot \begin{bmatrix} A \cdot X^I \\ \delta^I \end{bmatrix}^{-1}$$

Par contre le calcul du critère de candidature fait intervenir la matrice X des générateurs extrêmes de B. En effet :

$$d(I) = [f - u(I) \cdot A] \cdot X - u^0(I) \cdot \delta$$

Nous avons vu, au § I - 5, qu'un changement de base peut être effectué, si l'on connaît une composante positive du critère de candidature et la colonne correspondante du tableau simplicial. DANTZIG et WOLFE proposent de les obtenir en résolvant le programme auxiliaire suivant :

$$P_3 \quad \begin{cases} \text{Max } [f - u(I) \cdot A] \cdot x \\ B \cdot x = b \\ x \geq 0 \end{cases}$$

$P_3$  étant résolu par la méthode simpliciale, on obtiendra un générateur extrême  $X^S$  et  $d^S = [f - u(I) \cdot A] \cdot X^S - u^0(I) \cdot \delta^S$ .

A l'optimum deux cas sont possibles :

cas 1 : Optimum fini

Soit  $X^S$  la solution optimale trouvée. Dans ce cas, on a :

$i * [f - u(I) \cdot A] \cdot X^S - u^0(I) > 0$  - Alors s est candidat à entrer dans la base.

ou

ii - [ f - u (I) . A ] . X<sup>s</sup> - u° (I) ≤ 0 - Alors la base I est optimale.

Cas 2 : Optimum infini.

Soit X<sup>s</sup> la direction d'infinitude trouvée. On a alors :

[ f - u (I) . A ] . X<sup>s</sup> > 0 et s est candidat à entrer dans la base.

La résolution de P<sub>3</sub> nous conduit donc à l'un des deux résultats suivants:

1 - La solution de base réalisable λ (I) de P<sub>2</sub> est optimale

On a alors la solution optimale  $\hat{x}$  de P<sub>1</sub> :

$$\hat{x} = X^I \cdot \begin{bmatrix} A & X^I \\ & I \\ & \delta \end{bmatrix}^{-1} \cdot \begin{bmatrix} a \\ 1 \end{bmatrix}$$

2 - Obtention d'un candidat s permettant d'effectuer le changement de base à l'aide de la colonne dite candidate :

$$\begin{bmatrix} A \cdot X^s \\ \delta^s \end{bmatrix}$$

En effet, on peut alors déterminer l'indice r quittant la base.

- Remarque :

Une base initiale peut être obtenue en effectuant une phase I sur le problème P<sub>2</sub>.

II - 2 - 3 - Algorithme :

a - Détermination d'une base réalisable de P<sub>2</sub>.

b - I étant une base réalisable de P<sub>2</sub>

i - Détermination des variables duales w (I) = [ u (I), u° (I) ]

ii - Résolution de  $P_3$  - obtention de la colonne candidate

$$\begin{bmatrix} A \cdot X^s \\ \delta^s \end{bmatrix}$$

Deux cas :

- $d^s(I) \leq 0$  . FIN
- $d^s(I) > 0$  . Changement de base - Retour à b

II - 2 - 4 - Convergence :

$P_3$  étant résolu par la méthode simpliciale, à chaque étape, on obtient un générateur de  $B$  . Leur nombre étant fini, moyennant l'hypothèse de non cyclage, la convergence est assurée en un nombre fini d'étapes .

- Remarques :

1 - L'optimisation complète de  $P_3$  n'est pas nécessaire. Il suffit de déterminer  $X^s$  tel que  $d^s(I) > 0$ .

2 - On peut noter que, dans le cas où  $B$  est borné, on a l'encadrement suivant :

$$0 \leq f \cdot \hat{x} - f \cdot x(I) \leq d^s$$

Ce type d'encadrement joue un rôle primordial dans la décomposition barycentrique des programmes non linéaires.

II - 2 - 5 - Applications :

II - 2 - 5 - 1-Décomposition des systèmes angulaires

Il peut arriver que la matrice  $B$  se décompose en sous-matrices parmi lesquelles les seules qui contiennent des termes non nuls sont situées sur une





$$[f^1 - u(I) \cdot A_1] \cdot x^1 + \dots + [f^h - u(I) \cdot A_h] \cdot x^h - u^*(I) \cdot \delta^s > 0$$

b - A la matrice A on associe les h matrices B<sub>i</sub>

Désignons par X<sub>i</sub> la matrice des générateurs de:

$$B_i = \{ x^i \mid B_i \cdot x^i = b^i, x^i \geq 0 \}$$

P2 s'écrit alors :

$$\left[ \begin{array}{l} \text{Max } f^1 \cdot X_1 \cdot \lambda^1 + \dots + f^h \cdot X_h \cdot \lambda^h \\ A_1 \cdot X_1 \cdot \lambda^1 + \dots + A_h \cdot X_h \cdot \lambda^h = a \\ \delta_1 \cdot \lambda^1 = 1 \\ \dots \\ \delta_h \cdot \lambda^h = 1 \\ \lambda^i \geq 0, i = 1, h \end{array} \right.$$

Les h sous-programmes P3 sont encore définis par :

$$\left[ \begin{array}{l} \text{Max } [f^i - u(I) \cdot A_i] \cdot x^i \\ B_i \cdot x^i = b^i \quad i = 1, h \\ x^i \geq 0 \end{array} \right.$$

x<sup>i</sup> est candidat si :

$$[f^i - u(I) \cdot A_i] \cdot x^i - u_i^*(I) \cdot \delta^s > 0$$

Remarque:

Un exemple de ce type est fourni par le problème de transport simple .

II - 2 - 5 - 2 - Paramétrisation de la fonction économique [ 14 ]

GAUTHIER a appliqué cet algorithme pour résoudre le problème paramétrique suivant :

$$\left[ \begin{array}{l}
 \text{Pour chaque valeur de } \theta : \\
 \\
 \text{Max } [ f1 + \theta \cdot f2 ] \cdot x \\
 \\
 A \cdot x = a \\
 \\
 B \cdot x = b \\
 \\
 x \geq 0
 \end{array} \right.$$

II - 2 - 5 - 3 - Problème du transport simple [ 32 ]

Le problème du transport simple peut se formuler de la façon suivante:

$$\left[ \begin{array}{l}
 \text{Min } \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \\
 \\
 \sum_{j=1}^n x_{ij} = S_i \quad i = 1, m \quad (1) \\
 \\
 \sum_{i=1}^m x_{ij} = D_j \quad j = 1, n \quad (2) \\
 \\
 x_{ij} \geq 0 \quad (3) \\
 \\
 \text{avec } \sum_{i=1}^m S_i = \sum_{j=1}^n D_j
 \end{array} \right.$$

WILLIAMS applique le principe de décomposition en utilisant les contraintes (2) et (3) .

Le problème se ramène alors à :

$$\left[ \begin{array}{l} \text{Min } \sum_{l=1}^p f^l \lambda_l \\ \sum_{l=1}^p s_i^l \lambda_l = S_i \quad i = 1, m \\ \lambda_l \geq 0 \end{array} \right.$$

avec  $X^1 = [X_{ij}^1]$  générateur de (2) + (3)

$$s_i^1 = \sum_{j=1}^n X_{ij}^1$$

L'obtention du candidat se fait par un calcul très simple pour lequel la grille des coûts n'est examinée qu'en colonnes.

## II - 3 - METHODE DUALE [ 1 ]

### II - 3 - 1 - Principe

Elle repose essentiellement sur la méthode duale-simpliciale. On résout le problème sous sa forme équivalente P2. On a besoin de connaître une base I "duale optimale" [c'est-à-dire  $d(I) \leq 0$ ] ainsi que :

a - l'inverse de la matrice de base

b -  $\lambda(I)$ ,  $w(I) = [u(I), u^*(I)]$

Supposons qu'il existe  $r \mid \lambda_r(I) < 0$

Désignons par  $[\alpha_i(I), \alpha_i^0(I)]$  la  $i^{\text{ème}}$  ligne de l'inverse de la matrice de base.

Les problèmes à résoudre sont alors les suivants :

1 - Recherche d'un pivot négatif, soit ici :

$$X^j \text{ tel que } \alpha_r(I) \cdot A \cdot X^j + \alpha_r^0(I) \cdot \delta^j < 0$$

2 - Recherche de la colonne candidature. Soit

$$\text{Min} \left\{ \frac{[f - u(I) \cdot A] \cdot X^j - u_0(I) \cdot \delta^j}{\alpha_r(I) \cdot A \cdot X^j + \alpha_r^0(I) \cdot \delta^j} \mid \alpha_r(I) \cdot A \cdot X^j + \alpha_r^0(I) \cdot \delta^j < 0 \right\}$$

La base I étant "duale optimale" les rapports sont tous, dans ce cas, positifs ou nuls.

Remarque

Les mêmes problèmes se posent qu'au paragraphe II - 2 . Une difficulté supplémentaire apparaît : dans la recherche de la colonne candidate, la fonction intervenant est fractionnaire. Toutefois, l'algorithme qui suit ne prend en compte que les programmes linéaires.

Pour simplifier l'exposé, on supposera B borné.

II - 3 - 2 - Algorithme partiel de recherche de s

1) On résoud d'abord

$$\begin{array}{l}
 P4 \quad \left[ \begin{array}{l}
 \text{Min } \alpha_r(I) \cdot A \cdot x \\
 B \cdot x = b \\
 x \geq 0
 \end{array} \right.
 \end{array}$$

En désignant par  $X^t$  le point optimal trouvé on a :

- Soit  $\alpha_r(I) \cdot A \cdot X^t + \alpha_r^0(I) \geq 0$ , le problème initial n'a pas de solution.

- Soit  $\alpha_r(I) \cdot A \cdot X^t + \alpha_r^0(I) < 0$

---

Posons :  $v_t = h [X^t] = \frac{[f - u(I) \cdot A] X^t - u_0(I)}{\alpha_r(I) A \cdot X^t + \alpha_r^0(I)}$

Alors :

- si  $v_t = 0$ ,  $X^s = X^t$  est le générateur candidat

- si  $v_t > 0$  aller à 2

2 - On résoud le programme linéaire suivant :

P5  $\left\{ \begin{array}{l} \text{Min } v_t [ \alpha_r(I) \cdot Ax + \alpha_r^0(I) ] - [ (f - u(I) \cdot A) \cdot x - u_0(I) ] \\ Bx = b \\ x \geq 0 \end{array} \right.$

Désignons par  $X^j$  la solution optimale trouvée.

Deux cas sont alors possibles :

Cas 1 :

$$v_t [ \alpha_r(I) \cdot A \cdot X^j + \alpha_r^0(I) ] - [ (f - u(I) \cdot A) \cdot X^j - u_0(I) ] < 0$$


---

On prend :  $X^{t+1} = X^j$

$$v_{t+1} = h [ X^{t+1} ]$$

- Si  $v_{t+1} = 0$ ,  $X^s = X^{t+1}$  est le générateur candidat

- Si  $v_{t+1} > 0$  retour à 2 avec  $t = t+1$

Cas 2 :

$$v_t [ \alpha_r(I) \cdot A \cdot X^j + \alpha_r^0(I) ] - [ (f - u(I) \cdot A) X^j - u_0(I) ] = 0$$


---

Dans ce cas  $X^s = X^t$  est le candidat cherché.

- La suite  $v_t$  étant strictement décroissante, l'algorithme est fini si on résoud P4 et P5 par la méthode simpliciale (moyennant l'hypothèse de non-cyclage)

- Remarques :

1 - Dans la méthode primale, la colonne candidate est obtenue en résolvant un seul sous-programme P3. Par contre, dans la méthode duale, on a à résoudre une séquence (finie) de programmes linéaires P5 en plus de la résolution de P4.

2 - On peut noter que cet algorithme est utilisable lorsqu'il s'agit de résoudre des programmes linéaires paramétriques (fonction économique ou second membre), pour lesquels des problèmes du même type se posent.

II - 4 - METHODE PRIMALE-DUALE [ 2 ]

Nous nous contenterons de donner l'idée directrice de la méthode. A chaque itération, on résoud d'abord le sous-programme P'3 (u) suivant (avec initialement  $u = 0$ ) :

$$P'3(u) \quad \left[ \begin{array}{l} \text{Max } [ f - u \cdot A ] \cdot x \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

On définit alors un programme directeur P'2 dans lequel  $X^K$  est une matrice dont les colonnes  $j$  sont des points  $X^j$  solutions optimales de P'3 (u) .

$$P'2(u) \quad \left[ \begin{array}{l} \text{Max} - \sum_i (\mu_i^1 + \mu_i^2) \\ A \cdot X^K \cdot \lambda_K + \mu^1 - \mu^2 = a \\ \delta^K \cdot \lambda_K = 1 \\ \lambda_K \geq 0, \mu^1 \geq 0, \mu^2 \geq 0 \end{array} \right.$$

Remarque

- Cette méthode diffère des précédentes à 2 points de vue :

a)  $u$  n'est pas défini comme multiplicateur de KUHN et TUCKER relatif au programme principal P'2.

b) Les solutions optimales de P'2 ( $u$ ) ne donnent pas en général des points  $x$  réalisables de P1.

On réajuste la valeur du paramètre  $u$  s'il y a lieu.

$$\text{Posons } Z(u) = + \sum_i (\mu_i^1 + \mu_i^2)$$

Il est montré que  $Z(u)$  est décroissant. De plus la solution optimale est atteinte quand  $Z(u) = 0$ .

Si P1 a des solutions réalisables, l'algorithme converge moyennant l'hypothèse de non-cyclage.

CHAPITRE III

DECOMPOSITION BARYCENTRIQUE  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CAS NON LINEAIRE  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX



### III - 1 - INTRODUCTION

Toute une famille d'algorithmes a été obtenue en introduisant, dans les programmes mathématiques non linéaires, le principe de décomposition barycentrique. CHANDRA [ 5 ] a étudié les programmes de type fractionnaire en utilisant l'algorithme de KANTI SWARUP [21] . Le cas des programmes quadratiques a été abordé par WHINSTON ( [29] , [30] ). Des variantes de la méthode simpli-  
ciale sont alors adaptées, pour résoudre les nouveaux problèmes posés.

Dans le cas général, pour les programmes dits convexes, des algorithmes ont été définis par DANTZIG [8] puis HUARD [20] . Une décomposition barycentrique partielle du domaine est réalisée. Les conditions de KUHN et TUCKER [22] permettent alors " d'agrandir " cet ensemble à chaque étape. Citons également les travaux de GEOFFRION [12] .

Afin de préciser cette généralisation, nous présenterons dans ce chapitre:

- Un algorithme de programmation quadratique ( § III - 2 )
- La méthode des colonnes de DANTZIG ( § III - 3 )
- La méthode du barycentre de HUARD ( § III - 4 )

Ces rappels permettront de situer " l'algorithme mixte ", que nous allons définir au § IV - 3 . Nous introduirons de la même façon , le principe de décomposition barycentrique dans l'algorithme de FRANK et WOLFE [13] .

### III - 2 - CAS QUADRATIQUE ( [29] , [30] )

#### III - 2 - 1 - Problème posé

Considérons le programme quadratique suivant:

$$P1 \quad \left[ \begin{array}{l} \text{Max } \bar{p} \cdot x - \frac{1}{2} \bar{x} \cdot Q \cdot x \\ A \cdot x = a \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

où Q est une matrice symétrique semi-définie positive.

B = { x ∈ ℝ^n | B . x = b , x ≥ 0 } est borné et non vide.

En introduisant une décomposition barycentrique de B , P1 est équivalent au problème suivant :

$$\begin{array}{l}
 \text{P2} \\
 \left[ \begin{array}{l}
 \text{Max } \bar{p} \cdot X \cdot \lambda - \frac{1}{2} \bar{\lambda} \cdot \bar{X} \cdot Q \cdot X \cdot \lambda \\
 A \cdot X \cdot \lambda = a \quad (1) \\
 \delta \cdot \lambda = 1 \quad (2) \\
 \lambda \geq 0 \quad (3)
 \end{array} \right.
 \end{array}$$

On résoud alors P2 en utilisant l'algorithme de WHINSTON et VAN DE PANNE [31] . Il s'agit d'une variante de la méthode simpliciale adaptée aux programmes quadratiques [29] ( une méthode duale est abordée dans [30] ) .

III - 2 - 2 - Conditions de KUHN et TUCKER relatives à P2 .

Soit λ une solution réalisable de P2 . Les conditions nécessaires et suffisantes d'optimalité relatives à P2 s'écrivent :

$$\exists ( v , v^{\circ} ) \text{ et } \exists u = ( u_k | k \in G_0 ) \text{ tels que } \forall k \in G_0 :$$

$$\bar{p} \cdot X^k - \bar{\lambda}^k \cdot Q \cdot X \cdot \lambda - \bar{v} \cdot A \cdot X^k - v^{\circ} + u_k = 0 \quad (4)$$

$$u_k \cdot \lambda_k = 0 \quad (5)$$

$$u_k \geq 0 \quad (6)$$

On cherche ( λ , u , v , v° ) satisfaisant les conditions de (1) à (6) . L'algorithme progresse par des changements de base de type simplex de telle sorte que:

a - Les conditions de (1) à (4) soient vérifiées

b - La fonction économique soit croissante

Une solution de base est de type standard lorsqu'on a :

$\lambda_k$  variable de base  $\longrightarrow$   $u_k$  variable hors base et vice versa.

Remarque :

On retrouve le même problème que précédemment, X et  $\delta$  n'étant pas explicitement connus.

### III - 2 - 3 - ALGORITHME

On initialise l'algorithme avec une solution de base réalisable de type standard. On va à l'étape 1.

- Etape 1:

On détermine le minimum des  $u_k$ . Soit  $u_{k_1}$  cette valeur.

Deux cas sont alors possibles :

.  $u_{k_1} \geq 0$  . On est à l'optimum

.  $u_{k_1} < 0$  . Aller à l'étape 2.

- Etape 2:

On introduit la variable  $\lambda_{k_1}$  dans la base. La variable quittant la base est choisie parmi les  $\lambda_k$  de base et  $u_{k_1}$  ; Deux cas sont alors possibles :

.  $u_{k_1}$  quitte la base - Retour à l'étape 1

. sinon  $\lambda_{k_2}$  quitte la base. Aller à l'étape 3.

- Etape 3:

On introduit  $u_{k_2}$  dans la base. La variable quittant la base est choisie

parmi les  $\lambda_k$  de base et  $u_{k1}$ .

Deux cas sont alors possibles :

- .  $\lambda_k$  quitte la base. Répéter l'étape 3.
- .  $u_{k1}$  quitte la base. Aller à l'étape 1.

Remarques :

- 1 - Il est montré, qu'en se donnant  $|L| + 1$  variables  $\lambda_k \geq 0$ , on peut en principe, déterminer les valeurs initiales de  $v$  et  $v^0$ .
  - 2 - On revient toujours à l'étape 1 avec un tableau de type standard.
  - 3 -  $X \cdot \lambda$  est déterminé par les variables  $\lambda_k$  de base.
  - 4 - L'algorithme exige de connaître le minimum des  $\{ u_k, k \in G_0 \}$ , soit  $u_{k1}$ .
- Il suffit pour cela, de résoudre le programme linéaire suivant :

$$\left[ \begin{array}{l} \text{Max } \bar{p} \cdot x - \bar{x} \cdot Q \cdot X \cdot \lambda - \bar{v} \cdot A \cdot x - v^0 \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

En résolvant le problème par la méthode simpliciale, on obtient à la fois,  $u_{k1}$  et  $X^{k1}$ .

- 5 - Enfin, on peut vérifier que la colonne candidate est explicitement connue. On peut donc dire que l'algorithme peut se dérouler, sans que l'on ait à déterminer tous les points extrêmes de  $B$ .

Il est montré que, moyennant l'hypothèse de non cyclage, l'algorithme est fini.

III - 2 - 4 - Conclusion.

Le principe de décomposition barycentrique est donc applicable aux programmes quadratiques. De la même façon, l'étude du problème a été ramenée à la résolution de programmes linéaires de taille plus réduite, définis à partir des conditions de KUHN et TUCKER.

Il est à noter que, si la partie quadratique est nulle, l'algorithme se réduit à celui de DANTZIG et WOLFE.

III - 3 - CAS CONVEXE - METHODE DES COLONNES [8]

III - 3 - 1 - Problème traité - hypothèses

DANTZIG a généralisé aux programmes dits convexes, le principe de décomposition barycentrique.

Considérons en effet le problème suivant :

$$P1 \quad \begin{cases} \text{Max } f(x) \\ a(x) \geq 0 \\ b(x) \geq 0 \end{cases}$$

Posons :  $A = \{ x \mid a(x) \geq 0 \}$

$B = \{ x \mid b(x) \geq 0 \}$

Prenons, en outre, comme hypothèses initiales :

H1 -  $A \cap B \neq \emptyset$

H2 -  $B$  compact

H3 -  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  concave sur  $\mathbb{R}^n$

a :  $\mathbb{R}^n \rightarrow \mathbb{R}^p$  concave sur  $\mathbb{R}^n$

b :  $\mathbb{R}^n \rightarrow \mathbb{R}^q$  concave sur  $\mathbb{R}^n$

Remarque :

D'après l'hypothèse H3, f, a et b sont continues sur  $\mathbb{R}^n$ .

A chaque itération, après avoir déterminé une solution approchée de P1, on cherche un nouveau générateur  $X^S$ . Le programme directeur aura un nombre croissant de colonnes tout en restant linéaire. Par contre, les sous-programmes de recherche de  $X^S$ , définis à partir des conditions de KUHN et TUCKER, ne sont pas linéaires.

III - 3 - 2 - Notations

Désignons par :

- K un ensemble fini d'indices.
- $X^j$ ,  $j \in K$  un ensemble de points de B

Posons :

- $X^K$  matrice dont la colonne j est  $X^j$
- $f^K$  vecteur ligne tel que  $f^j = f(X^j)$
- $e^K$  vecteur ligne tel que  $\forall j \in K, e^j = 1$
- $A^K$  matrice dont la colonne j est  $A^j = a(X^j)$

On sait que, x est barycentre des  $\{ X^j \mid j \in K \}$  si et seulement si :

$\exists \lambda_K$  tel que :  $x = X^K \cdot \lambda_K$  avec  $e^K \cdot \lambda_K = 1$  et  $\lambda_K \geq 0$

III - 3 - 3 - Programme directeur :

On remplace alors P1 par un programme linéaire approché, noté P2 (K).

$$P2 (K) \quad \left[ \begin{array}{l} \text{Max } f^K \cdot \lambda_K \\ A^K \cdot \lambda_K \geq 0 \\ e^K \cdot \lambda_K = 1 \\ \lambda_K \geq 0 \end{array} \right.$$

On désignera par  $\lambda(K)$  une solution optimale de P2 (K) et par  $x(K)$  =  $x^K \cdot \lambda(K)$  le point associé.

Remarques :

1 - L'ensemble des générateurs de P2 (K) est initialement choisi de telle sorte que l'ensemble des solutions réalisables soit non vide.

2 - P2 (K) est une approximation de P1 par défaut :

En effet :

- Posons :

$$B (K) = \{ x \mid x = x^K \cdot \lambda_K \text{ avec } A^K \cdot \lambda_K \geq 0, e^K \cdot \lambda_K = 1, \lambda_K \geq 0 \}$$

- Du fait de la concavité de a et b, on a :

$$B (K) \subset A \cap B$$

- En outre, f étant aussi concave :

$$f^K \cdot \lambda_K \leq f [ x (K) ] \leq f (\hat{x})$$

où  $\hat{x}$  désigne une solution optimale de P1.

III - 3 - 4 - Conditions de KUHN et TUCKER relatives à P2 (K)

A l'optimum,  $\lambda(K)$  est liée à la variable duale  $w(K) = [u(K), u^*(K)]$  par les relations suivantes :

- (1)  $u(K) \geq 0$
- (2)  $u(K) \cdot A^K + f^K + u^*(K) \cdot e^K \leq 0$
- (3)  $u(K) \cdot A^K \cdot \lambda(K) = 0$
- (4)  $f^K \cdot \lambda(K) = -u^*(K)$

Il est à noter que, si P2 (K) est résolu par la méthode simpliciale,  $w(K)$  est fourni avec la solution optimale  $\lambda(K)$ . DANTZIG introduit alors, comme nouveau générateur de P2 (K), le point de B donnant la valeur maximale de :  $u(K) \cdot a(x) + f(x) + u^*(K)$ .

III-3-5- Programme auxiliaire

La variable duale  $u(K)$  étant considérée comme constante, on définit le sous-programme suivant :

$$P3(K) \quad \left[ \begin{array}{l} \text{Max } f(x) + u(K) \cdot a(x) \\ b(x) \geq 0 \end{array} \right.$$

Désignons par  $X^B$  une solution optimale de P3 (K) .

Posons :

$$A^B = a(X^B) ; \quad f^B = f(X^B) ; \quad e^B = 1 .$$

Notons enfin :

$$d^B = f^B + u(K) \cdot A^B + u^*(K) .$$



Remarques :

- 1 - P3 (K) ne comporte que q contraintes alors que P1 en comprenait (p + q).  
Comme dans le cas linéaire, on réalise un gain en lignes sur la taille du problème à résoudre.
- 2 - Toutefois, P3 (K) reste un problème entièrement non linéaire.

III - 3 - 6 - Encadrement de l'optimum :

Par définition de  $X^S$  on a :

$$f(\bar{x}) + u(K) \cdot a(\bar{x}) + u^0(K) \leq d^S$$

Comme  $u(K) \cdot a(\bar{x}) \geq 0$ , on déduit que :

$$f(\bar{x}) + u^0(K) \leq d^S$$

$$f(\bar{x}) - f^K \cdot \lambda(K) \leq d^S \quad (\text{KUHN et TUCKER})$$

$$\text{Or } f^K \cdot \lambda(K) \leq f[x(K)] \quad (f \text{ concave})$$

$$\longleftarrow \quad 0 \leq f(\bar{x}) - f[x(K)] \leq d^S$$

On en déduit que :

- (a)  $d^S \geq 0$
- (b)  $d^S = 0 \longrightarrow x(K)$  solution optimale de P1.

Remarque : Cet encadrement est intéressant sur le plan pratique. On en trouvera une étude au chapitre VII.

III - 3 - 7 - Algorithme.

Etant donné un ensemble fini d'indices K et  $\{x^j \mid j \in K, x^j \in B\}$   
les générateurs correspondants étant bien choisis (remarque 1 du III - 3 - 3) :

- 1 - Optimisation de P2 (K).
- 2 - Détermination de  $w$  (K) associé.
- 3 - Résolution de P3 (K) à l'aide de  $u$  (K) et calcul de  $d^s$ .

Deux cas sont alors possibles :

- (a)  $d^s = 0$  alors  $x$  (K) est solution optimale de P1.
- (b)  $d^s > 0$  alors on agrandit l'ensemble d'indices:  $K' = K + s$ .  
Retour en 1 avec  $K = K'$ .

Remarque:

L'algorithme est tel que la suite des valeurs  $\{ f[x(K)] \}$  est monotone et non décroissante.

III - 3 - 8 - Proposition

Comme  $K$  augmente d'un élément à chaque étape, on peut noter:

$$K = \{1, 2, \dots, k\} \quad \text{et} \quad s = k + 1.$$

Par ailleurs, notons:

$$\overset{k}{x} = x(K); \quad \overset{k}{u} = u(K); \quad \overset{k}{u}^o = u^o(K); \quad \overset{k}{f} = f[x(K)].$$

$$\overset{k+1}{d} (K) = f(\overset{k+1}{x}) + \overset{k}{u} \cdot a(\overset{k+1}{x}) + \overset{k}{u}^o.$$

Désignons par  $S$  la suite des itérations.

- Supposons que,  $\forall k \in S, \overset{k+1}{d} (K) > 0$ . L'algorithme est alors infini.

