

THESIS / THÈSE

MASTER EN SCIENCES MATHÉMATIQUES

Sur l'algorithme de H. TUY en programmation concave

DU FOUR, Philippe

Award date:
1983

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES N.D. DE LA PAIX
NAMUR
FACULTE DES SCIENCES

Sur l'algorithme de H. TUY
en programmation concave

Memoire présenté pour l'obtention
du grade de Licencié en Sciences
mathématiques
par

Année 1982-83

DU FOUR

Promoteur :

Philippe

NGUYEN V. HIEN

Je tiens à exprimer toute ma gratitude à Messieurs Nguyen V. Hien et J.J. Strodiot pour l'aide et les conseils qu'ils m'ont prodigués tout au long de l'élaboration de ce mémoire.

Je remercie également mes parents et toute ma famille pour leur aide lors de la rédaction de ce mémoire et pour leurs encouragements tout au long de mes études.

TABLE DES MATIERES

Introduction

Chapître I : Description de la méthode

I.1. Problème	1
I.2. Méthode	3
I.3. Interprétation de la méthode	13

Chapître II : Démonstration théorique de la convergence

II.1. Convergence	15
II.2. Processus exhaustif de subdivision	17
II.3. Estimation consistante d'une borne	24
II.4. Cas particuliers	34

Chapître III : Implémentation de l'algorithme

III.1. Algorithme pour une solution ξ -optimale	35
III.2. Programme algorithmique en FORTRAN VII	41

Chapître IV : Amélioration de la méthode

IV.1. Interprétation	55
IV.2. Théorème de convergence	56
IV.3. Modifications engendrées	57

Chapitre V : Exemples numériques

V.1. Exemples testés 62

V.2. Interprétation des résultats obtenus 69

Références

INTRODUCTION

En programmation convexe, on a l'habitude d'examiner des problèmes de minimisation de fonction convexe sur un ensemble fermé. Ceux-ci se résolvent à l'aide d'algorithmes basés sur une propriété-clé, à savoir que tout minimum local est global.

Par contre, la maximisation d'une fonction convexe sur un ensemble convexe fermé ne peut se résoudre de la même façon. Cependant, on sait que pour ce type de problème qui est équivalent à minimiser une fonction concave sur un ensemble convexe fermé, le maximum si il existe, est atteint en au moins un des points extrêmes de l'ensemble admissible.

La programmation concave n'a pas qu'un intérêt théorique, notamment les problèmes "d'économie d'échelle" et de maximisation de la fonction distance euclidienne sur un polyèdre. D'autre part, elle recouvre une classe assez large de problèmes de programmation mathématique:

1) Tout problème complémentaire linéaire et convexe de la forme :

$$\text{Trouver } x \in \mathbb{R}_+^n, y \in \mathbb{R}_+^n \text{ tel que } y=w(x), \langle x, y \rangle = 0 \quad (1)$$

où $w_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ($i=1, \dots, n$) sont des fonctions concaves,

est équivalent au problème de programmation concave :

$$\text{Minimiser } \sum_{i=1}^n \min \{x_i, w_i(x)\}, \text{ sous contraintes } x \geq 0, w(x) \geq 0, \quad (2)$$

c'est-à-dire que toute solution de (1) est une solution optimale de (2), de valeur optimale zéro, et réciproquement.

2) Le problème de minimisation de la différence de 2 fonctions convexes :

$$\text{Minimiser } f(x) - g(x), \text{ s.c. } x \in D,$$

où f et g sont des fonctions convexes et D est un sous-ensemble convexe fermé de \mathbb{R}^n ,

peut être réécrit sous la forme :

Minimiser $z - g(x)$, s.c. $f(x) \leq z$, $x \in D$,

qui est un problème de programmation concave.

3) En programmation bilinéaire, un problème du type :

Minimiser $f(x,y)$, s.c. $x \in D$, $y \in E$,

où D et E sont des sous-ensembles convexes fermés de \mathbb{R}^n et de \mathbb{R}^m respectivement, $f(x,y)$ est une fonction affine en x quand y est fixé et en y quand x est fixé,

est équivalent à :

Minimiser $g(x)$, s.c. $x \in D$,

où $g(x) = \min_{y \in E} f(x,y)$ est une fonction concave.

Diverses recherches ont été effectuées à ce sujet, notamment la méthode des coupes internes de H. TUY (1964) [1], la méthode des coupes externes de FALK & HOFFMAN (1976) [2] et la méthode de subdivision des cônes de H. TUY (1980) [3].

C'est cette dernière méthode qu'on va examiner ici. Elle est basée à la fois sur un processus de subdivision et sur l'estimation d'une borne inférieure d'un cône.

Le cône initial est le plus petit cône convexe contenant le domaine admissible. Si la borne inférieure estimée n'est pas raisonnable, alors on fractionne le cône suivant une procédure de bisection, en plusieurs sous-cônes plus petits; sinon il est éliminé. Tant qu'il subsistera des cônes, la meilleure solution admissible ne sera pas jugée optimale.

Dans un premier chapitre, on exposera la méthode pour ensuite

la démontrer théoriquement.

Au troisième chapitre, on détaillera le programme algorithmique codé en FORTRAN VII.

Par la suite, on proposera une amélioration de la méthode qui concerne la subdivision et qui tiendra compte des informations fournies dans l'estimation de la borne.

Dans le dernier chapitre, on comparera par des exemples numériques, les deux méthodes et on donnera une interprétation des résultats obtenus.

CHAPITRE I.

Description de la méthode.

=====

I.1. Problème :

On se propose de résoudre le problème suivant :

$$(P) \quad \begin{cases} \text{MIN } f(x) \\ \text{s.c. } x \in D \end{cases}$$

où f est une fonction réelle concave (ou quasi-concave) et continue sur \mathbb{R}^n

et D est un ensemble polyédral convexe de \mathbb{R}^n .

Ce type de problème ne peut être résolu par les méthodes classiques d'optimisation locale. Toutefois on sait que si il existe, le minimum est atteint en au moins un des points extrêmes de l'ensemble admissible défini par les contraintes.

La méthode proposée est basée sur une classe de procédures "branch and bound" dans laquelle un cône contenant l'ensemble admissible, est successivement subdivisé en sous-cônes de plus en plus petits. Sur chacun de ces sous-cônes, on estime une borne inférieure suivant la technique de coupage de plan développé par H. TUY. Par la "compacité" de cette méthode d'estimation, on sait que l'erreur d'estimation approche zéro tel que le sous-

cône devient de plus en plus mince et tend vers un rayon. Pour assurer la convergence, on applique le processus de "branching" de telle façon qu'on génère une séquence de sous-cônes qui tendent vers un rayon.

Avant de décrire la méthode, il est nécessaire de fixer certaines hypothèses :

- | | |
|-----|---|
| (H) | <p>(i) $\forall a \in \mathbb{R} : S = \left\{ x \in \mathbb{R}^n \mid f(x) \geq a \right\}$ est <u>borné</u>;</p> <p>(ii) l'ensemble admissible $D = \left\{ x \in \mathbb{R}^n \mid A \cdot x \leq b \right\}$ est <u>borné</u> (où $A =$ matrice $m \times n$ et $b =$ vecteur $m \times 1$);</p> <p>(iii) l'origine $0 \in \mathbb{R}^n$ est un sommet du polyèdre D.</p> |
|-----|---|

Remarques :
.....

(ii) le domaine admissible D est un polyèdre convexe de \mathbb{R}^n c'est-à-dire est défini par un système de contraintes linéaires d'inégalités : $A \cdot x \leq b$;

(iii) $s^0 \equiv 0$ est un point extrême de D , adjacent à exactement n autres points extrêmes s^1, s^2, \dots, s^n où l'ensemble $\left\{ s^1, s^2, \dots, s^n \right\}$ est linéairement indépendant.

Au départ, on a un cône convexe M_0 engendré par les n rayons issus de 0 et passant par s^1, s^2, \dots, s^n respectivement, qui est le plus petit cône issu de 0 , contenant l'ensemble admissible D .

I.2. Méthode :

La procédure du type "branch and bound" comprend 2 opérations de base :

(a) le processus de subdivision :

A chaque étape k , on fractionne M_k en un nombre fini de cônes plus petits.

(b) l'estimation de la borne :

Pour chaque cône $M_{k,i}$, on calcule une borne inférieure, notée $\mu(M_{k,i})$, pour $f(x)$ sur $D \cap M_{k,i}$ basée sur la propriété suivante :

Pour chaque suite décroissante infinie de cônes $\{M_{k_q}\}$ tendant vers un rayon

c'est-à-dire tel que $M_{k_{q+1}} \subset M_{k_q}$ ($q = 1, 2, \dots$),

$$\bigcap_{q=1}^{\infty} M_{k_q} = \left\{ \theta x^* : \theta \geq 0 \right\} \text{ pour un certain } x^*,$$

la différence $\gamma_{k_q} - \mu(M_{k_q})$ tend vers 0 pour $q \rightarrow \infty$

(où $\gamma_{k_q} = f(x_{k_q})$)

et M_{k_q} désigne le cône qui est divisé à l'étape k_q).

I.2.a. Méthode générale :- au départ :

On a un cône convexe M_0 déterminé par n points extrêmes s^1, s^2, \dots, s^n linéairement indépendants et l'origine $s^0 \equiv 0$.

On choisit une solution initiale admissible x_0 comme l'un des des points s^0, s^1, \dots, s^n qui atteint la plus petite valeur de la fonction concave : $f(s^i)$ ($i = 0, 1, \dots, n$).

Notons cette valeur $f(x^0) = \gamma_0$ et l'ensemble des cônes

$$\mathcal{C}_0 = \{M_0\}.$$

A partir de ces données, on peut estimer une borne inférieure

$\mu(M_0)$ pour f sur $D \cap M_0$.

- pour $k = 1, 2, \dots$:

On a une solution admissible x^k à l'itération k , qui est la meilleure solution courante avec $\gamma_k = f(x^k)$.

L'ensemble des cônes \mathcal{C}_k est constitué de membres de \mathcal{C}_{k-1} et de sous-cônes d'un membre de \mathcal{C}_{k-1} .

A chaque cône $M \in \mathcal{C}_k$, est associé un nombre $\mu(M)$ qui est l'estimation d'une borne inférieure pour la fonction f sur $D \cap M$.

On supprime tout cône $M \in \mathcal{C}_k$ tel que $\mu(M) \geq \gamma_k$.

Notons \mathcal{C}'_k l'ensemble des cônes restants.

Si $\mathcal{C}'_k = \emptyset$, alors STOP : x^k est une solution optimale;

sinon on cherche $M \equiv M_k \in \mathcal{C}'_k$ tel que $\mu(M)$ soit le plus petit.

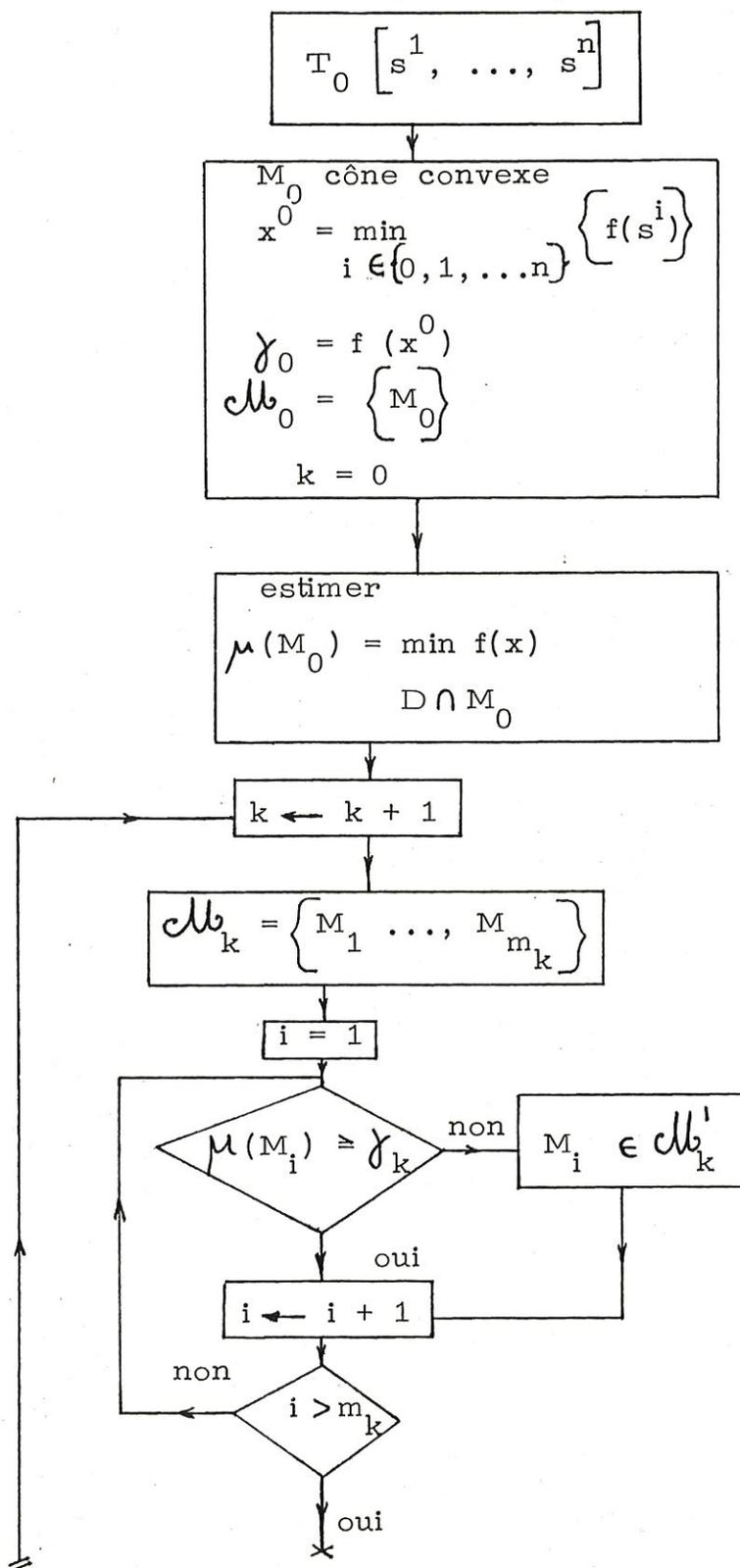
On divise ce cône M_k en un nombre fini de cônes plus petits $M_{k,1}, \dots, M_{k,r_k}$ et on estime sur chaque sous-cône $M_{k,i}$, une borne inférieure $\mu(M_{k,i})$ pour la fonction f sur $D \cap M_{k,i}$.

