REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES

Mémoire de fin d'étude

En vue d'obtention du diplôme master en recherche opérationnelle

Option : **METHODES ET MODELES DE DECISION**

THÈME

La méthode QBB, de l'optimisation globale deux fois différentiable

Présenté par :

Mr:ALLIK BOUBEKER

Mr: MAMOU IDRIS

Encadré par :

Mr: OUANES

Devant le jury d'éxamen composé de :

Mm : Rabia
Mr : Chebah

Promotion juin 2014

Table des matières

Remerciements

D'abord nous remercions

le bon dieu de nous avoir donné santé, courage et foie

pour réaliser ce travail avec volonté.

Tous nos vifs remerciements ; nos profondes reconnaissances

s'adressent à notre promoteur Mr **OUANES Mohand** à qui nous tenons

témoigner notre sincère gratitude de nous avoir confié ce sujet intéressant pour

les conseils et son aide pour accomplir notre travail.

Que Monsieur le président et Messieurs les membres de jury

trouvent ici l'expression de notre gratitude et respect de nous avoir

fait l'honneur d'examiner et juger notre travail.

Nous remercions nos très chers parents pour leurs

soutiens et leurs encouragements durant

notre cycle d'étude.

Enfin, merci à tous ceux qui ont

contribué de près ou de loin à ce mémoire, du point de vue

scientifique ou administratif.

Introduction générale

L'optimisation est une branche mathématique qui cherche à analyser et à résoudre les problèmes qui consistent à déterminer le meilleur élément d'un ensemble, au sens d'un critère donné.

L'optimisation est devenue une discipline incontournable du monde moderne dans lequel nous vivons, car celui-ci est sujet à une compétition internationale excessive et croissante. Dés lors, il devient nécessaire, voir vital pour les entreprises comme pour les gouvernements, de maximiser ou de minimiser toutes sortes de choses ; par exemple maximiser les profits tout en minimisant les pertes, améliorer si possible de façon optimale certaines processus de fabrication ou les fonctionnalités, de certaines processus objets ou produits.

L'optimisation va consister à rechercher dans le domaine initiale une solution qui maximise ou minimise une fonction objectif, pour un domaine continu et discret, on distingue classiquement deux type d'optimisation :

-L'optimisation locale : cherche une solution qui est la meilleure localement, cette solution est appelée un optimum local.

-L'optimisation globale : cherche quant à elle la meilleure solution du domaine en entier, c'est à dire que dans tout le domaine. il n'existe aucune solution qui lui soit meilleur tout en respectant les contraintes. Cette solution est appelée globale.

L'intérêt de l'optimisation globale par rapport à l'optimisation locale est patent. Elle garantit en effet que personne ne peut avoir une solution meilleure que celle trouvée. Or, pour une entreprise, cette information a son importance, car la différence entre la solution globale et une solution locale est bien souvent significative. Mais l'intérêt n'est pas que compétitif. Dans de nombreux problèmes, l'optimum global est la seule solution mathématique corres-

pondant à une réalité physique. C'est ce qu'illustre par exemple la recherche de la quantité de chaque élément présent dans un mélange chimique à l'équilibre.

De nos jours, afin de résoudre des problèmes d'optimisation globale avec contraintes, de nombreux travaux ont été effectués sur les différentes approches (tel que les méthodes multi-start,les algorithmes révolutionnaires, Les méthodes métaheuristiques,les méthodes dites stochastiques et les méthodes déterministes globales).

Nous allons nous intéresser aux méthodes déterministes ,parmi elles la méthode Branch and- Bound qui nous offres la certitude d'obtenir l'optimum global(si celui ci existe).Ce type d'algorithme dit de séparation et évaluation qui est plus connue sous le nom Branch and Bound qui peut théoriquement ressoudes n'importe quel problème.

Notre étude se base sur les problèmes d'optimisations globales des fonctions deux fois differentiable.

Le premier chapitre de notre travail commence par un rappel des notions élémentaires de la programmation mathématique ainsi que les notions et définitions qui en découlent afin d'établir clairement les bases sur lesquelles repose notre approche . Nous y détaillerons le principe de l'algorithme de base Branch-and-Bound en optimisation globale .

Le deuxième chapitre sera consacré à la présentation de la méthode QBB ,dont les techniques de calcul des fonctions de borne inférieure ,ainsi que l'algorithme explécite de la méthode et en fin un éxample numérique connu sous le nom "le problème de Colville".

Quand au dernier chapitre , on va faire une description de logiciel LINGO,qui est un logiciel d'application util pour la résolution des problèmes d'optimisation.

Nous finirons notre travail par une conclusion générale.

Chapitre 1

Présentation et généralités

1.1 Introduction:

Nous proposons dans ce premier chapitre introductif les notions de base sur l'étude de l'optimisation et les notions d'analyse d'intervalles ainsi que de l'algèbre. On décrira aussi une technique principale fréquente utilisée dans l'optimisation globale qui est la méthode de séparation et évaluation (Branch and Bound).afin d'établir clairement les bases sur lesquelles repose notre travail.