Ajoutons l'hypothèse supplémentaire suivante :

$$H4 - \forall k \in S, (\overset{k}{u}, \overset{k}{u}^o) \in U \text{ compact}$$

$$\text{On a } \forall k \in S, (\overset{k+1}{x}, \overset{k}{u}, \overset{k}{u}^o) \in B \times U \text{ compact.}$$

Soit  $(\tilde{x}, \tilde{u}, \tilde{u}^o)$  un point d'accumulation de cette suite. On peut extraire de  $S$  une sous-suite infinie  $\tilde{S}$  telle que :

$$(X^{k+1}, \bar{u}^k, \bar{u}^{k*}) \rightarrow (\bar{X}, \bar{u}, \bar{u}^*) \in \mathcal{B} \times \mathcal{U}$$

$$k \rightarrow \infty$$

$$k \in \mathcal{S}$$

Or

$$\forall k \in \mathcal{S}, \forall d^{k+1}(K) = f(X^{k+1}) + \bar{u}^k \cdot a(X^{k+1}) + \bar{u}^{k*} > 0 \text{ (par hypothèse)}$$

et

$$\forall k \in \mathcal{S}, \forall k' > k, \forall d^{k+1}(K') = f(X^{k+1}) + \bar{u}^{k'} \cdot a(X^{k+1}) + \bar{u}^{k'*} \leq 0$$

$f$  et  $a$  étant continues sur  $\mathbb{R}^n$ , on en déduit en passant à la limite que :

$$f(\bar{X}) + \bar{u} \cdot a(\bar{X}) + \bar{u}^* = 0$$

Comme la suite des valeurs  $f$  est monotone non décroissante, on a :

$$\lim_k f = f(\bar{x})$$

$$k \rightarrow \infty$$

$$k \in \mathcal{S}$$

### Proposition

Avec les notations précédentes et sous les hypothèses H1, H2, H3 et H4 si à une étape on a  $d^{k+1}(K) = 0$  alors  $\bar{x}^k$  est solution optimale du problème P1.

Si cette éventualité ne se produit jamais, la suite infinie  $\{\bar{x}^k \mid k \in \mathcal{S}\}$  fournie par l'algorithme est telle que :

$$\lim_k f(\bar{x}^k) = f(\bar{x})$$

$$k \rightarrow \infty$$

$$k \in \mathcal{S}$$

### Remarque 1 :

L'hypothèse H4 est en particulier vérifiée dans les 2 cas suivants :

H4-A - A une certaine itération  $K = K_0$ , P2 ( $K_0$ ) écrit sous forme canonique admet une base optimale  $I = I_0$  non dégénérée.

H'4 - initialement  $\exists j^0 \in K$  tel que  $a(x^{j^0}) > 0$  et  $b(x^{j^0}) \geq 0$ .

Cette condition suppose que  $A$  est d'intérieur non vide.

Remarque 2

La proposition est encore vraie si on se contente de  $X^{k+1}$  tel que :

$$d^{k+1}(K) = f(x^{k+1}) + u \cdot a(x^{k+1}) + \bar{u} \cdot b(x^{k+1}) > 0$$

et

$$f(x^{k+1}) + u \cdot a(x^{k+1}) \geq \text{Max} \{f(x) + u \cdot a(x) \mid x \in B\} - \epsilon_k \quad (\epsilon_k > 0)$$

si toutefois :  $\epsilon_k \longrightarrow 0$  quand  $k \rightarrow \infty$

III - 4 - CAS CONVEXE - METHODE DU BARYCENTRE [20]

III - 4 - 1 - Problème posé - Hypothèses

Il s'agit de traiter le même problème que précédemment :

$$P1 \quad \left[ \begin{array}{l} \text{Max } f(x) \\ a(x) \geq 0 \\ b(x) \geq 0 \end{array} \right.$$

avec les mêmes notations.

Les hypothèses choisies sont plus fortes, à savoir :

H1 -  $A \cap B \neq \emptyset$

H2 -  $B$  compact

H'3 -  $f, a, b$  concaves et continûment différentiables sur  $\mathbb{R}^n$

H4 - en posant :

$$a(x) = \begin{bmatrix} a'(x) \\ C \cdot x - c \\ -C \cdot x + c \end{bmatrix}$$

(C et c peuvent ne pas exister)

On suppose que :

$$\exists \bar{x} \mid a'(\bar{x}) > 0, C \cdot \bar{x} - c = 0, b(\bar{x}) \geq 0$$

et

C non singulière

III - 4 - 2 - Programme directeur

On utilise la même méthode que dans l'algorithme de DANTZIG en introduisant les points  $\{ x^j \in B \mid j \in K \}$ .

Toutefois, on ne procède pas à des linéarisations de f et a. Le programme directeur s'écrit :

$$P2(K) \begin{cases} \text{Max } f(x^K, \lambda_K) \\ a(x^K, \lambda_K) \geq 0 \\ e^K \cdot \lambda_K = 1 \\ \lambda_K \geq 0 \end{cases}$$

Les  $x^j$  étant choisis de telle sorte que le domaine de P2 (K) soit non vide.

Remarque : P2 (K) est un programme non linéaire. Il s'agit encore d'une approximation par défaut de P1:

III - 4 - 3 - Conditions de KUHN et TUCKER relatives à P2 (K)

A l'optimum,  $\lambda(K)$  est liée aux variables duales  $[ u(K), u^o(K) ]$

par les conditions suivantes :

$$(1) \quad u(K) \geq 0$$

$$(2) \quad g(K) \cdot X^K + u^\circ(K) \cdot e^K \leq 0$$

$$(3) \quad u(K) \cdot a[x(K)] = 0$$

$$(4) \quad g(K) \cdot x(K) = -u^\circ(K)$$

$$\text{où } g(K) = u(K) \cdot \nabla a[x(K)] + \nabla f[x(K)]$$

III - 4 - 4 - Programme auxiliaire - Encadrement de l'optimum

- [ u(K), u°(K) ] étant supposés connus, on définit alors le sous-programme suivant :

$$P3(K) \quad \begin{cases} \text{Max } g(K) \cdot x \\ b(x) \geq 0 \end{cases}$$

A noter que P3(K) reste partiellement non linéaire.

- L'encadrement de l'optimum s'obtient immédiatement :

$$\begin{aligned} f(\hat{x}) - f[x(K)] &\leq f(\hat{x}) + u(K) \cdot a(\hat{x}) - (f[x(K)] + u(K) \cdot a[x(K)]) \\ &\leq g(K) \cdot [\hat{x} - x(K)] \\ &\leq g(K) \cdot [X^S - x(K)] \\ &\leq g(K) \cdot X^S + u^\circ(K) \end{aligned}$$

$$\text{Soit en posant : } d^S = g(K) \cdot X^S + u^\circ(K) \cdot e^S$$

avec  $e^S = 1$

$$0 \leq f(\hat{x}) - f[x(K)] \leq d^S$$

L'encadrement est du même type que précédemment. Comme au paragraphe III - 3 - 8, on peut poser :

$s = k + 1$  ;  $\bar{x} = x(k)$  et  $\bar{g} = g(k)$ ,  $\bar{u}^0 = u^0(k)$ ,

$$k_d^{k+1}(k) = \bar{g} \cdot \bar{x}^{k+1} + \bar{u}^0.$$

III - 4 - 5 Proposition

On définit alors le même algorithme qu'au paragraphe III - 3 - 7.

D'après l'hypothèse H'3,  $\|\nabla f(x)\|$  et  $\|\nabla a(x)\|$  sont bornés supérieurement sur B.

On obtient la proposition suivante :

Proposition

Avec les notations précédentes et sous les hypothèses (H1), (H2), (H'3), (H4) si à une étape on a  $k_d^{k+1}(k) = 0$  alors  $\bar{x}^k$  est solution optimale de P1. Si cette éventualité ne se produit jamais, la suite infinie  $\{\bar{x}^k \mid k \in S\}$  fournie par l'algorithme est telle que :

$$\begin{aligned} \lim_{k \rightarrow \infty} f(\bar{x}^k) &= f(\bar{x}) \\ k &\in S \end{aligned}$$

Remarque : L'optimisation complète de P3 (K) n'est pas nécessaire.

III - 4 - 6 - Lien entre la méthode du barycentre et l'algorithme de FRANK

ET WOLFE [13]

Soit à résoudre le programme suivant :

$$P1 \quad \begin{cases} \text{Max } f(x) \\ b(x) \geq 0 \end{cases}$$

III - 4 - 6 - 1 - Application de l'algorithme de FRANK et WOLFE :

Cet algorithme développé au § IV - 2 donne dans ce cas particulier la séquence suivante à l'étape k :

- a - Soit  $\hat{x}^k$  tel que  $b(\hat{x}^k) \geq 0$
- b - Soit  $\hat{z}^k$  solution optimale de :

$$\left[ \begin{array}{l} \text{Max } \nabla f(\hat{x}^k) \cdot x \\ b(x) \geq 0 \end{array} \right.$$

- c - Soit  $\hat{x}^{k+1}$  solution de  $\{ \text{Max } f(x) \mid x \in [\hat{x}^k, \hat{z}^k] \}$

(Retour à - a - avec  $\hat{x}^k = \hat{x}^{k+1}$ )

Et à chaque étape on a l'encadrement suivant :

$$0 \leq f(\hat{x}) - f(\hat{x}^k) \leq \nabla f(\hat{x}^k) \cdot (\hat{z}^k - \hat{x}^k)$$

III - 4 - 6 - 2 Application de la méthode du barycentre

Le programme directeur devient :

$$P2(K) \left[ \begin{array}{l} \text{Max } f(x^K) \cdot \lambda_K \\ e^K \cdot \lambda_K = 1 \\ \lambda_K \geq 0 \end{array} \right.$$

A l'optimum  $x(K) = x^K \cdot \lambda(K)$  est lié à la variable duale  $u^\circ(K)$  par les relations :

$$(2') \quad \nabla f[x(K)] \cdot x^K + u^\circ(K) \cdot e^K \leq 0$$

$$(4') \quad \nabla f[x(K)] \cdot x(K) = -u^\circ(K)$$

On a donc à résoudre le programme auxiliaire suivant :



$$P3 (K) \quad \left[ \begin{array}{l} \text{Max } \nabla f [ x (K) ] \cdot x \\ b (x) \geq 0 \end{array} \right.$$

A l'optimum,  $X^s$ , de  $P3 (K)$  on a :

$$0 \leq f (x) - f [ x (K) ] \leq \nabla f [ x (K) ] \cdot [ X^s - x (K) ]$$

### III - 4 - 6 - 3 - Généralisation de l'algorithme de FRANK et WOLFE

A l'étape suivante de la méthode du barycentre, on résoud  $P2 (K')$  avec  $K' = KU \{s\}$ . On maximise  $f$  dans un ensemble convexe contenant le segment  $[ x (K), X^s ]$ .

Il s'agit d'une généralisation de  $-c-$  dans l'algorithme de FRANK et WOLFE.

### III - 4 - 6 - 4 - Remarques

- 1 - On retrouve dans ce cas particulier, l'algorithme de HOLLOWAY [15] appliqué au programme suivant :

$$\left[ \begin{array}{l} \text{Max } f (x) \\ x \in B \end{array} \right.$$

- 2 - On peut encore noter que, dans les mêmes conditions, la méthode des colonnes [III - 3] de DANTZIG donne la solution après une étape de calculs.

### III - 5 - CONCLUSION

L'intérêt principal de ces deux méthodes réside dans le fait que les programmes à traiter sont de taille plus réduite, en général. Cependant, elles ne nous affranchissent pas de la non linéarité de tous les problèmes.

à envisager ; ceci en particulier au niveau des sous-programmes.

Notons enfin qu'elles semblent bien convenir pour les programmes mathématiques du type suivant :

$$\left[ \begin{array}{l} \text{Max } f(x) \\ a(x) \geq 0 \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

avec des hypothèses bien choisies.

- Remarque

Supposons de plus  $a$  quadratique ; on peut noter qu'aucun des deux algorithmes ne permet de tirer profit de cette propriété particulière.

"L'algorithme mixte", que nous allons définir au § IV - 3, se particularisera dans ce cas.

CHAPITRE IV

ALGORITHME MIXTE

\*\*\*\*\*

IV - 1 - SOUS-OPTIMISATION UNIDIMENSIONNELLE :

IV - 1 - 1 Problème posé

Soit à résoudre le problème suivant :

$$P \quad \left[ \begin{array}{l} \text{Max } f(x) \\ x \in A \cap B \end{array} \right.$$

On suppose que :

H1 - A est un ensemble convexe et fermé

H2 - B compact et convexe

H3 -  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  telle que :

i - f est concave sur  $\mathbb{R}^n$

ii - f est continûment différentiable sur  $\mathbb{R}^n$

H4 -  $A \cap B \neq \emptyset$

IV - 1 - 2 - Principe d'un algorithme de sous-optimisation unidimensionnelle

A une solution réalisable  $x^k$ , obtenue à une étape k, on associe une partie  $\Delta^k(x^k)$  de  $A \cap B$

On choisit arbitrairement  $z^k \in \Delta^k(x^k)$

On maximise alors la fonction f sur le segment  $[x^k, z^k]$

Soit  $x^{k+1}$  le point ainsi obtenu.

On réitère le processus avec  $x^{k+1}$  au lieu de  $x^k$

Initialement on choisit  $x^0 \in A \cap B$

IV - 1 - 3 Propriétés

Moyennant les hypothèses indiquées, l'algorithme génère une suite infinie  $\{ (x^k, z^k) \mid k \in \mathbb{N} \}$  vérifiant les propriétés suivantes :

- Propriété 1 :  $\forall k \in \mathbb{N}, f(x^k) \leq f(x^{k+1})$

- Propriété 2 :  $\forall k \in \mathbb{N}, x^k \in A \cap B$

- Propriété 3 :

Quel que soit le point d'accumulation  $(x^*, z^*)$  de la suite  $\{ (x^k, z^k) \mid k \in \mathbb{N} \}$  générée par l'algorithme on a :

$$\nabla f(x^*) \cdot (z^* - x^*) \leq 0$$

- Démonstration :

Soit  $N' \subset \mathbb{N}$  définissant une sous-suite infinie convergente :

$$\{ (x^k, z^k) \mid k \in N' \} \rightarrow (x^*, z^*) \in B \times B$$

Par définition de  $x^{k+1}$  on a :

$$\forall \theta, 0 \leq \theta \leq 1, f(x^{k+1}) \geq f[x^k + \theta(z^k - x^k)]$$

Donc  $\forall k \in N'$  et  $\forall k' \in N'$  avec  $k' > k + 1$  on a :

$$\forall \theta, 0 \leq \theta \leq 1, f(x^{k'}) \geq f[x^k + \theta(z^k - x^k)]$$

Soit en passant à la limite [  $f$  étant continue sur  $\mathbb{R}^n$  ] :

$$\forall \theta, 0 \leq \theta \leq 1, f(x^*) \geq f[x^* + \theta(z^* - x^*)]$$

On en déduit que  $x^*$  maximise  $f$  sur le segment  $[x^*, z^*]$

On a donc, en désignant par  $M$  le cône tangent en  $x^*$  au segment  $[x^*, z^*]$  :

$$\nabla f(x^*) \in \Gamma(M) \text{ cône polaire de } M \quad (\text{§ I - 3 - 5})$$

$$\text{Or } [z^* - x^*] \in M$$

Par définition du cône polaire  $\Gamma(M)$ , on a donc :

$$\nabla f(x^*) \cdot (z^* - x^*) \leq 0$$

Remarque: Pour toute partie infinie  $N''$  de  $\mathbb{N}$  telle que :

$$\forall k \in N'', (x^k, z^k) = (x^{k'}, z^{k'}) \text{ on a aussi : } \nabla f(x^*) \cdot (z^{k'} - x^{k'}) \leq 0$$

Propriété 4 :

$\bar{x}$  étant une solution optimale de P, si  $\forall k \in \mathbb{N}$ , on a :

$$0 \leq f(\bar{x}) - f\left(\frac{k}{x}\right) \leq \nabla f\left(\frac{k}{x}\right) \cdot \left(\frac{k}{x} - \bar{x}\right)$$

alors  $\lim_{k \rightarrow \infty} f\left(\frac{k}{x}\right) = f(\bar{x})$

En effet :  $\forall k \in \mathbb{N}, \nabla f\left(\frac{k}{x}\right) \cdot \left(\frac{k}{x} - \bar{x}\right) \geq 0$

D'après la propriété 3,  $(\bar{x}, \bar{z})$  étant un point d'accumulation de la suite, on a dans ce cas :

$$\nabla f(\bar{x}) \cdot (\bar{z} - \bar{x}) = 0.$$

$\implies \lim_{k \rightarrow \infty, k \in \mathbb{N}'} f\left(\frac{k}{x}\right) = f(\bar{x})$ . Donc:  $\lim_{k \rightarrow \infty} f\left(\frac{k}{x}\right) = f(\bar{x})$  la suite

suité étant monotone non décroissante.

Pour un point  $(\bar{x}^*, \bar{z}^*)$  [ remarque précédente ] on a le même résultat.

IV - 2 - ALGORITHME DE FRANK ET WOLFE [ 13 ]

IV - 2 - 1 - Algorithme :

1 - Initialement, on choisit  $\bar{x} \in A \cap B$

2 - A l'étape k, on dispose de  $\frac{k}{x}$ .

$$\Delta\left(\frac{k}{x}\right) = \left\{ z \in A \cap B \mid z \text{ maximise } \nabla f\left(\frac{k}{x}\right) \cdot x \text{ avec } x \in A \cap B \right\}$$

On choisit  $\frac{k}{z} \in \Delta\left(\frac{k}{x}\right)$

3 - Soit  $\frac{k+1}{x}$  le point qui maximise f sur  $\left[ \frac{k}{x}, \frac{k}{z} \right]$

Retour à 2 avec  $k = k + 1$ .

IV - 2 - 2 - Proposition

Par définition de  $\frac{k}{z}$ , on a :

$$\nabla f\left(\frac{k}{x}\right) \cdot \left(\bar{x} - \frac{k}{z}\right) \leq \nabla f\left(\frac{k}{x}\right) \cdot \left(\frac{k}{z} - \frac{k}{x}\right)$$

D'après la propriété 2 et compte tenu du fait que f est concave on peut écrire :

$$0 \leq f(\bar{x}) - f\left(\frac{k}{x}\right) \leq \nabla f\left(\frac{k}{x}\right) \cdot \left(\frac{k}{x} - \bar{x}\right)$$

La propriété 4 nous donne :

$$\lim_{k \rightarrow \infty} f\left(\frac{k}{x}\right) = f(\bar{x})$$

Proposition

Avec les notations précédentes et sous les hypothèses (H1), (H2), (H3), (H4), la suite infinie  $\left\{ \frac{k}{x} \mid k \in \mathbb{N} \right\}$  fournie par l'algorithme est telle que :

$$\lim_{k \rightarrow \infty} f\left(\frac{k}{x}\right) = f(\bar{x})$$

IV - 2 - 3 - Remarques

- 1 . Lors de la maximisation sur le segment, on peut se contenter d'une solution à  $\epsilon$  près. Nous précisons dans quelles conditions au chapitre VI .
- 2 . A chaque itération, on possède un encadrement de l'optimum très satisfaisant sur le plan numérique.
- 3 . Dans le cas particulier suivant :

$$\left[ \begin{array}{l} \text{Max } f(x) \\ A \cdot x = a \\ x \geq 0 \end{array} \right.$$

Cet algorithme conduit à une séquence infinie de programmes linéaires et de maximisations sur un segment, ce qui est intéressant sur le plan numérique.

- 4 . On trouvera dans [ 16 ] et [ 33 ] une étude générale des algorithmes de sous-optimisation unidimensionnelle utilisant en particulier la notion de fonction multivoque. On peut aussi en trouver une démonstration différente dans [ 11 ] mais moyennant des hypothèses plus fortes.

IV - 3 - ALGORITHME MIXTE

IV - 3 - 1 - Problème posé - Hypothèses

- Il s'agit de résoudre :

$$P1 \quad \begin{cases} \text{Max } f(x) \\ a(x) \geq 0 \\ b(x) \geq 0 \end{cases}$$

A et B sont définis comme au chapitre III.

- On supposera que :

H1 :  $A \cap B \neq \emptyset$

H2 : B compact

H3 :  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  concave et continûment différentiable sur  $\mathbb{R}^n$

$a : \mathbb{R}^n \rightarrow \mathbb{R}^p$  concave sur  $\mathbb{R}^n$

$b : \mathbb{R}^n \rightarrow \mathbb{R}^q$  concave sur  $\mathbb{R}^n$

Pour la suite, nous introduisons les mêmes notations qu'au chapitre III.

IV - 3 - 2 - Principe de l'algorithme mixte.

L'algorithme de FRANK et WOLFE détermine  $\bar{x}^k$  à partir du programme suivant :

$$\begin{cases} \text{Max } \nabla f(\bar{x}^k) \cdot x \\ a(x) \geq 0 \\ b(x) \geq 0 \end{cases}$$

pour lequel les contraintes sont non linéaires.



Nous allons introduire le principe de décomposition barycentrique.

Le point  $\overset{k}{z}$  sera alors défini à partir d'un programme linéaire dont le domaine des solutions réalisables sera inclus dans  $A \cap B$ . Il s'agira donc d'une approximation par défaut.

Les conditions de KUHN et TUCKER nous permettront "d'agrandir" cet ensemble à chaque itération en résolvant un programme non linéaire en général.

IV - 3 - 3 - Programme directeur de l'algorithme mixte

- A l'étape  $k$ , on dispose d'un ensemble fini d'indices  $K$ , et, d'un ensemble de générateurs  $\{ X^j \mid X^j \in B, j \in K \}$

Désignons par :  $B(K) = \{ x \in \mathbb{R}^n \mid x = X^K \cdot \lambda_K, A^K \cdot \lambda_K \geq 0, e^K \cdot \lambda_K = 1, \lambda_K \geq 0 \}$  où  $A^K$  est une matrice dont la colonne  $j$  est  $A^j = a(X^j)$ .

Soit  $\overset{k}{x} \in B(K)$

On peut définir le programme suivant :

$$\begin{array}{l}
 \text{P2 (k, K)} \quad \left[ \begin{array}{l}
 \text{Max } \nabla f(\overset{k}{x}) \cdot X^K \cdot \lambda_K \\
 A^K \cdot \lambda_K \geq 0 \\
 e^K \cdot \lambda_K = 1 \\
 \lambda_K \geq 0
 \end{array} \right.
 \end{array}$$

Si  $\lambda(K)$  est une solution optimale de P2 (k, K) on notera :

$$\overset{k}{z} = X^K \cdot \lambda(K)$$

- Remarques :

- 1)  $K$  est choisi tel que  $B(K) \neq \emptyset$  à l'étape 1.
- 2) P2 (k, K) est programme entièrement linéaire - S'il est supposé résolu par la méthode simpliciale, on obtient les variables duales avec  $\lambda(K)$ .

IV - 3 - 4 - Conditions de KUHN et TUCKER relatives à P2 (k, K)

- Désignons par  $(\overset{k}{u}, \overset{k_0}{u})$  les variables duales associées à la solution optimale  $\lambda(K)$ .

On obtient les relations suivantes :

- (1)  $\overset{k}{u} \geq 0$
- (2)  $\nabla f(\overset{k}{x}) \cdot X^K + \overset{k}{u} \cdot A^K + \overset{k_0}{u} \cdot e^K \leq 0$
- (3)  $\overset{k}{u} \cdot A^K \cdot \lambda(K) = 0$
- (4)  $\nabla f(\overset{k}{x}) \cdot \overset{k}{z} = -\overset{k_0}{u}$

IV - 3 - 5 - Programme auxiliaire de l'algorithme mixte

Désignons par  $X^S$  une solution optimale du programme suivant :

$$P3(k, K) \quad \left[ \begin{array}{l} \text{Max } \nabla f(\overset{k}{x}) \cdot x + \overset{k}{u} \cdot a(x) \\ b(x) \geq 0 \end{array} \right.$$

A noter que P3 (k, K) n'est pas un programme linéaire.

Posons enfin:  $d^S = \nabla f(\overset{k}{x}) \cdot X^S + \overset{k}{u} \cdot a(X^S) + \overset{k_0}{u}$

La résolution de P3 (k, K) va permettre "d'agrandir" l'ensemble B (K) .

IV - 3 - 6 - Encadrement de l'optimum

- On peut obtenir alors un encadrement de l'optimum comme dans les méthodes présentées au chapitre III.

En effet on a :  $\nabla f(\overset{k}{x}) \cdot \hat{x} + \overset{k}{u} \cdot a(\hat{x}) + \overset{k_0}{u} \leq d^S$

Comme  $\bar{u}^k \cdot a(\bar{x}) \geq 0$  et  $\nabla f(\bar{x}) \cdot \bar{z}^k = -\bar{u}^k$ , on en déduit que :

$$\nabla f(\bar{x}) \cdot (\bar{x} - \bar{z}^k) \leq d^s.$$

Soit encore :  $\nabla f(\bar{x}) \cdot (\bar{x} - \bar{z}^k) \leq d^s + \nabla f(\bar{x}) \cdot (\bar{z}^k - \bar{x})$

Or  $f$  est concave sur  $\mathbb{R}^n$  :

$$0 \leq f(\bar{x}) - f(\bar{z}^k) \leq d^s + \nabla f(\bar{x}) \cdot (\bar{z}^k - \bar{x})$$

- Remarque

On peut noter que l'algorithme mixte fait intervenir les majorants relatifs aux deux méthodes dont elle est issue.

Pour la suite nous poserons :  $D^s = d^s + \nabla f(\bar{x}) \cdot (\bar{z}^k - \bar{x})$

D'où :

$$0 \leq f(\bar{x}) - f(\bar{z}^k) \leq D^s$$

IV - 3 - 7 - Algorithme

1 - Initialement on choisit  $\bar{x}^k \in B(K)$

2 - A l'étape  $k$ , on dispose de  $\bar{x}^k$  et  $K$ .

a) On détermine  $\bar{z}^k$  solution optimale de P2 ( $k, K$ )

b) On calcule  $(\bar{u}^k, \bar{u}^k)$

c) On résout alors P3 ( $k, K$ ) qui nous donne  $\bar{x}^s$  et  $D^s$

3 - Deux cas sont alors possibles :

a)  $D^s = 0$  alors  $\bar{x}^k$  est solution optimale de P1 .

b)  $D^s > 0$  On détermine  $\bar{x}^{k+1}$  qui maximise  $f$  sur  $[\bar{x}^k, \bar{z}^k]$

On agrandit l'ensemble d'indices  $K' = K + s$

Retour à 2 avec  $k = k+1$  et  $K = K'$

- Remarques :

Il résulte de l'algorithme que :

-  $\forall k \in S, x \in B(K)$

- La suite  $\{ f(x^k) \mid k \in S \}$  est monotone non décroissante. (S étant la suite des itérations)

IV - 3 - 8 - Condition nécessaire et suffisante pour que  $D^S = 0$

- Montrons tout d'abord que  $d^S \geq 0$ . En effet :

$$\begin{aligned}
d^S &= \nabla f(x^k) \cdot x^S + \overset{k}{u} \cdot a(x^S) + \overset{k}{u} \cdot 0 \\
&= \nabla f(x^k) \cdot x^S + \overset{k}{u} \cdot a(x^S) - \nabla f(x^k) \cdot \overset{k}{z} \\
&> [\nabla f(x^k) \cdot x^S + \overset{k}{u} \cdot a(x^S)] - [\nabla f(x^k) \cdot \overset{k}{z} + \overset{k}{u} \cdot a(\overset{k}{z})] \geq 0
\end{aligned}$$

car  $\overset{k}{z} \in B$  et  $x^S$  est solution optimale de P3 (k, K)

- Comme  $\overset{k}{x} \in B(K)$  et  $\overset{k}{z}$  est solution optimale de P2 (k, K)

on a :  $\nabla f(x^k) \cdot (\overset{k}{z} - \overset{k}{x}) \geq 0$

- On peut alors écrire que :

$$D^S = 0 \quad \iff \quad \begin{cases} d^S = 0 \\ \text{et} \\ \nabla f(x^k) \cdot (\overset{k}{z} - \overset{k}{x}) = 0 \end{cases}$$

IV - 3 - 9 - Proposition

- D'après l'encadrement, si, à une étape k, on a  $D^S = 0$ , alors l'optimum est atteint.

- Sinon l'algorithme est infini.

Supposons que :  $\forall k \in S, D^S > 0$ .

Ajoutons l'hypothèse suivante :

H4 -  $\forall k \in S, (\overset{k}{u}, \overset{k}{u} \cdot 0) \in U$  compact

S étant la suite des itérations, posons pour les mêmes raisons qu'au chapitre

III :

$$K = \{ 1, 2, \dots, k \} \quad s = k+1$$

$$\forall k \in S \text{ et } k' > k : K' = \{ 1, 2, \dots, k' \}$$

$$k_D^{k+1}(K) = \nabla f(\bar{x}^k) \cdot X^{k+1} + \bar{u}^k \cdot a(X^{k+1}) + \bar{u}^k$$

$$k_D^{k+1}(K) = k_D^{k+1}(K) + \nabla f(\bar{x}^k) \cdot (\bar{z}^k - \bar{x}^k)$$

$$\forall k \in S, (\bar{x}^{k+1}, \bar{x}^k, \bar{z}^k, \bar{u}^k, \bar{u}^0) \in B^3 \times U \text{ compact.}$$

Désignons par  $(\bar{x}^*, \bar{x}, \bar{z}, \bar{u}, \bar{u}^0) \in B^3 \times U$  un point d'accumulation de cette suite. On peut extraire de S une sous-suite infinie S' telle que :

$$(\bar{x}^{k+1}, \bar{x}^k, \bar{z}^k, \bar{u}^k, \bar{u}^0) \rightarrow (\bar{x}^*, \bar{x}, \bar{z}, \bar{u}, \bar{u}^0)$$

$$k \rightarrow \infty$$

$$k \in S'$$

$$\text{Or : } \forall k \in S', k_D^{k+1}(K) = \nabla f(\bar{x}^k) \cdot X^{k+1} + \bar{u}^k \cdot a(X^{k+1}) + \bar{u}^0 \geq 0 \quad (\text{IV-3-8})$$

$$\text{et } \forall k \in S', \forall k' > k, k_D^{k+1}(K') = \nabla f(\bar{x}^{k'}) \cdot X^{k+1} + \bar{u}^{k'} \cdot a(X^{k+1}) + \bar{u}^0 \leq 0$$

(K. et T.)