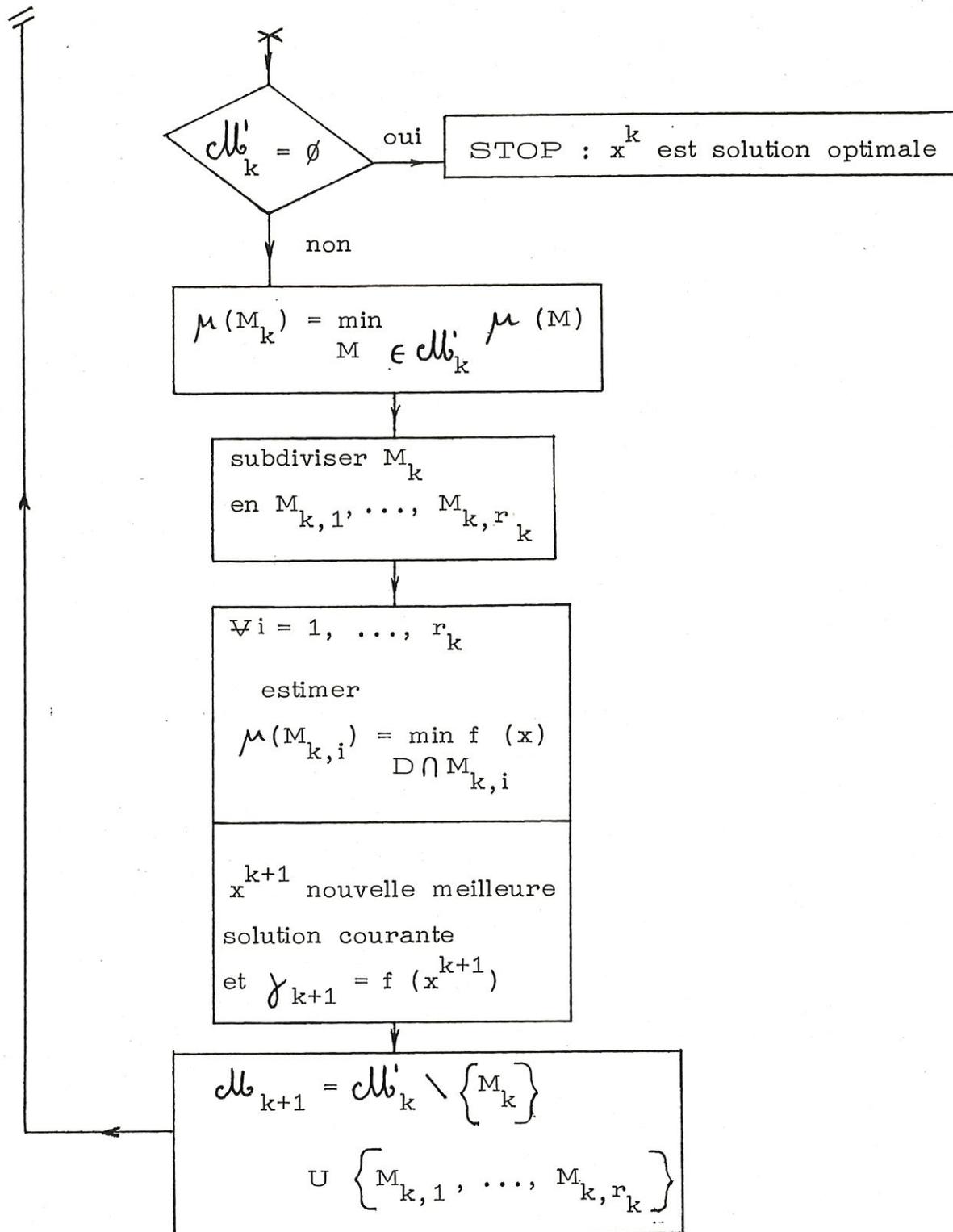
Ces opérations génèrent de nouvelles solutions admissibles :

x^{k+1} est la nouvelle meilleure solution admissible courante parmi x^k et toutes les solutions admissibles nouvellement générées. De ce fait, $\gamma_{k+1} = f(x^{k+1})$ est la nouvelle meilleure valeur courante.

A partir du nouvel ensemble \mathcal{C}_{k+1} (en substituant $M_{k,1}, \dots, M_{k,r_k}$ pour M_k dans \mathcal{C}'_k), on va à l'étape $k+1$.

I.2.b. Organigramme :





I.2.c. Estimation consistante d'une borne :

Pour un cône M issu de l'origine 0 et engendré par n rayons passant par n points linéairement indépendants v^1, \dots, v^n du simplexe

$$T_0 = [s^1, \dots, s^n],$$

on estime une borne inférieure $\mu(M)$ pour $f(x)$ sur $D \cap M$.

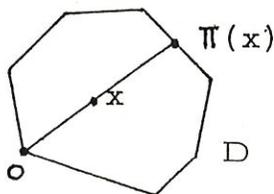
description :

1) la fonction Π :

On définit une transformation

$$\begin{array}{ccc} \Pi : T_0 & \longrightarrow & \partial D \\ x & \rightsquigarrow & \Pi(x) = t x \end{array}$$

$$\text{où } t = \max \left\{ \theta : A \cdot (\theta \cdot x) \leq b, \theta \geq 0 \right\}$$



La fonction Π envoie tout point $x \in T_0$ sur la frontière de D suivant le rayon issu de 0 et passant par x . De plus, ce point $\Pi(x)$ est unique (et il est admissible).

2) la borne supérieure β :

Soit γ un nombre donné (la meilleure valeur courante : $\gamma = f(x)$).

On définit

$$\beta = \beta(M, \gamma) = \min \left\{ \gamma, f(0), f(\Pi(v^1)), \dots, f(\Pi(v^n)) \right\}$$

c'est-à-dire la plus petite valeur parmi la meilleure valeur courante et les valeurs de la fonction en n points-frontières.

3) les points y^i :

Pour chaque $i = 1, \dots, n$ on construit le point $y^i = y^i(M, \gamma)$

tel que

$$y^i = \Theta_i v^i$$

avec

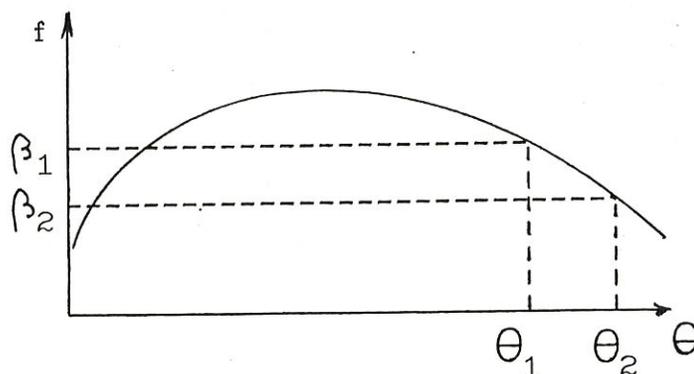
$$\Theta_i = \max \left\{ \Theta : \Theta \geq 1, f(\Theta.v^i) \geq \beta \right\}$$

Par l'hypothèse (i), on sait que Θ_i est fini.

Par construction, le segment $[0, y^i]$ contient $\pi(v^i)$ car

$$f(\Theta_i.v^i) = f(y^i) \geq \beta = \min \left\{ \gamma, f(v^j) \text{ où } j = 0, 1, \dots, n \right\}.$$

On recherche le point le plus éloigné de l'origine, situé sur le rayon $[0, v^i]$ tel que $f(y^i) \geq \beta$.



Ces points y^1, \dots, y^n déterminent un hyperplan, noté H , qui sépare l'espace en 2.

4) le point z :

On recherche un point $z = z(M, \gamma) \in D \cap M$ qui se trouve à une distance maximale de H .

où la distance d'un point à H est prise négative si ce point se trouve dans le demi-espace contenant 0,

c'est-à-dire si l'hyperplan ne coupe pas $D \cap M$, on prend le point le plus proche de H ;

la distance d'un point à H est prise positive si ce point se trouve dans l'autre demi-espace,

c'est-à-dire si l'hyperplan coupe $D \cap M$, on prend le point le plus éloigné de H dans ce demi-espace.

Notons $z^i = z^i(M, \gamma)$ le point où le rayon issu de O et passant par v^i , coupe l'hyperplan H' // à H et qui passe par z .

Pour cela, on résout le sous-problème suivant :

$$(SP) \begin{cases} \alpha^* = \text{Max } \alpha = \sum_{i=1}^n \lambda_i / \theta_i \\ \text{s.c. } \begin{cases} A \cdot B \cdot \lambda \leq b \\ \lambda \geq 0 \end{cases} \end{cases}$$

où $\left\{ \begin{array}{l} A \text{ est la matrice des contraintes (m \times n)} \\ B = \begin{pmatrix} | & & | \\ v^1 & \dots & v^n \\ | & & | \end{pmatrix} \in \mathbb{R}^{n \times n} \\ \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} \in \mathbb{R}^n \\ \theta_i \text{ est le nombre qui définit } y^i \end{array} \right.$

$$\text{et } \begin{cases} z = B \cdot \lambda^* \\ z^i = \alpha^* \cdot y^i \end{cases}$$

5) procédure d'estimation :

au départ :

On a $M = M_0$,

$$\gamma_0 = f(x^0) = \min_{i \in \{0, 1, \dots, n\}} \{f(s^i)\}$$

et on estime la borne inférieure $\mu(M_0)$ sur le cône initial par

$$\mu(M_0) = \nu(M_0, \gamma_0) \triangleq \min_{i \in \{1, \dots, n\}} \{f(z^i)\}$$

à l'étape k :

Soit $M_{k,i}$ un sous-cône résultant de la subdivision de M_k .

Notons γ_k la meilleure valeur courante de $f(x) = f(x^k)$

et $\mu(M_k)$ la borne inférieure de $f(x)$ sur $D \cap M_k$

(déjà calculée à une étape précédente)

En vertu du lemme 3 qu'on verra plus loin, on pose :

$$\mu(M_{k,i}) = \begin{cases} \beta_{k,i} = \beta(M_{k,i}, \gamma_k) & \text{si } \alpha_{k,i}^* \leq 1 \\ \max \{ \mu(M_k), \nu(M_{k,i}, \gamma_k) \} & \text{sinon} \end{cases}$$

où $\alpha_{k,i}^*$ est la valeur optimale de la fonction-objectif du sous-problème (SP), associé à $M_{k,i}$, $\gamma_k, \beta_{k,i}$.

I.2.d. Processus de subdivision (ou procédure de bisection)

A chaque étape, on fractionne M_k en un nombre fini de sous-cônes plus petits.

description :

Soit M_k le cône subdivisé à l'étape k .

On note $v^{k,i}$ le point où le i^e côté de M_k rencontre l'hyperplan passant par s^1, \dots, s^n .

On choisit un des plus grands côtés du $(n-1)$ - simplexe T_k formé par $v^{k,1}, \dots, v^{k,n}$:

soit v^{k,i_1}, v^{k,i_2} tel que

$$\|v^{k,i_1} - v^{k,i_2}\| = \max_{i,j \in \{1, \dots, n\}} \{ \|v^{k,i} - v^{k,j}\|, i \neq j \}$$

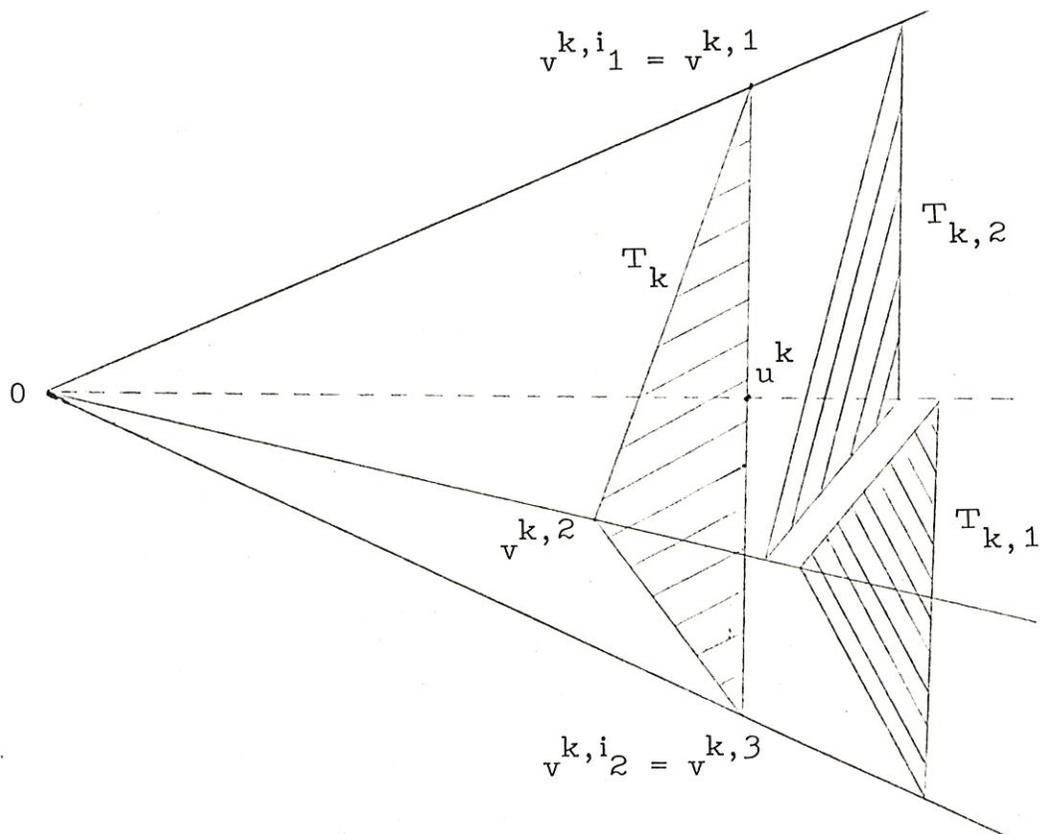
où $\forall x \in \mathbb{R}^n$ on prend $\|x\| = \max_{i \in \{1, \dots, n\}} |x_i|$

Le milieu du segment $[v^{k,i_1}, v^{k,i_2}]$ est noté u^k tel que

$$u^k = \frac{v^{k,i_1} + v^{k,i_2}}{2}$$

De ce fait, les cônes $M_{k,1}$ et $M_{k,2}$ sont obtenus par bisection de M_k . De même, $T_{k,1}$ et $T_{k,2}$ sont obtenus par bisection du simplexe T_k où dans l'un, v^{k,i_1} est remplacé par u^k et dans l'autre v^{k,i_2} par u^k .

Dans \mathbb{R}^3 :



I.3. Interprétation de la méthode :

La méthode repose essentiellement sur l'estimation d'une borne inférieure pour la partie admissible d'un cône; la façon de subdiviser les cônes étant choisie afin d'assurer la convergence de la méthode.

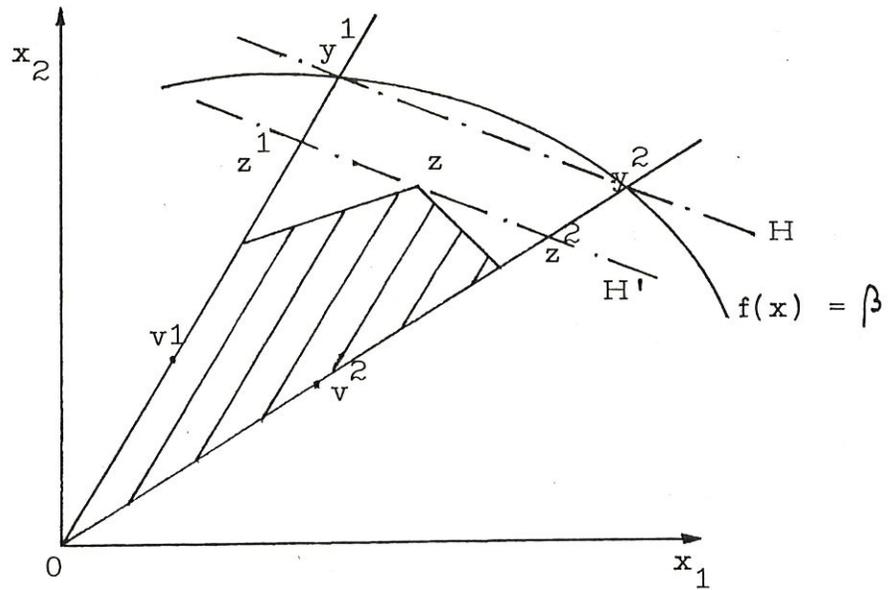
A partir de n points v^i linéairement indépendants qui déterminent notre cône, on évalue une borne supérieure β . Les intersections de la surface $f(x) = \beta$ avec les n rayons issus de 0 et passant par v^i , nous donnent n points y^i (pour $i = 1, \dots, n$). Ceux-ci déterminent un hyperplan H . En résolvant le sous-problème (SP), on obtient un point admissible z se trouvant à une distance maximale de H , qui est l'intersection de l'hyperplan $H'(z^1, \dots, z^n)$ // à H , avec le cône admissible considéré.