1.2 Forme générale des programmes mathématiques

La programmation mathématique se fait en optimisant une fonction á plusieurs variables soumise á un ensemble d'équations ou d'inequations appelées contraintes. On appelle fonction coût ou fonction critère ou encore fonction objectif la fonction á minimiser ou á maximiser et on nomme respectivement contraintes d'égalité et contraintes d'inégalités les équations et les inequations. Lorsque toutes ces fonctions sont linéaire on parle de la programmation linéaire ,dans le cas contraire ,on parle de la programmation non linéaire. De façon générale, un programme non linéaire s'écrit selon la forme standard suivante :

$$\begin{cases} minf(x) & \\ g_i(x) \leq 0 & i = 1,,,,m \\ h_i(x) = 0 & j = 1,,,,n \\ x \in \mathbb{R}^n & \end{cases}$$

Où f,g,h de classe C^2,x un vecteur de taille n. Presque tous les programmes mathématiques peuvent être mis á la forme standard par quelques transformations élémentaires. Ainsi, un problème de maximisation se transforme en problème de minimisation par l'égalité :

$$maxf(x) = -min(-f(x)).$$

Une contrainte d'inégalité dans le mauvais sens qui comporte un second membre non nul,soit :

$$g_i(x) \geq b_i$$

Peut s'écrire sous la forme:

$$-g_i(x) + b_i \leq 0.$$

Une contrainte d'inégalité:

$$g_i(x) \leq 0.$$

Peut être transformée en contrainte d'égalité par l'ajout d'une variable d'écart non négative selon l'une ou l'autre formulation suivante :

$$g_i(x) + x_{n+i} = 0; x_{n+i} \geq 0$$

On transforme une contrainte d'égalité en deux contraintes d'inégalité, par exemple :

$$h_j(x) = 0 \Leftrightarrow \begin{cases} h_j(x) \leq 0 \\ h_j(x) \geq 0 \end{cases}$$

Définition 1.1. Un minimum global est un point admissible x^*qui est selon la fonction critère meilleur ou aussi bon que n'importe quel autre point réalisable c'est-à-dire :

$$f(x^*) \leq f(x) \forall x \in S$$

Ce minimum est seulement local si l'inégalité précédente n'est valable que dans un voisinage immédiat de x^* c'est-à-dire :

$$f(x^*) \leq f(x) \forall x \in v(x^*)$$

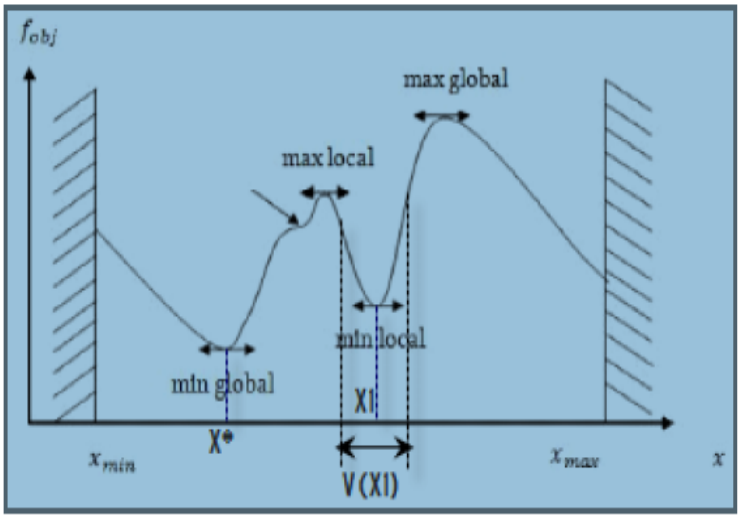


FIG. 1.1 – *Optimum local et optimum global*

1.3 Rappel de notions d'analyse et d'algèbre

On note $M_{m,n}(K)$ l'ensemble de matrices de la forme

$$A = \begin{pmatrix} a_{11} & a_{12} & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & a_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{m1} & a_{m2} & . & . & a_{mn} \end{pmatrix}$$

, à coefficients dans $K = (\mathbb{Z},\mathbb{Q},\mathbb{R})$. On écrira pour simplifier $A = (a_{ij})$ ou a_{ij} est l'élément de A situé à l'intersection de la ligne i et de la colonne j. La somme de deux matrices $A + B = (a_{ij} + b_{ij})$ et le produit d'une matrice A par un nombre $\alpha \in \mathbb{R}$ défini par $\alpha A = (\alpha a_{ij})$ vérifient toutes les propriétés,ce qui nous permet de dire : $M_{m,n}(R)$ a une structure de $\mathbb{R}$ espace vectoriel.

Matrice identité

La matrice identité s'écrit sous forme $In = \begin{pmatrix} 1 & 0 & . & . & 0 \\ 0 & 1 & . & . & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & . & . & . & 1 \end{pmatrix} \in M_{n,n}(R)$.

Matrice inversible

On dit qu'une matrice $A \in M_{n,n}$ est inversible s'il existe une matrice $B \in M_{n,n}$ telle que $AB = BA = In$, Dans ce cas B est unique.

Proposition. *Soit $A \in M_{n,n}(R)$. Les affirmations suivantes sont équivalentes:*

1. *A est inversible.*
2. $\det(A) \neq 0$
3. *Il existe $B \in M_{n,n}(R)$ tel que $AB = In$*
4. *Il existe $C \in M_{n,n}(R)$ tel que $CA = In$*

Les matrices ligne de $M_{1,n}(R)$ on les appelle vecteurs ligne et les matrices colonne de $M_{n,1}(R)$ sont appelées vecteurs colonne. Dans la suite, quand on fait des opérations avec des vecteurs on sous-entends vecteurs de même type.