Comme a est continue sur  $\mathbb{R}^n$  et f continûment différentiable sur  $\mathbb{R}^n$ , on obtient, en passant à la limite :

$$\nabla f(\bar{x}^*) \cdot \bar{x} + \bar{u} \cdot a(\bar{x}) + \bar{u}^0 = 0$$

De plus  $\forall k \in S', \nabla f(\bar{x}^k) \cdot (\bar{z}^k - \bar{x}^k) \geq 0$ . Compte-tenu de la propriété 3 du paragraphe IV - 1 - 3 on a aussi :

$$\nabla f(\bar{x}^*) \cdot (\bar{z} - \bar{x}) = 0$$

$$\text{D'où : } \lim_{k \rightarrow \infty} k_D^{k+1}(K) = 0$$

$$k \in S'$$

Comme la suite des valeurs  $f(\bar{x}^k)$  est monotone non décroissante, on a :

$$\lim_{k \rightarrow \infty} f(\bar{x}^k) = f(\bar{x}^*)$$

$$k \in S$$

Proposition Avec les notations précédentes et sous les hypothèses H1, H2, H3, H4 si à une étape de l'algorithme on a  $D^k(K) = 0$  alors  $\frac{k}{x}$  est une solution optimale du problème P1. Si cette éventualité ne se produit jamais, la suite infinie  $\{ \frac{k}{x} \mid k \in S \}$  fournie par l'algorithme est telle que :

$$\lim_{\substack{k \rightarrow \infty \\ k \in S}} f\left(\frac{k}{x}\right) = f(\bar{x})$$

IV - 3 - 10 - Cas où l'hypothèse H4 est satisfaite

H'4 - Initialement l'un des générateurs  $x^{j_0}$  est tel que :

$$a(x^{j_0}) > 0 \text{ et } b(x^{j_0}) \geq 0.$$

- En effet :  $\forall k \in S, \forall f\left(\frac{k}{x}\right) \cdot x^{j_0} + \frac{k}{u} \cdot a(x^{j_0}) + \frac{k_0}{u^0} \leq 0$

$$\Rightarrow \frac{k}{u} \cdot a(x^{j_0}) \leq \nabla f\left(\frac{k}{x}\right) \cdot (z - x^{j_0})$$

$$\Rightarrow \sum_1^{k_i} u^i a_i(x^{j_0}) \leq \nabla f\left(\frac{k}{x}\right) \cdot (z - x^{j_0})$$

Or

$$0 \leq \sum_1^{k_i} u^i a_i(x^{j_0}) \leq \sum_1^{k_i} u^i a_i(x^{j_0})$$

$$0 \leq \sum_1^{k_i} u^i \leq \frac{\nabla f\left(\frac{k}{x}\right) \cdot (z - \frac{k}{x})}{a_i(x^{j_0})}$$

- En outre

$$\frac{k_0}{u^0} = -\nabla f\left(\frac{k}{x}\right) \cdot z$$

Comme  $f$  est continûment différentiable et que  $\forall k \in S, (\frac{k}{x}, \frac{k}{z}) \in B \times B$  compact, on a :

$$\forall k \in S, (\frac{k}{u}, \frac{k_0}{u^0}) \in U \text{ compact}$$

H''4 : A une certaine itération  $k = k_0$  le programme directeur P2 ( $k, K$ ) supposé écrit sous forme canonique admet une base optimale non dégénérée  $I$ .

Mettons P2 ( $k, K$ ) sous forme canonique.

$$\left[ \begin{array}{l} \text{Max } \nabla f(x) \cdot X^{K+L} \cdot \lambda_{K+L} \\ A^{K+L} \cdot \lambda_{K+L} = 0 \\ e^{K+L} \cdot \lambda_{K+L} = 1 \\ \lambda_{K+L} \geq 0 \end{array} \right.$$

avec  $X^L = 0$  ;  $e^L = 0$  ;  $A^L = -U$

Il suffit de montrer que  $\forall k \in S, k \geq k_0$  on a  $(\overset{k}{u}, \overset{k}{u}^0) \in U$  compact.

Posons :

$$d(K) = \nabla f(x) \cdot X^{K+L} + \overset{k}{u} \cdot A^{K+L} + \overset{k}{u}^0 \cdot e^{K+L} \leq 0, \forall k \in S$$

d'où  $d^I(K) = \nabla f(x) \cdot X^I + \overset{k}{u} \cdot A^I + \overset{k}{u}^0 \cdot e^I \leq 0, \forall k \geq k_0, k \in S$

- Comme  $\overset{\circ}{\lambda}_I > 0$  on a donc :

$$\begin{aligned} d^I(K) \cdot \overset{\circ}{\lambda}_I &= \nabla f(x) \cdot X^I \cdot \overset{\circ}{\lambda}_I + \overset{k}{u} \cdot A^I \cdot \overset{\circ}{\lambda}_I + \overset{k}{u}^0 \cdot e^I \cdot \overset{\circ}{\lambda}_I \leq 0 \\ &= \nabla f(x) \cdot X^I \cdot \overset{\circ}{\lambda}_I + \overset{k}{u}^0 \\ &= \nabla f(x) \cdot [X^I \overset{\circ}{\lambda}_I - \overset{k}{z}] \end{aligned}$$

Or  $d^I \cdot \overset{\circ}{\lambda}_I = \sum_{j \in I} d^j \cdot \overset{\circ}{\lambda}_j \leq 0$  . Posons :  $\overset{\circ}{y} = X^I \cdot \overset{\circ}{\lambda}_I$  .

D'où  $\nabla f(x) \cdot (\overset{\circ}{y} - \overset{k}{z}) \leq d^j \cdot \overset{\circ}{\lambda}_j \leq 0$  pour  $j \in I$

Soit  $\frac{\nabla f(\overset{\circ}{x}) \cdot (\overset{\circ}{y} - \overset{\circ}{z})}{\overset{\circ}{\lambda}_j} \leq d^j \leq 0, \forall j \in \overset{\circ}{I}, \forall k \in S, k \geq k_0$

On peut donc affirmer que  $d^{\overset{\circ}{I}}(K)$  est borné,  $\forall k \in S, k \geq k_0$

$$\text{Comme } d^{\overset{\circ}{I}}(K) - \nabla f(\overset{\circ}{x}) \cdot X^{\overset{\circ}{I}} = \begin{bmatrix} \overset{\circ}{A} \\ \overset{\circ}{I} \\ e \end{bmatrix}$$

$$\rightarrow (\overset{\circ}{u}, \overset{\circ}{u}^0) = [d^{\overset{\circ}{I}}(K) - \nabla f(\overset{\circ}{x}) \cdot X^{\overset{\circ}{I}}] \cdot \begin{bmatrix} \overset{\circ}{A} \\ \overset{\circ}{I} \\ e \end{bmatrix}^{-1}$$

Or  $\overset{\circ}{x} \in B$  et  $f$  est continûment différentiable :

$$\rightarrow (\overset{\circ}{u}, \overset{\circ}{u}^0) \in U \text{ compact } \forall k \in S, k \geq k_0.$$

- Remarques :

1 - Les deux cas proposés sont identiques à ceux envisagés dans la méthode des colonnes.

2 - L'optimisation complète de  $P3(k, K)$  n'est pas nécessaire, ce qui est très intéressant sur le plan pratique (voir chapitre VI).

IV - 3 - 11 - Conclusion

- Comme dans les deux algorithmes précédents, tous les sous-problèmes à traiter ne sont pas linéaires.

- On peut toutefois remarquer que, dans le cas particulier qui va suivre et moyennant une hypothèse supplémentaire, l'algorithme mixte se ramène à une séquence théoriquement infinie de programmes linéaires et de maximisations



sur un segment.

- En effet, supposons que l'on ait à résoudre :

$$P1 \quad \left[ \begin{array}{l} \text{Max } f(x) \\ a(x) \geq 0 \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

P2 (k, K) conserve la même structure linéaire tandis que P3 (k, K) devient un problème à contraintes linéaires et à fonction économique partiellement non linéaire.

$$P3(k, K) \quad \left[ \begin{array}{l} \text{Max } \nabla f^k(\bar{x}) \cdot x + \bar{u} \cdot a(x) \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

Si  $a$  est de plus continûment différentiable sur  $\mathbb{R}^n$ , l'algorithme de FRANK et WOLFE peut être utilisé et donnera une séquence infinie de problèmes linéaires et de maximisations sur un segment. En outre, à chaque étape, on possédera un encadrement de l'optimum, ce qui s'avère très intéressant si on veut résoudre P3 (k, K) à  $\epsilon$  près. On en trouvera l'étude détaillée au chapitre VI.

Remarquons enfin que, dans le cas où  $a$  est quadratique, P3 (k, K) est un programme quadratique. Les maximisations sur un segment peuvent se faire de façon exacte. On peut même résoudre P3 (k, K) par un algorithme fini de programmation quadratique. Les deux algorithmes présentés au chapitre III, ne présentent pas dans ce cas cette particularité intéressante.

CHAPITRE V

ETUDE D'UN CAS PARTICULIER

\*\*\*\*\*

V - 1 - ETUDE D'UN CAS PARTICULIER

V - 1 - 1 - Problème traité - Hypothèses

Nous allons maintenant nous intéresser au problème particulier suivant :

$$P \quad \begin{cases} \text{Max } f(x) \\ c(x) = 0 & (1) \\ b(x) \geq 0 & (2) \end{cases}$$

Posons :  $A' = \{ x \mid c(x) = 0 \}$

$B = \{ x \mid b(x) \geq 0 \}$

Les hypothèses sont les suivantes :

H1 :  $A' \cap B \neq \emptyset$

H2 :  $B$  compact

H3 :  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  concave et continûment différentiable sur  $\mathbb{R}^n$

$b : \mathbb{R}^n \rightarrow \mathbb{R}^q$  concave sur  $\mathbb{R}^n$

H4 : i -  $c : \mathbb{R}^n \rightarrow \mathbb{R}$  convexe sur  $\mathbb{R}^n$

ii -  $\forall x \in \mathbb{R}^n, c(x) \geq 0$

- Remarque 1 :

(1) ne comporte qu'une seule contrainte. Si on introduit le principe de décomposition dans  $B$ , le programme principal ne contiendra que deux contraintes, ce qui est très intéressant sur le plan numérique.

- Remarque 2 :

A titre d'exemple, le problème suivant peut se ramener à la forme P :

$$\begin{cases} \text{Max } f(x) \\ A \cdot x = a \quad (1) \\ b(x) \geq 0 \end{cases}$$

Si (1) comporte p contraintes, on peut poser :

$$c(x) = \sum_{i=1}^p (A_i \cdot x - a_i)^2$$

où c vérifie l'hypothèse H4.

V - 1 - 2 - Forme canonique du problème

Ramenons (P) à la forme (P1) des chapitres III et IV .

En utilisant l'hypothèse H4 - ii on peut écrire que :

$$c(x) = 0 \iff -c(x) \geq 0$$

Posons donc :  $\forall x \in \mathbb{R}^n, a'(x) = -c(x) \implies \forall x \in \mathbb{R}^n, a'(x) \leq 0$

Notons de même  $A = \{ x \mid a'(x) \geq 0 \}$

P s'écrit alors

$$(P1) \quad \begin{cases} \text{Max } f(x) \\ a'(x) \geq 0 \\ b(x) \geq 0 \end{cases}$$

V - 1 - 3 - Difficultés rencontrées

Compte-tenu des hypothèses :  $\nexists x \in \mathbb{R}^n \mid a'(x) > 0$

L'hypothèse H4 de la méthode du barycentre (§ III - 4) n'est pas satisfaite.

Envisageons maintenant, la méthode des colonnes (§ III - 3) et l'algorithme mixte (§ - IV - 3).

On peut déjà noter que l'hypothèse H'4 n'est pas vérifiée. Montrons qu'il en est de même pour l'hypothèse H''4 -

En effet, pour ces deux méthodes, le programme directeur est un programme linéaire à deux contraintes. Mis sous forme canonique, il s'écrit :

$$\begin{cases} A^K \cdot \lambda_K - \lambda_0 = 0 \\ e^K \cdot \lambda_K + 0 \cdot \lambda_0 = 1 \\ \lambda_K \geq 0, \lambda_0 \geq 0 \end{cases}$$

$$\text{Or } A^K \cdot \lambda_K - \lambda_0 = 0 \iff \sum_{j \in K} c(x^j) \cdot \lambda_j + \lambda_0 = 0$$

$$\iff \lambda_0 = 0 \text{ et } (\forall j \in K, c(x^j) \cdot \lambda_j = 0)$$

On en déduit que :

$$[\lambda_j \neq 0 \implies c(x^j) = 0] \iff [c(x^j) \neq 0 \implies \lambda_j = 0]$$

Pour avoir une base non dégénérée on devrait avoir comme matrice de base

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \text{ ce qui est impossible. Il n'existe donc aucune base non dégénérée.}$$

Remarque 1 :

Les deux variables de base ont pour valeurs :

$$\lambda_{j1} = 1 \quad \text{et} \quad \lambda_{j2} = 0$$

La solution du programme directeur est donc un des générateurs  $X^{j1}$  tel que  $a'(X^{j1}) = 0$

Remarque 2

D'après la remarque précédente, la solution du programme directeur dans la méthode des colonnes est le générateur  $X^{j1}$  tel que :

$$f(X^{j1}) = \text{Max} \{ f(X^j) \mid j \in K, b(X^j) \geq 0 \text{ et } a'(X^j) = 0 \}$$

D'une étape sur l'autre, la solution du programme directeur reste inchangée tant que le générateur entré  $X^{k+1}$  est tel que  $a'(X^{k+1}) < 0$

V - 1 - 4 - Méthode des colonnes - lien avec les méthodes de pénalisations extérieures [23] [25] [34]

A chaque étape de la méthode des colonnes, on introduit un nouveau générateur  $X^{k+1}$  solution optimale de P3 (K) :

$$\begin{cases} \text{Max } f(x) + u^k \cdot a'(x) \\ b(x) \geq 0 \end{cases}$$

Deux cas sont alors possibles :

1 -  $a'(X^{k+1}) = 0$  : Alors  $X^{k+1}$  est solution optimale du problème posé P .

2 -  $a'(X^{k+1}) < 0$  . Alors la solution optimale du programme directeur P2 (K) reste inchangée à l'étape suivante (§ V - 1 - 3 - remarque 2) .

Supposons que :  $\forall k \in S, a'(X^{k+1}) < 0$

Initialisons l'algorithme avec :

$K = \{1\}$  et  $X^1 \in A \cap B$

En général, on a :  $f(X^1) < f(\bar{x})$

Dans ces conditions on a :  $\forall k \in S, \bar{x}^k = X^1$  (cas 2)

Donc  $\forall k \in S, \bar{u}^{k^0} = -f(X^1)$  (condition de Kuhn et Tucker)

L'encadrement donne dans ce cas :

$$0 < f(\bar{x}) - f(X^k) \leq d^{k+1}(K)$$

$$\text{D'où : } \forall k \in S, d^{k+1}(K) = f(X^{k+1}) + \bar{u}^k \cdot a'(X^{k+1}) + \bar{u}^{k^0} \geq \rho > 0$$

$$\text{Or : } \forall k \in S, f(X^{k+1}) + \bar{u}^k \cdot a'(X^{k+1}) + \bar{u}^{k^0} \leq 0$$

(Kuhn et Tucker)

$$\begin{aligned} \rightarrow (u^k - u^{k+1}) \cdot a'(X^{k+1}) &> 0 \\ a'(X^{k+1}) &< 0 \end{aligned} \quad \Rightarrow \quad \bar{u}^{k+1} > \bar{u}^k \geq 0$$

La suite des valeurs  $\{\bar{u}^k \mid k \in S\}$  est monotone strictement croissante.

Supposons qu'elle soit bornée.

Dans ce cas, on aurait :  $\forall k \in S, (\bar{u}^k, \bar{u}^{k^0}) \in U$  compact

En utilisant les résultats du § - III - 3 - 8 - On aurait :

$$\exists \bar{S} \subset S \text{ tel que : } d^{k+1}(K) \rightarrow 0$$

$$k \rightarrow \infty$$

$$k \in \bar{S}$$

ce qui est en contradiction avec le fait que :  $\forall k \in S, d^{k+1}(K) \geq \rho > 0$

$$\rightarrow \lim_{k \rightarrow \infty} \bar{u}^k = +\infty$$

On peut alors établir un lien entre la méthode des colonnes et les méthodes de pénalisation extérieure.

En effet, reprenons le problème P :

$$P \quad \begin{cases} \text{Max } f(x) \\ c(x) = 0 \\ b(x) \geq 0 \end{cases}$$

Rappelons que  $A' = \{ x \in \mathbb{R}^n \mid c(x) = 0 \}$

Introduisons une suite de fonctions pénalisantes, relatives à  $A'$ , soit  $p_k$ ,  
 $k = 1, \dots, n, \dots$

On résoud le programme pénalisé suivant :

$$P_k \quad \begin{cases} \text{Max } f(x) - p_k(x) \\ b(x) \geq 0 \end{cases} \quad k = 1, \dots, n, \dots$$

On a la proposition suivante :

Si les problèmes  $(P_k)$  admettent tous des solutions optimales  $Z_k$ , alors tout point d'accumulation de la suite  $\{ Z_k \mid k \in \mathbb{N} \}$  est solution optimale de P et la limite de  $p_k(Z)$  est nulle.

Or le sous-programme  $P3(K)$  relatif à la méthode des colonnes s'écrit :

$$\begin{cases} \text{Max } f(x) - u \cdot c(x) \\ b(x) \geq 0 \end{cases}$$

avec  $\forall k \in S, \quad u^{k+1} > u^k \geq 0$

et

$$\lim_{k \rightarrow \infty} u^k = +\infty$$



Remarque :

A partir de l'étape 2 on a donc  $\frac{k}{u} > 0$

Par un changement d'indices, on peut démarrer à l'étape 2.

Alors la fonction continue :  $p_k(x) = \frac{k}{u} \cdot c(x)$  est une fonction pénalisante relative à  $A'$ . En effet :

(i)  $p_k(x) = \frac{k}{u} \cdot c(x) = 0 \iff x \in A'$

(ii)  $p_k(x) = \frac{k}{u} \cdot c(x) > 0 \iff x \notin A'$

(iii)  $\forall x \notin A', p_k(x) = \frac{k}{u} \cdot c(x) < p_{k+1}(x) = \frac{k+1}{u} \cdot c(x)$

car  $\frac{k}{u} < \frac{k+1}{u}$  et  $(x \notin A' \iff c(x) > 0)$

(iv)  $p_k(x) \xrightarrow[k \rightarrow \infty]{} \infty$  pour tout  $x$  fixé,  $x \notin A'$

car  $\lim_{k \rightarrow \infty} \frac{k}{u} = +\infty$

On en déduit donc que la décomposition barycentrique du problème se ramène à une pénalisation extérieure du problème (P).

On a donc, dans ces conditions, la proposition suivante :

- Proposition

Avec les notations précédentes et sous les hypothèses H1, H2, H3, H4, tout point d'accumulation  $X^*$  de la suite  $\{ X^{k+1} \mid k \in S \}$  générée par la "méthode des colonnes" est solution optimale du problème P

V - 1 - 5 - Méthode des colonnes - Etude d'un exemple

Afin d'illustrer le résultat précédemment trouvé, nous allons résoudre, en utilisant la méthode des colonnes, le programme mathématique suivant :

$$\text{Max } -x_1^2 - x_2^2$$

$$(x_1 + x_2 - 1)^2 = 0$$

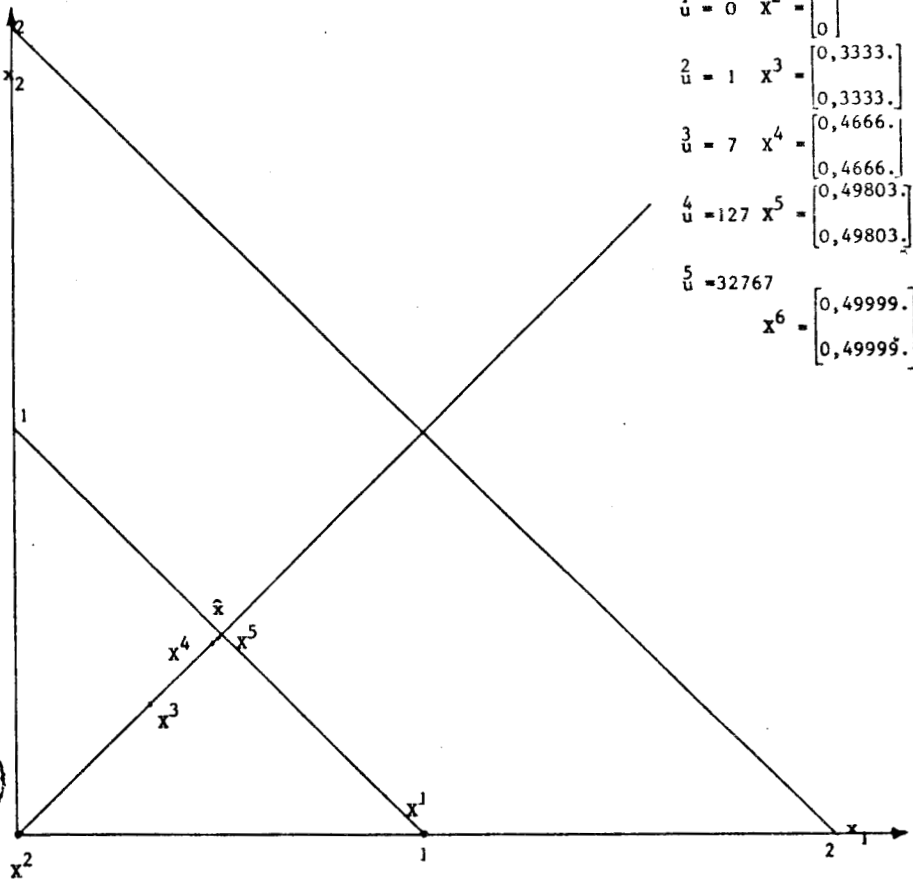
$$x_1 + x_2 \leq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Graphiquement, on obtient :

$$\bar{x} = \begin{bmatrix} 0,5 \\ 0,5 \end{bmatrix} \quad \text{et} \quad f(\bar{x}) = -0,5$$



$$\begin{aligned} \frac{1}{u} = 0 \quad x^2 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \frac{2}{u} = 1 \quad x^3 &= \begin{bmatrix} 0,3333. \\ 0,3333. \end{bmatrix} \\ \frac{3}{u} = 7 \quad x^4 &= \begin{bmatrix} 0,4666. \\ 0,4666. \end{bmatrix} \\ \frac{4}{u} = 127 \quad x^5 &= \begin{bmatrix} 0,49803. \\ 0,49803. \end{bmatrix} \\ \frac{5}{u} = 32767 \quad x^6 &= \begin{bmatrix} 0,49999. \\ 0,49999. \end{bmatrix} \end{aligned}$$



Choisissons initialement  $X^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ . On en déduit que :

$$\bar{u} = X^1 \text{ et } c(X^1) = 0.$$

A l'étape 1, on a nécessairement  $\bar{u} = 0$ . ( $X^1$  seul générateur de  $P_2(K)$ )

On sait (remarque 2 - § V - 1 - 3) que  $X^1$  restera solution optimale du programme directeur tant que le générateur entré  $X^{k+1}$  est tel que :

$$c(X^{k+1}) > 0$$

Comme  $\bar{u} = 0$ , on a :  $X^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$  et  $c(X^2) = 1$  ;  $f(X^2) = 0$

Si  $c(X^{k+1}) > 0$ , alors la matrice de base à l'étape suivante vaut :

$$\begin{bmatrix} -c(X^1) & -c(X^{k+1}) \\ 1 & 1 \end{bmatrix}$$

$$\text{d'où : } \bar{u}^{k+1} = \frac{f(X^{k+1}) - f(X^1)}{c(X^{k+1})}$$

Le sous-programme donnant  $X^{k+1}$  s'écrit :

$$P_3(K) \quad \begin{cases} \text{Max } f(x) - \bar{u}^k \cdot c(x) \\ x_1 + x_2 \leq 2 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Remarquons qu'à partir de l'étape 2, le point de gradient nul est solution optimale de  $P_3(K)$ . En effet, par raison de symétrie, on a  $x_1 = x_2$ .

$$\nabla f(x) - \bar{u}^k \nabla c(x) = 0 \quad \Leftrightarrow \quad x_1 = x_2 = \frac{\bar{u}^k}{1 + 2\bar{u}^k} = X_1^{k+1} = X_2^{k+1}$$

$$\Rightarrow \quad X^{k+1} = \begin{bmatrix} X_1^{k+1} \\ X_2^{k+1} \end{bmatrix} \in B \text{ et } X^{k+1} \notin A'$$

Les points  $\{ X^{k+1} \mid k \in S \}$  générés par la méthode des colonnes sont situés sur le segment  $[ X^2, \hat{x} ]$ . On vérifie que la suite  $\{ \frac{k}{x} \mid k \in S \}$  est stationnaire :  $\frac{k}{x} = X^1$

L'algorithme est infini et on a :

$$\lim_{k \rightarrow \infty} X_1^{k+1} = \lim_{k \rightarrow \infty} X_2^{k+1} = \frac{1}{2}$$

$X^* = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$  point d'accumulation de la suite  $\{ X^{k+1} \mid k \in S \}$  est solution optimale de P

On peut encore noter que :

$$c(X^{k+1}) = \frac{1}{(1 + 2 \frac{k}{u})^2} \quad \text{et} \quad f(X^{k+1}) = -\frac{2 \frac{k}{u^2}}{(1 + 2 \frac{k}{u})^2}$$

et  $\frac{k}{u} = 2 \frac{k}{u^2} + 4 \frac{k}{u} + 1$

Dans ce cas, la convergence est très rapide. En effet :

$$X^5 = \begin{bmatrix} 0,4999923 \\ 0,4999923 \end{bmatrix} \quad \text{et} \quad f(X^5) = -0,499985\dots$$

l'optimum est atteint à  $10^{-5}$  près.

Remarque

Dans le cas général, la convergence numérique s'avère lente. Ceci est dû au fait que les valeurs de  $\frac{k}{u}$  ne sont pas contrôlées mais calculées à partir des conditions de Kuhn et Tucker.

V - 2 - PARAMETRISATION DU PROBLEME P

V - 2 - 1 - Paramétrisation du problème

Nous avons indiqué au paragraphe précédent les principales difficultés rencontrées en utilisant les méthodes barycentriques. La propriété particu-

lière de la fonction  $c$  (H4 - ii) va toutefois nous permettre d'introduire une paramétrisation du problème équivalent P1.

Définissons le problème P1 ( $\alpha$ ) suivant :

$$P1(\alpha) \quad \begin{cases} \text{Max } f(x) & \alpha > 0 \\ a'(x) \geq -\alpha & \text{avec } \alpha \rightarrow 0 \\ b(x) \geq 0 \end{cases}$$

On notera  $x(\alpha)$  une solution optimale de P1 ( $\alpha$ )

$$\begin{aligned} \text{Remarque : } \underline{1} \quad a'(x) \geq -\alpha & \iff 0 \leq c(x) \leq \alpha \\ \underline{2} \quad \alpha \rightarrow 0 & \implies c(x) \rightarrow 0 \end{aligned}$$

### V - 2 - 2 - Algorithme de paramétrisation

1 - On initialise l'algorithme avec  $\alpha = \alpha_0 > 0$

2 - A l'étape 1, pour  $\alpha = \alpha_1 > 0$  , on résout P1 ( $\alpha$ )

3 - On choisit alors  $\alpha = \alpha_{1+1} < \alpha_1$  retour à 2 avec  $l = l + 1$

l'algorithme génère une suite infinie  $\{ (x[\alpha_l], \alpha_l) \mid l \in \mathbb{N} \}$

### V - 2 - 3 - Proposition

- Avec les notations précédentes et sous les hypothèses H1, H2, H3, H4, si  $(x^*, \alpha^*)$  est un point d'accumulation de la suite  $\{ (x(\alpha_l), \alpha_l) \mid l \in \mathbb{N} \}$  générée par l'algorithme, alors on a :

$$\alpha^* = 0 \implies x^* \text{ est solution optimale de P}$$

En effet, désignons par  $\tilde{x}$  une solution optimale de P.