Si $\alpha \leq 1$, alors la borne inférieure égale la borne supérieure et le cône est rejeté, c'est-à-dire le point z qui pour la fonction f , a la plus petite valeur dans ce cône admissible, a cependant une valeur trop grande par rapport à la meilleure valeur courante. Si $\alpha > 1$, alors la borne inférieure est strictement plus petite que β , c'est-à-dire il est encore possible de trouver une meilleure solution admissible dans ce cône ou ailleurs, ou du moins approcher sa valeur. (voir figures page suivante)

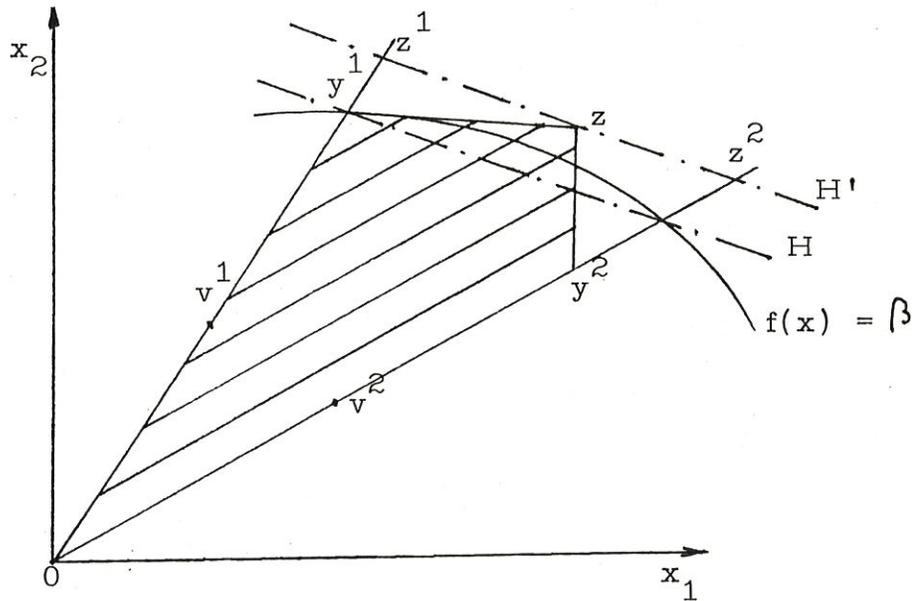
En ce qui concerne le choix du cône à subdiviser, on prend celui dont la borne inférieure est la plus petite, car c'est celui qui en principe, peut contenir une solution optimale. Ceci est fait afin de pouvoir rejeter plus rapidement un cône, puisque le but est de faire croître la borne inférieure et décroître la borne supérieure de telle façon qu'elles coïncident.

Dans \mathbb{R}^2 :

$\alpha \leq 1$
 ■■■■■■



$\alpha > 1$
 ■■■■■■



CHAPITRE II.

Démonstration théorique de la convergence.
 =====

II.1. Convergence :

On cherche par le processus de subdivision, une suite décroissante infinie de cônes $\{M_{kq}\}$ tendant vers un rayon issu de l'origine et passant par un certain x^* , de telle manière que la différence

$$\gamma_{kq} - \mu(M_{kq}) \text{ tende vers } 0 \text{ pour } q \longrightarrow \infty$$

où $\gamma_{kq} = f(x_{kq})$ est la valeur courante

et $\mu(M_{kq})$ est la borne estimée du cône divisé à l'étape kq .

Notre méthode se justifie par le théorème suivant :

théorème 1 :

Si le processus de subdivision est exhaustif et si l'estimation de la borne est compatible

Alors : on a - soit que la procédure se termine à une certaine étape k avec une solution optimale x^k ;
 - soit qu'elle génère une suite infinie $\{x^k\}$ où chaque point d'adhérence de cette suite est une solution optimale.

Chaque point x^k est une solution optimale approximée

avec une erreur n'excédant pas $\gamma_k - \mu_k$
 où $\mu_k = \min \left\{ \mu(M) : M \in \mathcal{M}_k \right\} = \mu(M_k)$

démonstration :

.....

Notons $\mu^* = \min_{x \in D} f(x)$.

On voit que $\mu_k \leq \mu^*$.

En effet, on peut montrer par induction qu'à chaque étape k , chaque point $x \in D$ qui est $\subset M_0$, est contenu soit dans un cône $M \in \mathcal{C}_k$, soit dans un cône M' qui a été supprimé en une étape précédente $h < k$ parce que $\mu(M') \geq \gamma_h$:

- $M \in \mathcal{C}_k$:

$$\mu(M_k) = \mu_k \leq \mu(M) \leq f(x) \quad \forall M \in \mathcal{C}_k ;$$

- M' supprimé à l'étape $h < k$:

$$\text{on sait que } \mu(M') \geq \gamma_h \text{ et } \gamma_k \geq \mu_k = \mu(M_k),$$

$$\text{or } \gamma_{k-2} \geq \gamma_{k-1} \geq \gamma_k \geq \gamma_{k+1} \dots \Rightarrow \gamma_h \geq \gamma_k, \text{ car } h < k$$

$$\Rightarrow f(x) \geq \gamma_h \geq \gamma_k \geq \mu_k.$$

$$\text{Donc, on a } f(x) \geq \mu_k \quad \forall x \in D$$

$$\Rightarrow \mu^* \geq \mu_k.$$

1) Si la procédure s'arrête à l'étape k ($\mathcal{C}'_k = \emptyset$)

$$\text{c'est-à-dire } \mu(M) \geq \gamma_k \quad \forall M \in \mathcal{C}_k,$$

$$\text{alors on a } \mu_k \geq \gamma_k \Rightarrow \mu^* \geq \gamma_k.$$

$$\text{Or on a } \mu^* \leq \gamma_k,$$

ce qui entraîne que x^k est une solution optimale : $\mu^* = \gamma_k = f(x^k)$.

2) Si la procédure ne s'arrête jamais :

Comme le processus de subdivision est exhaustif, il y a une suite infinie décroissante de cônes $\{M_{kq}\}$ tendant vers un rayon.

A partir de l'uniformité de l'estimation de borne, on peut alors avoir $\gamma_{kq} - \mu_{kq} \longrightarrow 0$ pour $q \longrightarrow \infty$ et par conséquent

$$\gamma_{kq} - \mu^* \longrightarrow 0.$$

Mais comme la suite $\{\gamma_k\}$ est évidemment non croissante, on a $\gamma_k - \mu^* \searrow 0$ ou $f(x^k) \searrow \mu^*$.

Par la compacité de D , la suite $\{x^k\}$ a au moins un point adhérent et chacun de ces points adhérents x^* satisfait $f(x^*) = \mu^*$, car $\mu_k \leq \mu^* \leq \gamma_k$ où $\mu_k = \min\{\mu(M) : M \in \mathcal{M}_k\} = \mu(M_k)$ et $\gamma^* = f(x^*)$.

II.2. Processus exhaustif de subdivision :

Il y a différentes façons de construire ce processus. On choisira celui qui subdivise le cône M_k à l'étape k , en 2 sous-cônes $M_{k,1}$ et $M_{k,2}$ par la procédure de bisection suivante : si on note $v^{k,i}$ le point où le i^e côté de M_k rencontre l'hyperplan formé de s^1, \dots, s^n ,

on choisit un des plus grands côtés du $(n-1)$ -simplexe T_k formé par ces $v^{k,1}, \dots, v^{k,n}$, qu'on note $[v^{k,i_1}, v^{k,i_2}]$, et on prend le milieu de ce côté :

$$u^k = \frac{(v^{k,i_1} + v^{k,i_2})}{2}$$

$$\begin{aligned} \text{où } \left\| v^{k,i_1} - v^{k,i_2} \right\| &= \max \left\{ \left\| v^{k,i} - v^{k,j} \right\| : i \neq j \right\} \\ &= \max \left\{ \max_1 \left| v^{k,i} - v^{k,j} \right| : i \neq j \right\}. \end{aligned}$$

théorème 2 :

Le processus de subdivision par bisection successive est exhaustif.

Pour démontrer ce théorème, on a besoin au préalable de 2 lemmes :

lemme 1 :

.....

Soit M_k le cône subdivisé à l'étape k ,
 considérons $[v^{k,i_1}, v^{k,i_2}]$ un des plus grands côtés du
 $(n-1)$ - simplexe T_k ,
 et u^k le milieu de ce segment,

Alors pour $h = 1, 2$:

l'ensemble obtenu à partir de $\{v^{k,1}, \dots, v^{k,n}\}$
 en remplaçant v^{k,i_h} par u^k , est linéairement indé-
 pendant et génère un cône $M_{k,h}$ de sommet 0 et
 formé de n côtés exactement.

De plus $M_k = M_{k,1} \cup M_{k,2}$.

démonstration :

Par construction des $v^{k,i}$, ils sont linéairement indépendants, car s^1, \dots, s^n le sont.

Si on remplace v^{k,i_1} par u^k dans $v^{k,1}, \dots, v^{k,n}$, on a encore un ensemble de points linéairement indépendants par construction de u^k , qui génère le cône $M_{k,1}$.
 De même, on obtient $M_{k,2}$ en remplaçant v^{k,i_2} par u^k .

lemme 2 :

A partir du simplexe initial $T_0 = [s^1, \dots, s^n]$,
 considérons une suite infinie de $(n-1)$ sous-simplexes :
 $S_1 \supset S_2 \supset \dots \supset S_q \supset \dots$ tel que chaque S_q est obtenu à
 partir de S_{q-1} par bisection.

Alors : $\text{diam } S_q \xrightarrow{q \rightarrow \infty} 0$,
 c'est-à-dire $\bigcap_{q=1}^{\infty} S_q$ est un singleton.

démonstration :

Notons $a_{q,1}, \dots, a_{q,h}$ les longueurs des côtés de S_q par
 ordre décroissant : $a_{q,1} \geq a_{q,2} \geq \dots \geq a_{q,h}$
 où $h = n(n-1) / 2$ car chaque côté est déterminé par 2 des
 n points linéairement indépendants.

On a évidemment $a_{q,1} \geq a_{q+1,1}$, car $S_{q-1} \subset S_q$.

Donc, $\exists \alpha$ tel que la plus grande longueur $a_{q,1} \searrow \alpha$ quand $q \rightarrow \infty$.

Montrons que $\alpha = 0$?

Par l'absurde, supposons que $\alpha > 0$.

Notons l'indice de la plus petite longueur de S_q qui tend vers α :

$$p = \max \left\{ i : a_{q,i} \xrightarrow{q \rightarrow \infty} \alpha \right\}$$

or $p \geq 1$ car $1 \leq i \leq h$

donc pour $q \rightarrow \infty$: $\lim a_{q,i} = \alpha$ pour $i = 1, \dots, p$. (1)

De plus, on sait que $a_{q,p} \geq a_{q,p+1}$

$\Rightarrow \lim a_{q,p} \geq \lim a_{q,p+1}$

$$\Rightarrow \alpha \geq \lim a_{q, p+1}$$

$$\Rightarrow \alpha > \underline{\lim} a_{q, p+1} .$$

Alors il est toujours possible de trouver un nombre ε et un indice q tel que par (1) :

$$\forall q' \geq q : \alpha - \varepsilon \leq a_{q', i} \leq \alpha + \varepsilon \quad (i = 1, \dots, p) \quad (2)$$

$$\text{et } 0 < \underbrace{a_{q, p+1}}_{\leq a_{q, p}} < \alpha - 3\varepsilon < \alpha . \quad (3)$$

Cela signifie pour (2) que le simplexe S_{q+1} a au moins p côtés de longueur $\geq \alpha - \varepsilon$ car $i = 1, \dots, p$.

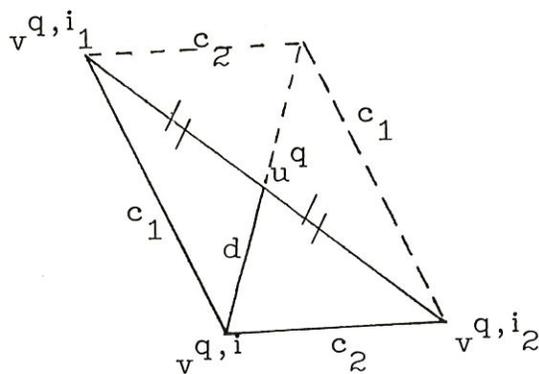
Donc il suffit de montrer que S_{q+1} a au plus $p-1$ côtés de longueur $\geq \alpha - \varepsilon$ et on arrive à une contradiction $\Rightarrow \alpha = 0$.

Démontrons cet affirmation :

Soient $v^{q, i}$ ($i = 1, \dots, n$) les sommets de S_q et considérons $[v^{q, i_1}, v^{q, i_2}]$ le plus long côté de S_q par lequel on a effectué une bissection pour obtenir S_{q+1} . Le point-milieu de ce segment est noté u^q .

Alors l'ensemble des sommets de S_{q+1} est constitué des $v^{q, i}$ ($i \neq i_1$) et de u^q (remplaçant v^{q, i_1}).

Supposons que pour un indice $i \neq i_2$, le côté $[u^q, v^{q, i}]$ a une longueur $d \geq \alpha - \varepsilon$.



$$c_1 \equiv \left\| v^{q,i_1} - v^{q,i_2} \right\| \leq \left\| v^{q,i_1} - v^{q,i_2} \right\| \equiv 1$$

$$c_2 \equiv \left\| v^{q,i_1} - v^{q,i_2} \right\| \leq 1$$

$$d \equiv \left\| u^q - v^{q,i} \right\| \geq \alpha - \varepsilon$$

on construit le parallélogramme de côtés c_1 et c_2
 $\Rightarrow 2d \leq c_1 + c_2$.

Donc, on doit avoir soit $c_1 \geq \alpha - \varepsilon$, soit $c_2 \geq \alpha - \varepsilon$,
 car $d \geq \alpha - \varepsilon$.

Prenons $c_1 \geq \alpha - \varepsilon$:

or par (2) : $c_1 \leq \alpha + \varepsilon$ car $c_1 = a_{q+1,i}$

mais $c_1 \leq a_{q,1}$.

De plus $2(\alpha - \varepsilon) \leq 2d \leq c_1 + c_2 \leq \alpha + \varepsilon + c_2$

$$\Rightarrow \alpha - 3\varepsilon \leq c_2$$

ce qui permet de dire que $\alpha - \varepsilon \leq c_2$

car par (3) : $a_{q,p+1} < \alpha - 3\varepsilon \leq c_2 \Rightarrow c_2 > a_{q,p+1}$

et donc $c_2 = a_{q,i}$ pour un $i \leq p$ (les $a_{q,i}$ sont rangés
 par ordre décroissant)

Par conséquent, si $d \geq \alpha - \varepsilon$, alors $\begin{cases} c_1 \geq \alpha - \varepsilon \\ c_2 \geq \alpha - \varepsilon \end{cases}$

Prenons pour convention de noter :

- "blanc" chaque côté de S_{q+1} qui $\notin S_q$ et de longueur $\geq \alpha - \varepsilon$,
- "noir" chaque côté de S_q qui $\notin S_{q+1}$ et de longueur $\geq \alpha - \varepsilon$.

On peut donc dire si pour un $i \neq i_2$:

le côté $[u^q, v^{q,i}]$ est blanc, alors le côté $[v^{q,i}, v^{q,i_1}]$ est noir, c'est-à-dire si $[u^q, v^{q,i}] \in S_{q+1}$, mais $\notin S_{q+1}$ (de longueur $d \geq \alpha - \varepsilon$)

alors $\underbrace{[v^{q,i}, v^{q,i_1}]} \in S_q$, mais $\notin S_{q+1}$ (car u^q remplace v^{q,i_1})
de longueur $c_1 \geq \alpha - \varepsilon$

De plus, $[v^{q,i_1}, v^{q,i_2}]$ est noir (étant le plus long côté de S_q)
car $\in S_q$, mais $\notin S_{q+1}$ et de longueur $l \geq c_1 \geq \alpha - \varepsilon$

puisque $[u^q, v^{q,i_2}]$ n'est pas blanc,

$$\in S_{q+1}, \notin S_q$$

$$\text{car sa longueur } 1/2 \left\| v^{q,i_1} - v^{q,i_2} \right\| = a_{q,1} \leq \alpha + \varepsilon$$

$$\leq 1/2 (\alpha + \varepsilon)$$

$$< \alpha - \varepsilon \text{ car } 0 < \alpha - 3\varepsilon \Rightarrow \varepsilon < \alpha/3$$

$$\Rightarrow 1/2(\alpha + \varepsilon) < \frac{2}{3}\alpha = \alpha - \alpha/3 < \alpha - \varepsilon.$$

Par conséquent, on peut dire que le nombre de côtés blancs est au plus égal au nombre de côtés noirs moins un, ce qui signifie que

S_{q+1} a au plus $(p-1)$ côtés de longueur $\geq \alpha - \varepsilon$.

En effet, $[v^{q,i}, v^{q,i_1}]$ et $[v^{q,i_1}, v^{q,i_2}]$ sont noirs;

$[u^q, v^{q,i}]$ est blanc, mais $[u^q, v^{q,i_2}]$ n'est pas blanc

(u^q remplace v^{q,i_1})

et à cause de (2), S_q a au moins p côtés noirs.