Définition 1.2. Une famille de vecteurs $(v_i)_{i\in I}$ (avec I un ensemble fini) sera dite

- **Liée** : s'il existe $i \in I$ et $\alpha_j \in \mathbb{R}, j \neq i$ tels que $v_i = \sum_{j\in I\setminus\{i\}} \alpha_j v_j$,
- **Libre** : si elle n'est pas liée, c'est-à-dire : $\alpha_i \in \mathbb{R}, \sum_{i\in I} \alpha_i v_i = 0 \Rightarrow \alpha_i = 0 \ \forall i \in I$.
- **Génératrice** : si tout vecteur v s'écrit comme une combinaison linéaire des vecteurs de la famille, i.e. pour tout v il existe des nombres α_i tels que $v = \sum_{i\in I} \alpha_i v_i$.

Définition 1.3. L'espace vectoriel V engendré par une famille de vecteurs $(v_i)_{i\in I}$ avec $I = \{1,...,p\}$ est l'ensemble de toutes les combinaisons linéaires de cette famille et sera notée Vect $(v_1,...,v_p)$. Donc $V = Vect(v_1,...,v_p) = \{\alpha_1 v_1 + ... + \alpha_p v_p / \alpha_1,...,\alpha_p \in \mathbb{R}\}$.

Théorème. *Soit V un espace vectoriel* ($V \subset M_{n,1}(\mathbb{R})$ *ou* $V \subset M_{1,n}(\mathbb{R})$) *qui possède une base ayant p éléments. Alors :*

1. *Toute autre base de V a le même nombre p d'éléments. Ce nombre sera dit la dimension de V et sera noté* $p = \dim V$.
2. *Toute famille libre de V a au plus p éléments.*
3. *Toute famille génératrice de V a au moins p éléments.*
4. *Toute famille libre de V ayant exactement p éléments est une base de V .*
5. *Toute famille génératrice de V ayant exactement p éléments est une base de V .*
6. *Toute famille libre est contenue dans une base (i.e. on peut compléter une famille libre pour obtenir une base).*
7. *De toute famille génératrice on peut extraire une base.*

Définition 1.4. Le rang d'une matrice A est la dimension de l'espace vectoriel engendré par les colonnes de A. Donc si $A = (A_1, A_2,...,A_n) \in M_{m,n}(\mathbb{R})$,avec $A_j \in M_{p,1}(\mathbb{R})$ alors $rang(A) = dimvect(A_1,A_2,...,A_n)$.

Remarque 1.1. evidament on a : $rang(A) \leq min(p,n)$.

Théorème. *soit* $A \in M_{m,n}(\mathbb{R})$,*Les affirmations suivantes sont équivalentes :*

- $rang(A) = r$.
- *Il existe un mineur d'ordre r de A non nul et tous les mineurs d'ordre* $r+1$ *(s'il y en a) sont nuls.*
- *L'espace engendré par les lignes de A est de dimension r.*
- *Il existe une sous-matrice carrée de A d'ordre r inversible et toutes les sous-matrices carrées d'ordre* $r+1$ *(s'il y en a) sont non inversibles.*

Remarque 1.2. soit $A \in M_{n,n}(\mathbb{R})$ une matrice carée alors:
$rang(A) = n \Leftrightarrow det(A) \neq 0 \Leftrightarrow A$ est inversible.

1.4 opérations sur les matrices

1. **la somme:** soient A et B deux matrices de même ordre $(m \times n)$ alors la somme $A + B$ est égale à la matrice C, dans laquelle tout élément est la somme des éléments correspondants de A et B.
2. **le produit:** soient A et B deux matrices d'ordre $(m \times p)$ et $(p \times n)$ respectivement, le produit A.B est une matrice C d'ordre $(m \times n)$,dont chaque élément c_{ij} est la somme des produits des éléments de la $i^{ème}$ ligne de A par les éléments correspondants de la $j^{ème}$ colonne de B.
   $c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$,
   (a) Ce produit n'est possible que si le nombre de colonnes de A est égale au nombre de lignes de B.
   (b) Le produit A.B peut *être* défini sans que le produit B.A le soit.
3. **Matrice transposée:** En remplaçant dans une matrice A d'ordre $(m \times n)$ les lignes par les colonnes respectives on obtient une matrice $A' = A^t$ qui s'appelle matrice transposée de la matrice A, d'ordre $(n \times m)$ et vérifie les propriétés suivantes :
   (a) $(A^t)^t = A$
   (b) $(A + B)^t = A^t + B^t$
   (c) $(AB)^t = B^t A^t$

Matrice carrée

La matrice carrée s'écrit sous forme $A = \begin{pmatrix} a_{11} & a_{12} & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & a_{2n} \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{n1} & . & . & . & a_{nn} \end{pmatrix} \in M_{n,n}(R)$.

Avec $a_{ij} \in R; i = 1...n; j = 1...n$.

On a n lignes et n colonnes.

Le vecteur $(a_{i1},...,a_{in})$ est appelé i^{eme} ligne.

Le vecteur $\begin{pmatrix} a_{1j} \\ . \\ . \\ a_{nj} \end{pmatrix}$,est appelé j^{eme} colonne.