On a :  $s'(\tilde{x}) = 0 \leq \alpha_l$  ,  $\forall l \in \mathbb{N}$ .

$\rightarrow \hat{x}$  est solution réalisable de P1 ( $\alpha_1$ ),  $\forall 1 \in \mathbb{N}$ .

Il s'ensuit que :  $\forall 1 \in \mathbb{N}, f(\hat{x}) \leq f(x(\alpha_1))$

Comme  $f$  est continue on a donc  $f(\hat{x}) \leq f(x^*)$

En outre :

$\forall 1 \in \mathbb{N}, x(\alpha_1) \in B$   
}  $\rightarrow x^* \in B$   
 $B$  compact

De plus :

$\forall 1 \in \mathbb{N}, 0 \leq a'(x(\alpha_1)) \leq \alpha_1$   
 $\alpha^* = 0$   
 $a'$  continue sur  $B$  }  $\rightarrow a'(x^*) = 0$

On en déduit que :  $x^* \in A' \cap B$  d'où  $f(x^*) \leq f(\hat{x})$

Soit finalement  $f(\hat{x}) = f(x^*)$

$x^*$  est bien solution optimale de P

Remarque 1

En utilisant les fonctions multivoques, cette proposition est évidente.

Remarque 2

- L'algorithme n'est pas modifié si on cherche une solution à  $\epsilon$  près de P  
(cf chap. VI)

- On peut noter au niveau des propriétés une similitude avec les méthodes de pénalisation extérieure ( [23] [25] [34] )

- a -  $\forall 1 \in \mathbb{N}, f(\hat{x}) \leq f(x(\alpha_1))$

- b -  $\forall 1 \in \mathbb{N}, f(x(\alpha_{1+1})) \leq f(x(\alpha_1))$

- c -  $a'(x^*) = 0$

Remarque 3

Si  $\exists l_0 \in \mathbb{N} \mid a'(x(\alpha_{l_0})) = 0$  alors  $x(\alpha_{l_0})$  est solution optimale du problème P. On peut donc modifier l'algorithme comme suit.

3' - Deux cas sont alors possibles :

a -  $a'(x(\alpha_1)) = 0$  -  $x(\alpha_1)$  est solution optimale de P - FIN

b -  $a'(x(\alpha_1)) < 0$  - retour à 2 - avec  $\alpha = \alpha_{1+l} < \alpha_1$  et  $l = l+1$

V - 2 - 4 - Résolution de P1 (α)

Posons :  $\forall \alpha > 0, \forall x \in \mathbb{R}^n, a(x) = a'(x) + \alpha = \alpha - c(x)$   
 d'après H4 - i, a est concave (et continue) sur  $\mathbb{R}^n$ .

P1 (α) peut alors se mettre sous la forme :

$$P'1(\alpha) \left[ \begin{array}{l} \text{Max } f(x) \\ a(x) \geq 0 \\ b(x) \geq 0 \end{array} \right.$$

On voit de façon immédiate que les hypothèses H1, H2, H3 de l'algorithme mixte sont vérifiées. Le programme directeur s'écrit :

$$P2(\alpha, k, K) \left[ \begin{array}{l} \text{Max } \nabla f(x) \cdot x^K \cdot \lambda_K \\ A^K \cdot \lambda_K \geq -\alpha \\ e^K \cdot \lambda_K = 1 \\ \lambda_K \geq 0 \end{array} \right.$$

$A^K$  matrice dont la colonne j est  $A^j = a'(x^j) = -c(x^j)$

Il s'agit d'un programme linéaire à deux contraintes - Sa résolution par la méthode simpliciale nous donne de façon immédiate les variables duales. Les conditions de KUHN et TUCKER donnent dans ce cas :

$$\exists (k_u \geq 0, k_o) \text{ tels que :}$$

$$1 - \forall j \in K, \forall f(x) \cdot x^j + k_u \cdot a'(x^j) + k_o \leq 0$$

$$2 - \forall f(x) \cdot k_u = k_u \alpha - k_o$$

$$3 - k_u \cdot A^K \cdot \lambda(K) = -k_u \cdot \alpha$$

et le sous-programme s'écrit :

$$P3(\alpha, k, K) \left[ \begin{array}{l} \text{Max } \nabla f(x) \cdot x + k_u \cdot a'(x) \\ b(x) \geq 0 \end{array} \right]$$

Enfin :  $A' \neq \emptyset$  et  $\forall x \in \mathbb{R}^n, c(x) \geq 0$ .

Compte tenu des hypothèses on peut affirmer que :

$$\exists \bar{x} \in B \mid c(\bar{x}) < \alpha \text{ ceci } \forall \alpha > 0$$

Donc :  $\forall \alpha > 0, \exists \bar{x} \mid a(\bar{x}) > 0$  et  $b(\bar{x}) \geq 0$ .

l'hypothèse H'4 de l'algorithme mixte est satisfaite si on introduit un tel point  $\bar{x}$  dans l'ensemble des générateurs initiaux.

Nous utiliserons donc l'algorithme mixte pour résoudre  $P1(\alpha), \forall \alpha > 0$ .

### V - 3 - CONCLUSION

Le problème particulier P peut donc être résolu en introduisant l'algorithme précédent de paramétrisation. Toutefois la résolution de  $P1(\alpha)$  par l'algorithme mixte nécessite une séquence théoriquement infinie de problèmes à résoudre.



Au chapitre suivant, nous allons nous intéresser à la résolution approchée des sous-programmes à traiter. Des solutions approchées seront obtenues après un nombre fini de calculs.

CHAPITRE VI

DETERMINATION DE

XXXXXXXXXXXXXXXXXXXX

SOLUTIONS APPROCHEES

XXXXXXXXXXXXXXXXXXXX

## VI- 1 - PARAMETRISATION

- Dans l'algorithme de paramétrisation, défini au paragraphe V - 2 - 2, modifions comme suit la phase 2 :

2' - A l'étape 1, pour  $\alpha = \alpha_1 > 0$ , on résoud  $p_1(\alpha)$  à  $\epsilon_1$  près.

Soit  $x(\alpha_1)$  la solution approchée trouvée.

- On obtient alors le résultat suivant :

### - Proposition

Avec les notations et sous les mêmes hypothèses qu'aux paragraphes V - 1 et V - 2, si  $(x^*, \alpha^*, \epsilon^*)$  désigne un point d'accumulation de la suite

$\{x(\alpha_1), \alpha_1, \epsilon_1 \mid 1 \in \mathbb{N}\}$  générée par l'algorithme modifié de paramétrisation, alors on a :

$\alpha^* = 0 \Rightarrow x^*$  est solution approchée à  $\epsilon^*$  près de (P)

En effet :  $\forall 1 \in \mathbb{N}, f[x(\alpha_1)] + \epsilon_1 \geq f(\bar{x})$

Comme  $f$  est continue sur  $B$  compact :

$$f(x^*) + \epsilon^* \geq f(\bar{x})$$

En outre  $\alpha^* = 0 \Rightarrow a(x^*) = 0$ . Par définition de  $\bar{x}$  on a donc :

$$f(x^*) \leq f(\bar{x})$$

Soit finalement :

$$f(x^*) \leq f(\bar{x}) \leq f(x^*) + \epsilon^*$$

## VI- 2 - SOUS-OPTIMISATION UNIDIMENSIONNELLE

Dans l'algorithme de sous-optimisation unidimensionnelle défini au paragraphe IV - 1 - 2, définissons le point  $x^{k+1}$  de la façon suivante :

$$- a - \frac{k+1}{x} \in \left[ \frac{k}{x}, \frac{k}{z} \right]$$

$$- b - f\left(\frac{k+1}{x}\right) \geq f\left(\frac{k}{x}\right)$$

$$- c - f\left(\frac{k+1}{x}\right) \geq \text{Max} \{ f(x) \mid x \in \left[ \frac{k}{x}, \frac{k}{z} \right] \} - \epsilon_k \quad (\epsilon_k > 0)$$

On obtient la proposition suivante :

Propriété 3'

Moyennant les hypothèses indiquées au paragraphe IV - 1 - 1, quel que soit le point d'accumulation  $(x^*, z^*, \epsilon^*)$  de la suite infinie  $(\frac{k}{x}, \frac{k}{z}, \epsilon_k \mid k \in \mathbb{N})$  générée par l'algorithme, on a :

$$\epsilon^* = 0 \quad \rightarrow \quad \forall f(x^*) \cdot (z^* - x^*) \leq 0$$

- Démonstration

Soit  $N' \in \mathbb{N}$  définissant toujours une sous-suite infinie convergente :

$$\left\{ \left( \frac{k}{x}, \frac{k}{z}, \epsilon_k \right) \mid k \in N' \right\} \rightarrow (x^*, z^*, \epsilon^*)$$

Par définition de  $\frac{k+1}{x}$  on a :

$$\forall \theta, 0 \leq \theta \leq 1, f\left(\frac{k+1}{x}\right) \geq f\left(\frac{k}{x} + \theta \left[ \frac{k}{z} - \frac{k}{x} \right]\right) - \epsilon_k$$

D'où  $\forall k \in N'$  et  $\forall k' \in N'$  avec  $k' \geq k+1$  :

$$\forall \theta, 0 \leq \theta \leq 1, f\left(\frac{k'}{x}\right) \geq f\left[\frac{k}{x} + \theta \left(\frac{k'}{z} - \frac{k}{x}\right)\right] - \epsilon_k$$

Soit en passant à la limite :

$$\forall \theta, 0 \leq \theta \leq 1, f(x^*) \geq f[x^* + \theta(z^* - x^*)] - \epsilon^*$$

On en déduit que, pour  $\epsilon^* = 0$ ,  $x^*$  maximise  $f$  sur le segment  $[x^*, z^*]$

La suite du raisonnement se poursuit alors de la même façon.

Conséquences

- En choisissant  $x^{k+1}$  comme on vient de l'indiquer, on en déduit que :
- a) La proposition IV - 2 - 2 relative à l'algorithme de FRANK et WOLFE est encore vérifiée.
  - b) La proposition IV - 3 - 9 relative à l'algorithme mixte également.
- $d^{k+1}(K)$  tend vers zéro indépendamment du choix de  $x^{k+1}$ .

Ces résultats sont très importants sur le plan numérique, les maximisations sur un segment s'effectuant en général de façon approchée après un nombre fini de calculs.

VI - 3 - ALGORITHME MIXTE - RESOLUTION APPROCHEE DU PROGRAMME AUXILIAIRE

Comme dans les méthodes de décomposition rappelées au chapitre III, on peut, dans l'algorithme mixte, se contenter d'une solution approchée de  $P_3(k, K)$  définie comme suit :

$$a - \forall f \left( \begin{matrix} k \\ \bar{x} \end{matrix} \right) . X^s + \overset{k}{u} . a (X^s) + \overset{k_0}{u} \geq 0$$
$$b - \forall f \left( \begin{matrix} k \\ \bar{x} \end{matrix} \right) . X^s + \overset{k}{u} . a (X^s) \geq \text{Max} \{ \forall f \left( \begin{matrix} k \\ \bar{x} \end{matrix} \right) . x + \overset{k}{u} . a (x) \mid b (x) \geq 0 \} - \beta_k$$

avec  $\beta_k > 0$   
 $\beta_k \rightarrow \beta^*$  quand  $k \rightarrow \infty$

En effet l'encadrement obtenu au paragraphe IV - 3 - 6 s'écrit cette fois :

$$0 \leq f(\bar{x}) - f \left( \begin{matrix} k \\ \bar{x} \end{matrix} \right) \leq d^s + \nabla f \left( \begin{matrix} k \\ \bar{x} \end{matrix} \right) . \left( \bar{z} - \begin{matrix} k \\ \bar{x} \end{matrix} \right) + \beta_k.$$

Dans ces conditions :

- i si  $\beta^* = 0$  on a encore :  $\lim_{k \rightarrow \infty} f \left( \begin{matrix} k \\ \bar{x} \end{matrix} \right) = f(\bar{x})$

ii - si  $\beta^* > 0$  alors  $\lim_{k \rightarrow \infty} f(x^k) \geq f(\hat{x}) - \beta^*$

On obtient une solution approchée à  $\beta^*$  près du problème P

#### VI-4-CONSEQUENCES

1 - L'algorithme mixte nécessite à chaque étape le traitement de deux programmes en général non linéaires, à savoir P3 (k, K) et une maximisation sur un segment.

D'après ce qui précède, ces deux sous-programmes peuvent être résolus de façon approchée. Chaque étape de cet algorithme peut donc être ramenée à une séquence finie de calculs.

2 - Dans les mêmes conditions, chaque étape de l'algorithme de paramétrisation nécessitera un nombre fini de calculs lors de la résolution de P1 ( $\alpha_1$ ) à  $\epsilon_1$  près.

#### VI - 5 - ALGORITHME MIXTE - RESOLUTION APPROCHEE DU PROGRAMME AUXILIAIRE DANS UN CAS PARTICULIER

##### VI - 5 - 1 - Cas particulier étudié

Sur le plan numérique, nous nous intéressons tout particulièrement au cas suivant :

$$b(x) \geq 0 \iff \{ B \cdot x = b, x \geq 0 \}$$

Dans ces conditions l'algorithme mixte (IV - 3) (et par voie de conséquence l'algorithme de paramétrisation (V - 2)) demande de résoudre un sous-programme du type :

$$P3(k, K) \left[ \begin{array}{l} \text{Max } \nabla f \left( \begin{smallmatrix} k \\ \bar{x} \end{smallmatrix} \right) \cdot x + \begin{smallmatrix} k \\ \bar{u} \end{smallmatrix} \cdot a(x) \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

Moyennant une hypothèse supplémentaire sur la fonction  $a$ , on peut chercher une solution approchée à  $\beta_k > 0$  près en utilisant l'algorithme de FRANK et WOLFE (IV - 2). Ce choix se justifie par le fait qu'à chaque étape, on possède un encadrement de l'optimum.

VI - 5 - 2 - Résolution approchée de P3 (k, K)

Aux hypothèses retenues au paragraphe IV - 3 ajoutons :

H5 -  $a : \mathbb{R}^n \rightarrow \mathbb{R}^p$  continûment différentiable sur  $\mathbb{R}^n$

On peut appliquer l'algorithme de FRANK et WOLFE comme suit :

i - On initialise avec  $\bar{y} \in B$

ii - A l'étape  $m$  on dispose de  $\bar{y}^m \in B$

a - On résoud le programme linéaire suivant :

$$\left[ \begin{array}{l} \text{Max } \nabla f \left( \begin{smallmatrix} k \\ \bar{x} \end{smallmatrix} \right) \cdot x + \begin{smallmatrix} k \\ \bar{u} \end{smallmatrix} \cdot \nabla a \left( \begin{smallmatrix} m \\ \bar{y} \end{smallmatrix} \right) \cdot x \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

Soit  $\bar{y}'^m$  la solution trouvée.

b - On détermine  $\bar{y}^{m+1}$  vérifiant :

$$* \bar{y}^{m+1} \in [\bar{y}^m, \bar{y}'^m]$$

$$* \nabla f \left( \begin{smallmatrix} k \\ \bar{x} \end{smallmatrix} \right) \cdot \bar{y}^{m+1} + \begin{smallmatrix} k \\ \bar{u} \end{smallmatrix} \cdot a \left( \begin{smallmatrix} m+1 \\ \bar{y} \end{smallmatrix} \right) \geq \nabla f \left( \begin{smallmatrix} k \\ \bar{x} \end{smallmatrix} \right) \cdot \bar{y}^m + \begin{smallmatrix} k \\ \bar{u} \end{smallmatrix} \cdot a \left( \begin{smallmatrix} m \\ \bar{y} \end{smallmatrix} \right)$$

$$* \nabla f \left( \begin{smallmatrix} k \\ \bar{x} \end{smallmatrix} \right) \cdot \bar{y}^{m+1} + \begin{smallmatrix} k \\ \bar{u} \end{smallmatrix} \cdot a \left( \begin{smallmatrix} m+1 \\ \bar{y} \end{smallmatrix} \right) \geq \text{Max} \{ \nabla f \left( \begin{smallmatrix} k \\ \bar{x} \end{smallmatrix} \right) \cdot x + \begin{smallmatrix} k \\ \bar{u} \end{smallmatrix} \cdot a(x) \mid x \in [\bar{y}^m, \bar{y}'^m] \}$$

avec  $\epsilon_m > 0$

$\epsilon_m$

Retour à ii avec 
$$\begin{cases} \bar{m} = \bar{m}+1 \\ \varepsilon_{\bar{m}} = \varepsilon_{\bar{m}+1} \end{cases}$$

On sait (VI - 2) que si  $\varepsilon^* = 0$  alors la suite :

$$\{ \nabla f(\bar{x}) \cdot \bar{y} + \bar{u} \cdot a(\bar{y}) \mid m \in \mathbb{N} \} \text{ converge vers l'optimum de P3 (k, K)}$$

L'algorithme est infini, montrons que l'on peut trouver après un nombre fini de calculs, une solution approchée de P3 (k, K) à  $\beta_k > 0$  près.

Comme nous l'avons indiqué au paragraphe VI - 3,  $X^s$  doit d'abord vérifier :

$$\nabla f(\bar{x}) \cdot X^s + u^k \cdot a(X^s) + \bar{u}^0 \geq 0$$

Notons d'abord que par définition de  $\bar{y}^{m+1}$  la suite :

$$\{ \nabla f(\bar{x}) \cdot \bar{y} + \bar{u} \cdot a(\bar{y}) + \bar{u}^0 \mid m \in \mathbb{N} \} \text{ est monotone non décroissante.}$$

$$\left. \begin{aligned} \text{Or : } \nabla f(\bar{x}) \cdot \bar{z} + \bar{u}^0 = 0 \\ a(\bar{z}) \geq A^k \cdot \lambda(K) \geq 0 \end{aligned} \right\} \Rightarrow \nabla f(\bar{x}) \cdot \bar{z} + \bar{u} \cdot a(\bar{z}) + \bar{u}^0 \geq 0$$

Si on initialise l'algorithme précédent avec :

$$\bar{y} = \bar{z} \text{ on a :}$$

$$\forall m \in \mathbb{N}, \nabla f(\bar{x}) \cdot \bar{y} + \bar{u} \cdot a(\bar{y}) + \bar{u}^0 \geq 0$$

En outre on a, à chaque étape, l'encadrement suivant :

$$\begin{aligned} 0 \leq \{ \text{Max } \nabla f(\bar{x}) \cdot x + \bar{u} \cdot a(x) \mid x \in B \} + \bar{u}^0 - [\nabla f(\bar{x}) \cdot \bar{y} + \bar{u} \cdot a(\bar{y}) + \bar{u}^0] \\ \leq [\nabla f(\bar{x}) + \bar{u} \cdot \nabla a(\bar{y})] \cdot [\bar{y}' - \bar{y}] \end{aligned}$$

de plus, si  $(\bar{y}^*, \bar{y}')$  est un point d'accumulation de la suite :

$$\{ (\bar{y}^m, \bar{y}'^m) \mid m \in \mathbb{N} \} \text{ on a :}$$

$$[\nabla f(\bar{x}) + \bar{u} \cdot \nabla a(\bar{y}')] \cdot [\bar{y}'^* - \bar{y}^*] = 0 \quad (\varepsilon^* = 0) \quad \text{[SIV-1]}$$

On peut donc affirmer qu'après un nombre fini d'étapes,  $m_0$ , on aura :

$$0 \leq [\nabla f(\bar{x}) + \bar{u} \cdot \nabla a(\bar{y}^0)] \cdot [\bar{y}^{m_0} - \bar{y}^0] \leq \beta_k$$

Il suffit alors de prendre :

$$X^s = \bar{y}^0$$



On en déduit donc qu'une solution approchée à  $\beta_k > 0$  près de  $P_3(k, K)$  peut être trouvée après un nombre fini de calculs.

ESSAIS NUMERIQUES  
XXXXXXXXXXXXXXXXXXXX

Le code des méthodes a été réalisé en FORTRAN IV ( toutes les variables étant en précision étendue ) et exploité sur un ordinateur IBM 360-40 de 128 K-octets de mémoire centrale.

Pour ce matériel le temps de résolution du programme test de COLVILLE , en précision étendue, est de 710 secondes. Nous appellerons temps de résolution le rapport obtenu en divisant le temps réel de résolution par 710 .

La méthode des colonnes ( § III - 3 ) et l'algorithme mixte ( § IV - 3 ) ont été étudiés numériquement ( § VII - 1 ). Au paragraphe VII - 2 les deux méthodes ont été comparées. Pour les problèmes tests envisagés ( annexe I ) l'algorithme mixte est plus performant que la méthode des colonnes. Toutefois la convergence numérique s'avère lente dans certains cas.

Au chapitre VIII nous étudierons la paramétrisation ( § V - 2 ). Les problèmes tests sont regroupés dans l'annexe II. Des variantes seront apportées afin d'améliorer la convergence numérique ( § VIII - 4 ).

CHAPITRE VII

ALGORITHME MIXTE

XXXXXXXXXXXXXXXXXXXX

METHODE DES COLONNES

XXXXXXXXXXXXXXXXXXXX

VII - 1 - CONSIDERATIONS NUMERIQUES

VII - 1 - 1 - Introduction

La programmation des deux algorithmes a nécessité la résolution des problèmes suivants :

- Choix des générateurs initiaux et
- Résolution du programme auxiliaire à  $\epsilon$  près.

Au paragraphe VII - 1 - 2, nous allons indiquer la méthode choisie pour sélectionner les générateurs initiaux et en montrer l'intérêt. La résolution approchée du programme auxiliaire sera abordée au paragraphe VII- 1 - 3.

Les deux algorithmes fournissent à chaque étape un encadrement de l'optimum. Son efficacité comme test d'arrêt sera étudiée au paragraphe VII - 1 - 4.

Les caractéristiques des sept problèmes tests utilisés figurent en annexe I.

VII - 1 - 2 - Choix des générateurs initiaux.

Dans le cas particulier étudié numériquement (1 seule contrainte non linéaire) l'option retenue a été une création automatique de générateurs initiaux. On sait que l'un d'entre eux, soit  $X^1$ , doit être réalisable ; il a été obtenu, lorsqu'il existe, en résolvant le programme suivant :

$$\left[ \begin{array}{l} \text{Max } a(x) \\ b(x) \geq 0 \end{array} \right.$$

La convergence numérique des deux algorithmes s'est avérée relativement lente en partant de ce seul point. Une amélioration a été obtenue en introduisant les deux générateurs supplémentaires suivants (s'ils existent) :

- $X^2$  solution à  $\epsilon$  près de : 
$$\begin{cases} \text{Max } f(x) \\ b(x) \geq 0 \end{cases}$$
- $X^3$  tel que  $a(X^3) = 0$  à  $\epsilon$  près,  $X^3 \in [X^1, X^2]$

VII - 1 - 3 - Résolution approchée du programme auxiliaire.

Le programme auxiliaire de la méthode des colonnes ou de l'algorithme mixte a été résolu à  $\epsilon$  près en utilisant l'algorithme de FRANK et WOLFE. Pour cet algorithme on a fait varier le nombre maximum d'itérations; soit  $J$  ce nombre. Des essais numériques, il ressort que la précision des résultats obtenus pour le problème posé ( P ) varie avec les valeurs de  $J$ .

VII - 1 - 3 - 1 - Cas de l'algorithme mixte.

Dans un premier temps, le sous programme a été résolu à  $\epsilon$  près sans limitation, les temps de résolution étaient alors relativement élevés (voisin d'un temps standard pour une précision de  $10^{-3}$ ).

On a été amené à étudier une limitation du nombre d'étapes. Il en est ressorti que les faibles valeurs de  $J$  (inférieures à 30) permettaient d'obtenir bien plus rapidement la même précision (temps moyen : 0,35 pour  $10^{-3}$ ). La valeur  $J = 1$  a donné pour certains problèmes de très bons résultats.  $J$  supérieur à 30 n'apporte aucun gain, ni en précision, ni en temps. On a envisagé l'alternance  $J = 1, J = J_0$  pour des valeurs de  $J_0$  inférieures à 30.

Le tableau suivant, caractéristique des résultats obtenus dans l'ensemble, donne l'illustration du gain de temps.

TEMPS DE RESOLUTION NECESSAIRE POUR ATTEINDRE

UNE PRECISION ABSOLUE DE  $\Delta f = 10^{-3}$

Temps maximum de résolution 0, 282 (200 secondes)

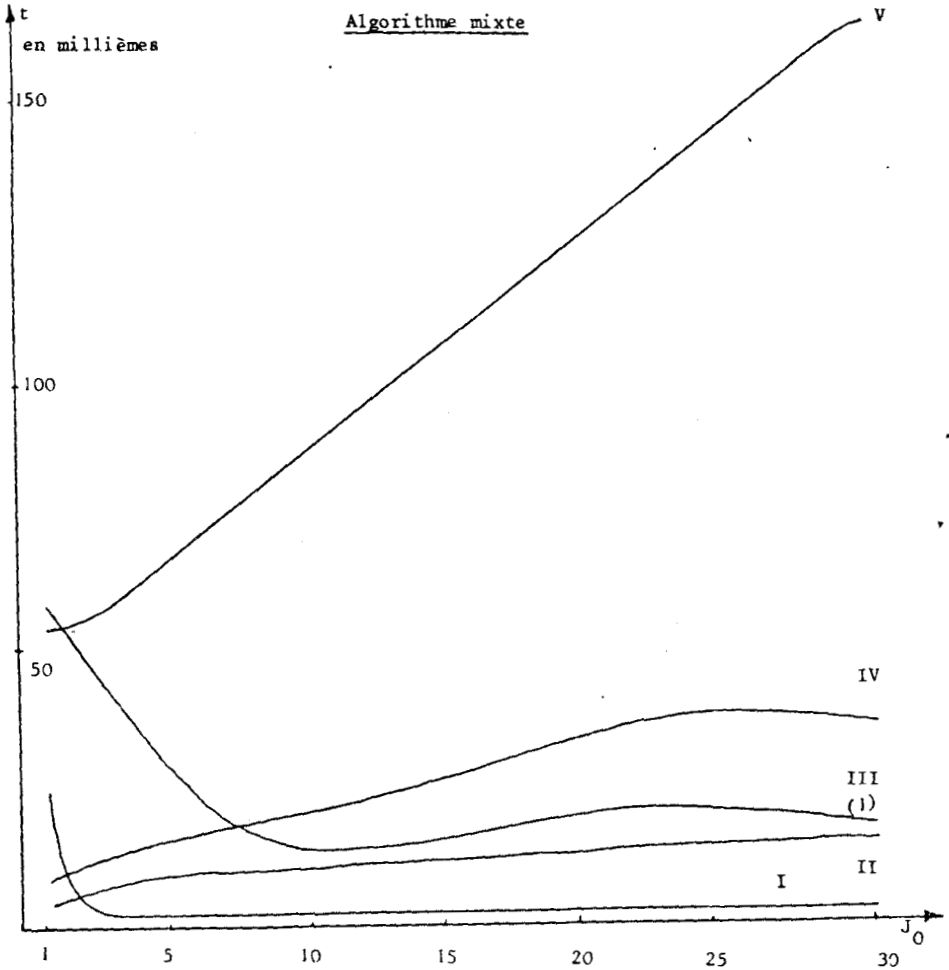
TYPE		A	B	B / A
PROBLEME	f ( $\hat{x}$ )	ALTERNANCE 1 - 15	J CONSTANT = 15	
I	3,7321	0,001	0,011	11
II	-0,2540	0,011	0,033	3
IV	14,9706	0,025	0,039	1,5
V	-2,1806	0,106	0,262	2,5
VI	16,1231	0,080	pas atteint (1)	
VII	175,5996	0,123	pas atteint (2)	

1) Meilleure valeur atteinte : 16,0635

2) Meilleure valeur atteinte : 175,5848



Le graphique ci-dessous donne l'évolution du temps de résolution nécessaire à l'obtention d'une précision de  $10^{-3}$  dans le cas de l'alternance pour des valeurs de  $J_0$  comprises entre 1 et 30 ( l'étude a porté sur les problèmes I à V ) .