Il ne reste plus qu'à démontrer le théorème.

démonstration :

Notons T_{k_q} l'intersection de M_{k_q} avec $T_0 = [s^1, \dots, s^n]$.

Si le processus de subdivision par bisection successive est infini, alors il génère une suite infinie de cônes $\{M_{k_q}\}$ tel que

T_{k_q} soit obtenu à partir de $T_{k_{q-1}}$ par une bisection.

Par le lemme précédent, $\text{diam } T_{k_q} \xrightarrow{q \rightarrow \infty} 0$

et $\bigcap_{q=1}^{\infty} T_{k_q} = \{v^*\}$.

La suite $\{M_{k_q}\}$ est évidemment décroissante et tend vers un rayon issu de 0 et passant par v^* .

II.3. Estimation consistante d'une borne :

Après avoir décrit la procédure d'estimation, il reste à démontrer le théorème suivant :

théorème 3 :

L'estimation de borne est compatible.

Mais pour cela, on a besoin de certains lemmes.

lemme 3 :

.....

La transformation $\Pi : T_0 \rightarrow \partial D$ est continue
 $x \rightsquigarrow \Pi(x) = tx$
 où $t = \max \{ \theta : A \cdot (\theta x) \leq b, \theta \geq 0 \}$.

démonstration :

Par l'hypothèse (iii), on sait que l'origine $0 \in D$

$$\Rightarrow A \cdot 0 = 0 \leq b \Rightarrow b = (b_1, \dots, b_m)^t \geq 0.$$

Considérons l'ensemble des b_i non nuls :

$$I = \{ i : b_i > 0 \}$$

et la fonction

$$g(x) = \max_{i \in I} g_i(x)$$

$$\text{où } g_i(x) = \frac{\sum_j a_{ij} x_j}{b_i} \leq 1 \quad \text{car } Ax \leq b$$

et a_{ij} est le j^e élément de la i^e ligne de A .

Par construction, cette fonction $g(x)$ est continue et positivement

homogène : $g(tx) = tg(x) \quad \forall t \geq 0$.

De plus, $g(x) > 0$ pour $x \in T_0$.

En effet, supposons par l'absurde que $g_i(tx) \leq 0 \quad (\forall i \in I) \quad \forall t \geq 0$

$$\Rightarrow t \sum_j a_{ij} x_j \leq 0 < b \quad \forall i \in I, \forall t > 0$$

$$\Rightarrow t.x \in D \quad \forall t \geq 0$$

$$\Rightarrow D \text{ contient tout point } t.x \quad (t \geq 0)$$

ce qui contredit le fait que l'ensemble D est borné (hypothèse (ii)).

Comme $\pi(x) \in \partial D = \{x \in \mathbb{R}^n \mid A.x = b\}$

$$\left. \begin{array}{l} \Rightarrow g(\pi(x)) = 1 \\ \text{or } g(\pi(x)) = g(tx) = t.g(x) \end{array} \right\} \Rightarrow g(x) = 1/t$$

$$\Rightarrow \pi(x) = t.x = \frac{x}{g(x)}$$

Par conséquent, on peut affirmer que π est continue car x et $g(x)$ continue et $g(x) > 0$.

lemme 4 :
.....

Si λ^* est une solution optimale du sous-problème:

$$\text{Max } \left(\sum_{i=1}^n \lambda_i / \theta_i \right)$$

$$\text{s.c. } \begin{cases} A.B.\lambda \leq b \\ \lambda \geq 0 \end{cases}$$

Alors 1) $z = B.\lambda^*$, $z^i = \alpha^*.y^i$ où $\alpha^* = \sum_{i=1}^n \lambda_i^* / \theta_i$

2) si $\alpha^* \leq 1$, alors $f(x) \geq \beta \quad \forall x \in D \cap M$,

sinon $\forall (M, \gamma) = \min \{ f(z^i) : i = 1, \dots, n \} \leq f(x)$

$$\forall x \in D \cap M$$

démonstration :

- 1) Chaque cône M est déterminé par n rayons issu de l'origine et passant par n points v^i linéairement indépendants. Donc chaque point $x \in M$ s'écrit sous la forme suivante :

$$x = \sum_{i=1}^n \lambda_i \cdot v^i = B \cdot \lambda \quad \text{avec } \lambda \geq 0$$

$$\text{où } \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} \quad \text{et } B = \begin{pmatrix} | & & | \\ v^1 & \dots & v^n \\ | & & | \end{pmatrix}$$

car x est combinaison linéaire positive des v^i puisque x se trouve à l'intérieur du cône.

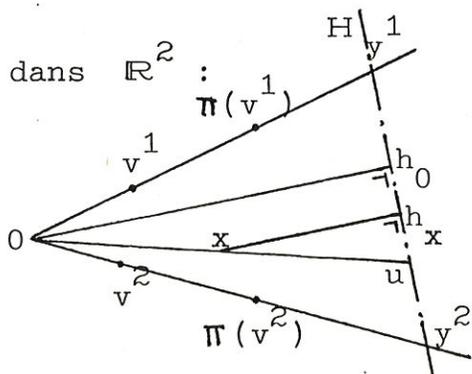
Dans la procédure d'estimation, on détermine les $y^i = \theta_i \cdot v^i$ avec $\theta_i = \max\{\theta : \theta \geq 1, f(\theta \cdot v^i) \geq \beta\}$ pour $i = 1, \dots, n$, qui forment l'hyperplan H .

Notons u le point où le rayon issu de 0 et passant par x rencontre H :

u est combinaison convexe des y^i :

$$u = \sum_{i=1}^n \eta_i y^i \quad \text{avec } \sum_i \eta_i = 1$$

$$\text{or } v^i = y^i / \theta_i \Rightarrow x = \sum_{i=1}^n \frac{\lambda_i}{\theta_i} \cdot y^i$$



D'autre part $x = \alpha \cdot u$ où α est un scalaire à déterminer.

Par construction de u et de x , on peut écrire :

$$\sum_i \frac{\lambda_i}{\theta_i} y^i = \sum_i \alpha \cdot \eta_i \cdot y^i,$$

mais les y^i sont linéairement indépendants :

$$\sum_i \left(\frac{\lambda_i}{\theta_i} - \alpha \cdot \eta_i \right) \cdot y^i = 0 \Rightarrow \frac{\lambda_i}{\theta_i} - \alpha \cdot \eta_i = 0 \quad \forall i$$

$$\Rightarrow \frac{\lambda_i}{\theta_i} = \alpha \cdot \eta_i \quad \forall i$$

Par conséquent,

$$\alpha = \sum_{i=1}^n \frac{\lambda_i}{\theta_i}.$$

De plus, $x - u = (\alpha - 1) \cdot u$

ce qui montre que pour $x \in M$:

$$d(x, H) = d(0, H) \cdot (1 - \alpha)$$

car la distance de 0 à H est prise négative par convention.

En effet, le triangle $0h_0u$ est semblable au triangle xh_xu

$$\Rightarrow \frac{d(x, H)}{d(0, H)} = \frac{d(x, h_x)}{d(0, h_0)} = \frac{d(u, x)}{d(u, 0)} = \frac{u-x}{u} = 1 - \alpha$$

Or on cherche z , le point $D \cap M$ qui se trouve à une distance maximale de H . Par conséquent, si λ^* est une solution optimale du sous-problème,

alors $z = B \cdot \lambda^*$ correspond au plus grand α : α^* c.à.d. à λ^*
 et $z^i = \alpha^* \cdot y^i$ où $\alpha^* = \sum_{i=1}^n \lambda_i^* / \theta_i$,

car z^i est le point où le rayon issu de 0 et passant par y^i (donc par v^i) coupe l'hyperplan $H' //$ à H et passant par z .

2) $\alpha^* \leq 1$:

Il est facile de voir que $D \cap M \subseteq T [0, y^1, \dots, y^n]$
 car $z^i = \alpha^* \cdot y^i$.

Mais on sait qu'une fonction f quasi-concave atteint son minimum en un des points extrêmes de l'ensemble admissible (ici du simplexe T) c.à.d. en un des points $0, y^1, \dots, y^n$.
 Or par construction des y^i : $f(y^i) \geq \beta$, on doit avoir
 $f(x) \geq \beta$ sur $D \cap M$
 où $\beta = \min \{ \gamma, f(0), f(\pi(v^i)) : i = 1, \dots, n \}$
 c.à.d. $f(0) \geq \beta$.

$\alpha^* > 1$:

Montrons que $f(x) \geq \gamma(M, \gamma) \quad \forall x \in D \cap M$.

Similairement, on voit que $D \cap M \subseteq T [0, z^1, \dots, z^n]$ et que f atteint son minimum en un des points suivants $0, z^1, \dots, z^n$.

Par définition de β , on a $f(0) \geq \beta$

et par construction des $z^i = \alpha^* \cdot y^i$ avec $\alpha^* > 1$

et où $f(y^i) \geq \beta$, on a $f(z^i) < \beta$ pour $i = 1, \dots, n$,

on trouve que :

$$f(x) \geq \gamma(M, \gamma) = \min \{ f(z^i) : i = 1, \dots, n \} \quad \forall x \in D \cap M.$$

lemme 5 :

.....
 Si $\alpha^* > 1$

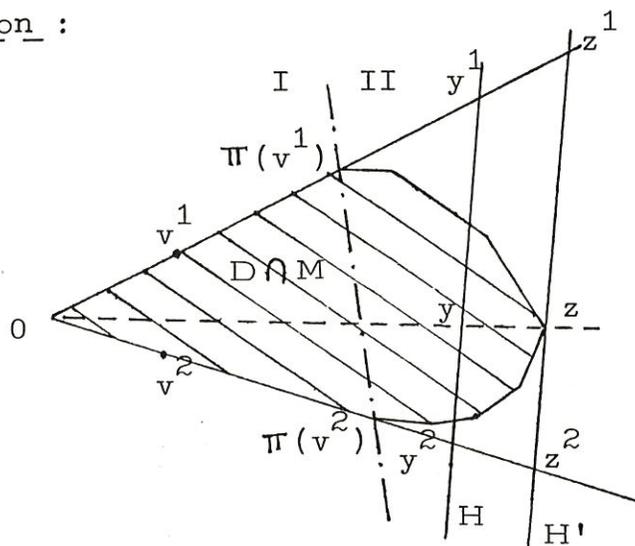
alors le segment-ligne $[0, z]$ rencontre l'hyperplan $H(y^1, \dots, y^n)$
 en un point y

tel que $y \in$ à l'enveloppe convexe de l'ensemble

$$\{z, \pi(v^1), \dots, \pi(v^n)\}$$

démonstration :

dans \mathbb{R}^2 :



L'hyperplan passant par $\pi(v^1), \dots, \pi(v^n)$ divise le cône M en 2 parties :

- l'une contenant l'origine 0 , est un simplexe I ,
- l'autre est un cône tronqué II .

En vertu de la construction des points $y^i = \theta_i v^i$ ($\theta_i \geq 1$), il est clair que tous ces points se trouvent dans II ,

car $\forall i : f(y^i) \geq \beta = \min\{\gamma, f(0), f(\pi(v^1)), \dots, f(\pi(v^n))\}$.

Comme $z \in M$, le point y où le segment-ligne $[0, z]$ rencontre l'hyperplan H , doit se trouver entre z et le point où le segment rencontre le simplexe $[\pi(v^1), \dots, \pi(v^n)]$ car $\alpha^* > 1$ et $z^i = \alpha^* \cdot y^i$.

Il reste à montrer le théorème de compatibilité de l'estimation.

démonstration :

Considérons une sous-suite décroissante de cônes $\{M_{k_q}\}$ tendant

vers un rayon issu de l'origine 0 et passant par un certain $x^* \in \partial D$.

Alors l'hyperplan passant par T_0 détermine sur ces cônes des sections $T_k = [v^{kq,1}, \dots, v^{kq,n}]$ qui forment une suite décroissante de sous-simplices de T_0 . De plus, on a par le lemme 2 que l'intersection $\bigcap_{q=1}^{\infty} T_{k_q} = \{v^*\}$, où v^* satisfait la relation

$$\Pi(v^*) = x^* .$$

Soit un nombre positif ε donné.

Par l'hypothèse (i), il existe une constante C tel que $\|x\| \leq C$ pour tout x satisfaisant l'ensemble $S = \{x \in \mathbb{R}^n \mid f(x) \geq \alpha\}$ et en particulier pour $\alpha = \mu^* \equiv \min_{x \in D} f(x)$.

Par continuité de f et par la compacité de la boule fermée de centre 0 et de rayon C , on peut trouver un nombre positif δ tel que chaque fois qu'on prend un x de telle manière que $\|x\| \leq C$ et $\|x' - x\| \leq \delta$, on ait $|f(x') - f(x)| < \varepsilon$.

On peut évidemment supposer que M_{k_q} est obtenu à partir de M_{k_q-1} . Alors $\mu(M_{k_q})$ est calculé à l'étape k_q-1 .

Notons $y^{kq,i}$, z^{kq} , $z^{kq,i}$, α_{kq}^* les points y^i , z , z^i et le nombre α^* construits pour $\mathcal{J} = \mathcal{J}_{k_q-1}$, $M = M_{k_q}$ comme décrit préalablement.

Notons aussi y^{kq} et v^{kq} les points où le segment-ligne $[0, z^{kq}]$ intercepte les simplexes $[y^{kq,1}, \dots, y^{kq,n}]$ et T_{k_q} respectivement.

$$\text{Alors } \|v^{kq,i} - v^*\| \xrightarrow{q \rightarrow \infty} 0$$

$$\text{et } \|v^{kq} - v^*\| \xrightarrow{q \rightarrow \infty} 0$$

$$\text{car } \bigcap_{q=1}^{\infty} T_{k_q} = \{v^*\} .$$

Par le lemme 3, on sait que la transformation Π est continue. Ceci implique que $\forall q$ assez grand :

$$\left\| \Pi(v^{kq,i}) - x^* \right\| < \frac{\delta}{4C} \cdot \|x^*\| \quad \text{pour } i = 1, \dots, n$$

$$\text{où } x^* = \Pi(v^*) \quad \text{car } \|x^*\| \leq C$$

$$\text{et } \|z^{kq} - x^*\| < \frac{\delta}{4C} \cdot \|x^*\|$$

c'est-à-dire la boule de rayon $(\delta/4C) \cdot \|x^*\|$ de centre x^* contiendra tous les points $\Pi(v^{kq,1}), \dots, \Pi(v^{kq,n})$ et z^{kq} .

Remarquons que $\alpha_{kq}^* > 1$,

sinon on aurait que $\mu(M_{kq}) = \beta(M_{kq}, \gamma_{kq-1}) \geq \gamma_{kq}$

si $\alpha_{kq}^* \leq 1$ (par définition de β)

mais cela contredirait que M_{kq} est 1 cône qui doit être subdivisé

à l'étape k_q ($\mu(M_{kq}) < \gamma_{kq}$).

En vertu du lemme 5, on sait que y^{kq} appartient à l'enveloppe convexe de $\Pi(v^{kq,1}), \dots, \Pi(v^{kq,n})$ et z^{kq} et donc aussi à la boule fermée $B_F(x^*, \frac{\delta}{4C} \cdot \|x^*\|)$.

Puisque y^{kq} et z^{kq} se trouvent dans cette boule fermée, on aura :

$$\|y^{kq} - z^{kq}\| < \frac{\delta}{2C} \cdot \|x^*\| \quad (2x \text{ le rayon})$$

$$\|y^{kq} - x^*\| < \frac{\delta}{4C} \cdot \|x^*\|.$$

D'autre part, on peut supposer $\delta < 2C$:

$$\Rightarrow \|y^{kq} - x^*\| < \frac{\delta}{4C} \cdot \|x^*\| < \frac{\|x^*\|}{2} \quad)$$

$$\text{or } \|y^{kq} - x^*\| \geq \|x^*\| - \|y^{kq}\|$$

$$\Rightarrow \frac{\|x^*\|}{2} < \|y^{kq}\|$$

$$\text{tandis que } \|y^{kq,i}\| \leq C \quad \text{car } f(y^{kq,i}) \geq \gamma_{kq} \geq \mu^*.$$

On sait aussi que les hyperplans passant par $y^{kq,1}, \dots, y^{kq,n}$ et par $z^{kq,1}, \dots, z^{kq,n}$ respectivement sont parallèles.