Définition 1.5. Soit A une matrice carrée symétrique ,on dit que A est:

1. **définie positive** si:
$$X^T AX > 0 \quad ,\forall X \neq 0.$$
2. **définie négative** si :
$$X^T AX < 0 \quad ,\forall X \neq 0.$$
3. **semi-définie positive** si:
$$X^T AX \geq 0 \quad ,\forall X \neq 0.$$
4. **semi-définie négative** si:
$$X^T AX \leq 0 \quad ,\forall X \neq 0.$$

1.5 Critère de Sylvester (mineurs principaux)

Soit A une matrice carrée symétrique($n \times n$),on dit que A est:

1. **définie positive** si:
$$detA_K > 0 \quad ,\forall K = 1,...,n.$$
2. **définie négative** si :
$$det(-1)^k A_K > 0 \quad ,\forall K = 1,...,n.$$
3. **semi-définie positive** si:
$$detA_K \geq 0 \quad ,\forall K = 1,...,n.$$
4. **semi-définie négative** si:
$$det(-1)^k A_K \geq 0 \quad ,\forall K = 1,...,n.$$

, $\forall K = 1,...,n.$

1.6 le critère des valeurs propres

soit A une matrice carrée symétrique et λ_i les valeurs propres de A,on dit que A est:

1. **déf inie positive** si:
$$\lambda_i > 0, \forall i = 1,...,n.$$
2. **définie négative** si :
$$\lambda_i < 0, \forall i = 1,...,n.$$
3. **semi-définie positive** si:
$$\lambda_i \geq 0, \forall i = 1,...,n.$$
4. **semi-définie négative** si:
$$\lambda_i \leq 0, \forall i = 1,...,n.$$

1.7 Fonctions de plusieurs variables

Définition 1.6. Soit D un domaine de R^n, une application f de D à valeurs dans R est dite fonction réelle de n variables:

$f : D \subseteq R^n \to R$

$(x_1,...,x_n) \to f(x_1,...,x_n)$

Définition 1.7. soit $f : D \subseteq R^n \to R$

f est continue en P_0 tel que $P_0 \in D$ si:$\forall \epsilon > 0, \exists \delta > 0$ tel que :

$\| P - P_0 \| < \delta \Rightarrow |f(P) - f(P_0)| < \epsilon.$

On écrit: $\lim_{p \to p_0} f(P) = f(P_0)$.

Définition 1.8. On dit que f est continue sur D si elle est continue en chaque point de D.

Définition 1.9. On définit la dérivée partielle de f par rapport à x_i au point $P = (x_1,...,x_n)$ par la limite suivante: $\lim_{h \to 0} \frac{f(x_1,...,x_i+h,...x_n)-f(x_1,...,x_i,...x_n)}{h}$ quand elle existe et on la note :$\frac{\partial f}{\partial x_i}(p)$

Définition 1.10. On dit que f est continûment differentiable sur D (classe C^1) si ses dérivées partielles sont continues sur D.

Définition 1.11. Si $f \in C^1(R)$, on appelle gradient de f en x le vecteur de R^n :

$$\nabla f(x) = (\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, ..., \frac{\partial f(x)}{\partial x_n})$$

avec $x = (x_1,...,x_n)$

Exemple 1.1. $f(x_1,....,x_n) = \sum_{i=1}^{n} x_i^2$

$\frac{\partial f}{\partial x_i}(x_1,...,x_n) = 2x_i, i = 1,...,n \Rightarrow$ *continue sur* $R^n \Rightarrow f$ *est de classe* C^1.

Définition 1.12. Soit $F : D \subseteq R^n \to R^n$ on définit la matrice Jacobienne de F par:

$$JF(x) = \begin{pmatrix} \nabla f_1(x) \\ . \\ . \\ \nabla f_n(x) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & . & . & . & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & . & . & . & \frac{\partial f_2}{\partial x_n} \\ . & . & . & . & . \\ . & . & . & . & . \\ \frac{\partial f_n}{\partial x_1} & . & . & . & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Définition 1.13. Soit f de classe C^2, on définit la hessienne de f par :

$$H_f(x) = J(\nabla f) = \begin{pmatrix} \frac{\partial^2 f_1}{\partial x_1^2} & . & . & . & \frac{\partial^2 f_1}{\partial x_1 \partial x_n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \frac{\partial^2 f_n}{\partial x_n \partial x_1} & . & . & . & \frac{\partial^2 f_n}{\partial x_n^2} \end{pmatrix}$$

Remarque 1.3. $H_f(x)$ est une matrice carrée symétrique.

Exemple 1.2. *Ecrire la Hessienne de:*$f(x,y,z) = 2x^2 + y^2 + z^2 + 2xy + 4yz + 2zx$

$$H_f = J(4x + 2y + 2z, 2y + 2x + 4z, 2z + 4y + 2x)$$

$$H_f = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 2 & 4 \\ 2 & 4 & 2 \end{pmatrix}$$

1.8 Condition d'optimalité pour les extremums d'une fonction

ce qu'on cherche ici c'est le moyen avec lequel on poura faire la différence entre un optimum et tout les autres points voisins . Le cas sans contrainte constitue une bonne introductive à ce genre de résultats théoriques bien utiles. D'autre part, dans le cas où les variables x_1,...,x_n sont soumises à des conditions supplémentaires ($g_i(X) \leq 0$,$i = 1,...,m; h_j(X) = 0$,$j = 1,...,p$) peuvent dans certaines conditions se ramener à la résolution de problèmes d'optimisation sans contrainte.