(1) courbe correspondant à une précision de 0,006

Les valeurs de  $J_0$  comprises entre 1 et 10 semblent donner les meilleurs résultats .



VII - 1 - 3 - 2 - Cas de la méthode des colonnes

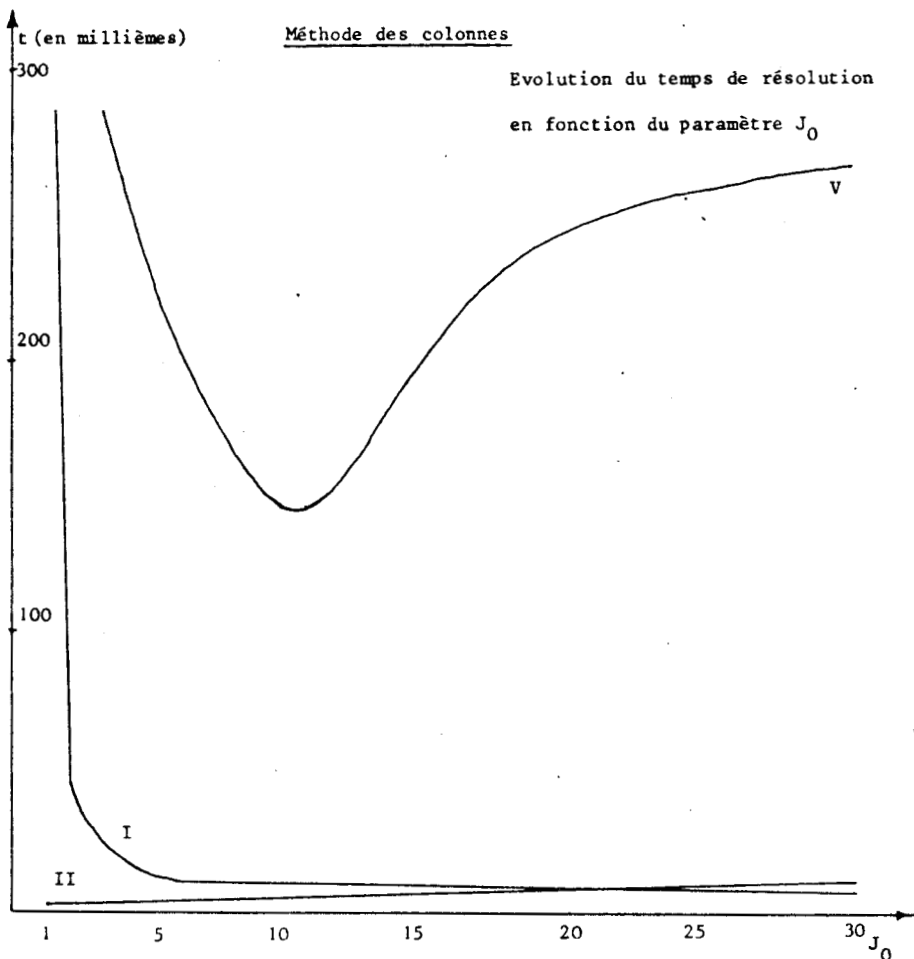
Une recherche des meilleurs paramètres a été également réalisé pour cette méthode. Il en ressort que les principes de la limitation de J et de l'alternance sont également améliorants.

Pour les problèmes I, II et V (les seuls ayant donné des résultats dans un temps limité à 0,282) les résultats obtenus apparaissent dans le tableau et le graphique ci-après.

Temps d'obtention d'une précision  $\Delta f = 10^{-3}$

Temps maximum de résolution 0,282 (200 secondes)

PROBLEME / TYPE	A ALTERNANCE 1 - 15	B J CONSTANT = 15	B / A
I	0,009	0,012	1,3
II	0,006	0,019	3,2
V	0,215	pas atteint	



Les résultats les plus satisfaisants semblent être obtenus pour

$J_0$  voisin de 10.



VII - 1 - 4 - Encadrement de l'optimum

L'étude a porté sur tous les problèmes tests, les tableaux qui vont suivre donnent les résultats pour les problèmes I et IV caractéristiques de tous les autres.

VII - 1 - 4 - 1 - Cas de l'algorithme mixte

Problème n° I

$$f(\bar{x}) = 3,732050$$

$$J_0 = 30$$

k N° de l'itération	A $f(\bar{x}) - f(x^k)$	B $D^S$	B / A
1	0,957	3,745	4
2	0,1320	0,4837	4
3	0,0381	0,1340	4
4	0,0122	0,0212	2
5	0,0043	0,0062	1
6	0,0016	0,0062	4
7	0,0012	0,0035	3
8	0,0010	0,0575	57
9	0,0002	0,0008	4

Problème n° IV

$$f(\bar{x}) = 14,97056$$

$$J_0 = 1$$

k N° de l'itération	A $f(\bar{x}) - f(x^k)$	B $D^s$	B / A
1	3,900	6,502	2
2	0,151	1,469	10
3	0,058	1,798	31
4	0,048	1,557	32
5	0,017	1,363	80
6	0,016	1,322	83
7	0,003	0,603	201
8	0,002	0,399	200
9	0,002	0,351	176
10	0,0007	0,185	264
11	0,0006	0,166	276
12	0,0004	0,100	250



L'encadrement est beaucoup plus fin lorsque  $J_0$  est grand, ceci est à rapprocher du fait que dans les mêmes conditions la convergence numérique est moins rapide en général.

VII - 1 - 4 - 2 - Cas de la méthode des colonnes.

Problème n° I

$$f(\bar{x}) = 3,732050$$

$$J_0 = 15$$

k N° de l'itération	A $f(\bar{x}) - f(x^k)$	B $D^s$	B / A
1	0,0710	1,2961	18
11	0,0012	0,0373	31
21	0,0006	0,0016	3
31	0,0004	0,0109	27
41	0,0003	0,0007	2
51	0,0002	0,0063	32
61	0,00017	0,00050	3
71	0,00014	0,00440	31
81	0,00013	0,00040	3
91	0,00011	0,00030	3



Problème n° IV

$f(\hat{x}) = 14,97056$

$J_0 = 1$

k N° de l'itération	A $f(\hat{x}) - f(x^k)$	B $D^s$	B / A
1	0,1511	3,162	21
2	0,0500	1,425	29
3	0,0240	2,417	100
4	0,0240	1,247	52
5	0,0118	1,252	106
6	0,0116	0,907	78
7	0,0048	0,745	155
8	0,0046	0,364	79
9	0,0045	0,329	73
10	0,0035	0,416	119
11	0,0019	0,264	139
12	0,0013	0,269	207
13	0,0012	0,225	188
14	0,0008	0,160	200
15	0,0005	0,137	274



On peut noter de même que l'encadrement est d'autant plus fin que  $J_0$  est grand.

VII - 1 - 4 - 3 - Conclusion

On peut estimer que l'encadrement obtenu à chaque étape est au moins 10 fois plus important que l'écart réel.

Toutefois, étant donné la très grande disparité des résultats, il ne peut être utilisé efficacement comme test d'arrêt pour des valeurs meilleures que  $10^{-2}$ .



VII - 2 - ETUDE COMPARATIVE DES DEUX METHODES

VII - 2 - 1 - Introduction

L'algorithme mixte et la méthode des colonnes ont été utilisés pour résoudre les mêmes problèmes tests regroupés dans l'annexe I.

Après les avoir comparés en précision absolue dans un temps très limité ( § VII - 2 - 2 ) on effectuera une analyse des résultats obtenus en précision relative (§VII - 2 - 3) dans un temps beaucoup plus important. En outre, on évaluera (§ VII - 2 - 4) le nombre d'itérations nécessaires pour obtenir une précision relative donnée.

VII - 2 - 2 - Etude comparative en précision absolue

Le temps de résolution maximum a été limité à 0,282 (soit 200 sec.) Les temps de résolution figurant dans le tableau ci-dessous, sont ceux qui ont été nécessaires pour atteindre la précision absolue choisie dans un temps de résolution inférieur à 0,282.

De l'analyse des résultats il ressort que, l'algorithme mixte est plus performant que la méthode des colonnes. En effet, les précisions recherchées sont atteintes dans deux fois plus de cas. En outre, l'algorithme mixte converge, numériquement, en moyenne quatre fois plus vite que la méthode des colonnes (dans les cas où les deux méthodes atteignent les résultats.)

Toutefois, on doit reconnaître qu'elle reste une méthode à convergence numérique lente, les résultats cherchés n'ayant pu être atteints dans tous cas pour des problèmes de taille modeste.



COMPARAISON EN PRECISION ABSOLUE

TEMPS DE RESOLUTION MAXIMUM T = 0,282

Précision $\Delta f$	$10^{-3}$				$10^{-4}$		
	Problème	$f(\bar{x})$	Dantzig	Alg. mixte	k	Dantzig	Alg. mixte
I	3,7321	0,008	0,001	8	0,152	0,030	5
II	-0,2540	0,001	0,001	1	0,004	0,001	4
IV	14,9706	0,015	0,008	1,9	0,030	0,018	1,7
V	-2,1806	0,146	0,054	2,7	Pas atteint (1)	0,225	
VI	16,1231	Pas atteint (2)	0,080		Pas atteint	0,161	
VII	175,5996	Pas atteint (3)	0,123		Pas atteint	0,205	

Aucune des deux méthodes n'a permis d'atteindre la précision souhaitée dans le temps fixé pour le problème test n° III .

Meilleure valeur atteinte : - (1) - 2,1811

- (2) 15,9530

- (3) 175,5792

$k = \frac{\text{Temps de résolution Dantzig}}{\text{Temps de résolution Alg. mixte}}$



VII - 2 - 3 - Comparaison des méthodes en précision relative

On a déterminé cette fois les temps de résolution nécessaires pour atteindre une précision relative donnée. Le temps de résolution maximum a été fixé à 1,9 (soit 1350 secondes).

La précision relative maximale atteinte en général est égale à  $10^{-6}$ . Les tableaux qui suivent donnent les résultats pour les précisions relatives  $10^{-3}$ ,  $10^{-4}$  puis  $10^{-5}$ ,  $10^{-6}$ .

COMPARAISON EN PRECISION RELATIVE

TEMPS DE RESOLUTION MAXIMUM T = 1,9

Précision $\frac{\Delta f}{f}$		$10^{-3}$			$10^{-4}$		
Pb.	f ( $\hat{x}$ )	Dantzig	Alg. mixte	k	Dantzig	Alg. mixte	k
I	3,7321	0,004	0,001	4	0,032	0,007	4,6
II	-0,25403	0,001	0,001	1	0,004	0,002	2
III	12,6969	0,128	0,033	3,9	1,830	0,669	2,7
IV	14,9706	0,007	0,004	1,8	0,012	0,006	2
V	-2,1806	0,153	0,040	3,8	0,500	0,199	2,5
VI	16,1231	Pas atteint (1)	0,043		Pas atteint (1)	0,071	
VII	175,5996	0,022	0,018	1,2	0,222	0,050	4,4

Meilleure valeur atteinte (1) = 16,0655



COMPARAISON EN PRECISION RELATIVE  
TEMPS DE RESOLUTION MAXIMUM 1,9

Précision $\frac{\Delta f}{f}$		$10^{-5}$			$10^{-6}$		
Problème	f (x̂)	Dantzig	alg.mixte	k	Dantzig	Alg.mixte	k
I	3,732051	0,338	0,060	5,6	1,876	0,649	2,9
II	0,2540331	0,006	0,004	1,5	0,014	0,007	2
IV	14,970562	0,032	0,013	2,5	0,280	0,023	12,2
VI	16,12305	Pas atteint (1)	0,100		Pas atteint (1)	0,132	
VII	175,5996	Pas atteint (2)	0,118		Pas atteint (2)	0,192	

Remarque:

Les précisions  $10^{-5}$  et  $10^{-6}$  n'ont été atteintes par aucune des deux méthodes pour les problèmes III et V .

(1) Meilleure valeur atteinte 16,0655

(2) Meilleure valeur atteinte 175,5935

On peut faire les remarques suivantes :

1 - L'algorithme mixte atteint dans tous les cas les précisions relatives  $10^{-3}$  et  $10^{-4}$ .

2 - Le rapport  $k$  croît de la manière suivante quand la précision augmente:

$\frac{\Delta f}{f}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
$k$	2,6	3	3,2	5,7

3 - Si l'algorithme mixte converge numériquement en moyenne 4 fois plus vite que la méthode des colonnes, cette convergence est toutefois lente au voisinage de l'optimum.

VII - 2 - 4 - Détermination du nombre d'itérations nécessaires pour atteindre un résultat donné.

Dans le tableau suivant, nous avons indiqué le nombre d'itérations nécessaires pour atteindre une précision relative déterminée. Les représentations graphiques qui suivent en donnent l'illustration.

Les valeurs retenues correspondent aux cas donnant les meilleurs résultats pour chaque méthode.

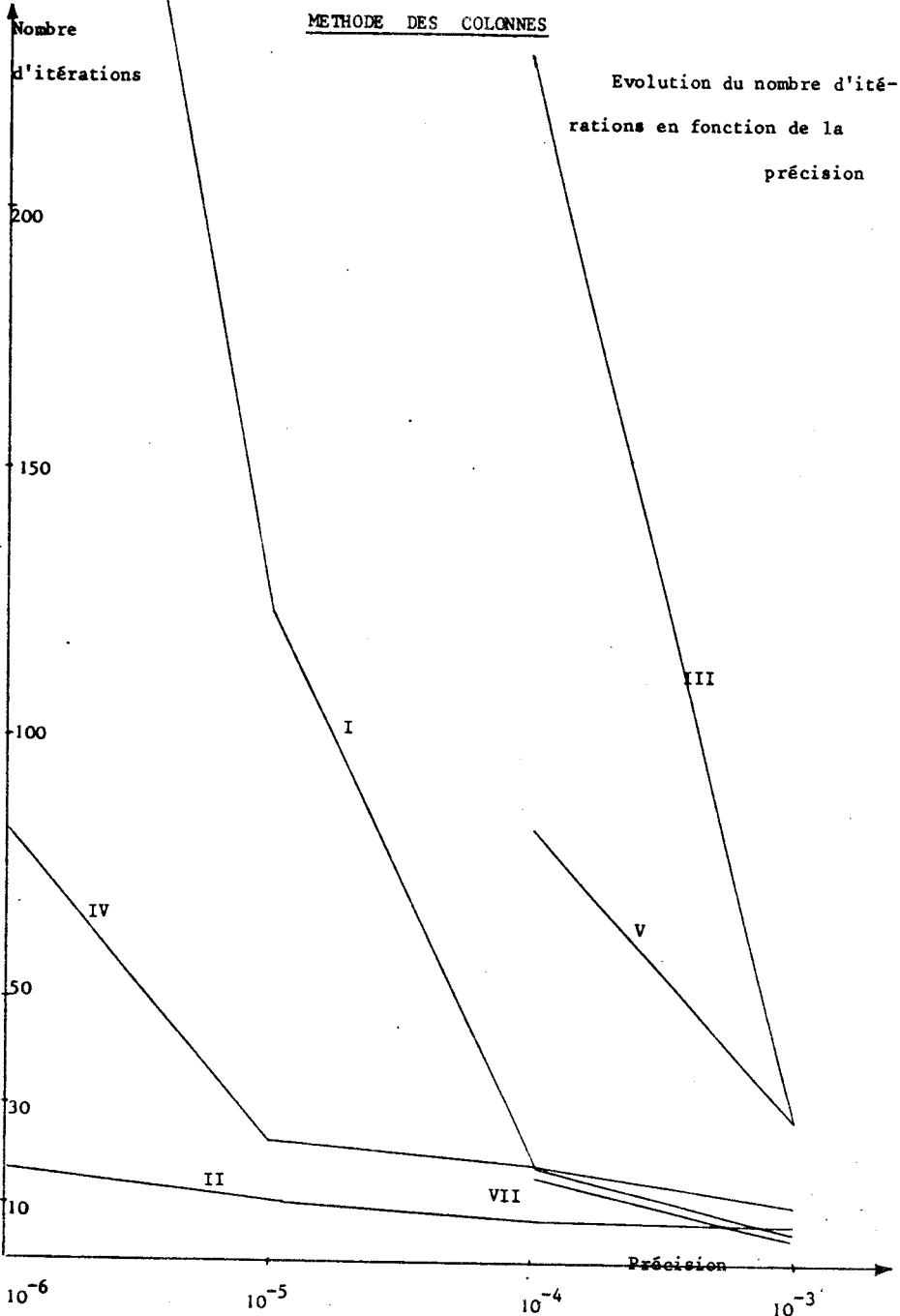
NOMBRE D'ITERATIONS NECESSAIRES POUR ATTEINDRE UNE PRECISION DONNEE

Précision relative $\frac{\Delta f}{f}$		$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
PB	ALGO.				
I	Dantzig	5	18	125	402
I	Alg.mixte	5	6	34	150
II	Dantzig	6	7	11	17
II	Alg.mixte	9	14	24	31
III	Dantzig	27	230		
III	Alg.mixte	17	83		
IV	Dantzig	10	17	23	81
IV	Alg.mixte	5	7	13	18
V	Dantzig	26	82		
V	Alg.mixte	16	75		
VI	Dantzig				
VI	Alg.mixte	16	22	29	37
VII	Dantzig	4	18		
VII	Alg.mixte	7	35	36	46

BUS  
1111

METHODE DES COLONNES

Evolution du nombre d'itérations en fonction de la précision



ALGORITHME MIXTE

nombre  
d'itérations

Evolution du nombre d'itérations  
en fonction de la précision

200

150

100

50

30

10

I

III

V

VII

VI

II

IV

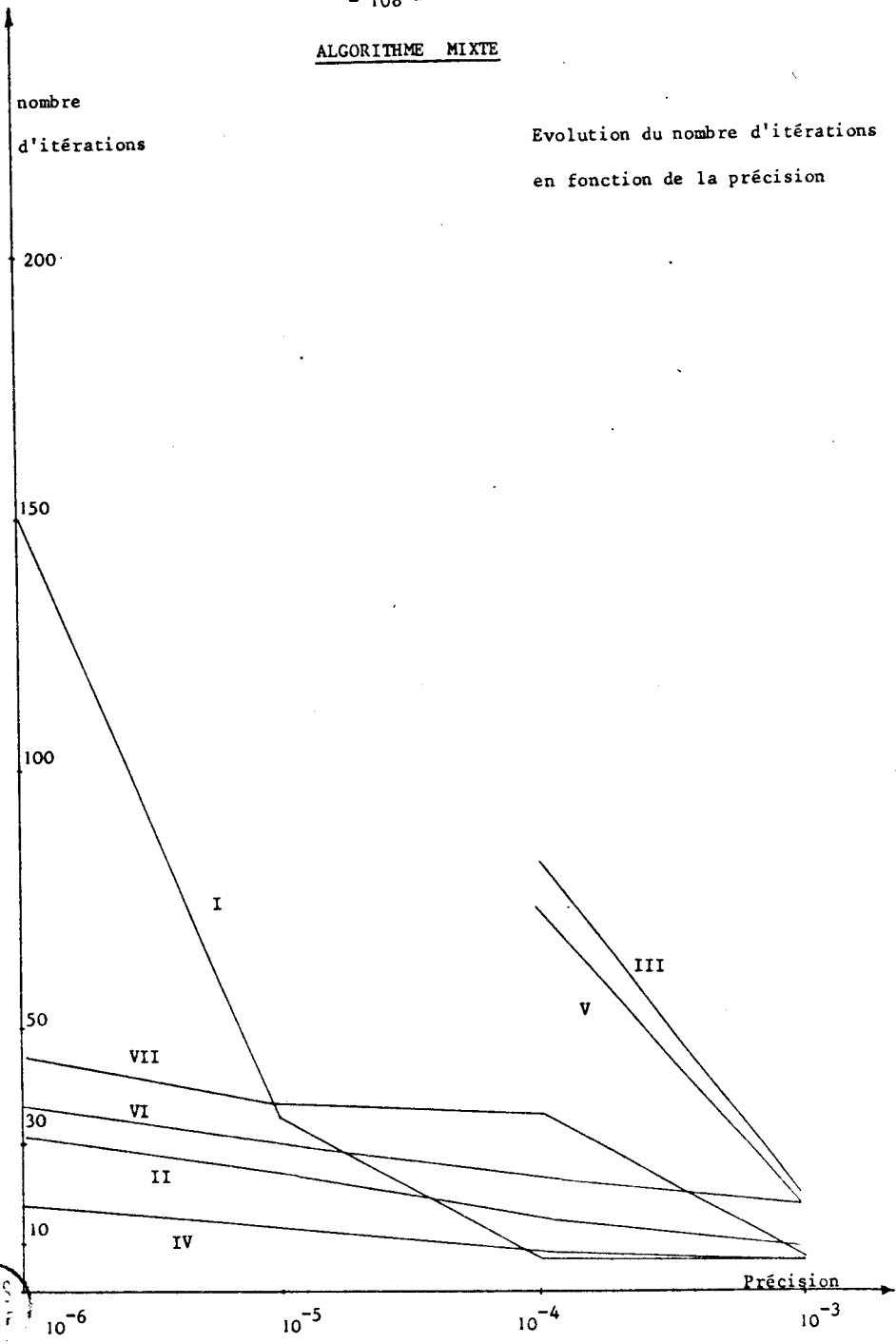
Précision

$10^{-6}$

$10^{-5}$

$10^{-4}$

$10^{-3}$





On peut noter que, globalement, les résultats sont plus homogènes pour l'algorithme mixte.

Toutefois, dans les deux cas, se confirme la convergence lente au voisinage de l'optimum. En particulier, pour l'algorithme mixte, 50 itérations en moyenne sont nécessaires pour atteindre la précision  $10^{-6}$ .

CHAPITRE VIII

PARAMETRISATION

XXXXXXXXXXXXXXXXXXXX

VIII - 1 - INTRODUCTION.

Soit à résoudre le problème suivant :

$$P \quad \left[ \begin{array}{l} \text{Max } f(x) \\ c(x) = 0 \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

en moyennant les hypothèses du § V - 1 - 1 et B étant une matrice de dimension (q, n).

Pour les essais numériques la contrainte c(x) a été obtenue à partir d'un système linéaire :

$$A \cdot x = a \quad \rightarrow \quad c(x) = \sum_{i=1}^p (A_i \cdot x - a_i)^2$$

Le problème initial à résoudre était donc le suivant :

$$P' \quad \left[ \begin{array}{l} \text{Max } f(x) \\ A \cdot x = a \\ B \cdot x = b \\ x \geq 0 \end{array} \right.$$

- Nous avons résolu le problème sous sa forme (P') et sous sa forme équivalente (P).

Au § VIII - 2 nous montrerons que la taille des problèmes à résoudre par paramétrisation de (P) peut être très inférieure à celle des problèmes intervenant dans la résolution de (P') par la méthode de FRANK et WOLFE.

L'étude de la convergence numérique de la paramétrisation sera abordée au § VIII - 3. Les variantes définies au § VIII - 4 améliorent notablement la convergence pratique de la paramétrisation.

VIII - 2 - EVALUATION DE LA TAILLE DES PROBLEMES A RESOUDRE.

Résoudre ( P ) par paramétrisation demande la résolution, en alternance, du programme directeur, qui est linéaire, et du programme auxiliaire. Pour ce dernier, l'algorithme de FRANK et WOLFE se ramène à la résolution d'une suite de programmes linéaires. Il en est de même pour ( P' ).

Nous allons déterminer le nombre de mémoires nécessaires à la résolution de ces programmes linéaires en fonction des diverses techniques de calcul utilisées.

- Méthode "manuelle". Pour toute base I on doit disposer de :  $T(I)^{\bar{I}}$ ,  $t(I)$  et  $d(I)^{\bar{I}}$ .

- Méthode de l'inverse explicite. On conserve en mémoire l'inverse de la matrice de base  $(A^I)^{-1}$ , elle est modifiée à chaque changement de base. La connaissance de cette matrice permet de déterminer :  $\forall j \in \bar{I}, d^j(I)$  puis  $T^s(I)$  et  $t(I)$ .

- Méthode de la factorisation de l'inverse. Soit  $I_0$  la base de départ,  $A^{I_0}$  est généralement une matrice unité. Pour toute base  $I_k$  :

$$(A^{I_k})^{-1} = (E_{I_k}^{I_{k-1}}) \cdot (A^{I_{k-1}})^{-1}$$

$E_{I_k}^{I_{k-1}}$  matrice de passage de la base  $I_{k-1}$  à la base  $I_k$  est de même dimension que  $A^{I_0}$ . Elle est complètement déterminée par la connaissance d'une seule de ses colonnes. Les calculs se ramènent à une succession de produits scalaires de deux vecteurs de dimension  $|I_0|$ .

Le tableau suivant donne, pour chaque méthode, le nombre de mémoires nécessaires pour résoudre les différents programmes linéaires.

On a indiqué :

- sur la ligne A le nombre de mémoires nécessaires en mémoire centrale,
- sur la ligne B le nombre de mémoires nécessaires sur périphériques.

On désigne par :

- k le nombre d'itérations lors de la résolution du programme directeur dans l'algorithme de paramétrisation,
- $l_1, l_2, l_3$  le nombre de changements de base lors de l'utilisation de la méthode simpliciale.

		M E T H O D E		
		"manuelle"	inverse explicite	factorisation de l'inverse
PARAMETRISATION	Programme (A) directeur	$2k - 4$	6	4
	(B)	$kn \quad (1)$	$kn \quad (1)$	$\begin{cases} kn & (1) \\ 4l_3 & (2) \end{cases}$
	Programme (A) auxiliaire	$q(n-q)$	$q^2$	2q
	(B)			$l_1(q+2) \quad (2)$
Résolution de (P')	(A)	$(p+q)(n-p-q)$	$(p+q)^2$	$2(p+q)$
	(B)			$l_2(p+q+2) \quad (2)$

(1) Stockage d'un générateur à chaque étape.

(2) Stockage d'une colonne de  $E_{I_k}^{I_{k-1}}$  à chaque itération.

En général, la résolution de (P) par l'algorithme de paramétrisation nécessite globalement moins de mémoire que la résolution de (P') par la méthode directe. De plus, le taux d'occupation de la mémoire centrale est plus faible avec la paramétrisation. Ces avantages croissent avec la taille des problèmes. On remarquera aux paragraphes suivants que cet avantage se traduit par une augmentation du temps de résolution. Le tableau suivant donne les résultats obtenus, pour les problèmes de l'annexe II, par la méthode directe.

Probleme N°	f (x̄)	f (x*)	PRECISION		c (x*)	Temps de résolution
			Absolute	Relative		
1	0,75	0,75	0	0	0	0,001
2	6,66666...	6,6666..	0.4 10 <sup>-13</sup>	0.6 10 <sup>-14</sup>	0.1 10 <sup>-14</sup>	0,003
3	6,66666...	6,6666..	0.4 10 <sup>-13</sup>	0.6 10 <sup>-14</sup>	0.1 10 <sup>-14</sup>	0,003
4	6	6	0	0	0	0,003
5	14166,6666..	14166,66..	0.1 10 <sup>-9</sup>	0.1 10 <sup>-13</sup>	0.1 10 <sup>-14</sup>	0,004
6	48,706136	48,706136	0.1 10 <sup>-9</sup>	0.2 10 <sup>-11</sup>	0.1 10 <sup>-14</sup>	0,056



VIII - 3 - PARAMETRISATION.

VIII - 3 - 1 - Considérations numériques.

La paramétrisation a été abordée comme une application directe de l'algorithme mixte. Le code a été testé à l'aide des problèmes tests figurant en annexe II.

La création automatique de générateurs initiaux ne nous a pas permis d'obtenir numériquement au moins un générateur réalisable. Nous nous sommes donc donnés des générateurs initiaux dont un réalisable.

La paramétrisation nécessite la connaissance d'une suite :

$\{ \alpha_1 \mid 1 \in \mathbb{N} \}$  telle que  $\lim_{1 \rightarrow \infty} \alpha_1 = 0$ . Les meilleurs résultats ont été obtenus pour une suite  $\{ \alpha_1 \}$  telle que  $\alpha_{1+1} = \mu \cdot \alpha_1$  avec  $0,3 \leq \mu \leq 0,7$ .

L'algorithme mixte converge uniquement pour  $\alpha > 0$ . Numériquement des problèmes se sont posés pour  $\alpha$  voisin de 0 (environ  $10^{-3}$ ).

Une première option choisie a été de cumuler les générateurs des étapes précédentes.

Le tableau suivant récapitule les résultats obtenus.