Alors on peut écrire pour $i = 1, \dots, n$:

$$\frac{\|y^{kq,i} - z^{kq,i}\|}{\|y^{kq,i}\|} = \frac{\|y^{kq} - z^{kq}\|}{\|y^{kq}\|}$$

$$\Rightarrow \|y^{kq,i} - z^{kq,i}\| = \|y^{kq} - z^{kq}\| \cdot \frac{\|y^{kq,i}\|}{\|y^{kq}\|} \quad \text{pour } i = 1, \dots, n$$

$$< \left(\frac{\delta}{2C} \cdot \|x^*\|\right) \cdot \frac{C}{\|x^*\|/2} = \delta.$$

Donc on a trouvé un nombre positif δ , ce qui entraîne que :

$$|f(y^{kq,i}) - f(z^{kq,i})| < \varepsilon \quad \text{pour } i = 1, \dots, n.$$

Par le lemme 4, on prend si $\alpha_{kq}^* > 1$,

$$\mu(M_{kq}) = \max \left\{ \mu(M_k), \nu(M_{kq}, \gamma_{kq-1}) \right\}$$

$$\Rightarrow \nu = \min \{ f(z^{kq,i}) : i = 1, \dots, n \} \leq \mu(M_{kq})$$

$$\Rightarrow |\gamma_{kq} - \mu(M_{kq})| = |f(y^{kq,i}) - f(z^{kq,i})| < \varepsilon$$

$$\text{c'est-à-dire } \gamma_{kq} - \mu(M_{kq}) \xrightarrow{q \rightarrow \infty} 0.$$

Remarquons encore une dernière chose. Par le processus de subdivision, on aboutit à un 'cône' réduit à un rayon issu de l'origine 0 et passant par la solution optimale x^* , où x^* est un

point extrême de l'ensemble admissible D . Mais l'origine est aussi un point extrême du domaine admissible. Ceci nous conduit à énoncer le fait suivant :

$$\boxed{\text{Min } \{ f(0), f(x^*) \} = \text{Min } \{ f(x) : x \in D \}}$$

démonstration :

Comme la suite $\{y^{kq,i}\}$ ($i = 1, \dots, n$) est bornée, il existe une sous-suite de $\{k_q\}$ tel que $y^{kq,i} \longrightarrow y^{*i}$ ($i=1, \dots, n$).

Comme $v^{kq,i} \longrightarrow v^*$, tous ces points y^{*i} se trouvent sur le rayon passant par x^* .

D'un autre côté, $\pi(v^{kq,i})$ pour $i = 1, \dots, n$ et z^{kq} tendent vers x^* et donc $y^{kq} \longrightarrow x^*$. Or y^{kq} appartient à l'enveloppe convexe des $y^{kq,i}$ ($i = 1, \dots, n$), on aura de ce fait x^* qui appartiendra à l'enveloppe convexe de y^1, \dots, y^n et pour au moins un i , le segment $[0, x^*]$ contiendra y^{*i} .

Comme $f(0) \geq \mu^*$ et $f(y^{*i}) = \lim_{q \rightarrow \infty} f(y^{kq,i}) \leq \lim_{q \rightarrow \infty} \gamma_{kq-1} = \mu^*$,

on peut conclure à partir de la quasi-concavité de f que

$$\mu^* = \min \{ f(0), f(x^*) \}.$$

Donc, quand $q \rightarrow \infty$, le cône M_{kq} tend vers un rayon passant par x^* , pendant que la borne inférieure estimée $\mu(M_{kq})$ pour la fonction f sur $D \cap M_{kq}$, tend vers le minimum de $f(x)$ sur le segment $[0, x^*]$.

II.4. Cas particuliers :

La méthode a été décrite sous certaines hypothèses. Cependant, celles-ci ne sont pas toujours vérifiées.

(i) Supposons qu'il existe un $\alpha \in \mathbb{R}$ tel que l'ensemble

$$S = \{x \in \mathbb{R}^n \mid f(x) \geq \alpha\}$$

n'est pas borné.

Ceci impliquera essentiellement que dans le calcul de y^i , le nombre θ_i ne soit plus nécessairement fini. On peut contourner cette difficulté en modifiant les définitions de y^i et de $\mathcal{V}(M, \gamma)$.

On choisit un nombre positif C assez grand tel que la boule de rayon C autour de l'origine contient le domaine admissible D en son intérieur. De ce fait, on prendra

$$y^i = \theta_i v^i \quad \text{pour } i=1, \dots, n$$

$$\text{avec } \theta_i = \left\{ \max \theta : \theta \geq 1, f(\theta v^i) \geq \beta, \|\theta v^i\| \leq C \right\}$$

$$\text{et } \mathcal{V}(M, \gamma) = \min \left\{ f(0), f(z^1), \dots, f(z^n) \right\}.$$

Comme on le voit, l'hypothèse (i) n'est pas indispensable à notre algorithme.

(ii) La condition (ii) est essentielle et assez naturelle dans bon nombre de problèmes pratiques.

(iii) La dernière hypothèse peut être satisfaite assez facilement (voir méthode de BALAS [6] ou technique de perturbation de programmation linéaire de SIMONNARD [7]).

CHAPITRE III.

Implémentation de l'algorithme.
 =====

III.1. Algorithme pour une solution ϵ -optimale :

III.1.a. On se donne un nombre $\epsilon > 0$.

Une solution admissible x^* est ϵ -optimale si

$$f(x) \geq f(x^*) - \epsilon \quad \forall x \in D.$$

Le cône convexe initial, noté M_0 , de sommet $s^0 = 0$, est généré par N points linéairement indépendants v^1, \dots, v^n du simplexe initial $T_0 = [s^1, \dots, s^n]$.

Initialisation :

On pose : - x^0 la meilleure solution admissible initiale :

$$x^0 = \min \{ f(s^i) : i = 0, 1, \dots, n \} \quad ;$$

- γ_0 la meilleure valeur initiale :

$$\gamma_0 = f(x^0) \quad ;$$

- \mathcal{M}_0 l'ensemble initial de cônes convexes :

$$\mathcal{M}_0 = \{ M_0 \} .$$

On calcule une borne inférieure, notée $\mu(M_0)$:

$$\mu(M_0) = \nu(M_0, \gamma_0) = \min \{ f(z^i) : i = 1, \dots, n \}$$

où les z^i ont été préalablement définis et sont relatifs au cône convexe M_0 .

Etapes $k = 1, 2, \dots$:

On supprime tout cône $M \in \mathcal{C}_k$ si $\mu(M) \geq \gamma_k - \epsilon$
et on note \mathcal{C}'_k l'ensemble des cônes restant de \mathcal{C}_k .

Si \mathcal{C}'_k est vide, alors x^k est une solution ϵ -optimale,

sinon on choisit $M_k \in \mathcal{C}'_k$ tel que $\mu(M_k)$ soit le plus petit parmi tous les $\mu(M)$, pour $M \in \mathcal{C}'_k$.

Considérons ce cône M_k de sommet 0 et généré par n points linéairement indépendants $v^{k,1}, v^{k,2}, \dots, v^{k,n}$.

On choisit le plus long côté de $T_k = [v^{k,1}, \dots, v^{k,n}]$

et s'il y en a plusieurs, on prend n'importe lequel.

On note u^k le point milieu de ce côté et on subdivise M_k en 2 sous-cônes $M_{k,1}$ et $M_{k,2}$ à partir de la bissection de ce côté.

Sur ces 2 sous-cônes, on va estimer une borne inférieure.

Comme borne supérieure, on prend $\beta_k = \min \{ \gamma_k, f(\pi(u^k)) \}$.

Pour chaque $h = 1, 2$, on résout le sous-problème (SP) correspondant à $\gamma = \gamma_k$ et $M = M_{k,h}$. Celui-ci nous donne

$\alpha_{k,h}^*$ et $\lambda_{k,h}^*$ respectivement comme valeur optimale et solution optimale. On obtient alors un point admissible $z^{k,h} = B \cdot \lambda_{k,h}^*$

qui est une combinaison linéaire des éléments de

$\{v^{k,1}, \dots, u^k, \dots, v^{k,n}\}$ et une borne $\nu_{k,h} = \nu(M_{k,h}, \gamma_k)$.

Finalement, la borne inférieure estimée $\mu(M_{k,h})$ est égale à β_k si $\alpha_{k,h}^* \leq 1$ ou au $\max\{\mu(M_k), \nu_{k,h}\}$ sinon.

A partir de là, on forme M_{k+1} en substituant dans M_k , M_k par $M_{k,1}$ et $M_{k,2}$.

La nouvelle meilleure valeur $\gamma_{k+1} = \min\{\beta_k, f(z^{k,1}), f(z^{k,2})\}$

et la nouvelle meilleure solution admissible est alors x^{k+1} tel que $f(x^{k+1}) = \gamma_{k+1}$.

Ensuite, on va à l'étape $k+1$.

III.1.b. Théorème :

Cet algorithme se termine après un nombre fini d'étapes et donne une solution ε -optimale.

Démonstration :

On sait à partir des théorèmes 1, 2 et 3, que

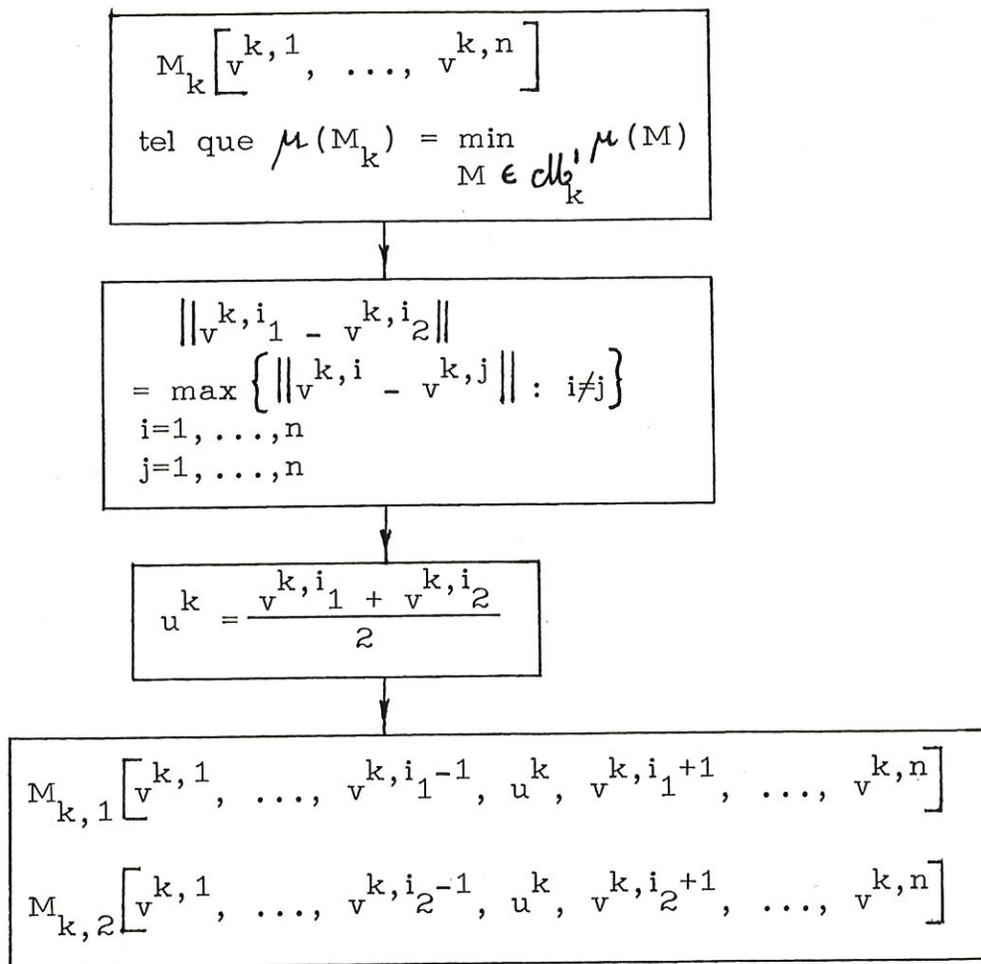
$\gamma_k - \mu(M_k) \rightarrow 0$ quand $k \rightarrow \infty$. Notons que $\mu(M_k)$ ne décroît pas avec k , puisque γ_k ne croît pas.

Donc pour un certain k , on doit avoir $\gamma_k - \mu(M_k) \leq \varepsilon$.

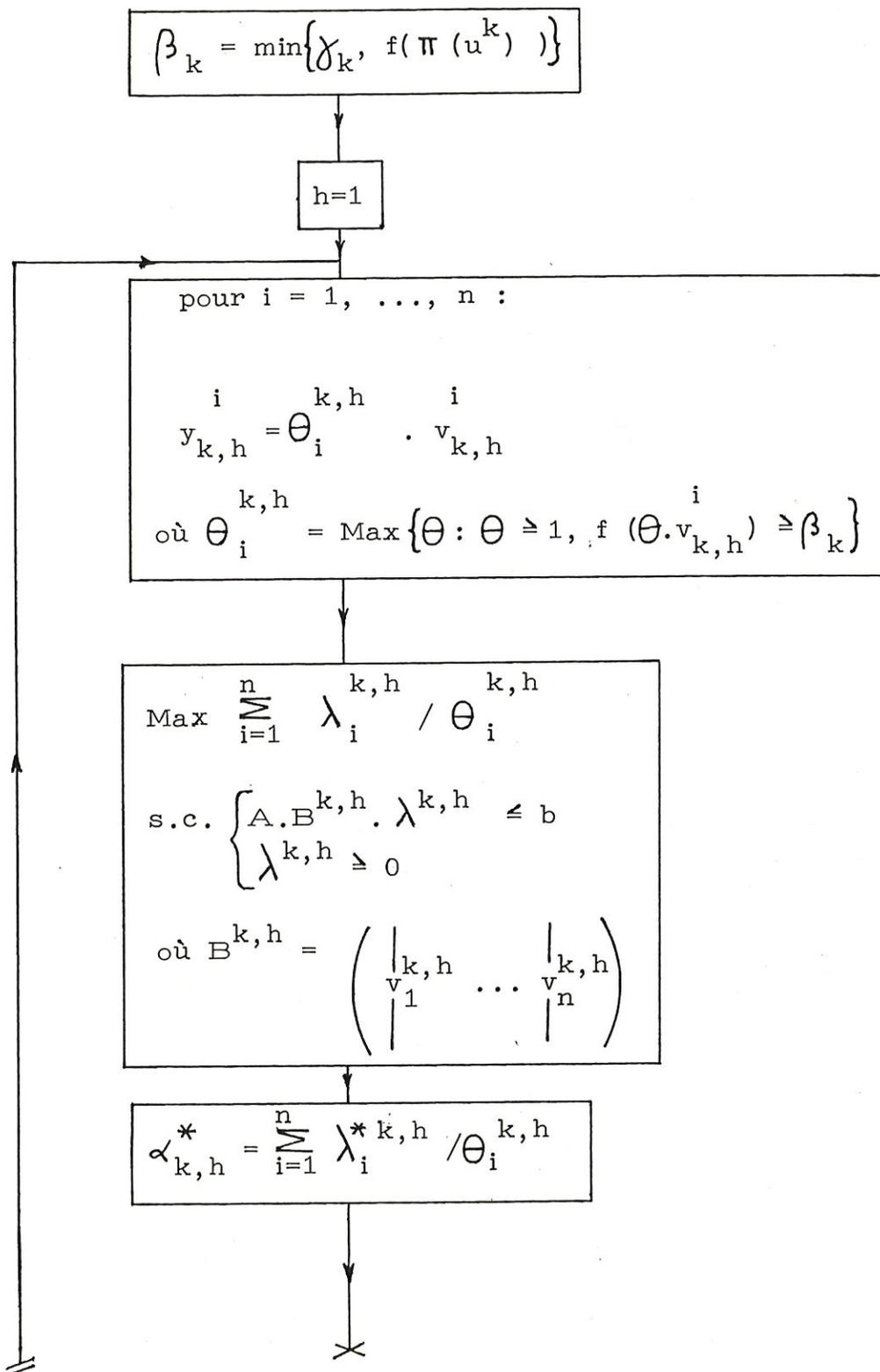
Or par le théorème 1, x^k donne une solution optimale approximée avec une erreur ne dépassant pas $\gamma_k - \mu(M_k)$, c'est-à-dire une solution ε -optimale.