1.8.1 Condition nécessaire d'optimalité locale

Soit X^* un minimum de la fonction f(x). pour tout $t > 0$ assez petit et $d \in R^n$ on écrit la formule de Taylor:

$$f(x^* + td) = f(x^*) + td\nabla f(x^*) + o(\| td \|)$$

$$\Rightarrow f(x^* + td) - f(x^*) = td\nabla f(x^*) + o(\| td \|)$$

$$\Rightarrow \frac{f(x^* + td) - f(x^*)}{t} = \frac{td\nabla f(x^*) + o(\| td \|)}{t}$$

D'où :

$$lim_{t\to 0^+} \frac{f(x^* + td) - f(x^*)}{t} = lim_{t\to 0^+}(d\nabla f(x^*) + o(\| td \|))$$

En particulier avec : $d = -\nabla f(X^*)$, on obtient:

$$\nabla f(x^*)(-\nabla f(x^*)) \geq 0 \Rightarrow - \| \nabla f(x^*) \|^2 \geq 0$$

$\Rightarrow - \parallel \nabla f(x^*) \parallel^2 = 0$

$\Rightarrow \parallel \nabla f(x^*) \parallel^2 = 0$

D'où :

$\nabla f(x^*) = 0$

Ceci a donné l'énoncé d'une première condition d'optimalité dite condition nécessaire du premier ordre :

Un minimum x^* de $f(x)$ sur R^n annule le gradient de la fonction , c'est-à-dire que:

$\nabla f(x^*) = 0.$

Définition 1.14. Les points qui vérifient $\nabla f(x^*) = 0$ sont appelés points critiques ou points stationnaires.

Théorème. *Une condition nécessaire du deuxième ordre pour que x^* soit un minimum (local ou global)de f est :*

1. $\nabla f(x^*) = 0$ *(stationnarité).*
2. $H_f(x^*)$ *est une matrice semi-définie positive.*

Preuve. *1/ déjà démontré.*

2/ comme f est deux fois continûment differentiable, le développement de Taylor à l'ordre 2 au voisinage de x^* donne :

$f(x^* + td) = f(x^*) + \nabla f(x^*)td + \frac{1}{2}(td)^T H_{f(x^*)}td + o(\parallel td \parallel^2)$

puisque $\nabla f(x^*) = 0$ alors:

$\frac{f(x^*+td)-f(x^*)}{t^2} = \frac{\frac{1}{2}t^2 d^T H_{f(x^*)}d + o(\|td\|^2)}{t^2}$

D'où :

$lim_{t\to 0}\frac{f(x^*+td)-f(x^*)}{t^2} = lim_{t\to 0}\frac{\frac{1}{2}t^2 d^T H_{f(x^*)}d + o(\|td\|^2)}{t^2}$

$\frac{1}{2}d^T H_{f(x^*)}d + lim_{t\to 0}\frac{o(t^2)}{t^2} \parallel td \parallel^2 = \frac{1}{2}d^T H_{f(x^*)}d \geq 0$

$\Rightarrow H_{f(x^*)}$ *est semi-définie positive.*

Condition suffisante d'optimalité locale

Une condition suffisante pour que x^* soit un optimum local de f sur R^n est :

1. $\nabla f(x^*) = 0$ *(stationnarité).*

2. $H_f(x^*)$ est une matrice définie positive.

Preuve. *On a le résultat d'algèbre :*

$\lambda_{min} = min_{X \neq 0} \frac{X^t AX}{\|X\|}^2$

λ_{min}*:La plus petite valeur propre de A.*

$\lambda_{max} = max_{X \neq 0} \frac{X^t AX}{\|X\|}^2$

λ_{max}*:La plus grande valeur propre de A.*

Alors on a:

$min_{X \neq 0} \frac{X^t AX}{\|X\|}^2 \leq \frac{X^t AX}{\|X\|}^2 \leq max_{X \neq 0} \frac{X^t AX}{\|X\|}^2$

$\lambda_{min} \parallel X \parallel^2 \leq X^t AX \leq \lambda_{max} \parallel X \parallel^2$

Le développement de Taylor de f à l'ordre 2 au niveau de X^*s'écrit alors :

$f(X^* + d) = f(X^*) + \nabla f(X^*) + \frac{1}{2} d^T H_{f(X^*)} d + o(\parallel d \parallel^2)$

$\Rightarrow f(X^* + d) - f(X^*) = \frac{1}{2} d^T H_{f(X^*)} d + o(\parallel d \parallel^2) \geq \frac{1}{2} \lambda_{min} \parallel d \parallel^2 + o(\parallel d \parallel^2)$

$\Rightarrow \frac{1}{2} \lambda_{min} \parallel d \parallel^2 + o(\parallel d \parallel^2) \geq 0$

Donc $f(X^* + d) \geq f(X^*)$

$\Rightarrow X^*$*est un minimum local.*

1.8.2 Cas des fonctions convexes : condition nécessaire et suffisante d'optimalité global

Dans le cas d'une fonction convexe f définie sur R^n une condition nécessaire et suffisante pour que X^* soit un minimum global de f est que ξ un sous gradient de f en X^*. Pour une fonction continûment differentiable, on obtient donc:

Théorème. *si f une fonction convexe continûment differentiable une condition nécessaire et suffisante pour que X^* soit un optimum global de f sur R^n est que :* $\nabla f(X^*) = 0$. *Autrement dit ; dans le cas convexe, la stationnarité à elle seule constitue une condition nécessaire et suffisante d'optimalité globale.*

Remarque 1.4. On appelle sous gradient de f au point X^0 tout vecteur $Y = (y_1,.....,y_n)^T \in R^n$ vérifiant :

$f(X) \geq f(X^0) + Y^T(X - X^0), \forall X \in R^n$.