Temps maximum de résolution : 1

Problème N°	f (x)	f (x) <sup>*</sup>	PRECISION		c (x) <sup>*</sup>	α	Temps de résolution
			absolue	relative			
1	0,75	Jamais atteint à 0,25 près ∀ α, 0 < α < 10 <sup>-3</sup>					
2	6,6666...	6,6395	0,03	0,0044	1.10 <sup>-7</sup>	4.10 <sup>-7</sup>	0,32
3	6,6666...	6,66732	0,001	0,0001	10 <sup>-7</sup>	4.10 <sup>-7</sup>	0,28
4	6	Jamais atteint à 0,25 près ∀ α, 0 < α < 10 <sup>-4</sup>					
5	14166,666...	14166,599	0,07	5.10 <sup>-6</sup>	5.10 <sup>-5</sup>	5.10 <sup>-5</sup>	0,06
6	48,706136	Jamais atteint à 0,75 près ∀ α, 0 < α < 10 <sup>-2</sup>					

PARAMETRISATION





VIII - 3 - 2 - Conclusion

La convergence numérique n'est pas assurée dans tous les cas . Nous n'avons pas obtenu de résultats satisfaisants, une fois sur deux, pour  $\alpha < 10^{-3}$ .

Dans les cas favorables le temps moyen de résolution est de 0,22 pour  $\alpha$  voisin de  $10^{-5}$ .

Nous avons amélioré la convergence numérique en introduisant, à chaque étape de l'algorithme mixte, des générateurs supplémentaires. Deux variantes ont été définies.

VIII - 4 - VARIANTES DE LA PARAMETRISATION

VIII - 4 - 1 - Idée générale des variantes

Nous avons utilisé l'algorithme mixte pour résoudre le problème P1 ( $\alpha$ ). On introduit à chaque étape un générateur  $X^S$ . Afin d'améliorer la convergence numérique, nous allons introduire deux générateurs supplémentaires. Ils seront obtenus par des maximisations unidimensionnelles dans  $B$  du type suivant:

$$P ( X_0 , d ) \quad \left[ \begin{array}{l} \text{Max } f (x) \\ x = X_0 + \theta \cdot d \quad \text{avec } X_0 \in B \\ x \geq 0 \\ \theta \geq 0 \\ B \cdot x = b \end{array} \right.$$

Remarque :

Ces variantes améliorent la convergence numérique de la paramétrisation sans accélérer la vitesse de convergence de l'algorithme mixte.

VIII - 4 - 2 - Variante n° 1.

Les deux points supplémentaires sont obtenus en prenant :

- 1 -  $X_0 = X^S$  et  $d = \nabla f ( X^S )$ .

- 2 -  $X_0 = X^S$  et  $d = \nabla a ( X^S )$ .

VIII - 4 - 3 - Variante n° 2.

Les deux points supplémentaires sont obtenus en prenant :

- 1 -  $X_0 = x^k$  et  $d = \nabla a ( x^k )$

- 2 -  $X_0 = X^S$  et  $d = \nabla f ( X^S )$

VIII - 4 - 4 - Illustration.

Afin d'illustrer ce qui précède, considérons le programme suivant :

$$\left[ \begin{array}{l} \text{Max} \quad -x_1^2 - x_2^2 + 2x_1 + 2x_2 \\ (x_1 + x_2 - 1)^2 = 0 \\ x_1 + x_2 \leq 2 \\ x_2 \leq 3/2 \\ x_1, x_2 \geq 0 \end{array} \right.$$

Il a pour solution optimale  $\hat{x}_1 = \hat{x}_2 = 0,5$

Choisissons comme générateur initial le point  $X^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

L'algorithme mixte ou de paramétrisation (ainsi que ses variantes)

donne  $\bar{u} = 0$ .

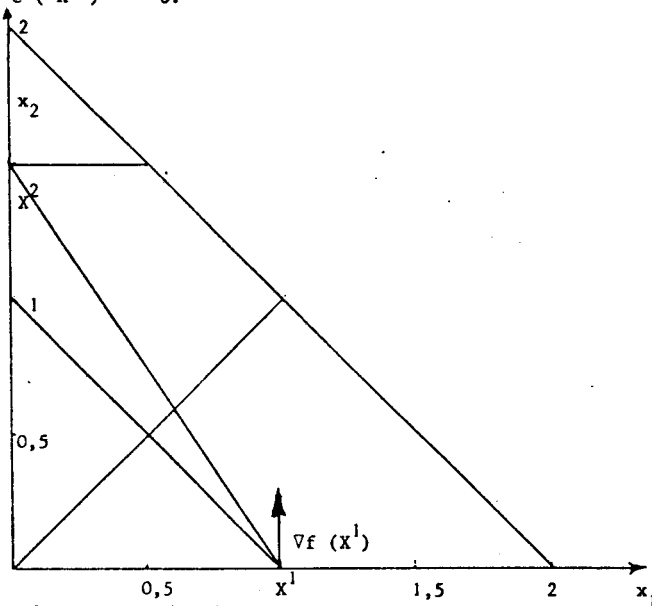
Dans tous les cas, le premier sous-programme à résoudre a pour solution

$$X^2 = \begin{bmatrix} 0 \\ 1,5 \end{bmatrix}$$

Les figures suivantes illustrent la solution au démarrage dans tous les cas.

- 1 - Algorithme mixte.

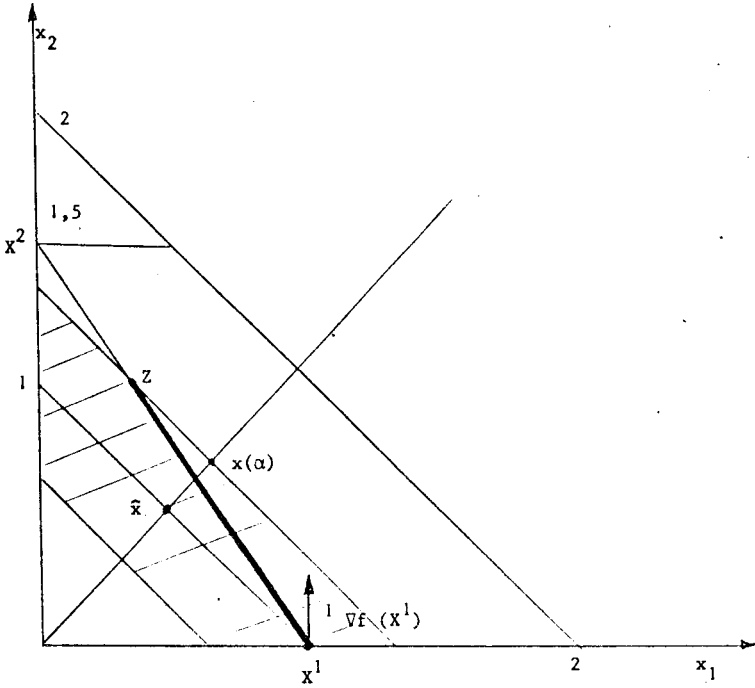
On sait que  $z^k = X^1$  tant que le générateur  $X^s$  entrant est tel que  $c(X^s) > 0$ .



- 2 - Paramétrisation.

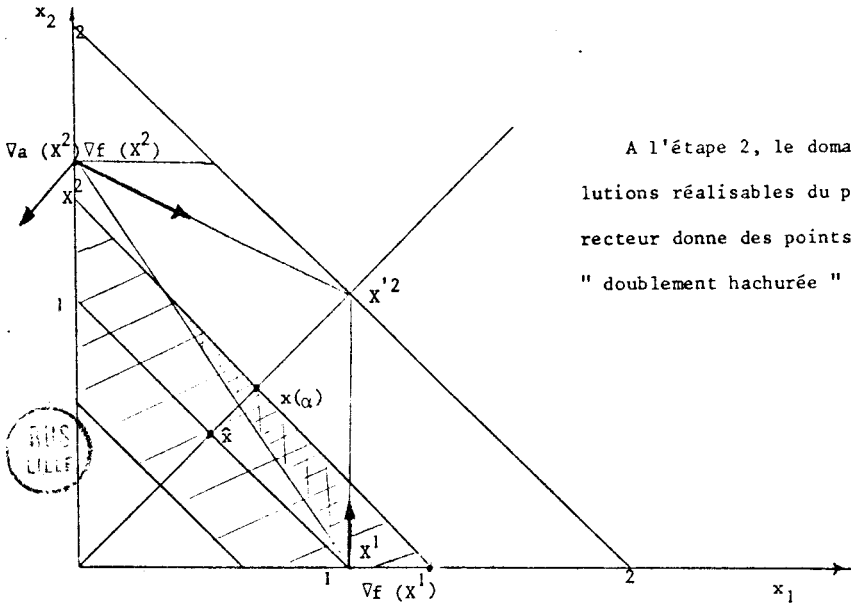
Pour  $\alpha > 0$ , on résoud :

$$\left[ \begin{array}{l} \text{Max} \quad -x_1^2 - x_2^2 + 2x_1 + 2x_2 \\ \alpha - (x_1 + x_2 - 1)^2 \geq 0 \\ x_1 + x_2 \leq 2 \\ x_2 \leq 1,5 \\ x_1, x_2 \geq 0 \end{array} \right.$$



Le domaine des solutions réalisables du programme directeur donne, à l'étape 2, des points du segment  $[x^1, Z]$

- 3 - Variante 1 ou 2 .



A l'étape 2, le domaine des solutions réalisables du programme directeur donne des points de la partie " doublement hachurée " .

UNIVERSITÄT  
LILLE

$$\forall f ( X^2 ) = \begin{pmatrix} - 2 x_1 + 2 \\ - 2 x_2 + 2 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} = d$$

On maximise alors  $f$  dans  $B$  sur la demi droite  $( X^2, d )$ ,

on obtient  $X^{12}$ .

$$- \forall c ( X^2 ) = \begin{pmatrix} -1 \\ -1 \end{pmatrix} = d'. \text{ La maximisation de } f \text{ dans } B \text{ sur } ( X^2, d' )$$

donne  $X^2$ .

L'introduction de ces générateurs supplémentaires "agrandit" l'ensemble des solutions du programme directeur. La maximisation de  $f$  dans  $B$  sur ces demi-droites a pour effet "d'agrandir" le domaine en " s'éloignant ", en général, de la contrainte  $c ( x ) = 0$ .

#### VIII - 4 - 5 - Résultats.

Pour les mêmes problèmes les deux variantes donnent les résultats suivants :

Temps de résolution maximum : 1

Problème N°	f (x̄)	f (x <sup>*</sup> )	P R E C I S I O N		c (x <sup>*</sup> )	α	Temps de résolution
			Absolute	Relative			
1	0,75	0,7489	0,001	0,001	2.10 <sup>-10</sup>	6.10 <sup>-6</sup>	0,56
2	6,6666...	6,6670	0,0004	0,00005	2.10 <sup>-6</sup>	4.10 <sup>-6</sup>	0,53
3	6,6666...	6,6659	0,0007	0,0001	4.10 <sup>-5</sup>	4.10 <sup>-6</sup>	0,42
4	6	5,9876	0,01	0,002	7.10 <sup>-8</sup>	7.10 <sup>-6</sup>	0,92
5	14166,666...	14161	5	0,0003	6.10 <sup>-5</sup>	4.10 <sup>-4</sup>	0,73
6	48,706136	48,707565	10 <sup>-3</sup>	3.10 <sup>-5</sup>	2.10 <sup>-4</sup>	10 <sup>-4</sup>	0,50

P A R A M E T R I S A T I O N V A R I A N T E N ° 1



Temps de résolution maximum : 1,0

Problème n°	f (x̄)	f* (x)	PRECISION		c* (x)	α	Temps de résolution
			Absolute	Relative			
1	0,75	0,7493	0,0007	0,001	10 <sup>-6</sup>	6.10 <sup>-6</sup>	0,66
2	6,6666.	6,6686	0,002	0,0003	9.10 <sup>-8</sup>	5.10 <sup>-6</sup>	0,59
3	6,6666.	6,6714	0,005	0,0007	10 <sup>-6</sup>	4.10 <sup>-6</sup>	0,54
4	6,0	6,0013	0,001	0,0002	3.10 <sup>-7</sup>	2.10 <sup>-5</sup>	0,47
5	14166,666.	14129	38	0,002	4.10 <sup>-5</sup>	6.10 <sup>-5</sup>	0,67
6	48,706136.	48,620	0,09	0,002	2.10 <sup>-4</sup>	10 <sup>-4</sup>	0,71

PARAMETRISATION VARIANTE N° 2



VIII - 4 - 6 - Conclusion.

On peut noter que les deux variantes donnent chacune des résultats satisfaisants pour tous les problèmes. De plus, dans tous les cas, l'une des deux variantes permet d'obtenir une solution avec une précision relative meilleure que  $10^{-3}$  pour  $\alpha$  de l'ordre de  $10^{-5}$ .

Les essais numériques ont permis de constater que  $10^{-6}$  est un seuil numérique pour le paramètre  $\alpha$  correspondant à la non convergence théorique de l'algorithme pour  $\alpha = 0$ .

La précision des résultats est obtenue au détriment du temps de résolution.

Les graphiques qui suivent, illustrent l'évolution des solutions des problèmes I et III lors de leur résolution par la paramétrisation et les deux variantes de la paramétrisation.



PROBLEME N° I

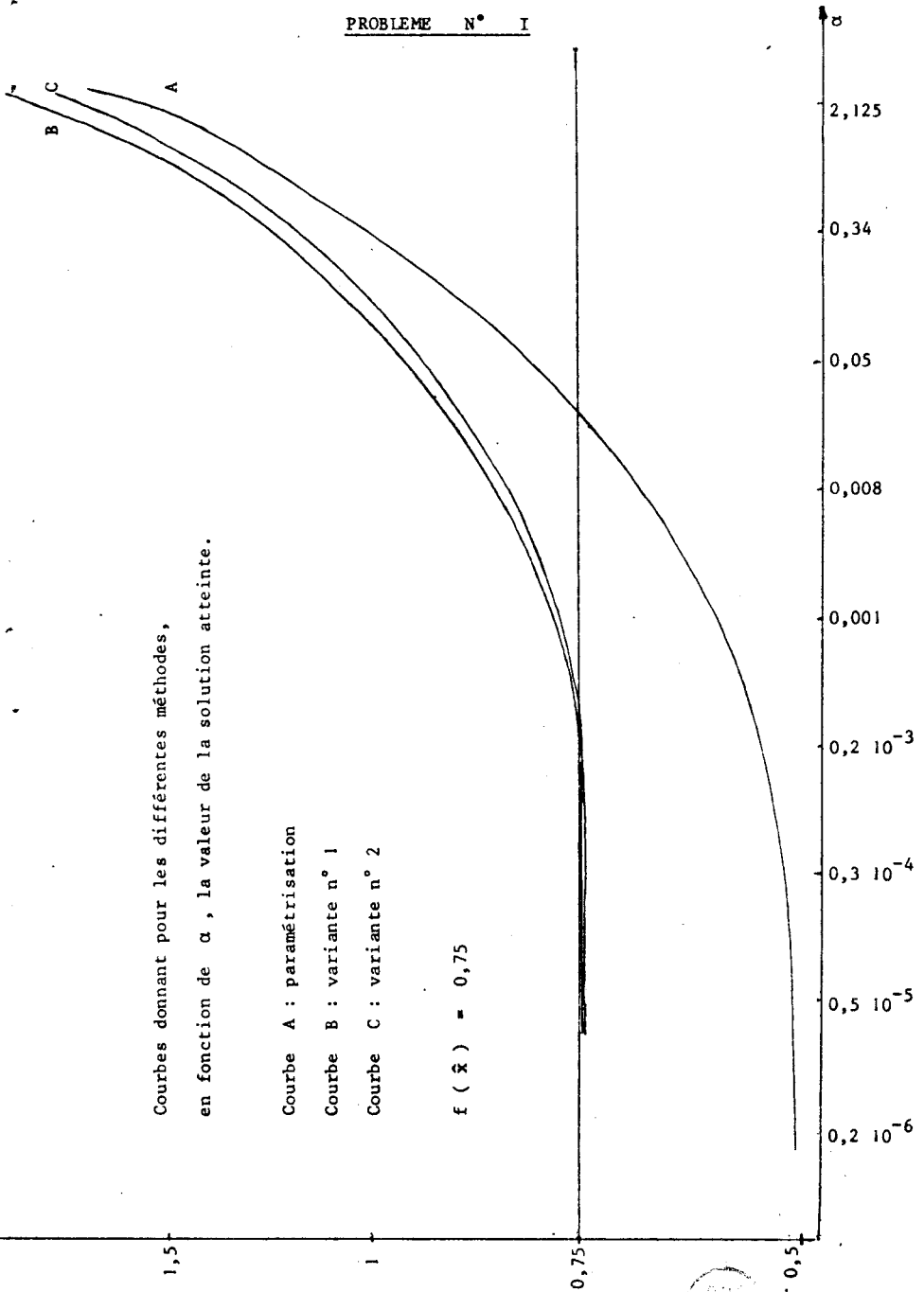
Courbes donnant pour les différentes méthodes,  
en fonction de  $\alpha$ , la valeur de la solution atteinte.

Courbe A : paramétrisation

Courbe B : variante n° 1

Courbe C : variante n° 2

$$f(\bar{x}) = 0,75$$



PROBLEME N° III



Courbes donnant pour les différentes méthodes,  
en fonction de  $\alpha$ , la valeur de la solution atteinte.

Courbe A : paramétrisation

Courbe B : variante n° 1

Courbe C : variante n° 2

$f(\hat{x}) = 6,666666...$

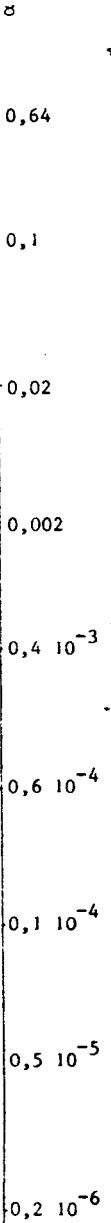
$f(x)$

7,5

7

6,666666

6,5



VIII - 5 - Conclusion.

La paramétrisation et ses variantes permettent la résolution de problèmes de grande taille. Le gain de place se fait au détriment du temps de résolution.

La paramétrisation ou ses variantes fournissent des résultats satisfaisants dans l'ensemble.

ANNEXE I  
\*\*\*\*\*

PROBLEME N° I

Maximiser :  $- X_1^2 - X_2^2 + 4 X_1$

Sous les contraintes :

$- X_1^2 - X_2^2 + 2 X_1 + 2 X_2 + 1 \geq 0$

$X_1 \leq 3/2$

$X_2 \leq 3/2$

$X_1, X_2 \geq 0$

Valeur du programme :  $f(\bar{x}) = 3,7320508$

PROBLEME N° II

Maximiser :  $-\sum_{i=1}^5 X_i^2$

Sous les contraintes :

$-\sum_{i=1}^5 X_i^2 + 2 X_1 + 4 X_2 - 2 \geq 0$

$\sum_{i=1}^5 X_i \leq 1$

$X_i \geq 0, i = 1, 5.$

Valeur du programme :  $f(\bar{x}) = 0,254033077$

PROBLEME N° III

Maximiser :  $-\sum_{i=1}^6 X_i^2 + 6 X_1 + 6 X_2 + 6 X_3$

Sous les contraintes :

$-\sum_{i=1}^6 X_i^2 + 2 \geq 0$

$\sum_{i=1}^6 X_i \geq 1$  et  $\sum_{i=1}^6 X_i \leq 3$

$X_i \geq 0, i = 1, 6.$

Valeur du programme :  $f(\bar{x}) = 12,69693846$



PROBLEME N° IV

$$\text{Maximiser : } - \sum_{i=1}^7 X_i^2 + 6 (X_1 + X_2 + X_3 + X_7)$$

Sous les contraintes :

$$- \sum_{i=1}^7 X_i^2 + 2 \geq 0$$

$$\sum_{i=1}^7 X_i \geq 1$$

$$\sum_{i=1}^7 X_i \leq 3$$

$$X_i \geq 0, i = 1, 7$$

$$\text{Valeur du programme : } f(\bar{x}) = 14,97056$$

PROBLEME N° V

$$\text{Maximiser : } - \sum_{i=1}^7 X_i^2 + 2 X_6 + 2 X_7$$

Sous les contraintes :

$$- \sum_{i=1}^7 X_i^2 + 4 \sum_{i=1}^7 X_i - 21 \geq 0$$

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 \leq 13$$

$$X_1 + 5 X_3 - X_2 - X_4 + 3 X_5 - X_6 + 4 X_7 \leq 12$$

$$4 X_1 - 5 X_2 + 7 X_3 - X_6 + X_7 \leq 6$$

$$3 X_1 + X_2 + 3 X_4 - X_5 - X_7 \geq 1$$

$$- X_1 + 2 X_2 - X_3 + X_6 \geq 0$$

$$2 X_1 - X_2 - 2 X_3 + 4 X_5 - X_6 + X_7 \geq 1$$

$$X_i \geq 0, i = 1, 7$$

$$\text{Valeur du programme : } f(\bar{x}) = -2,1806$$

PROBLEME N° VI

Maximiser :  $-\sum_{i=1}^{10} X_i^2 + 2 X_3 + 2 X_5 + 7 X_7 + 8 X_9$

Sous les contraintes :

$$\sum_{i=1}^{10} X_i^2 + 2 X_1 + 7 X_2 + 9 X_3 - 8 X_4 - 5 X_5 + 3 X_6 + 2 X_7 + X_8 + 4 X_9 + 2 X_{10} - 16 \geq 0$$

$$\sum_{i=1}^{10} X_i \leq 11$$

$$X_1 - X_2 + X_3 - X_4 - X_5 - X_6 + X_7 + X_8 - X_9 - X_{10} \leq 0$$

$$2 X_1 + 3 X_2 + X_3 - 2 X_4 + X_5 + 2 X_6 - 3 X_7 + X_8 + X_9 + X_{10} \leq 8$$

$$X_1 - 3 X_2 - X_3 + 2 X_4 - 11 X_5 + X_6 + 7 X_7 - 2 X_8 - X_9 + X_{10} \leq 20$$

$$-X_1 - 2 X_2 - 3 X_3 - 4 X_4 - 4 X_5 + 7 X_6 + 8 X_7 + 9 X_8 - 10 X_9 - 15 X_{10} \leq 0$$

$$2 X_1 + 3 X_2 + 4 X_3 - 7 X_4 - 5 X_5 - 10 X_6 - 7 X_7 + 4 X_8 - 3 X_9 + 2 X_{10} \leq 0$$

$$X_2 + X_4 + X_6 + X_8 + X_{10} = 5$$

$$X_1 + 2 X_2 + 5 X_3 + 4 X_4 + X_5 - 7 X_6 - 7 X_7 + 2 X_8 + 3 X_9 \geq 3$$

$$-2 X_1 + X_2 - 3 X_3 + 7 X_4 + 12 X_5 - 4 X_6 + 3 X_7 + 5 X_8 - 9 X_9 + 2 X_{10} \geq 10$$

$$5 X_1 - 2 X_2 + 7 X_3 + 11 X_4 - 18 X_5 + 9 X_6 + 2 X_7 + 3 X_8 + X_9 - 4 X_{10} \geq 10$$

$$-3 X_1 - 2 X_2 + X_3 - 7 X_4 + 12 X_5 + 15 X_6 - 3 X_7 + 21 X_8 + 2 X_9 + 5 X_{10} \geq 35$$

$$X_i \geq 0, \quad i = 1, 10$$

Valeur du programme :  $f(\bar{x}) = 16,12305\dots$



PROBLEME N° VII

$$\text{Maximiser : } - \sum_{i=1}^{10} X_i^2 + 20 X_1 + 60 X_2 + 20 X_3 + 4 X_4 + 6 X_7 + 20 X_{10} \\ - ( 6 X_5 + 8 X_6 + 2 X_8 + 6 X_9 )$$

Sous les contraintes :

$$\sum_{i=1}^{10} X_i \leq 11$$

$$3 X_1 - 2 X_2 + X_6 + X_9 - X_{10} \leq 3$$

$$X_1 - 2 X_2 + X_3 + X_4 \leq 2$$

$$2 X_1 + X_2 - X_3 + 4 X_4 - X_5 \leq 6$$

$$- X_1 - X_2 + X_3 - X_4 + X_6 \leq 0$$

$$5 X_1 - X_2 - 2 X_3 + X_4 - 2 X_5 + X_{10} \leq 3$$

$$3 X_3 + 2 X_4 + X_5 - X_6 - X_7 \leq 5$$

$$\sum_{i=1}^5 X_i = 5 \quad ; \quad X_6 + X_9 \geq 3 \quad ; \quad X_7 + X_{10} \geq 1$$

$$X_7 + X_8 \geq 1 \quad ; \quad X_4 - X_5 + 3 X_6 + X_9 \geq 3$$

$$X_1 - X_2 + 2 X_3 + X_6 + X_{10} \geq 3$$

$$X_1 - 2 X_2 + X_3 - X_4 + 3 X_5 - X_6 + X_7 - X_8 - X_9 + X_{10} \geq 0$$

$$- \sum_{i=1}^{10} X_i^2 + 12 X_1 + 12 X_2 + 4 X_3 + 6 X_4 + 10 X_5 + 2 X_6 - 37 \geq 0$$

$$X_i \geq 0 \quad , \quad i = 1, 10$$

Valeur du programme :  $f(\bar{x}) = 175,5996\dots$



ANNEXE I I  
\*\*\*\*\*

PROBLEME N° I

Maximiser :  $-x_1 + x_2$

Sous les contraintes :

$$2x_1^2 + 2x_2^2 + x_3^2 + x_4^2 + x_5^2 - 3,5x_1 - 3,5x_2 - 2x_3 - 1,5x_4 - 1,5x_5 + 2(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_5) + 2,125 = 0$$

$$x_1 + x_2 \leq 5$$

$$x_3 + x_4 + x_5 \leq 1000$$

$$x_1 + 2x_2 \geq 1$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Valeur du programme :  $f(\bar{x}) = 0,75$

PROBLEME N° II

Maximiser :  $-x_1^2 - x_2^2 - x_3^2 + 4x_1 + 4x_2 + 4x_3$

Sous les contraintes :

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + 2x_1x_2 + 2x_1x_3 + 2x_1x_4 + 2x_2x_3 + 2x_2x_4 + 2x_3x_4 - 4x_1 - 4x_2 - 4x_3 - 4x_4 + 4 = 0$$

$$x_1 + x_2 \leq 1,5$$

$$x_3 \leq 1$$

$$x_4 \leq 10$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Valeur du programme :  $f(\bar{x}) = 6,666666$

PROBLEME N° III

$$\text{Maximiser : } -x_1^2 - x_2^2 - x_3^2 + 4x_1 + 4x_2 + 4x_3$$

Sous les contraintes :

$$x_1^2 + x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 + 2x_2x_3 - 4x_1 - 4x_2 - 4x_3 + 4 = 0$$

$$x_1 + x_2 \leq 1,5$$

$$x_3 \leq 1$$

$$x_1, x_2, x_3 \geq 0$$

$$\text{Valeur du programme : } f(\bar{x}) = 6,666666$$

PROBLEME N° IV

$$\text{Maximiser : } -x_1^2 - x_2^2 + 4x_1 + 4x_2$$

Sous les contraintes :

$$2(x_1^2 + x_2^2 + x_3^2) + x_4^2 + x_5^2 + x_6^2 + x_7^2 + 2(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3$$

$$+ x_2x_5 + x_2x_6 - x_1x_7 - x_2x_7 - x_3x_7) - 6x_1 - 6x_2 - 6x_3 - 2x_4 - 2x_5$$

$$- 2x_6 + 4x_7 + 7 = 0$$

$$x_1 + x_2 + x_3 \leq 4,5$$

$$x_4 + x_5 + x_6 + x_7 \leq 100$$

$$x_1 \geq 0,5$$

$$x_2 \geq 0,5$$

$$x_3 \geq 0,5$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0$$

$$\text{Valeur du programme : } f(\bar{x}) = 6$$

PROBLEME N° V

Maximiser :  $-x_1^2 - x_2^2 - x_3^2 + 300x_1 + 300x_2 + 300x_3$

Sous les contraintes :

$$x_1^2 + x_2^2 + x_3^2 + 2(x_1x_2 + x_2x_3 + x_1x_3) - 100(x_1 + x_2 + x_3) + 2500 = 0$$

$$x_1 + x_2 \leq 100$$

$$x_3 \leq 150$$

$$x_1 + 2x_2 \leq 10$$

$$x_1, x_2, x_3 \geq 0$$

Valeur du programme :  $f(\bar{x}) = 14166,6666$

PROBLEME N° VI

Maximiser :  $-\sum_{i=1}^7 x_i^2 + 4x_1 - 6x_2 + 2x_3 - 10x_4 + 20x_5 + 12x_7$

Sous les contraintes :

$$5x_1^2 + 3x_2^2 + 5x_3^2 + 5x_4^2 + x_5^2 + x_6^2 + x_7^2 - 2x_1x_2 + 4x_1x_3 + 2x_1x_5 + 4x_1x_4$$

$$- 4x_1x_7 + 2x_2x_3 - 6x_2x_4 + 2x_2x_5 + 2x_2x_6 + 2x_2x_7 + 4x_3x_4 + 4x_3x_5$$

$$- 2x_3x_6 - 4x_4x_6 - 2x_4x_7 - 14x_1 - 6x_2 - 22x_3 - 6x_4 - 10x_5 + 2x_6$$

$$+ 2x_7 + 27 = 0$$

$$x_1 + 2x_2 + x_3 + 5x_4 + 2x_5 + x_6 + x_7 \leq 13$$

$$x_1 - 2x_2 + 5x_3 - x_4 + 3x_5 - x_6 - 4x_7 \leq 2$$

$$4x_1 - 5x_2 + 7x_3 - x_6 + x_7 \leq 2$$

$$3x_1 + x_2 + 3x_4 - x_5 - x_7 \geq 1$$

$$-x_1 + 2x_2 - x_3 + x_6 \geq 0$$

$$2x_1 - x_2 - 2x_3 + 4x_5 - x_6 + x_7 \geq 1$$

$$x_i \geq 0, i = 1, 7.$$

Valeur du programme :  $f(\bar{x}) = 48,7061368..$



ANNEXE III

\*\*\*\*\*

SUBROUTINE FWFC(T,F,IBASE,M,NN2,INART,IPHAS,ISQL,N,XPAX,ALPHA,FXNC  
-T,FGNCT,ICER,E,KCE,I22,C,ZZ,MAXLIN)