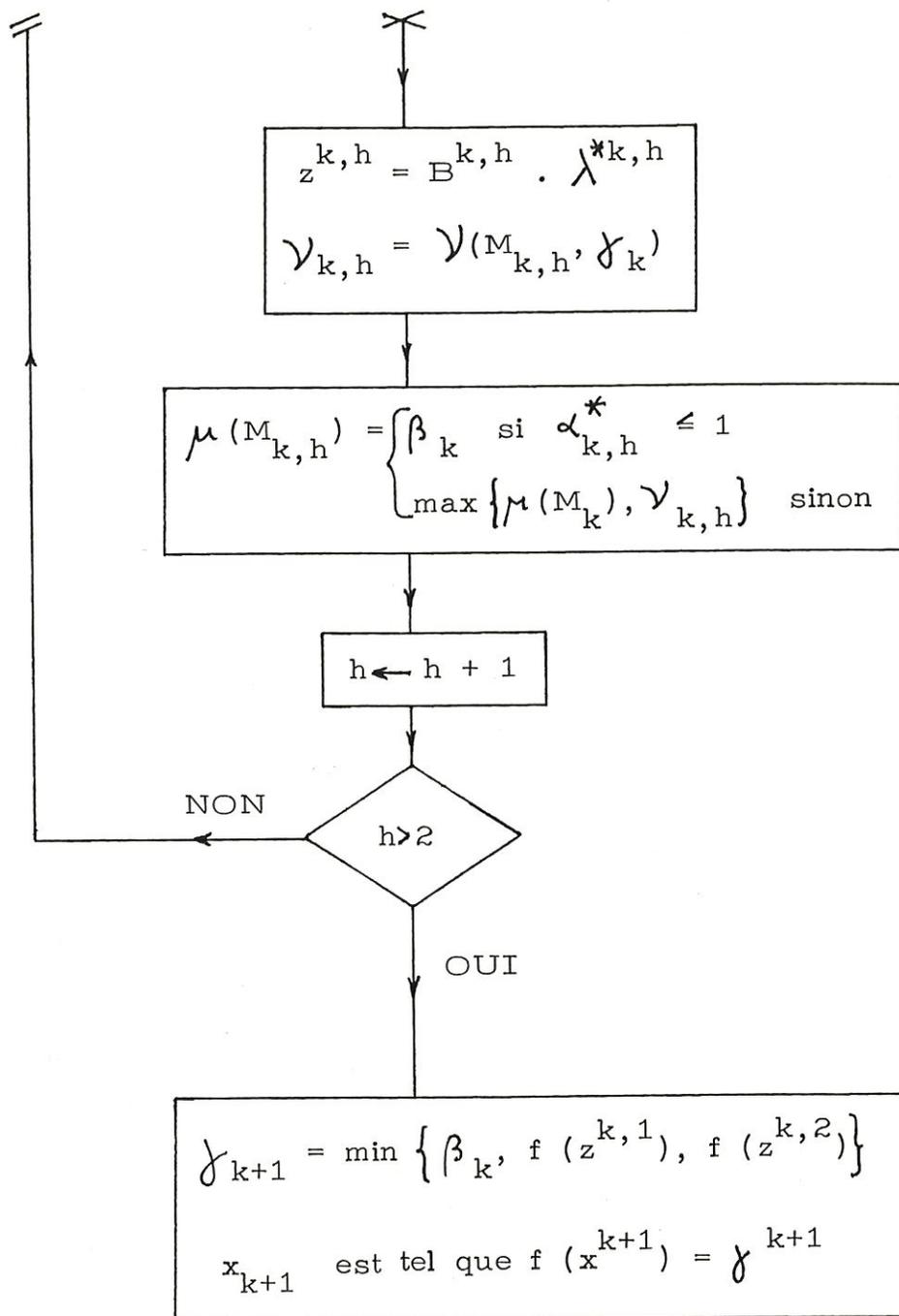
III.1.c. Organigrammes :

1) Subdivision du cône M_k :



2) Estimation d'une borne inférieure sur les 2 sous-cônes :





III.2. Programme algorithmique en FORTRAN VII :

III.2.a. Sous-routines :

1) SUBROUTINE PI (A, S, B, T, N, M, IN, IM)

Pour tout point admissible x de \mathbb{R}^n , on a $A \cdot x \leq B$

où $A = (a_{ij})$ est une matrice de dimension $M \times N$

et $B = (b_i)$ est un vecteur de dimension M .

IN et IM sont respectivement les dimensions maximales que peuvent prendre N et M .

Par la transformation Π , on envoie tout point S du simplexe T sur un point $\Pi(S) = t \cdot S$ de la frontière du domaine admissible δD .

Or par définition de t , c'est le plus grand réel positif tel que $t \cdot S$ reste admissible, c'est-à-dire $t \cdot S$ satisfait toutes les contraintes :

$$\forall i : \sum_{j=1}^n a_{ij} \cdot t \cdot S_j = t \cdot \left(\sum_{j=1}^n a_{ij} \cdot S_j \right) \leq b_i \quad \text{où } t \geq 0.$$

Cependant $b_i \geq 0 \quad \forall i$, car $0 \in D$. Donc si

$$\sum_{j=1}^n a_{ij} \cdot S_j \leq 0, \text{ alors } t = +\infty, \text{ mais si}$$

$$\sum_{j=1}^n a_{ij} \cdot S_j > 0, \text{ alors } t \leq \frac{b_i}{\sum_{j=1}^n a_{ij} S_j}.$$

Par conséquent, on calculera $T = \min \left\{ \frac{b_i}{\sum_j a_{ij} S_j} \mid \forall i \text{ tel}$

que $\sum_j a_{ij} S_j > \eta \right\}$ où $\eta = 10^{-7}$.

Pour éviter les erreurs d'arrondi, on multipliera T par un facteur correctif = 0.9999999999.

2) SUBROUTINE TETA I (V, S, BETA, N, I, TETA, IN)

Pour un i donné, on calcule $\Theta^i = \max\{\Theta : \Theta \geq 1, f(\Theta \cdot v^i) \geq \beta\}$
 = TETA où $\beta = BETA$, $v^i = V(., I)$ et $i = I$. (S est un vecteur de travail qui contiendra v^i).

Ceci se fait par une recherche linéaire uniforme, avec un pas $T = 1$.

Par la concavité (ou quasi-concavité) de la fonction f, on sait que pour des Θ croissants, la fonction va décroître à un certain moment (descendre sous la valeur β dans un intervalle $[TI, TS]$) et ne plus jamais remonter. Une fois cet intervalle déterminé, on entame une recherche plus précise. Pour $\Theta = (TI + TS) / 2$, tant que $(TS - TI) \geq \eta = 10^{-8}$ (test de sortie), si $f(\Theta v^i) \geq \beta$, alors $TI = \Theta$, sinon $TS = \Theta$. Si le test de sortie est satisfait pour un Θ tel que $f(\Theta v^i) < \beta$, on corrige Θ par un facteur multiplicatif = 0.9999999999.

3) SUBROUTINE RECC (V, MK, K, N, IN, IM, IT)

Le stockage des cônes est indépendant de la dimension N du problème.

Chaque cône M_k est stocké dans une ligne K de la matrice MK (de dimension maximale IM et IT) et est décrit par une suite de couples de la forme :

$$(i_1, j_1) (i_2, j_2) \dots (i_l, j_l) 0$$

qui indique le chemin pour construire le simplexe
 $S = M_k \cap \partial T \quad (T = [s^0, s^1, \dots, s^n])$.

En partant du simplexe initial $S_0 = M_0 \cap \partial T$, on remplace le i_1^e sommet de S_0 par le point milieu du côté (i_1, j_1) , c'est-à-dire le segment joignant le i_1^e au j_1^e sommet. Dans le simplexe résultant S_1 , on remplace le i_2^e sommet par le point milieu du côté (i_2, j_2) , etc... jusqu'au moment où $MK (K, 2(l+1) - 1) = 0$.

Notons que le nombre l de couples n'est pas nécessairement égal au nombre d'itérations, car un cône n'est pas forcément subdivisé à chaque étape.

Donc si le cône M_k est ultérieurement subdivisé suivant le côté (i_{l+1}, j_{l+1}) , on obtiendra 2 sous-cônes $M_{k,1}$ et $M_{k,2}$ décrits respectivement par les suites :

$$(i_1, j_1) \dots (i_l, j_l) (i_{l+1}, j_{l+1}) \quad 0 \quad \text{et}$$

$$(i_1, j_1) \dots (i_l, j_l) (j_{l+1}, i_{l+1}) \quad 0.$$

L'ordre des indices des couples a son importance.

Par conséquent, en utilisant cette méthode, il suffit de stocker les $N+1$ sommets de T , à savoir s^0, s^1, \dots, s^n , et une suite de couples pour chaque cône qui doit être enregistré. Et précisément, cette sous-routine recherche les N sommets qui déterminent le cône K et les stocke dans la matrice V de dimension $N \times N$. Une alternative est d'utiliser ce système après un certain nombre d'itérations, quand l'ensemble \mathcal{M}'_k devient trop grand.

4) SUBROUTINE BORN1(A, X,B,Z,V, N, M, GAMA, BETA, VMU, WM, IN, IM, ITE)

Cette sous-routine évalue une borne inférieure de la fonction f sur $D \cap M_k$ où le cône M_k est déterminé par N points linéairement indépendants, stockés dans la matrice V .

A l'aide de la borne supérieure BETA et de la sous-routine TETAI, on calcule les points y^i (pour $i = 1, \dots, N$) qui sont rangés dans la matrice Y .

Pour résoudre le sous-problème (SP), on fait appel à la sous-routine SIMPLX qui minimise une fonction linéaire sous des contraintes d'égalité $AB \cdot \lambda = S$ et de non-négativité $\lambda \geq 0$. La fonction-objectif est stockée dans la première ligne de la matrice AB et à partir des lignes suivantes, on met les coefficients des contraintes, c'est-à-dire de la matrice résultante du produit $A \cdot V$. Le second membre des contraintes est rangé dans le vecteur S à partir du deuxième élément, le premier étant nul.

La variable ALPHA et le vecteur VLAMDA sont respectivement la valeur et le vecteur optimal de ce sous-problème. A partir de ces solutions, on peut calculer le vecteur Z et les points z^i ($i = 1, \dots, N$) qui sont stockés dans la matrice ZI .

A l'itération initiale, la borne estimée VMU prend la valeur $VNU = \mathcal{V}(M_k, \gamma_k)$ qui est le minimum de la fonction aux points z^i . Par après, VMU sera égale à la borne supérieure BETA si ALPHA est inférieure à $1 + \xi$ (où $\xi = 10^{-6}$ est un facteur correctif) ou sinon, au maximum entre VNU et la borne WM du cône non encore subdivisé.

Si Z est une meilleure solution admissible, alors X enregistrera ses composantes et GAMA prendra la valeur de la fonction de ce point.

5) SUBROUTINE LECT (A, B, V, N, M, IN, IM)

Au début du programme s'effectuera la lecture des données du problème. Celles-ci seront la dimension N de l'espace considéré, les n points extrêmes v^i linéairement indépendants (rangés dans les n premières colonnes de la matrice V), le nombre M de contraintes d'inégalité " \leq " qui déterminent le domaine admissible. Ensuite on lira chaque contrainte qui sera stockée dans la matrice A et par après, le vecteur B , second membre de ces mêmes contraintes.

III.2.b. Programme principal :

Il est conçu en double précision pour une dimension N de l'espace considéré et un nombre M de contraintes, respectivement égal à 10 et à 20 au maximum.

La fonction concave f à minimiser est stockée dans un fichier, FOR indépendant.

Après l'initialisation, on estime d'abord une borne pour le cône initial. Ensuite, la deuxième partie est consacrée aux étapes $ITE = 1, 2, 3, \dots$ où on estime une borne sur les 2 sous-cônes résultant de la subdivision.

Les cônes sont stockés chacun sur une ligne de la matrice MK et à cet indice de ligne correspond dans le vecteur WMU , la borne inférieure μ pour ce cône.

Lors de l'élimination des cônes, on utilise un facteur correctif $\epsilon = 10^{-6}$ afin d'obtenir une solution ϵ -optimale.

Le test de sortie se fait simplement lorsque le nombre NC de cônes restants est nul. A ce moment, on sait que la valeur optimale est $GAMA$ au point X .

III.2.c. On trouvera ci-après le programme codé en FORTRAN VII. On a omis la sous-routine LECT qui ne présente guère d'intérêt ici.

```

=====
TABLE DES VARIABLES :
=====
X(.) : meilleure solution admissible courante
N : dimension de l'espace dans lequel on travaille
M : nombre de contraintes lineaires
A(.,.) : matrice des contraintes lineaires
B(.) : second membre des contraintes
V(.,1) : Vi, points lineairement independants, qui determinent
avec le sommet 0, un cone convexe
MK(.,.) : contient par ligne, les indices des points qui determi-
nent un cone
VM(.,.) : matrice de travail
S(.) : vecteur de travail
U(.) : point milieu du plus grand cote
WMU(1) : borne inferieure estimee du cone Mi
NC : nombre de cones restants
VMU : borne inferieure de F(X) sur un cone
BETA : borne superieure
T : variable qui definit la fonction PI(X)=t*X
GAMA : valeur de F(Xi)
F : fonction concave a minimiser
FS,FX : valeur de F(S), de F(X)
ITE : nombre d'iterations
VMIN : borne minimale en un point du cone qui va etre subdivise
MIN : indice du sommet dont la valeur par la fonction F est
minimale parmi les autres sommets du cone
VMAX : valeur du plus grand cote qui etre va "bissecte"
VKI : distance entre 2 points du cone
I1,I2 : indices des points qui forment le plus grand cote
IK : indice du cone qui va etre subdivise
I,J,K,L, : variables entieres de travail
=====

```

```

PROGRAMME PRINCIPAL :
=====

```

```

IMPLICIT DOUBLE PRECISION(A-H,O-Z)
DIMENSION X(10),A(20,10),B(20),V(10,10),MK(15,25)
DIMENSION VM(10,10),WMU(20),U(10),S(10),Z(10)

```

```

CALL LECT(A,B,V,N,M,10,20)

```

```

C      initialisation

```

```

30     NC=1
        IK=1
        ITE=0
        MK(1,1)=0

```

```

C      choix du vecteur initial Xo

```

```

50     DO 50 I=1,N
        X(I)=0.
        MIN=0
        FX=F(X)

```

```

DO 70 I=1,N
DO 60 J=1,N
S(J)=V(J,I)
60 CONTINUE
FS=F(S)
IF(FS.GE.FX) GO TO 70
MIN=I
FX=FS
70 CONTINUE
GAMA=FX
IF(MIN.EQ.0) GO TO 90
DO 80 J=1,N
80 X(J)=V(J,MIN)
C calcul de BETA
90 BETA=GAMA
DO 120 I=1,N
DO 100 J=1,N
100 S(J)=V(J,I)
CALL PI(A,S,B,T,N,M,10,20)
DO 110 J=1,N
110 S(J)=T*S(J)
FS=F(S)
IF(FS.GE.BETA) GO TO 120
BETA=FS
IF(FS.GE.GAMA) GO TO 120
DO 115 K=1,N
115 X(K)=S(K)
GAMA=FS
120 CONTINUE
C estimation d'une borne inferieure sur Mo
CALL BORNIA(X,B,Z,V,N,M,GAMA,BETA,VMU,WMU(1),10,20,ITE)
WMU(1)=VMU
C iteration suivante
200 ITE=ITE+1
DO 210 J=1,N
DO 210 K=1,N
210 VM(J,K)=V(J,K)
CALL RECC(VM,MK,IK,N,10,15,25)
C recherche du plus grand cote
VMAX=0.
DO 230 I=1,N-1
DO 225 J=I+1,N
VKI=0.
DO 220 K=1,N
220 VKI=VKI+(VM(K,I)-VM(K,J))**2
IF(VKI.LE.VMAX) GO TO 225
VMAX=VKI
I1=I
I2=J
225 CONTINUE
230 CONTINUE

```

C bisection

```
245        DO 250 I=1,N
          U(I)=(VM(I,I1)+VM(I,I2))/2
250        CONTINUE
          NC=NC+1
```