1.8.3 Optimisation convexe

Dans les problèmes d'optimisation, avec ou sans contraintes, la convexité joue un rôle trés important.En effet pour la plupart des algorithmes que nous décrirons, la convergence vers un optimum global ne pourra être démontré qu'avec des hypothèses de convexité. Dans ce premier chapitre nous introduisons rapidement quelques éléments de l'analyse convexe requis pour l'étude de l'optimisation.

Ensemble convexe

Un ensemble $S \subseteq R^n$ est dit convexe si et seulement si:

$$\forall x \in S, \forall y \in S \forall \lambda \in [0,1] \Rightarrow \lambda x + (1-\lambda)y \in S$$

C'est-à-dire S est convexe si et seulement si pour deux points quelconques x et y pris dans S le segment [x,y] est tout entier contenu dans S. Plus généralement, étant donné $p(x_1,...,x_p)$ points de R^n ,on dit que $x \in R^n$ est combinaison convexe de ces points s'ils existent des coefficients $U_1,...,U_p, U_i \geq 0, \forall i = 1,2,...,p$tel que

$$\sum_{i=1}^{p} U_i = 1 et \sum_{i=1}^{p} U_i x^i = x$$

1.8.4 Enveloppe convexe :

On remarque que l'intersection d'ensembles convexes est convexe. Etant donné un ensemble S on peut définir l'enveloppe convexe Conv (S) comme le plus petit ensemble convexe contenant S c-a-d:

$$S \subseteq conv(S)$$

Définition 1.15. un polyèdre P est un sous-ensemble de $\mathbb{R}^n$ tel qu'il existe une famille $\{a_1,a_2,...,a_n\}$ de vecteurs de $\mathbb{R}^n$ et une famille $\{b_1,b_2,...,b_m\}$ de réels vérifiant:
$\forall x \in P, \forall i \in \{1,...,m\}, a_i^t x \leq b_i$
On notera A la matrice dont les lignes sont les $(a_i)_{1 \leq i \leq n}$,et b le vecteur formé des $b_1,b_2,...,b_m$.
P peut alors s'écrire: $P = \{x \in \mathbb{R}^n / Ax \leq b\}$

Définition 1.16. Un polytope P est un sous-ensemble de $\mathbb{R}^n$ tel qu'il existe une famille $\{x_1,x_2,...,x_p\}$ de points de $\mathbb{R}^n$ vérifiant: $P = conv(\{x_1,x_2,...,x_p\})$
On introduit aussi la notion de sommet qui joue un role particulier pour ces structures.

Définition 1.17. Un point z d'un sous-ensemble S de $\mathbb{R}^n$ est un sommet s'il n'existe pas x,y$\in S$,tels que

$$z = \lambda.x + (1-\lambda).y, pour un \lambda \in]0,1[$$

On note qu'on aurait pu se contenter, pour la définition, du cas $\lambda = 1/2$
Nous nous concentrons maintenant sur le cas d'un polyèdre défini par A et b, avec $P = \{x \in \mathbb{R}^n / Ax \leq b\}$ Dans ce cas, étant donnée $z \in P$, on notera A_z la sous-matrice de A constituée des lignes d'indice i avec $a_i^t = b_i$.

résultat: soit $P = \{x \in \mathbb{R}^n / Ax \leq b\}$ un polyèdre de $\mathbb{R}^n$ Alors z est un sommet de P si et seulement si $rang(Az) = n$.

1.8.5 Fonction convexe:

Une fonction $f : R^n \to R$ est dite convexe si elle vérifie :

$$\forall x \in R^n, \forall y \in R^n, et \forall \lambda \in [0,1] : f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

Cette inégalité est appelée inégalité de convexité. Elle signifie que toute corde joignant deux points quelconque $(x^1,f(x^1))$et $(x^2,f(x^2))$ se trouve toujours au dessus de la courbe de f.