\*\*\*\*\*

XPAX EST LA SOLUTION DE  
MAXIMISER FXNC(X,N) A ALPHA PRES  
SUR UN ENSEMBLE DEFINI PAR DES CONTRAINTES LINEAIRES

ON UTILISE LA METHODE DE FRANCK ET WOLFE

FGNCT DEFINI LE GRADIENT DE FXNC  
I22 EST LE NOMBRE MAXIMUM D ITERATIONS  
SI ICER=0 ALORS E EST LE POINT REALISABLE DE DEPART  
T,F,IBASE,M,NN2,INART,IPHAS CARACTERISENT LES CONTRAINTES LINEAIRES  
ISCL = 6 INDIQUE QUE L ON A TROUVE UNE  
SOLUTION OPTIMALE A DISTANCE FINIE

\*\*\*\*\*

IMPLICIT REAL\*8(A-H),REAL\*8(C-Z)  
LOGICAL MAXLIN

REAL\*8KCE  
EXTERNAL ABS  
EXTERNAL FXNC,FGNCT  
DIMENSION E(20)  
DIMENSION C(15,15)  
DIMENSION XX(20)  
DIMENSION T(1),F(1),XPAX(1),IBASE(1)  
DIMENSION X(20),Z(20),GRAD(40)  
DIMENSION XK(20),ZK(20)  
COMMON/ICENT/C4(15,15),P3(15)  
COMMON L,LC,GXX(40)  
COMMON/FUN1/Q1(15,15),P1(15)  
COMMON/FUN3/CSTA,PA(15),QA(15,15)  
COMMON/AAAA/AAAA

NAMLIST/REC/221,I21,ALPHA1,TETA,XK,ZK  
RMAX=-1.E30  
RMIN=1.E 10  
IIP=3

\*\*\*EFS TOLERANCE POUR LA MAXIMISATION SUR LE SEGMENT  
EFS=C.1  
M2=NN2+M  
N1=N-2  
EFTE=0.CCGC1  
DO 204 I=1,N1  
4 X(I)=E(I)

\*\*\*DETENTION D UN POINT REALISABLE  
IF(ICER)1,322,1  
CALL SFELI(T,F,IBASE,M,NN2,INART,IPHAS,ISQL,FX,1)  
IF(ISCL=1)101,100,101  
3 CALL SOLPL(X,N1,M,NN2,IBASE,T)  
IV=1

\*\*\*CALCUL A PARTIR D UN POINT REALISABLE  
2 DO 22 I21=1,I22  
CC 9 I=NL,M2



```
GRAD(I)=0.0
CALL FGCT(X,N1,GRAD)
CC 10 I=3,M2
II=IBASE(I)
10 F(I)=GRAD(II)
CALL SFBLI(T,F,IBASE,M,NN2,INART,IPHAS,ISOL,FX,1)
IF(ISCL.NE.1)GCTC 101
CALL SCLPL(Z,N1,M,NN2,IBASE,T)
*****PROTECTION DES POINTS PRECELEMENT OBTENUS
CC 41 I=1,N1
XK(I)=X(I)
41 ZK(I)=Z(I)
*****EFS TEND VERS C
EFS=EFS*XCE
*****DETERMINATION D UNE NOUVELLE SOLUTION
CALL MAXSG(XK,ZK,GRAD,C,TETA,TETA2,N1)
CC 2222 I=1,N1
2222 XMAX(I)=TETA*ZK(I)+(1-TETA)*XK(I)
3 AX=FXACT(XMAX,N1)
1000 CONTINUE
IF (MAXLIN) GC TC 500C
IF (ABS(A2(XMAX,N1)).LT.1.E-6) GO TO 20
5000 CONTINUE
*****TEST C ARRÊT
EPTC=1.E-15
Z2=GTTEST(X,Z,N1,FGCT)
Z21=Z2
ALPHA1=ALPHA
IF(Z2-ALPHA) 20,20,30
20 CC 31 I=1,N1
31 X(I)=XMAX(I)
IF(TETA.LT.EPTC) GC TC 20
32 CONTINUE
20 ISCL=6
101 RETURN
END
```

SLERCUTINE AMELI (XK,C,N1,IEQ,INEQ,T,GRAD,Q1,XSCL,\*,M,BAS)

\*\*\*\*\*

XSCAL EST LE POINT MAXIMISANT F(X) AVEC  
X = XK +L.D L POSITIF OU =C ET X APPARTENANT  
A UN ENSEMBLE DEFINI PAR DES CONTRAINTES  
LINEAIRES . F EST QUADRATIQUE  
GRAD CONTIENT GRAD F(XK)  
C1 LA MATRICE DES DERIVEES SECONDES

\*\*\*\*\*

IMPLICIT REAL\*(A-H),REAL\*(C-Z)  
LOGICAL BAS  
EXTERNAL ABS

```
DIMENSION XK(1),C(1),T(1),GRAC(1),Q1(15,15),XSCL(1),Z(20)
EPS=G.CC01
TETA=1.E25
MAXIMISATION DE F SUR LA DEMI DROITE ( XK , D )
CC 1 I=1,N1
Z(I)=XK(I)+C(I)
IF(IBAS)GOTO 5
CALL MAXSG (XK,Z,GRAD,Q1,TETA,TETA2,N1)
TETA=TETA2
CONTINUE
M5=2*M
M6=M+1
M7=(N1+2)*M
K=1
**ON RAMENE LA SOLUTION DANS LE DOMAINE
**DEFINI PAR LES CONTRAINTES LINEAIRES
CC 4C I=1,M
S1=T(M6)
M6=M6+1
M5=M5+1
S2=0.
II=1
CC 3 J=M5,M7,M
S1= S1-XK(II)*T(J)
S2=S2+ C(II)*T(J)
II=II+1
CC TC (1C,21,31),K
IF(1.GE.IEC) CC TC 20
S2=-S2
IF(S2.GE.0.) CC TC 40
TETA1=-S1/S2
IF(TETA1.LT.TETA) TETA=TETA1
CC TC 4C
K=2
IF(1.GE.INEC) CC TC 3C
IF (ABS(S2).GT.EPS) RETURN1
CC TC 4C
K=3
S1=-S1
CC TC 12
CONTINUE
CC 5C I=1,N1
IF( C(I).GE.C) CC TC 50
TETA1=-XK(I)/C(I)
IF(TETA1.LT.TETA) TETA=TETA1
CONTINUE
CC 6C I=1,N1
XSCL(I)=XK(I)+TETA*C(I)
RETURN
END
```





SLERCLTINE SPELIIT,F,IBASE,M,N,INART,IPHAS,ISCL,FX,IMPR)

\*\*\*\*\*

\*\*\*\*\* N E T H C C E S I M P L I C I A L E

I  
 T CONTIENT DANS L'ORDRE F LE SECOND MEMBRE  
 DU TABLEAU SIMPLICIAL ET LE TABLEAU SIMPLICIAL  
 IBASE INDICE COLONNE DU TABLEAU SIMPLICIAL  
 LE GRADIENT DE LA FONCTION EST DEFINI PAR F  
 M NOMBRE DE CONTRAINTES  
 N NOMBRE DE VARIABLES NATURELLES ET  
 D ECARTS ASSOCIEES AUX CONTRAINTES EN SUP OU EGAL  
 INART INDICE DE LA PREMIERE VARIABLE ARTIFICIELLE

ISCL TYPE DE LA SOLUTION OBTENUE

ISCL=1 SOLUTION OPTIMALE A DISTANCE FINIE  
 ISCL=2 PROBLEME IMPOSSIBLE  
 ISCL=3 SOLUTION INFINIE  
 ISCL=4 VARIABLE ARTIFICIELLE NON SORTIE  
 EN FIN DE PHASE 1  
 ISCL=5 CYCLAGE  
 FX VALEUR DU PROGRAMME LINEAIRE  
 IMPR INDEX CONTROLANT LES IMPRESSIONS INTERMEDIAIRES  
 IMPR =3 ALORS IMPRESSION DE TOUS LES RESULTATS INTERMEDIAIRES

\*\*\*\*\*

IMPLICIT REAL\*8(A-F), REAL\*8(C-Z)  
 INTEGER W  
 REAL\*8 MAXMAC  
 EXTERNAL AES  
 DIMENSION T(1),F(1),IBASE(1)  
 IAP=1

\*\*\*\*\* EFSS TOLERANCE ASSOCIEE A L'ENTREE D'UNE VARIABLE  
 EFSS=1.E-6  
 \*\*\*\*\* EFSR TOLERANCE ASSOCIEE A LA SORTIE D'UNE VARIABLE  
 EFSR=1.E-6  
 \*\*\*\*\* EPX TOLERANCE ASSOCIEE A LA FIN DE LA PHASE 1  
 EPX=1.E-8  
 \*\*\*\*\* MAXMAC PLUS GRAND NOMBRE UTILISEE  
 MAXMAC=1.E+30  
 MN=M\*N  
 \*\*\*\*\* ISTD NOMBRE MAXIMUM D'ITERATIONS  
 ISTD=4\*M\*N  
 DO 1000 IIT=1,ISTD  
 \*\*\*\*\* CHARGEMENT DE F SUR LA BASE DANS LES  
 M PREMIERS ELEMENTS DE DE T  
 IF((IPHAS-1)1,2,1  
 2 N1=N  
 DO 30 I=1,M  
 N1=N1+1  
 IF((IBASE(N1)-INART)10,20,20

```
C T(I)=-1.
* GC TC 3C
C T(I)=0.
C CCNTINLE
  GC TC 3I
1 N1=N
  CC 32 I=1,M
  N1=N1+1
2 T(I)=F(N1)
**CALCUL DE FX
1 FX=0.
  CC 40 I=1,M
  III=1+I
  IF (ABS(T(III)).LT.1.E-30) GC TC 4C
  IF (ABS(T(I)).LT.1.E-30) GC TC 4C
  FX=FX+T(I)*T(III)
  CCNTINLE
  IF (IMP) 5,6,5
  CALL TEST(T,F,IBASE,III,FX,M,N,IMP)
  CCNTINLE
**DETERMINATION DE S ENTRANT DANS LA BASE
IS=C
N2=2+I
DS=EPS5
  CC 50 J= 3,n
  C=C.
  IF (IFBAS-1) 52,51,52
  IF (IBASE(J)-INART) 55,54,54
  C=-1.
  GC TC 55
  IF (IBASE(J)-INART) 53,56,56
  C=F(J)
  KN2=N2
  CC 60 I=1,M
  KN2=KN2+1
  C=C-T(KN2)*T(I)
  CCNTINLE
  IF (L-DS) 56,56,61
  DS=C
  IS=J
  N2=N2+I
  CCNTINLE
  IF (IS) 100,100,70
**DETERMINATION DE R SORTANT DE LA BASE
C MS=(IS-1)*M
FR=MAXMAC
IF=0
  CC 80 J=1,M
  MS=MS+1
  TMS=T(MS)
  IF (TMS-EP5R) 80,80,82
  IIZ=M+J
  RA=T(IIZ)/TMS
  IF (RR-RA) 80,80,81
1 FR=RA
  TRS=TMS
```



```
IR=J
8C CCNTINCE
IF(IR)118,118,200
18 ISCL=3
RETURN
*****VERIFICATION TABLEAU SIMPLICIAL DEBUT
20C KK=2
201 PF=M+IR
N1=M+IR
21C ZZ=T(N1)
T(N1)=ZZ+/TRS
30C CCNTINCE
N1=N1+M
IF(N1-M)210,210,211
211 MS=(IS-1)*M
MSR=MS+IR
T(MSR)=1./TRS
*****CHANGEMENT LIGNE COURANTE
CC 220 I=1,M
IF(I-IR)221,220,221
221 N1=M+I
N2=M+IR
I11=M+I
T(I11)
CC 222 J=N1,M,N,M
T(J)=T(J)-T(I11)*T(N2)
223 CCNTINCE
222 N2=N2+M
I1=M+I
T(I1)=C.
T(I1)=-T(I1)/TRS
22C CCNTINCE
*****VERIFICATION TABLEAU SIMPLICIAL FIN
IFA=IF+M
IF=IEASE(IFN)
IEASE(IFN)=IEASE(IS)
IEASE(IS)=IR
Z=F(IFN)
F(IFN)=F(IS)
F(IS)=Z
*****FIN DE CHANGEMENT DE BASE
GOTO(132,1000),KK
*****TRAITEMENT PHASE 1 CU PHASE 2
10C IF(IPHAS-1)120,110,120
120 ISCL=1
RETURN
*****TRAITEMENT FIN DE PHASE 1
11C IF(ABS(FX)-EPX)111,130,130
130 ISCL=2
RETURN
*****VERIFICATION -VARIABLES ARTIFICIELLES FORS-BASE
111 W=C
113 W=W+1
I13=N+W
IF(IEASE(I13)-INART) 112,114,114
114 IF=M+W+M
```

```

IR=W
TFS =1.E-6
ISS=2
IS=C
CC 117 J=I,M,N,M
IF(AES(T(J))-TFS) 117,117,116
6 TFS=ABS(T(J))
JW1=J
IS=ISS
7 ISS=ISS+1
TFS=T(JW1)
IF(IS)133,133,131
3 ISCL=4
RETURN
1 IF=W
KK=1
GOTO 201
2 KK=2
2 CONTINUE
IF(W.LT.N)GOTO 113
IF(IMP)7,8,7
WRITE(IMP,9)
FORMAT(/////***CHANGEMENT DE PHASE***'/////))
IFAS=2
GOTO 1010
C CONTINUE
ISCL=5
- RETURN
ENC

```

SLEPCLTINE MAXSG (XK,YK,GRAC,C1,TETA,TETA2,N1)

\*\*\*\*\*

XK + TETA(YK-XK) MAXIMISE F SUR LE  
SEGMENT ( XK , YK )  
XK + TETA2 (YK - XK) MAXIMISE F SUR LA  
DEMI-CERCLE ( XK , YK - XK )  
F EST QUADRATIQUE  
GRAC CONTIENT GRAC F(XK),  
C1 LA MATRICE DES DERIVEES SECONDES

\*\*\*\*\*

```

IMPLICIT REAL*8(A-F),REAL*8(G-Z)
EXTERNAL ABS
DIMENSION ANEX(20)
DIMENSION GRAC(1),C1(15,15),XK(1),YK(1)
C=C.GCC
CC 11111 I=1,N1
C=YK(1)-XK(1)
IF(ABS(C).LT.1.E-15)D=C.GCC
ANEX(I)=C

```



```
C=C+GRAC(I)*C
.111 CCNTINLE
C=AES(C)
B=C.CCG
CC 8C I=1,N1
Z=C.CCG
CC 8I J=1,N1
Z=Z+ANEX(J)*C1(J,I)
B1 CCNTINLE
B=E+ANEX(I)*Z
EC CCNTINLE
B=ABS(B)
TETA=1.0CG
IF(C.LT.3)TETA=C/B
TETA=AES(TETA)
TETA2=1.E30
IF(B.GT.1.E-15)TETA2=ABS(C/E)
RETURN
ENC
```

SLERCUTINE LECTU

\*\*\*\*\*

LECTU ACCUE A POINTS C ENTREE MULTIPLES  
RECRUPANT LES DIFFERENTES PHASES DE LECTURE  
C INITIALISATION ET DE CALCULS UTILILISEES PAR TOUS  
LES ALGORITHMES PROPOSES, LEUR ENCHAINEMENT  
ETANT CONTROLÉ PAR UN AUTRE PROGRAMME

\*\*\*\*\*

LE POINT C ENTREE LECTU CONTROLÉ LA LECTURE  
DES DIFFERENTES DONNEES CARACTERISANT LE  
PROBLEME A RESCURE

\*\*\*\*\*

IMPLICIT REAL\*8(A-F),REAL\*8(C-Z)  
LOGICAL ARCU  
LOGICAL PARAM  
LOGICAL BAS  
LOGICAL MAXLIN  
LOGICAL BICE  
LOGICAL DTZIG  
REAL\*8KCE  
INTEGER A21,A22,A23  
EXTERNAL AES  
EXTERNAL F3,FG3  
EXTERNAL A,FGA,F1,F2,FC1,FG2  
DIMENSION FG5(2C),FG6(2C)  
DIMENSION GENI(2C)

```
DIMENSION X(20),XK(20),YK(20),XPAX(20),TT(1000),FF(450),IBBASE(450
-),ANV(4),XGE(3000),XX(20)
DIMENSION Y(500),F(80),IBASE(80)
DIMENSION FXGE(450)
DIMENSION FG4(40)
DIMENSION C3(15,15)
DIMENSION XSA(50)
DIMENSION C2(15,15)
DIMENSION TPR(500),IBPR(80),FPR(80)
DIMENSION GFA(20)
COMMON/FUNA/CSTA,PA(15),QA(15,15)
COMMON U,LD,GRAC(40)
COMMON/FLA1/C1(15,15),P1(15)
COMMON/AAAA/AAAAA
COMMON/IDENT/C4(15,15),P3(15)
55 CONTINUE
   LEC=1
   READ(LEC,4433,ENC=5556)KOE
33 FCFMAT(FIC-2)
   IMF=3
```

\*\*\*CONTRAINTES LINEAIRES, LECTURE, MISE EN FORME

```
CC 37 I=1,EO
   F(I)=0.CCO
   F(I)=1000
   READ(LEC,100)M,N,IEC,INEQ,MAX
30 FCFMAT(5(12,1X))
101 FCFMAT(8(2X,F8.2))
12 FCFMAT(8(2X,F8.2))
   M2=2*M
   N=N+2
   MN=M*N
CC 10 I=1,M
   I1=M2+I
   READ(LEC,101)(T(J),J=I1,MN,M)
10 CONTINUE
   I1=M+1
   READ(LEC,102)(T(J),J=I1,M2)
   N1=M+1-INEQ
   NN2=N
```

CHARGEMENT DES COLONNES CORRESPONDANT AUX VARIABLES D'EGARTS  
NN2= NOMBRE DE COLONNES PHYSIQUEMENT UTILISEES

```
IF(N1)20,20,30
30 NN2=N+N1
   MN2=NN2*M
   MN=MN+1
CC 31 I=MN,MN2
31 T(I)=C.CCO
   M=M+INEQ-1
   M1=M+1
CC 32 I=MN,MN2,M1
```



```

32 T(I)=-1.000
    FIA DE CHARGEMENT
33 CONTINUE
    N3=NN2+M
    CC 33 I=3,N3
33 IEASE(I)=I-2

```

INART=INDICE PREMIERE VARIABLE ARTIFICIELLE  
 IFFAS=1 METHODE DES DEUX PHASES  
 IFFAS=2 ON PART DIRECTEMENT EN PHASE DEUX

```

    IFFAS=2
    INART=32000
    IF (IEC-M) 34,34,35
34 IFFAS=1
    INART=N-2+N1+IEC
35 CONTINUE
    EPS=0.00100
    NEGENE=C
    BETA=0.00100
    ALPHA=0.0100
    N1=N-2

```

\*\*\*\*\*LECTURE DES CARACTERISTIQUES DE LA FONCTION  
 \*\*\*\*\*ECONOMIQUE QUADRATIQUE ET DE LA CONTRAINTE  
 \*\*\*\*\* QUADRATIQUE

```

    CC 84 I=1,N1
E4 READ(LEG,101)(G1(I,J),J=1,N1)
    CC 85 I=1,N1
E5 READ(LEG,101)(CA(I,J),J=1,N1)
    READ(LEG,101)(P1(I),I=1,N1)
    READ(LEG,101)(FA(I),I=1,N1)
    READ(LEG,101) CSTA
    CC 88 I=1,N1
    CC 88 J=1,N1
    C1(I,J)=-C1(I,J)
E8 CA(I,J)=-CA(I,J)
    NG=N1+1
    RETURN

```

ENTRY FRW

\*\*\*\*\*

RESOLUTION DIRECTE DU PROGRAMME  
 MAXIMISER F(X) SOUS LES CONTRAINTES  
 A(X)=C , B.X = C , X POSITIF OU =C  
 F EST QUADRATIQUE , A EST QUADRATIQUE TOUJOURS NEGATIF OU =C

\*\*\*\*\*

```

CALL FCAL(ITIME)
MAXLIN=.TRUE.
ALPHA=1.E-5
I22=30

```

CALL FRMC(T,F,IGASE,M,AN2,INART,IPHAS,ISOL,N,XK,ALPHA,F1,FG1,I,  
IX,KCE,I22,C1,Z21,PAXLIN)  
CC TC 9642

ENTRY BICCN(NB4)

\*\*\*\*\*

LES ENTREES BICCN ET CHARGE PERMETTENT DE  
REACTUALISER LA VALEUR DE LA CCNTRAINTE  
NGN LINEAIRE POUR CHAQUE GENERATEUR

\*\*\*\*\*

BICE=.TRUE.  
NN=NEGENE  
NEEIS=NN  
GCTC 4C

ENTRY CHARGE(\*)

BICE=.FALSE.  
NN=NEGENE  
NEEIS=NN+1  
NEGENE=NN  
NGENE=NN\*AG+1  
N184=NGENE-1  
DC 11 I=1,N184,AG  
J=I  
DC 12 K=1,N1  
J=J+1  
X(K)=XGE(J)  
XGE(I)=A(X,N1)  
1 CCNTINLE  
IF(BICE)RETURN  
DC 17 I=1,N1  
X\*PAX(I)=XK(I)  
CCNTINLE  
IF(A(X\*PAX,N1).LT.C.)RETURN1  
IA=1  
ARCU=.FALSE.  
GCTC 18

ENTRY LECGEN(NAN,DTZIG)

\*\*\*\*\*

ON LIT NAN GENERATEURS DE DEPART  
SI DTZIG EST VRAI ON SAUVEGARDE LA VALEUR  
DE LA FONCTION ECONOMIQUE ASSOCIEE

\*\*\*\*\*

NEEIS=NN  
IP=NN





```

AREL=.TRUE.
NCGENE=G
NGENE=I
18  CONTINUE
   CC 14 I=1,IN
   IF (AREL) READ (LEC,IC1) (XMAX(J),J=1,N1)
   NCGENE=NCGENE+1
   IF (DTZIG) FXGE(NCGENE)=F1(XMAX,N1)
   XGE(NGENE)=A(XMAX,N1)
   NGENE=NGENE+1
   CC 15 J=1,N1
   XGE(NGENE)=XMAX(J)
15  NGENE=NGENE+1
14  CONTINUE
   IF (NCGENE.NE.ANN) GC TC9631
   CC 16 I=1,N1
   GEN(I)=XMAX(I)
16  XSA(I)=XMAX(I)
   FW=F1(XMAX,N1)
9631 RETURN

```

ENTRY FENEXT(A22,ALPHA)

\*\*\*\*\*

ON EFFECTUE A22 ETAPES POUR RESCUEUR  
 MAXIMISER F(X)+AAAAA\*A(X) A ALPHA PRES  
 SOUS LES CONTRAINTES E\*X=C , X POSITIF OU NUL

\*\*\*\*\*

```

   CC 11C I=1,N1
   CC 11G J=1,N1
11C GE(I,J)=C1(I,J)+AAAAA*CA(I,J)
   CALL FFWC(T,F,IBASE,M,AN2,INART,IPHAS,ISOL,N,XMAX,ALPHA,F3,FG3,1,
*   X,KCE,A22,C3,ZZ1)
   AX=A(XMAX,N1)
   FX=F1(XMAX,N1)
   WRITE(TMP,112)FX,AX,ZZ1
112 FCFMAT(' ',3E15.12)
   WRITE(IMP,113)(XMAX(I),I=1,N1)
113 FCFMAT(' ***',5E15.12)
   RETURN

```

ENTRY SAUF

C\*\*\*\*\*

C  
 C EFFECTUE LA SALVEGARDE DES DIFFERENTES DONNEES  
 C RELATIVES AUX CONTRAINTES LINEAIRES  
 C

C\*\*\*\*\*

```

   IFFH=IPHAS
   CC 8937 IX=1,500
8937 TFR(IX)=T(IX)

```

```

DC 8938 IX=1,8C
  FFF(IX)=F(IX)
538 IEFR(IX)=IBASE(IX)
  RETURN

```

ENTRY RESTO

\*\*\*\*\*

EFFECTUE LA RESTAURATION DES DIFFERENTES DONNEES  
RELATIVES AUX CCATRAINTES LINEAIRES

```

*****
  IPHAS=IPPW
  DC 8939 IX=1,5CC
939 T(IX)=TFR(IX)
  DC 894C IX=1,8C
  F(IX)=FPF(IX)
54C IEASE(IX)=IEFR(IX)
  RETURN

```

ENTRY INITI(ALPHA,A21,A22,A23,A24,\*,DTZIG,AAA1)

\*\*\*\*\*

DETERMINATION DES GENERATEURS DE DEPART  
UTILISES DANS L ALGORITHME MIXTE  
DEUX OU TROIS GENERATEURS SONT DETERMINES  
ALPHA PRECISION POUR FRANCK ET WOLFE  
A21 NOMBRES D ETAPES POUR LE GENERATEUR NO 1  
A22 NOMBRES D ETAPES POUR LE GENERATEUR NO 2  
A23 NOMBRES D ETAPES POUR LE GENERATEUR NO 3  
AA11, A24 TOLERANCES RELATIVES A LA DETERMINATION  
DU GENERATEUR NO3

```

*****
MAXLIN=.TRUE.
NECENE=0
NCENE=1

```

MAXIMISATION A ALPHA PRES DE F DANS B  
CETENTION DE XX

```

CALL FRC(T,F,IBASE,M,ANZ,INART,IPHAS,ISOL,N,XX,ALPHA,F1,FG1,1,X,K
*CE,A21,C1,ZZZ1,MAXLIN)
IF(ISOL.NE.6)RETURN1
IFAL=1
CC3333 I=1,5
IF(A(XX,N1).LT.0) CCTC 3334
DC 12333 I80=1,N1
X(I8C)=XX(I8C)