C stockage des sous-cones

```
          J=1
260        IF(MK(IK,J).EQ.0) GO TO 280
          MK(NC,J)=MK(IK,J)
          MK(NC,J+1)=MK(IK,J+1)
          J=J+2
          GO TO 260
280        MK(IK,J)=I1
          MK(IK,J+1)=I2
          MK(IK,J+2)=0
          MK(NC,J)=I2
          MK(NC,J+1)=I1
          MK(NC,J+2)=0
```

C calcul d'une borne inferieure sur les 2 cones subdivises

```
          CALL PI(A,U,B,T,N,M,10,20)
          DO 310 J=1,N
310        S(J)=T*U(J)
          FS=F(S)
          BETA=GAMA
          IF(FS.LT.BETA) BETA=FS
          IF(FS.GE.GAMA) GO TO 313
          DO 312 I=1,N
312        X(I)=S(I)
          GAMA=FS
313        DO 315 I=1,N
          S(I)=VM(I,I2)
315        VM(I,I2)=U(I)
          CALL BORNI(A,X,B,Z,VM,N,M,GAMA,BETA,VMU,WMU(IK),10,20,ITE)
          WMU(NC)=VMU
          DO 320 I=1,N
          VM(I,I2)=S(I)
          VM(I,I1)=U(I)
320        CONTINUE
          CALL BORNI(A,X,B,Z,VM,N,M,GAMA,BETA,VMU,WMU(IK),10,20,ITE)
          WMU(IK)=VMU
```

C choix du cone qui va etre subdivise

```
          VMIN=GAMA
          J=1
322        IF(WMU(J).GE.VMIN) GO TO 325
          VMIN=WMU(J)
          IK=J
```

C elimination des cones dont la borne inferieure est >ou= GAMA
C et restructuration des stockages

```
325        IF((WMU(J)+1.E-6).LT.GAMA) GO TO 400
```

```

C      test d'arret
      IF(NC.EQ.1) GO TO 2000
      NC=NC-1
      IF(J.EQ.(NC+1)) GO TO 200
      DO 360 L=J,NC
      I=1
      MK(L,I)=MK(L+1,I)
350    MK(L,I+1)=MK(L+1,I+1)
      I=I+2
      IF(MK(L+1,I).NE.0) GO TO 350
      MK(L,I)=0
360    WMU(L)=WMU(L+1)
      GO TO 322
400    J=J+1
      IF(J.LE.NC) GO TO 322
      GO TO 200

C      impression des resultats finaux

2000   WRITE(5,2010) ITE,GAMA
2010   FORMAT(/,2(1X,32('S')),//,'SS',28X,'SS',/, 'SS',9X,'ON OBTIENT'
1'9X,'SS',/, 'SS' APRES I2,' ITERATIONS SS',/, 'SS'
2' D24.17,' LA VALEUR OPTIMALE SS',/, 'SS',28X,'SS',/, 'SS'
3,'SS',/, 'SS',28X,'SS',/, 'SS',8X,'AU POINT X':',8X
4,'SS',/, 'SS',28X,'SS')
      DO 500 I=1,N
500    WRITE(5,2020) X(I)
2020   FORMAT('SS',D24.17,' SS')
      WRITE(5,2030)
2030   FORMAT('SS',28X,'SS',/,2(1X,32('S')),/,//)
      STOP
      END
C~~~~~

C      SOUS-ROUTINES :
C      =====

C      SUBROUTINE BORNIA(A,X,B,Z,V,N,M,GAMA,BETA,VMU,WM,IN,IM,ITE)
C      *****

C      estimation d'une borne inferieure de F(X) sur D^M

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C      TABLE DES VARIABLES :
C      =====
C      V(.,i) : Vi, points lineairement independants, qui deter-
C      minent avec le sommet 0, le cone convexe considere
C      AB(.,.) : matrice des contraintes du simplexe en LAMBDA
C      Y(.,i) : les points Yi qui determinent l'hyperplan H
C      ZI(.,i) : Zi =ALPHA*Yi
C      Z(.) : vecteur se trouvant a une distance maximale de H
C      INFIX(.) : vecteur contenant des informations sur le simplexe
C      TOL(.) : vecteur de precision
C      KOUT(.) : vecteur contenant des informations sur les resultats
C      ERR(.) : vecteur de precision
C      JH(.) : vecteur entier de travail dans SIMPLX
C      W(.) : vecteur reel de travail dans SIMPLX
C      YY(.) : contient la valeur optimale, duale, etc...
C      P(.) : vecteur solution optimal

```



```

C      # appel de la sous-routine SIMPLX
      CALL SIMPLX(INFIX,AB,S,TOL,PRM,KOUT,ERR,JH,Z,P,YY,KB,E)
      IF(KOUT(1).EQ.3) GO TO 180

C      # cas ou SIMPLX ne trouve pas de solution optimale
      WRITE(5,998)KOUT(1),KOUT(2)
998    FORMAT(/, ' KOUT=',I2, ' ITER.=',I2)
      WRITE(5,1000)
1000   FORMAT(/,2(1X,52('*'),/), ' **',48X, '**',/, ' ** LA SOLUTION DU
1      SIMPLEXE N'EST PAS OPTIMALE!! **',/, '**',48X, '**',/,2(1X
2,52('*'),/))
      STOP

C      # resultats de SIMPLX
180    DO 200 I=1,N
      VLAMDA(I)=0.
      IF(KB(I).EQ.0) GO TO 200
      II=KB(I)
      VLAMDA(I)=Z(II)
200    CONTINUE
      ALPHA=-YY(1)

C      calcul de Z et des Zi
      DO 220 I=1,N
      Z(I)=0.
      DO 210 J=1,N
      Z(I)=Z(I)+V(I,J)*VLAMDA(J)
      ZI(J,I)=ALPHA*Y(J,I)
210    CONTINUE
220    CONTINUE
      FS=F(Z)
      IF(FS.GE.GAMA) GO TO 223
      DO 222 I=1,N
222    X(I)=Z(I)
      GAMA=FS

C      calcul de la borne inferieure
223    IF(ITE.EQ.0) GO TO 225
      VMU=BETA
      IF(ALPHA.LE.(1.+1.E-6)) RETURN
225    VNU=1.E10
      DO 240 I=1,N
      DO 230 J=1,N
230    S(J)=ZI(J,I)
      FS=F(S)
      IF(FS.GE.VNU) GO TO 240
      VNU=FS
240    CONTINUE
      VMU=VNU
      IF(ITE.EQ.0) RETURN
      IF(VNU.LT.WM) VMU=WM
      RETURN
      END
C ~~~~~

```

```

C      SUBROUTINE RECC(V,MK,K,N,IN,IM,IT)
C      *****
C      recherche des points qui determinent un cone
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C      DIMENSION V(IN,IN),MK(IM,IT)
C
C      I=1
10     IF(MK(K,I).EQ.0) RETURN
C      DO 50 J=1,N
50     V(J,MK(K,I))=(V(J,MK(K,I))+V(J,MK(K,I+1)))/2.
C      I=I+2
C      GO TO 10
C      END

```

```

C      SUBROUTINE PI(A,S,B,T,N,M,IN,IM)
C      *****
C      calcul de la fonction PI(X)=tX
C
C      TABLE DES VARIABLES :
C      =====
C      D,SOM : variables reelles de travail
C      T : valeur de la fonction PI recherchee
C      au point S considere
C
C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C      DIMENSION A(IM,IN),S(IN),B(IM)
C
C      T=1.E10
C      DO 20 I=1,M
C      SOM=0.
10     DO 10 J=1,N
C      SOM=SOM+A(I,J)*S(J)
C      IF(SOM.LT.1.E-7) GO TO 20
C      D=B(I)/SOM
C      IF(D.LT.T) T=D
20     CONTINUE
C      T=.9999999999*T
C      RETURN

```

```

C~~~~~END~~~~~
C
SUBROUTINE TETAI(V,S,BETA,N,I,TETA,IN)
C *****
C maximiser TETA>=1 tel que F(TETA*Vi)>=BETA (recherche lineaire)
C
C
C TABLE DES VARIABLES :
C =====
C TI,TS : borne inferieure et superieure de l'intervalle
C de recherche
C FS : valeur de la fonction F en TETA*Vi
C T : pas de la recherche lineaire
C S(.) : vecteur de travail
C I : indice du point Vi
C TETA : valeur recherchee
C
C
C
C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C DIMENSION V(IN,IN),S(IN)
C
C TI=1.
C TS=2.
C T=1.
5 DO 10 J=1,N
10 S(J)=TS*V(J,I)
C FS=F(S)
C IF(FS.LT.BETA) GO TO 50
C TI=TS
C TS=TS+T
C IF(TS.LE.1.E7) GO TO 5
C WRITE(5,1000)
1000 FORMAT(/,' ERREUR DANS TETAI',/)
C STOP
C TETA=(TI+TS)/2
C DO 60 J=1,N
60 S(J)=TETA*V(J,I)
C FS=F(S)
C IF(FS.GE.BETA) GO TO 70
C TS=TETA
C IF((TS-TI).GT.1.E-08) GO TO 50
C TETA=.999999999*TETA
C IF(TETA.LT.1.) TETA=1.
C RETURN
C TI=TETA
70 IF((TS-TI).GT.1.E-08) GO TO 50
C RETURN
C END
C~~~~~

```

CHAPITRE IV.

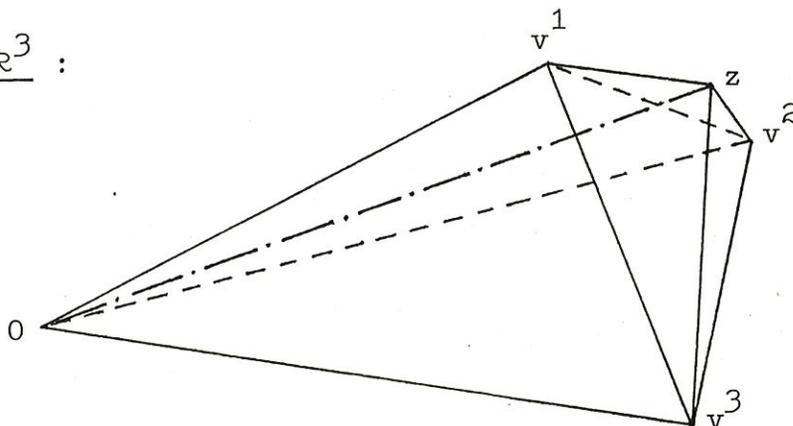
Amélioration de la méthode.

IV.1. Interprétation :

Dans l'algorithme précédent, on subdivise à chaque itération, le cône dont la borne inférieure μ est la plus petite, en 2 sous-cônes suivant le point-milieu du plus grand côté. Ceci permet de trouver une solution ξ -optimale en un nombre fini d'itérations.

Afin d'accélérer la vitesse de convergence, il serait intéressant de tenir compte de l'information obtenue lors de l'estimation de la borne inférieure du cône non encore fractionné. De plus, on sait que le minimum est atteint en un des sommets du domaine admissible D . Or si dans le sous-problème (SP), α est plus grand que 1, alors le point z obtenu est forcément un sommet de D . On subdivisera de ce fait le cône suivant le sommet z obtenu. Au maximum, on passera par tous les sommets de D et obligatoirement par la solution optimale.

dans \mathbb{R}^3 :



IV.2. Théorème de convergence :

$\forall k = 0, 1, 2, \dots$:

Considérons le point admissible z^k obtenu lors de la résolution du sous-problème (SP) dans l'estimation de $\mu(M_k)$ où

$$M_k = M_k [v^{k,1}, \dots, v^{k,n}].$$

On subdivise M_k en au plus n sous-cônes $M_{k,i}$ déterminés par

l'ensemble des points $\{v^{k,1}, \dots, v^{k,n}\}$ où on a substitué $v^{k,i}$ par z^k pour $i = 1, \dots, n$, à condition que l'ensemble

résultant de points déterminant chaque sous-cône, soit linéairement indépendant.

Alors le nombre de cônes considérés sera au plus égal à

$$n \cdot [s(m) - n - 1] + 1$$

où $s(m)$ est le nombre de sommets du domaine admissible (dépend du nombre m de contraintes)

et n est la dimension de l'espace dans lequel on travaille :

démonstration :

Au départ, on a un cône M_0 .

Pour évaluer $x^0 = \min_{i=0, \dots, n} \{f(s^i)\}$, on visite déjà $(n+1)$ sommets.

Supposons que $\alpha_0 > 1$, sinon M_0 est supprimé et x_0 est solution optimale.

Alors on obtient un point admissible z^0 à distance maximale de l'hyperplan $H[y^1, \dots, y^n]$:

z^0 est un sommet de D par construction, car z est un sommet du cône admissible, donc aussi du domaine admissible.

On crée de ce fait au plus n sous-cônes
 c'est-à-dire $(n - 1)$ cônes supplémentaires.
 Or il reste au plus $s(m) - (n+1)$ sommets à visiter.
 Par conséquent, on aura $s(m) - (n+1)$ cônes pour trouver ces
 sommets, plus $\lceil s(m) - (n+1) \rceil \cdot (n-1)$ sous-cônes supplémentaires
 engendrés. On n'en aura plus d'autres, car si un cône ne
 contient pas strictement un sommet, alors il est rejeté.
 Ceci nous donnera au total $\lceil s(m) - (n+1) \rceil \cdot n + 1$ le cône initial.

remarque :

Le nombre $n \cdot \lceil s(m) - n - 1 \rceil + 1$ est une borne supérieure car
 il se peut qu'un cône contenant un sommet non encore visité,
 soit rejeté :

En fait, on effectuera $n \cdot \lceil s(m) - n - 1 \rceil + 1$ estimations d'une
 borne inférieure au maximum.

IV.3. Modifications engendrées :

a) Stockage des cônes :

Chaque sommet trouvé sera stocké dans une colonne de la
 matrice V . De ce fait, on retiendra dans la matrice MK ,
 les n indices-colonnes de V qui déterminent les n points
 linéairement indépendants.

Ici, le stockage dépend de la dimension du problème.

b) Indépendance linéaire :

Celle-ci se vérifie en résolvant un système de n équations à n inconnues par élimination gaussienne (voir la sous-routine GAUSS). Les coefficients des équations sont les coordonnées des n points qui déterminent le cône. Si le système est indéterminé ou impossible, c'est-à-dire si le déterminant des n vecteurs de dimension n , est nul, alors ces points sont linéairement dépendants.

La sous-routine LIDP donnera dans ce cas-là, une valeur non nulle à la variable LI.

c) Programme algorithmique :

Dans le programme principal, on estimera la borne inférieure que d'un seul sous-cône par itération.

Après chaque sommet trouvé, on testera la dépendance linéaire des n points qui déterminent chaque sous-cône et on stockera s'il le faut, les indices de ces points suivant la procédure décrite préalablement.

Ce sont les seuls changements apportés au programme y compris les 2 sous-routines précitées et dont voici les détails.