Théorème. *si f est deux fois continûment differentiable, les conditions (1) (2) (3) ci-dessous sont équivalentes :*

1. *f est convexe.*
2. $\forall x \in R^n, \forall x^0 \in R^n, f(x) \geq f(x^0) + \nabla f^T(x^0).(x - x^0)$ *c'est-'a-dire la courbe C de f est au dessus de ses tangentes.*

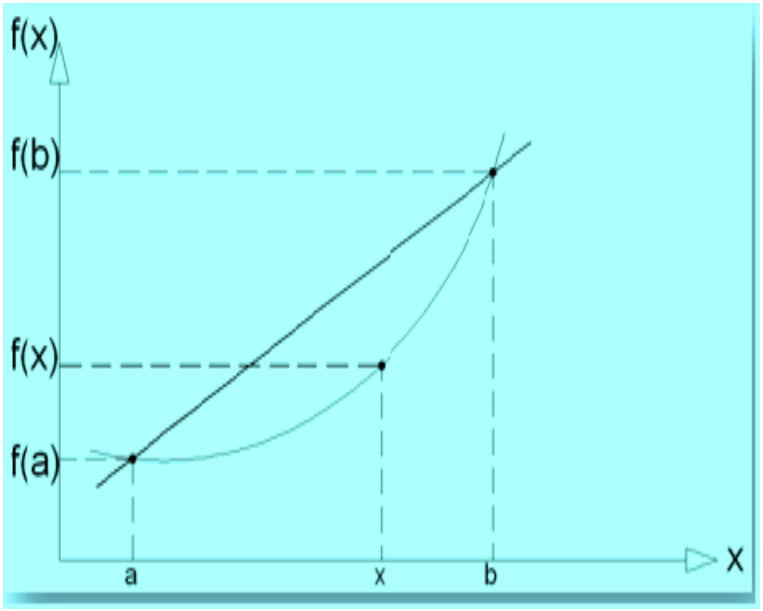


FIG. 1.2 – *Fonction convexe*

3. $\forall x$, *la hessienne* $\nabla^2 f(x)$ *est une matrice semi-définie positive c'est-à-dire:* $\forall y, y^T \nabla^2 f(x).y \geq 0$

Remarque 1.5. 1. f est dite strictement convexe si l'inégalité stricte est toujours vérifiée:

$$\forall x \in R^n, \forall y \in R^n, et \forall \lambda \in [0,1] : f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

2. f est dite fortement convexe ,s'il existe $\alpha > 0$ tel que:

$$\forall x \in R^n, \forall y \in R^n, et \forall \lambda \in [0,1] : f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y) - \frac{1}{2}\alpha\lambda(1-\lambda) \parallel x-y \parallel^2$$

On dit aussi que f est α convexe.

1.8.6 Fonction concave

Une fonction $f : R^n \to R$ *est dite concave si* $(-f)$ *est une fonction convexe i.e avec les mêmes notations , si* $\forall x \in R^n, \forall y \in R^n, \forall \lambda \in [0,1], f(\lambda x + (1-\lambda)y) = \lambda f(x) + (1-\lambda) f(y)$

Remarque 1.6. La convexité de la fonction est une caractéristique très importante en optimisation. En effet, lorsque la fonction n'est pas convexe ,il est pratiquement impossible d'identifierun optimum global d'un probléme de minimisation on utilise la notion de

convexité.

-Noter que la convexité et la concavité ne sont pas des propriétés complémentaires

*Une fonction peut n'étre ni convexe ni concave.

*les fonctions linéaires sont convexes et concaves au même temps.

1.8.7 L'estimation

Cést la connaissance des grandeurs initiales disponibles pour la résolution d'un probléme déxtremum est incompléte ,pourtant quoi qu'il en soit la position du probléme nous fournit quelques informations sur l'aspect physique et des renseignements correspondants sur les estimations de la précision des données déntrée appliquées aux problémes déxtremu tout en étudiant la rapidité de leur convergence.

Disons que les erreurs de calcul sont admissibles si elle ne dépasse pas un certain nombre ϵ.S'il existe dans une méthode des estimations qui définissent l'ordre de convergence.

Remarque 1.7. Sous-estimateur: c'est une fonction de borne inférieure.

1.9 Analyse d'intervalle:

L'analyse d'intervalle est un outil développé par Moore en 1966 pour contrôler les erreurs occasionnées par les calculs numériques. Au lieu d'approcher x par un nombre représentable en machine, nous le remplacerons par un intervalle qui le contient. Les bornes de cet intervalle seront des nombre représentables en machine. L'idée de l'analyse d'intervalle est de représenter tous les nombres réels par deux nombres flottants qui l'encadrent. L'analyse d'intervalle a rapidement été utilisée en optimisation, pour la résolution de systèmes linéaires et non linéaires. Grace à celle-ci, de nombreux algorithmes d'optimisation globale ont été développés, l'analyse d'intervalle utilisée ici permet de déterminer l'optimum global ainsi que tous les optimums d'un problème d'optimisation avec ou sans contraintes.

1.9.1 Opérations ensemblistes pures

1.9.2 Inclusion ensembliste

Définition 1.18. : Soit A et B deux sous-ensembles de R. On dit que A est inclus dans B si et seulement si tout élément de A appartient à B. Il vient de ce fait

$A \subset B \Leftrightarrow (\forall x \in A, x \in B)$

Les opérations suivantes s'appliquent sur des sous-ensembles de R^n en général. Elles regroupent l'union, l'intersection, le produit scalaire et la projection.L'union et l'intersection n d'ensemble sont deux notions trés utilisées. Etant donnés A et B deux sous-ensembles de R^n et R^m . Nous avons :

•$A \cup B = \{x \in R^n \mid x \in A \text{ ou } x \in B\}$ *avec* $n = m$

•$A \cap B = \{x \in R^n \mid x \in A \text{ou} x \in B\}$ *avec* $n = m$

•$A \cdot B = \{(x,y)^T \in R^{n+m} \mid x \in A \text{ou} y \in B\}$

•$proj_i(A) = \{x_i \in R \mid \exists (x_1,.....,x_i,.....,x_n)^T \in A\}$

D'une maniére générale , il va de soit que des principes valables pour les sous ensembles , le seront en particulier pour les pavés ainsi que les intervalles.