```

333 CCNTIALE



```
CALL FRWC(T,F,IBASE,M,NN2,INART,IPHAS,ISOL,N,XX,ALPHA,F1,FG1,C,X,K
*CE,A22,C1,ZZZ1,MAXLIN)
IF(ISCL.NE.6)RETURN1
33332 CONTINUE
IRAL=C
DC 07654 I=1,N1
E7654 XK(I)=XX(I)
33334 CONTINUE
NAGENE=NAGENE+1
IF(CTZIG)FXGE(NAGENE)=F1(XX,N1)
XGE(NAGENE)=A(XX,N1)
NAGENE=NAGENE+1
DC 50010 I=1,N1
X(I)=XX(I)
XGE(NAGENE)=XX(I)
NAGENE=NAGENE+1
50010 CONTINUE
C
C MAXIMISATION A ALPHA PRES DE A CANS B
C DETENTION DE XMAX
C
DC 22222 J=1,5
CALL FRWC(T,F,IBASE,M,NN2,INART,IPHAS,ISOL,N,XMAX,ALPHA,A,FGA,C,X,
*KCE,A23,C4,ZZZ1,MAXLIN)
IF(ISCL.NE.6)RETURN1
IF(A(XMAX,N1).GT.AAA1)GOTO 22223
DC 7897 I=1,N1
7897 X(I)=XMAX(I)
22222 CONTINUE
ISCL=21
RETURN1
22223 CONTINUE
NAGENE=NAGENE+1
IF(CTZIG)FXGE(NAGENE)=F1(XMAX,N1)
XGE(NAGENE)=A(XMAX,N1)
NAGENE=NAGENE+1
DC 80000 I=1,N1
XGE(NAGENE)=XMAX(I)
NAGENE=NAGENE+1
80000 CONTINUE
IF(IRAL.EQ.C)GOTO 55446
C
C XMAX FCINT DU SEGMENT ( XX , XMAX )
C TEL QUE A(XMAX)=C A A24 PRES
C
ECCC0 DC 80000 I=1,N1
ECCC0 XK(I)=(XMAX(I)+XX(I))/2.CCC
IF(A(XK,N1)) 80001,80001,ECCC2
ECCC2 DC 80003 I=1,N1
ECCC3 XMAX(I)=XK(I)
IF(A(XMAX,N1).LT.A24)GOTO 445
GOTO 80020
ECCC1 DC 80004 I=1,N1
ECCC4 XK(I)=XK(I)
GOTO 80020
445 CONTINUE
```

```

NGENE=NGENE+1
IF(DTZIG)FXGE(NGENE)=F1(XMAX,N1)
XGE(NGENE)=A(XMAX,N1)
NGENE=NGENE+1
CC 8345 I=1,N1
XGE(NGENE)=XMAX(I)
NGENE=NGENE+1
45 CCNTINLE
46 CCNTINLE
   IZZ=1
   CC 8542 I=1,N1
842 XSA(I)=XMAX(I)
   NEIS=NGENE
   RETURN
```

\*\*\*CHARGEMENT NOUVEAU GENERATEUR

```

CC KK=1
C1 XGE(NGENE)=A(XMAX,N1)
   NGENE=NGENE+1
   CC 2000 I=1,N1
   XGE(NGENE)=XMAX(I)
CC NGENE=AGENE+1
   NGENE=NGENE+1
   IF(DTZIG)FXGE(NGENE)=F1(XMAX,N1)
   BAS=.TRUE.
45 CCTC(32842,32843,18752,60000),KK
```

\*\*\*CHARGEMENT DES GENERATEURS SUPPLEMENTAIRES  
\*\*\*SELON LA VARIANTE

```

842 KK=2
   CALL FC1(X ,N1,FG4)
757 CCNTINLE
   CALL AP11 (X ,FG4,N1,IEQ,INEQ,TPR,FG4,Q1,XMAX,#32845,M,BAS)
   GC TC 4001
843 KK=3
   CALL FGA(XK,N1,FG4)
755 CALL AP11 (XK ,FG4,N1,IEQ,INEQ,TPR,FG4,Q1,XMAX,#32845,M,BAS)
   GC TC 4001
752 J=N1-1
   KK=4
   GC TC 18757
```



ENTRY CALCL(ALPHA,BETA,A21,A22,A24,I24,EPSU,DTZIG,PARAM,ISOL,A44,  
I K44)

```
C*****
C
C   CALCUL EFFECTUE L ENCHAINEMENT DES CALCULS
C   POUR LES DIFFERENTS ALGORITHMES DE TYPE
C   EFFYCENTRIQUE EN FONCTION DES INDICATEURS
C   DTZIG, PARAM ET DES DIFFERENTES FONCTIONS FOURNIES
C
C   ALPHA PRECISION POUR FRWC
C   A21,A22 NOMBRES DES ITERATIONS POUR FRWC
C   A24 TEMPS MAXIMUM POUR L ETAPE DE CALCUL
C   I24 NOMBRE MAXIMUM DE GENERATEURS
C   ISOL INDICATEUR DU TYPE DE LA SOLUTION
C   A44, K44 PARAMETRES RELATIFS A LA PROJECTION
C   EVENTUELLE DES GENERATEURS
C*****
      M2=NA2+N
      CC 6997 I=NI,P2
6997 FG4(I)=C.
      NGENE=NEBIS
      CC 8543 I=1,NI
8543 XMAX(I)=XSA(I)
      CC 444 I=1,NI
      XK(I)=XMAX(I)
      YK(I)=XMAX(I)
      444 XX(I)=XMAX(I)
      CALL HCR1(I,TIME)
6CCCC CCATINLE
      I22=A21
      IF(NEGENE/2*2.EC.NEGENE)I22=A22
      CC 333 I=NI,P2
      333 GRAD(I)=0.000
      CALL FQ1(XK,NI,GRAD)
C**** CHARGEMENT DU TABLEAU SIMPLICIAL DU PROGRAMME DIRECTEUR
446 J=1
      JJ=5
      CC 2010 I=1,NEGENE
      TT(JJ)=XGE(J)
      J=J+NG
      JJ=JJ+1
      TT(JJ)=1.000
2010 JJ=JJ+1
      TT(JJ)=-1.000
      JJ=JJ+1
      TT(JJ)=0.000
      TT(3)=0.000
      TT(4)=1.000
C**** FONCTION ECONOMIQUE
      JJ=3
      IF(DTZIG)GOTO 2032
      J=1
      CC 2020 I=1,NEGENE
      ZZ=0.000
```

```
J=J+1
  CC 2031 I1=1,N1
  ZZ=ZZ+XGE(J)*GRAC(I1)
31 J=J+1
  FF(JJ)=ZZ
20 JJ=JJ+1
  GCTC 2033
32 CC 2034 I=1,NBGENE
  FF(JJ)=FXGE(I)
34 JJ=JJ+1
33 N1C=NBGENE+2+3
  CC 2030 I=1,3
  FF(JJ)=0.0EG
30 JJ=JJ+1
  CC 2040 I=3,N1C
40 IEPASE(I)=I-2
```

```
**RESOLUTION DU PROGRAMME DIRECTEUR
  OBTECTION DU POINT XK
```

```
L1=1
L2=0
CALL SPBLI(I1,FF,IEBASE,2,NEGENE+3,NBGENE+2,L1,L2,FX,1)
ISCL=-L2
IF(L2.NE.1)GCTC SCCC
**CONSTRUCTION SOLUTION ET VARIABLES DUALES
N3=NEGENE+1
N4=IEBASE(NEGENE+4)
N5=IEBASE(N1C)
WC=C.CEO
W1=1.CEO
WM1=-1.CEO
IF(N4-N3)2100,2110,2100
CC IF(N5-N3)2120,2130,2120
10 N6=(N5-1)*NG+1
  CALL INV(WM1,WC,XGE(N6),W1,ANV,DET)
  AK=XGE(N6)*ANV(4)
  GCTC 2200
30 N6=(N4-1)*NG+1
  CALL INV(XGE(N6),W1,WM1,WC,ANV,DET)
  AK=XGE(N6)*ANV(3)
CC CC 2210 I=1,N1
  N6=N6+1
10 YK(I)=XGE(N6)
  GCTC 2220
20 N6=(N4-1)*NG+1
  N7=(N5-1)*NG+1
  AK1=XGE(N6)
  AK2=XGE(N7)
  CALL INV(XGE(N6),W1,XGE(N7),W1,ANV,DET)
  A1=ANV(3)
  IF(A1.LT.C.) A1=C.CEO
  IF(A1.GT.1.) A1=1.CEO
  CC 2125 I=1,N1
  N6=N6+1
  N7=N7+1
```



```
125 YK(I)=A1*XGE(N6)+(1.010-A1)*XGE(N7)
    AKK=AK1*ANV(3)+AK2*ANV(4)
220 IF(ABS(CET)-1.E-10)2222,2222,2221
222 ISCL=7
    GCTC 9CCC
221 CONTINUE
    A1=FF(NDGENE+4)
    A2=FF(N1C)
    U=A1*ANV(1)+A2*ANV(2)
    UC=A1*ANV(3)+A2*ANV(4)
    L=ABS(U)
    LC=-LC
```

RESOLUTION DU PROGRAMME AUXILIAIRE  
CALCUL DU NOUVEAU GENERATEUR XMAX

```
7 CC 4011 I=1,N1
(11 X(I)=YK(I)
    IF(DI2IG)GCTC 94
    CC 87 I=1,N1
    CC 87 J=1,N1
7 C2(I,J)=U*CA(I,J)
    CC TC 89
64 CC 90 I=1,N1
    CC 90 J=1,N1
60 C2(I,J)=C1(I,J)+U*CA(I,J)
    MAXLIN=.TRUE.
776 CALL FRC(I,F,IBASE,M,N2,INART,IPHAS,ISCL,N,XMAX,ALPHA,F2,FG2,C,X
    *,KCE,122,C2,221,MAXLIN)
    Z21=ABS(Z21)
    IF(ISCL.NE.6)GCTC 90C
****TEST D APRES LOGICLE
    AYK=A(YK,N1)
    IF(AYK.LT.C)GCTC 90C0
    CALL FCRL(11TIME)
    RTIME=(11TIME-ITIME)/200.
    IF(DI2IG) GCTC 91
777 DES=U*A(XMAX,N1)
848 CC 2345 I=1,N1
2345 DES=DES+GRAD(I)*(XMAX(I)-XK(I))
    CES=ABS(DES)
    CES=CES+Z21
    GCTC 92
51 CES=F2(XMAX,N1)
    CES=CES+Z21
    CC 93 I=1,N1
52 XK(I)=YK(I)
    IF(PARAN) GC TC 92
    AXS=A(XMAX,N1)
    FXS=F1(XMAX,N1)
    WRITE(IMP,456)AXS,AXS,FXS,FXS,RTIME,AYK
    IF(ABS(A(XMAX,N1)).GT.1.E-10) GC TO 3CCC
    CC 99 I=1,N1
99 XK(I)=XMAX(I)
    ISCL=12
```

```
GC TC 9000
- FCFMAT(' ',4E22.14,F1C.2,E12.4)
CONTINUE
F1X=F1(XK,N1)
IF(ABS(F1X).LT.1.E-10) F1X=1.E-10
AX=A(XK,N1)
CF1=CCS/F1X
WRITE(IMP,456)F1X,AX,CCS,CF1,RTIME,AYK
CCS=ABS(CCS)
IF(CCS-BETA)8500,8900,3000
C ISCL=6
GOTO 9000
C CONTINUE

MAXIMISATION DE F SUR LE SEGMENT (XK ,YK )
DETERMINATION DU NOUVEAU POINT DE LINEARISATION XK
IF(CTZIG)GOTO 5011
CC 3010 I=1,N1
XX(I)=XK(I)
C X(I)=XK(I)
CALL FAXSG (XK,YK,GRAD,Q1,TETA,TETA2,N1)
CC 22522 I=1,N1
2 XK(I)=TETA*YK(I)+(1.000-TETA)*XK(I)
1 CC 311 I=1,N1
X(I)=X*AX(I)
ISCL=3C
* TESTS D ARRÊT PHYSIQUE
IF(RTIME.GT.424) GC TC 9000
4 IF (ARGENE-I24)4000,4000,4010
C ISCL=9
C CONTINUE
IF(CTZIG) GC TC 9642
F1Y=F1(XK,N1)
ZK4=A2(XK,N1)
IF((F1X.LT.FK).CR.(ABS(ZW4).GT.1.E-9))GOTO 9642
CC 9642 I=1,N1
2 XSA(I)=XK(I)
FY=F1Y
C CONTINUE
IF(ABS(F1X).LT.1.E-10) F1X=1.E-10
CF1=CCS/F1X
CALL FCAL(II,TIME)
RTIME=(II,TIME-ITIME)/200.
AX=A(XK,N1)
F1X =F1(XK,N1)
WRITE(IMP,456)F1X,AX,CCS,CF1,RTIME
WRITE(IMP,1000)ISCL,ISCL,ISCL
WRITE(IMP,4434)NEGENE
WRITE(IMP,4435)KCE
WRITE(IMP,4002)(XK(I),I=1,N1)
4 FCFMAT(' NOMBRE D ITERATION',I5)
5 FCFMAT(' COEFFICIENT',F5.2)
2 FCFMAT(' LINEARISATION',3E26.16)
RETURN
6 CALL EXIT
FNC
```





FUNCTION GTEST(X,XBAR,N,FGNCT)

```
C*****
C
C   EVALUE LA QUANTITE
C   GTEST=FGNCT(X,N)*(XBAR-X)
C
C*****
C   IMPLICIT REAL*8(A-F), REAL*8(C-Z)
C   EXTERNAL ABS
C   DIMENSION X(1),XBAR(1)
C   DIMENSION GRAC(20)
C   COMMON U,UC,GXX(40)
C   COMMON/IDENT/G4(15,15),P3(15)
C   COMMON/FUN1/G1(15,15),P1(15)
C   COMMON/FLNA/CST4,PA(15),CA(15,15)
C   COMMON/AAAAA/AAAA
C   CALL FGNCT(X,N,GRAC)
C   GTEST=C.C
C   DO 10 I=1,N
C     Z1=XBAR(I)-X(I)
C     Z2=ABS(Z1)
C     IF(Z2.LT.E-10) Z1=0.0
10  GTEST=GTEST+GRAC(I)*Z1
C   GTEST=ABS(GTEST)
C   RETURN
C   END

SUBROUTINE SCLPL(X,N1,M,N2,IBASE,T)
```

```
C*****
C
C   X EST LA SOLUTION DE BASE REALISABLE
C   ASSOCIEE A LA BASE IBASE
C
C   M NOMBRE DE CONTRAINTES
C   N1 NOMBRE DE VARIABLES NATURELLES
C   N2=N1+NOMBRE DE VARIABLES D ECARTS EN SUP OU EGAL+2
C
C*****
C   IMPLICIT REAL*8(A-F), REAL*8(C-Z)
C   DIMENSION X(1),T(1),IBASE(1)
C   DO 10 I=1,N1
10  X(I)=0.0
C   I1=N2
C   DO 11 I=1,M
C     I1=I1+1
C     I2=IE/SE(I1)
C     IF(I2-N1)20,20,11
20  I2=M+I
C     X(I2)=T(I3)
11  CONTINUE
C   RETURN
C   END
```

SLERCUTINE TEST(T,F,IBASE,ITER,FX,M,N,IMP)

\*\*SOLS PROGRAMME D IMPRESSION DES  
\*\*RESULTATS INTERMEDIAIRES PROVENANT  
\*\*DE SPBLI (METHODE SIMPLICIALE)

```
IMPLICIT REAL*8(A-H),REAL*E(G-Z)
DIMENSION T(1),F(1),IBASE(1)
FORMAT('***TABLEAU SIMPLICIAL**/'-----'////)
FORMAT('*** SECCNC ** MEMBRE **/'-----'////)
FORMAT('***FONCTION ECCNOMIQUE/'-----'////)
FORMAT('*** ITERATION NUMERIC **/'-----'////)
FORMAT('***** E A S E ****/'-----'////)
FORMAT(1X,9E12.3)
FORMAT(1X,9I12)
FORMAT(///)
FORMAT(//)
FORMAT(12C(' '))
WRITE(IMP,144)
M3=M+1
M2=2*M
MM=M*N
NN=M*N
WRITE(IMP,130)
WRITE(IMP,141)ITER
WRITE(IMP,142)
IF(ITER-1)20,30,20
CONTINUE
WRITE(IMP,120)
WRITE(IMP,142)
WRITE(IMP,140)(F(I),I=3,MM)
WRITE(IMP,142)
WRITE(IMP,100)
DO 10 I=1,N
M1=M2+I
WRITE(IMP,140)(T(J),J=M1,NN,M)
WRITE(IMP,143)
CONTINUE
WRITE(IMP,142)
WRITE(IMP,110)
WRITE(IMP,140)(T(I),I=M3,M2)
WRITE(IMP,142)
WRITE(IMP,145)
WRITE(IMP,141)(IBASE(I),I=3,MM)
WRITE(IMP,142)
WRITE(IMP,120)
WRITE(IMP,140)FX
WRITE(IMP,142)
RETURN
END
```



FUNCTION FCP(X,N,C,P)

C\*\*\*\*\*

C  
C  
C  
C

$$F(X,N,F,C) = P * X + 1/2 * X * C * X$$

C\*\*\*\*\*

```

IMPLICIT REAL*8(A-F),REAL*8(C-Z)
DIMENSION X(1),P(1),Q(15,1)
DIMENSION R(15)
A=C.
DO 10 I=1,N
Z=C.
DO 11 J=1,N
IF(X(J).LT.1.E-10) X(J)=0.C
11 Z=Z+X(J)*C(J,I)
A=F(I)*X(I)+A
R(I)=Z/2.
10 CONTINUE
DO 12 I=1,N
A=A+P(I)*X(I)
12 CONTINUE
FCP=A
RETURN
END

```

SUBROUTINE GCP(X,N,GR,C,P)

C\*\*\*\*\*

C  
C  
C  
C

$$GF = P * X + C * X$$

C\*\*\*\*\*

```

IMPLICIT REAL*8(A-F),REAL*8(C-Z)
DIMENSION X(1),GR(1),P(1),C(15,1)
DO 10 I=1,N
Z=C.
DO 11 J=1,N
IF(X(J).LT.1.E-10) X(J)=0.C
11 Z=Z+X(J)*C(J,I)
GR(I)=F(I)+Z
10 CONTINUE
RETURN
END

```

\*\*\*\*\*  
F N C T I O N S     U T I L I S E E S  
\*\*\*\*\*

F N C T I O N   A ( X , N )

\*\*\* DETERMINATION DE LA CONTRAINTE A(X) POSITIF OU NUL

```

IMPLICIT REAL*8(A-H), REAL*8(C-Z)
DIMENSION X(1)
COMMON U, UC, GRAD(40)
COMMON/FUN1/C1(15,15), P1(15)
COMMON/FUNA/CSTA, PA(15), QA(15,15)
COMMON/AAAAA/AAAAA
COMMON/IDENT/C4(15,15), P3(15)
A=FCP(X,N,CA,PA)+CSTA     +AAAAA
RETURN

```

ENTRY A2(X,N)

```

A2=FCP(X,N,CA,PA)+CSTA
RETURN
ENTRY F1(X,N)

```

\*\*\* DETERMINATION DE LA FONCTION ECONOMIQUE

```

F1=FCP(X,N,C1,P1)
RETURN
END

```

F N C T I O N   F 2 ( X , N )

```

IMPLICIT REAL*8(A-H), REAL*8(C-Z)
DIMENSION X(1)
COMMON U, UC, GRAD(40)
COMMON/FUN1/C1(15,15), P1(15)
COMMON/FUNA/CSTA, PA(15), QA(15,15)
COMMON/AAAAA/AAAAA
COMMON/IDENT/C4(15,15), P3(15)
F2= U*A(X,N) + UC
DO 10 I=1,N
F2= F2 +GRAD(I)*X(I)
RETURN

```

ENTRY F3 (X,N)

```

F3=FCP(X,N,C4,P3)
RETURN
END

```



SLERCUTINE FG1(X,N,GR1)

```
*****DETERMINATION DU GRADIENT DE LA FONCTION ECONOMIQUE
IMPLICIT REAL*8(A-F), REAL*8(C-Z)
DIMENSION X(1),GR1(1)
COMMON U,UC,CFAC(4C)
COMMON/FLN1/C1(15,15),P1(15)
COMMON/FLNA/CSTA,PA(15),QA(15,15)
COMMON/AAAA/AAAA
COMMON/IDENT/C4(15,15),P3(15)
CALL GCF(X,N,GR1,C1,P1)
RETURN
```

ENTRY FGA(X,N,GR1)

```
*****DETERMINATION DU GRADIENT DE LA CONTRAINTE
*****NON LINEAIRE
CALL GCF(X,N,GR1,CA,PA)
RETURN
END
```

SLERCLTINE FG2(X,N,GR1)

```
IMPLICIT REAL*8(A-F), REAL*8(C-Z)
DIMENSION X(1),GR1(1)
DIMENSION CR2(15)
COMMON/FLNA/CSTA,PA(15),CA(15,15)
COMMON/FLN1/C1(15,15),P1(15)
COMMON/AAAA/AAAA
COMMON/IDENT/C4(15,15),P3(15)
COMMON U,UC,GRAC(4C)
CALL FGA(X,N,GR1)
DO 10 I=1,N
GR1(I)=GRAC(I)+U*CR1(I)
RETURN
```

ENTRY FG3(X,N,GR1)

```
CALL GCF(X,N,GR1,C4,P3)
RETURN
END
```

SEQUENCE D APPEL CORRESPONDANTE A  
UNE VARIANTE DE LA PARAMETRISATION

```

*****
IMPLICIT REAL*8(A-H), REAL*8(C-Z)
LOGICAL PARAM
LOGICAL DTZIG
EXTERNAL ABS
COMMON/AAAAA/AAAAA
COMMON/FUNAA/CSTA,PA(15),CA(15,15)
NAMELIST/BP/AAAAA,TEMP,PARAM,DTZIG,J,NNN,AV,A44,KW4,LMA,XWK
AV=0.300
KIT=250
ALPHA=1.E-6
EPSU=1.E-6
BETA=1.E-4
TEMP=30.
IMP=3
PARAM=.FALSE.
DTZIG=.TRUE.
AAAA=C.
J=5
NNN=1

```

\*\*\*FIN INITIALISATION STANDART

```

LEC=1
READ(LEC,BP)
WRITE(IMP,111)AAAAA,TEMP,PARAM,DTZIG,J,NNN
WRITE(IMP,BP)
TAN=TEMP
VV=AAAA
NE=NNN
CALL LECTU
KIT=4
TEMP=TAN
NN=N84
AAAA=VVV
AAAA=ABS(CSTA)
WRITE(IMP,9991)
CALL LECCEN(NNN,DTZIG)
WRITE(IMP,9991)
CC 15 L=1,LPA
IF(AAAAA.GT.XWK)LECTC 2
TEMP=300.
KIT=250
KIT = KIT + 12
C2 FORMAT('*****AAAAA=',E19.12)
WRITE(IMP,EC2)AAAAA
CALL CALCU(ALPHA,BETA,J,J,TEMP,KIT,EPSU,DTZIG,PARAM,ISEL,A44,KW4)
IF(AAAAA.LE.XWK)GO TO 1
AAAA=AAAA*AV

```



```
16 CALL BICCN(N84)
15 CONTINUE
   GOTO 1
9999 CALL EXIT
111 FCFMAT(1X,100(' ')/1X,'CCEF PARAM =',E19.12/1X,'TEMPS=',F10.2//1X
*, 'LANTZIG EXTERIEURE',L5/1X,'LANTZIG',L5/1X,'NB ETAPES CALCUL=',I4
*/1X,'NE GENERATEURS DE DEPART',I3//)
9991 FCFMAT('1','*****F(X)*****A(X)*****
*DES ***** C/F*****TEMPS*****')
   END
```

B I B L I O G R A P H I E

\* \* \* \* \*



- [1] ABADIE (J.M) , WILLIAMS (A.C)  
*Dual and parametric methods in decomposition*  
Recent advances in mathematical programming ( Graves et Wolfe )  
Mc Graw-Hill- 1963 - p. 149-158
- [2] BALAS (E.)  
*An infeasibility-pricing decomposition method for linear programs*  
Oper. Res. - 14 (1) - 1966 - p.847-872
- [3] BEN-ISRAEL (A.) , ROBERS (Ph.D)  
*A decomposition method for interval linear programming*  
Manag.Sc. - 16 (5) - 1970 - p. 374-387
- [4] BENDERS (J.)  
*Partitioning procedures for solving mixed-variables programming problems*  
Num. Math. - 4 (3) - 1962 - p. 238-252
- [5] CHANDRA (S.)  
*Decomposition principle for linear fractional functional programs*  
R.I.R.O. - 10 - 1968 - p. 65-72
- [6] COLVILLE (A.R)  
*A comparative study on nonlinear programming codes*  
Tech. Report No 320-2949, IBM New-York Scientific Center, 1968

- [7] COQUET (G.)  
*Ensembles convexes de  $\mathbb{R}^n$*   
Publication n°12, Laboratoire de calcul de Lille - 1968 -
- [8] DANTZIG (G.B.)  
*Linear programming and extensions*  
Princeton University Press, 1963
- [9] DANTZIG (G.B.) , WOLFE (Ph.)  
*Decomposition Principles for linear programs*  
Oper. Res. - 8 (1) - 1960 - p. 101-111
- [10] DANTZIG (G.B.) , WOLFE (Ph.)  
*The decomposition algorithm for linear programs*  
Econometrica - 29 (4) - 1961 - p. 767-778
- [11] DEM'YANOV (V.F.) , RUBINOV (A.M.)  
*The minimization of a smooth convex functional on a convex set*  
S.I.A.M. Control - 5 (2) - 1967 - p. 280-294
- [12] GEOFFRION (A.M.)  
*Primal resource-directive approaches for optimizing nonlinear decomposable systems*  
Oper. Res. - 18 (1) - 1970 - p. 375-403
- [13] FRANK (M.) , WOLFE (Ph.)  
*An algorithm for quadratic programming*  
Nav.Res.Logist.Quat. - 3 (1,2) - 1956 - p. 95-110

- [14] GAUTHIER (J.M.)  
*Paramétrisation de la fonction économique d'un programme linéaire*  
Rev.Fr. de Recherche Opérationnelle (22) - 1962 - p. 5-20
- [15] HOLLOWAY (C.A.)  
*An extension of the Frank and Wolfe method of feasible directions*  
Mathematical programming - 6 - 1974 - p. 14-27
- [16] HUARD (P.)  
*Optimisation dans  $\mathbb{R}^n$*   
Cours D.E.A. Traitement de l'information. Laboratoire de calcul  
Lille - 1972 -
- [17] HUARD (P.)  
*Tour d'horizon en programmation non linéaire*  
Bulletin de la direction des études et recherches de l'E.D.F.  
Série C - n°1 - 1971 - p.35-70
- [18] HUARD (P.)  
*Mathématiques des programmes économiques*  
Monographie A.F.I.R.O. - DUNOD - 1964
- [19] HUARD (P.)  
*Résolution des programmes mathématiques par la méthode des centres*  
Note E.D.F. n° HR 5467 du 26 déc. 1963
- [20] HUARD (P.), BROISE (P.), SENTENAC (J.)  
*Décomposition des programmes mathématiques*  
Monographie A.F.I.R.O. - DUNOD - 1968

- [21] KANTI SWARUP  
*Linear fractional functionals programming*  
Oper. Res. - 13 (6) - 1965 - p. 1029-1036
- [22] KUHN (H.W.) , TUCKER (A.W.)  
*Nonlinear programming*  
Proc. 2nd Berk. Symp. University of California  
Press . Berkeley - 1951 - p. 481-492
- [23] LOOTSMA (F.A)  
*Boundary properties of penalty functions for constrained minimization*  
Thèse - Université de Eindhoven - 1970
- [24] OBEL (B.)  
*A note on mixed Procedures for Decomposing Linear Programming Problems (unpublished)*  
Institute for History and Social Science, Odense University- 1977
- [25] ROODE (J.D.)  
*Generalized lagrangian functions in mathematical programming*  
Thèse - Université de Leiden - 1968
- [26] ROSEN (J.B.) , ORNEA (J.C.)  
*Solution of nonlinear programming problems by partitionning*  
Manag. Sc. - 10 (1) - 1963 - p.160-173
- [27] SIMMONARD  
*Programmation linéaire*  
DUNOD - 1973

- [28] UZAWA (H.)  
*Iterative methods for concave programming*  
in ARROW (K.), HURWICZ (L.), UZAWA (H.):  
*Studies in linear and Nonlinear Programming* - chap. 10  
Stanford University Press, Stanford, Calif. 1958
- [29] WHINSTON (A.)  
*a decomposition algorithm for quadratic programming*  
Cahiers du C.E.R.O. - 8 (2) - 1966 - p. 112-131
- [30] WHINSTON (A.)  
*A dual decomposition algorithm for quadratic programming*  
Cahiers du C.E.R.O. - 6 - 1964 - p. 188-201
- [31] WHINSTON (A.), VAN DE PANNE (C.)  
*The simplex and the dual method for quadratic programming*  
Oper. Res. Quart. - 15 - 1964 - p. 355-388
- [32] WILLIAMS (A.C.)  
*A treatment of transportation problem by decomposition*  
S.I.A.M. Vol 10 - n°1 - 1962 - p. 35-48
- [33] ZANGWILL (W.I.)  
*Nonlinear programming - An unified approach*  
Prentice Hall, Inc. Englewood Cliffs, New Jersey - 1969.
- [34] ZANGWILL (W.I.)  
*Nonlinear programming via penalty functions*  
Management Science - 13 (5) - 1967 - p. 344-358