PROGRAMME PRINCIPAL :
=====

IMPLICIT DOUBLE PRECISION(A-H, O-Z)
DIMENSION X(10), A(20,10), B(20), V(10,30), MK(20,10)
DIMENSION VM(10,10), WMU(30), Z(10), S(10)

CALL LECT(A, B, V, N, M, 10, 20)

```

C      initialisation
      NC=1
      IK=1
      LI=0
      ITE=-1
      NS=N
40     DO 40 I=1,N
      MK(I,I)=I
C      choix du vecteur initial X0
      DO 50 I=1,N
50     X(I)=0.
      MIN=0
      FX=F(X)
      DO 70 I=1,N
      DO 60 J=1,N
60     S(J)=V(J,I)
      CONTINUE
      FS=F(S)
      IF(FS.GE.FX) GO TO 70
      MIN=I
      FX=FS
70     CONTINUE
      GAMA=FX
      IF(MIN.EQ.0) GO TO 90
      DO 80 J=1,N
80     X(J)=V(J,MIN)
C      calcul d'une meilleure solution admissible
      DO 120 I=1,N
      DO 100 J=1,N
100    S(J)=V(J,I)
      CALL PI(A,S,B,T,N,M,10,20)
      DO 110 J=1,N
110    S(J)=T*S(J)
      FS=F(S)
      IF(FS.GE.GAMA) GO TO 120
      DO 115 K=1,N
115    X(K)=S(K)
      GAMA=FS
120    CONTINUE
C      iteration suivante
200    ITE=ITE+1
C      calcul d'une borne inferieure sur le cone IK
      DO 300 J=1,N
      DO 290 I=1,N
290    VM(I,J)=V(I,MK(IK,J))
      CONTINUE
      BETA=GAMA
      CALL BORNI(A,X,B,Z,VM,N,M,GAMA,BETA,VMU,WMU(IK),10,20,ITE)
300    WMU(IK)=VMU

```

```

C      elimination du cone si sa borne inferieure est >ou= GAMA
C      et restructuration des stockages
      IF((WMU(IK)+1.E-6).LT.GAMA) GO TO 370

C      test d'arret
      IF(NC.EQ.1) GO TO 2000
      NC=NC-1
      IF(IK.EQ.(NC+1)) GO TO 500
      DO 360 L=IK,NC
      DO 350 I=1,N
350    MK(L,I)=MK(L+1,I)
360    WMU(L)=WMU(L+1)
      GO TO 500

C      stockage des sous-cones dont les N points qui les determinent
C      sont lineairement independants
370    NS=NS+1
      DO 375 K=1,N
375    V(K,NS)=Z(K)
      DO 400 I=1,N-1
      DO 380 J=1,N
380    S(J)=VM(J,I)
      VM(J,I)=Z(J)
      IF(N.NE.2) CALL LIDP(VM,N,10,LI)
      IF(LI.EQ.1) GO TO 390
      NC=NC+1
      DO 385 J=1,N
385    MK(NC,J)=MK(IK,J)
      WMU(NC)=WMU(IK)
      MK(NC,I)=NS
390    DO 395 K=1,N
395    VM(K,I)=S(K)
400    CONTINUE
      DO 410 K=1,N
410    VM(K,N)=Z(K)
      IF(N.NE.2) CALL LIDP(VM,N,10,LI)
      IF(LI.EQ.1) GO TO 420
      MK(IK,N)=NS
      GO TO 500
420    IF(NC.EQ.1) GO TO 2000
      NC=NC-1
      IF(IK.EQ.(NC+1)) GO TO 500
      DO 460 L=IK,NC
      DO 450 I=1,N
450    MK(L,I)=MK(L+1,I)
460    WMU(L)=WMU(L+1)

C      choix du cone dont on va estimer la borne
500    VMIN=GAMA
      DO 525 J=1,NC
      IF(WMU(J).GE.VMIN) GO TO 525
      VMIN=WMU(J)
      IK=J
525    CONTINUE
      GO TO 200

```

Ensuite vient l'impression des résultats finaux (voir p. 50).

```

C      SUBROUTINE LIDP(VM,N,IN,LI)
C      *****
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION VM(IN,IN),V(10,10),S(10)

C      verifie si les N points qui determinent le cone VM sont
C      lineairement independants

      LI=0
      ZERO=1.E-7
      DO 20 I=1,N
      S(I)=1.
      DO 10 J=1,N
10      V(I,J)=VM(I,J)
20      CONTINUE
      CALL GAUSS(V,S,N,10,ZERO,LI)
      RETURN
      END
C~~~~~
C      SUBROUTINE GAUSS(A,B,N,IN,ZERO,IERR)
C      *****
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION B(IN),A(IN,IN)
      DIMENSION IL(10),IC(10)

C      resolution d'un systeme de n equations a n inconnues
C      par la methode de GAUSS

      IERR=0
      DO 100 I1=1,N
100      IL(I1)=0
      IC(I1)=0
      DO 200 IJ=1,N
      PIV=0
      DO 130 I1=1,N
      IF(IL(I1).NE.0) GO TO 130
      DO 120 I2=1,N
      IF(IC(I2).NE.0) GO TO 120
      AMEM=DABS(A(I1,I2))
      IF(AMEM.LE.DABS(PIV)) GO TO 120
      PIV=A(I1,I2)
      MEM1=I1
      MEM2=I2
120      CONTINUE
130      CONTINUE
      IF(DABS(PIV).GE.ZERO) GO TO 150
      IERR=1
      RETURN
150      IL(MEM1)=MEM2
      IC(MEM2)=MEM1
      DO 160 I3=1,N
160      A(MEM1,I3)=A(MEM1,I3)/PIV
      B(MEM1)=B(MEM1)/PIV
      DO 190 I4=1,N
      IF(I4.EQ.MEM1) GO TO 190
      BMM=A(I4, MEM2)
      DO 180 I5=1,N
      IF(I5.EQ.MEM2) GO TO 180
      A(I4,I5)=A(I4,I5)-A(MEM1,I5)*BMM
180      CONTINUE
      A(I4, MEM2)=0
      B(I4)=B(I4)-B(MEM1)*BMM
190      CONTINUE
200      CONTINUE
      DO 220 J2=1,N
220      A(J2,1)=B(J2)
      DO 230 J1=1,N
      IL1=IL(J1)
230      B(IL1)=A(J1,1)
      RETURN
      END

```

CHAPITRE V.

Exemples numériques.

=====

V.1. Exemples testés :

V.1.a. Fonctions quadratiques :

1) dans \mathbb{R}^{10} :

Q.10.1 :

$$f(X) = C \cdot X - (X^t \cdot D^t \cdot D \cdot X) / 10$$

$$\text{où } C = (10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10)$$

$$\text{et } D = \begin{bmatrix} 1 & 5 & 2 & 2 & 2 & 1 & 5 & 2 & 2 & 2 \\ 2 & 1 & 7 & 3 & 2 & 2 & 1 & 7 & 3 & 2 \\ 2 & 2 & 1 & 2 & 2 & 2 & 2 & 1 & 2 & 2 \\ 3 & 1 & 3 & 4 & 2 & 3 & 1 & 3 & 4 & 2 \\ 4 & 2 & 6 & 1 & 2 & 4 & 2 & 6 & 1 & 2 \\ 2 & 2 & 1 & 5 & 2 & 2 & 2 & 1 & 5 & 2 \\ 3 & 2 & 2 & 1 & 7 & 3 & 2 & 2 & 1 & 7 \\ 4 & 2 & 3 & 1 & 3 & 4 & 2 & 3 & 1 & 3 \\ 1 & 2 & 4 & 2 & 6 & 1 & 2 & 4 & 2 & 6 \end{bmatrix}$$

sous les contraintes :

$$\begin{cases} -x_i \leq 0 & \text{pour } i = 1, \dots, 10 \end{cases}$$

$$20x_1 + 20x_2 + 60x_3 + 60x_5 + 60x_6 + 5x_7 + 45x_8 + 55x_9 + 65x_{10} \leq 600.1$$

$$5x_1 + 7x_2 + 3x_3 + 8x_4 + 13x_5 + 13x_6 + 2x_7 + 14x_8 + 14x_9 + 14x_{10} \leq 310.5$$

$$100x_1 + 130x_2 + 50x_3 + 70x_4 + 70x_5 + 70x_6 + 20x_7 + 80x_8 + 80x_9 + 80x_{10} \leq 1800$$

$$200x_1 + 280x_2 + 100x_3 + 200x_4 + 250x_5 + 280x_6 + 100x_7 + 180x_8 + 200x_9 + 220x_{10} \leq 3850$$

$$2x_1 + 2x_2 + 4x_3 + 4x_4 + 4x_5 + 4x_6 + 2x_7 + 6x_8 + 6x_9 + 6x_{10} \leq 18.6$$

$$4x_1 + 8x_2 + 2x_3 + 6x_4 + 10x_5 + 10x_6 + 5x_7 + 10x_8 + 10x_9 + 10x_{10} \leq 198.7$$

$$60x_1 + 110x_2 + 20x_3 + 40x_4 + 60x_5 + 70x_6 + 10x_7 + 40x_8 + 50x_9 + 50x_{10} \leq 882$$

$$150x_1 + 210x_2 + 40x_3 + 70x_4 + 90x_5 + 105x_6 + 60x_7 + 100x_8 + 140x_9 + 180x_{10} \leq 4200$$

$$80x_1 + 100x_2 + 6x_3 + 16x_4 + 20x_5 + 22x_6 + 20x_8 + 30x_9 + 30x_{10} \leq 402.5$$

$$40x_1 + 40x_2 + 12x_3 + 20x_4 + 24x_5 + 28x_6 + 40x_9 + 50x_{10} \leq 327$$

La valeur optimale $\gamma^* = -348.099$ est obtenue

$$\text{en } X^* = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 9.3 \\ 0. \\ 0. \\ 0. \end{bmatrix} \quad \text{ou en } X^* = \begin{bmatrix} 0. \\ 4.025 \\ 0. \\ 0. \\ 0. \\ 0. \\ 5.275 \\ 0. \\ 0. \\ 0. \end{bmatrix}$$

Q.10.2 :
.....

$$f(X) = C.X - (X^t.D^t.D.X) / 10$$

$$\text{où } C = (10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10 \ 10)$$

$$\text{et } D = \begin{bmatrix} 0.1 & 0.3 & 0.2 & -0.1 & 0. & 0.2 & 0.6 & 0. & -0.1 & 1.0 \\ 0.2 & 0.5 & 0.3 & -0.2 & 0. & 0. & 0.7 & 0.2 & -0.2 & 0.1 \\ 0.3 & -0.3 & 0.5 & -0.3 & 0. & 0. & 0.4 & -0.4 & -0.3 & 0.2 \\ -0.4 & -0.2 & -0.3 & -0.4 & 0.3 & 0.3 & 0.1 & 0. & -0.4 & -1.0 \\ 0. & 0.4 & 0. & -0.5 & 0.2 & 0. & 0. & 0. & -0.5 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0. & 1.0 & 0.6 & 0.1 & 0.2 & 0.1 & 0. \\ 0.2 & 0.3 & 0.5 & 0. & 0.1 & 0.7 & 0.2 & 0. & 0.2 & 0.2 \\ 0.3 & 0.5 & -0.3 & 0. & 0.2 & 0.4 & 0.3 & 0. & 0.3 & -0.4 \\ -0.4 & -0.3 & -0.2 & 0.3 & -0.1 & 0.1 & 0.4 & 0.3 & 0.4 & 0. \\ 0. & 0. & 0.4 & 0.2 & 0.1 & 0. & 0.5 & 0. & 0.5 & 0. \end{bmatrix}$$

sous les mêmes contraintes que Q.10.1.

La valeur optimale $\gamma^* = 0.$ est obtenue

$$\text{en } X^* = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \\ 0. \end{bmatrix}$$

2) dans \mathbb{R}^3 :

Q.3.1 :

$$f(X) = C \cdot X + X^t \cdot D \cdot X + F$$

$$\text{où } C = (4 \ 10 \ -6) , F = -38$$

$$\text{et } D = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

sous les contraintes :

$$\begin{cases} 0.44540475x_1 - 0.391202x_2 + 0.3657663807x_3 \leq 3.312519666 \\ -1.06270001x_1 + 4.25080005x_2 + 2.6435429x_3 \leq 21.25400025 \\ 2.660372205x_1 - 8.230907284x_2 + 14.13871844x_3 \leq 40.7050876 \\ 21x_1 + 28x_2 + 24x_3 \leq 168 \\ -x_i \leq 0 \text{ pour } i=1, 2, 3 \end{cases}$$

La valeur optimale $y^* = -63.5625$ est obtenue

$$\text{en } X^* = \begin{pmatrix} 5.99999 \\ 0. \\ 1.75000875 \end{pmatrix}$$

Q.3.2 :

$$f(X) = C \cdot X + X^t \cdot D \cdot X + F$$

$$\text{où } C = (7 \ 0 \ 3) , F = -12$$

$$\text{et } D = \begin{pmatrix} -1.0 & -0.2 & 0.0 \\ -0.2 & -0.1 & 0.0 \\ 0.0 & 0.0 & -1.0 \end{pmatrix}$$

sous les mêmes contraintes que Q.3.1.

La valeur optimale $J^* = -17.87219052$ est obtenue
 en $X^* = \begin{pmatrix} 0. \\ 2.356457 \\ 4.250800167 \end{pmatrix}$

3) dans \mathbb{R}^2 :

$$f(X) = C \cdot X + X^t \cdot D \cdot X + F$$

Q.2.1 :

$$C = (16 \ -2) , F = -65 \text{ et } D = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

sous les contraintes :

$$\left\{ \begin{array}{l} -10x_1 + 3x_2 \leq 0 \\ -10x_1 + 14x_2 \leq 55 \\ -x_1 + 6x_2 \leq 40 \\ 0.2108x_1 + 2x_2 \leq 17.6864 \\ 1.2892x_1 + 3x_2 \leq 36.2596 \\ 1.5x_1 + 1.7268x_2 \leq 30.7242 \\ 2x_1 + 1.35095x_2 \leq 36.20835 \\ 2x_1 - 0.0775x_2 \leq 31 \\ x_1 - 16x_2 \leq 0 \end{array} \right.$$

La valeur optimale $J^* = -81.25004506$ est obtenue
 en $X^* = \begin{pmatrix} 16.07775 \\ 3. \end{pmatrix}$

Q.2.2 :

$$C = (14 \ -4) , F = -53 \text{ et } D = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$$

sous les contraintes :

$$\begin{cases} -x_1 \leq 0 \\ -3x_1 + 11x_2 \leq 77 \\ 3x_1 + 5x_2 \leq 83 \\ 4x_1 + 3x_2 \leq 85 \\ 2x_1 - 9x_2 \leq 11 \\ x_1 - 10x_2 \leq 0 \end{cases}$$

La valeur optimale $\gamma^* = -169$ est obtenue

en $X^* = \begin{pmatrix} 19 \\ 3 \end{pmatrix}$

Q.2.3 :

$C = (26 \ -6)$, $F = -178$ et $D = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$

sous les contraintes :

$$\begin{cases} x_1 - 4x_2 \leq 0 \\ -10x_1 + 7x_2 \leq 0 \\ -2x_1 + 3x_2 \leq 16 \\ 3x_1 + 5x_2 \leq 114 \\ 7x_1 + 2x_2 \leq 150 \end{cases}$$

La valeur optimale $\gamma^* = -307.7950139$ est obtenue

en $X^* = \begin{pmatrix} 13.78947368 \\ 14.52631579 \end{pmatrix}$

V:1.b. Fonctions linéaires par morceaux :

dans \mathbb{R}^2 :

L:2:1 :

$$f(X) = \begin{cases} x_1 - 7x_2 & \text{si } x_1 \leq 11.5 \\ -7.5x_1 - 7x_2 + 97.75 & \text{si } 11.5 \leq x_1 \leq 15 \\ -8x_1 - 7x_2 + 105.25 & \text{si } x_1 \leq 15 \end{cases}$$

sous les mêmes contraintes que Q:2:1.

La valeur optimale $\gamma^* = -48$ est obtenue

$$\text{en } X^* = \begin{pmatrix} 8 \\ 8 \end{pmatrix}$$

L:2:2 :

$$f(X) = \begin{cases} 2x_1 - 11x_2 & \text{si } x_1 \leq 18 \\ -5.5x_1 - 11x_2 + 135 & \text{si } 18 \leq x_1 \leq 22 \\ -10x_1 - 11x_2 + 234 & \text{si } x_1 \leq 22 \end{cases}$$

sous les contraintes :

$$\begin{cases} -2x_1 + 11x_2 \leq 96. \\ 0.822553x_1 + 2x_2 \leq 38.805954 \\ 1.177447x_1 + 2x_2 \leq 45.903834 \\ -10x_1 + 7x_2 \leq 0. \\ 5x_1 - 2x_2 \leq 90. \\ x_1 - 4x_2 \leq 0. \end{cases}$$

La valeur optimale $\gamma^* = -97.9519170$ est obtenue

$$\text{en } X^* = \begin{pmatrix} 20. \\ 11.177447 \end{pmatrix}$$

REFERENCES.

- [1] H. TUY (1964). Concave programming under linear constraints. Soviet Math. 5, 1437-1440.
- [2] FALK & HOFFMAN (1976). A successive underestimation method for concave minimisation problems. Math. Oper. Res. 1, 251-259.
- [3] H. TUY & NG.V. THOAI (1980). Convergent algorithms for minimizing a concave function. Math. Oper. Res. 5, 556-566.
- [4] HOFFMAN (1981). A method for globally minimizing concave functions over convex sets. Math. Programming 20, 22-32.
- [5] REZA (1981). Minimisation of a quasi-concave function on bounded or unbounded polyhedral sets. PH.D. dissertation.
- [6] BALAS (1971). Intersection cuts - A new type of cutting plane for integer programming. Operations Res. 19, 19-39.
- [7] SIMONNARD (1972). Programmation linéaire. Dunod-Paris.