1.9.3 Intervalles, pavés et sous-pavages:

Cette section a pour objet un rappel des notions sur les intervalles et les pavés. Ces notions constituent des éléments fondamentaux de l'analyse par intervalles.

1.9.4 Définitions et notations

Intervalle

Un intervalle $[x]$ *est un sous-ensemble convexe, borné et fermé de réels.Nous avons*

$$[x] = [x^L, x^U] = \{x \in R^n, x^L \leq x \leq x^U\}$$

où x^L *et* x^U *sont respectivement les bornes inférieures et supérieures de* $[x]$*. L'ensemble des intervalles est noté* $\mathbb{I} = \{x = [x^L, x^U] | x^L, x^U \in R^n \text{et} x^L \leq x^U\}$.

– *La largeur d'un intervalle est donnée par* $w([x]) = x^L - x^U$

– *Point milieu d'un intervalle est donnée par* $m([x]) = \frac{x^U + x^L}{2}$

– *Un intervalle est dég´enéré quand sa largeur est nulle c-à-d un intervalle dont les deux borne sont égales* $x^U = x = x^L$

Lúnion de deux intervalles ne donne pas toujours un intervalle, par exemple l'union des intervalles [1,2]*et*[3,4] *donne pour résultat un ensemble disjoint de deux intervalles. Bien que ce résultat soit intéressant, il rend difficile la manipulation de l'union, le nombre d'intervalles disjoints lors de la résolution des problémes pouvant croitre rapidement .il faut donc introduire une notion d'union différente qui renvoie le plus petit intervalle contenant lénsemble des solutions de l'union. L'intervalle solution est donc léncadrement extérieur de l'union défini par la formule* $[x] \cup [y] = [[x] \cup [y]]$.

Ou l'opérateur [..] renvoie le plus petit intervalle encadrant l'union. Ces opérations sont codées par les formules suivantes pour des intervalles non vides :

$$[x] \cup [y] = [min(x^L; y^L); max(x^U; y^U)]$$

$$[x] \cap [y] = \begin{cases} \phi si(x^U < y^L) \vee (y^U < x^L) \\ [max(x^L; y^L); min(x^U; y^U)] \ sinon \end{cases}$$

Pavé(Boite)

:De la même façon que nous représentons un réel par un intervalle, il est possible de concevoir une structure de compact simple afin de représenter un vecteur à composantes réelles. Un pavé $[X]$ *(ou vecteur n-dimensionnel d'intervalle qui définit léspace n de recherche dans lequel se trouvent les vecteur des inconnues) est un compact de* $\mathbb{R}^n$ *défini par le produit cartésien de n intervalles. On note*

$$[X] = [x_1^U; x_1^L] \times [x_2^U; x_2^L] \times ... \times [x_n^U; x_n^L] = [x_1] \times [x_2] \times ... \times [x_n]$$

$$[X] = ([x_1^U; x_1^L] \times [x_2^U; x_2^L] \times ... \times [x_n^U; x_n^L])^T$$

Pour tout i ∈ 1,2,...,n, l'intervalle $[x_i]$ correspond à la i^{eme} composante de $[X]$. Les caractéristiques du pavé sont décrites comme suit :

1. *La longueur du pavé [X] est donnée par $w([X]) = max_i^n(w[x_i])$*
2. *Le centre de [X] est défini par $m([X]) = (\frac{x_1^U+x_1^L}{2}.....\frac{x_i^U+x_i^L}{2}.....\frac{x_n^U+x_n^L}{2})$*
3. *Le volume de $[X]$ est donné par:*

$$Vol([X]) = \prod_i^n w(x_i) = (x_1^U + x_1^L)(x_2^U + x_2^L).....(x_n^U +_n^L)$$

Sous-pavage

: Un sous pavage est défini par une union de pavés. En particulier, un sous- pavage régulier est constitué d'un ensemble de pavés disjoints ou des pavés qui ne partagent que leurs frontiéres.

Remarque 1.8. C'est cette structure de pavé qui permet d'approximer des vecteurs incer- tains. Chaque composante du vecteur appartient à un intervalle. *Cependant la représentation par pavé implique une perte de précision. Un ensemble S de forme quelconque peut être encadré par un pavé [X]. L'encadrement réalisé contient tous les points de S mais aussi tous les points du pavé n'appartenant pas à S.*

1.10 Arithmétique d'Intervalles

La premiére de ces arithmétiques présentées dans ce rapport est l'arithmétique d'intervalle développé par R.E.Moore.Sa théorie fit une énorme avancée en mathématique appliquée car ce fut la premiére fois que l'on put écrire des algorithmes exacts entiérement

déterministes; c'est à dire que d'une exécution à l'autre on est sûr d'obtenir le même résultat et que ce résultat est prouvé exact mathématiquement à la précision prés.

1.10.1 Principe de l'arithmétique d'intervalles

Le principe consiste à représenter tout réel (ou calcul intermédiaire) par un intervalle de deux nombres flottants représentables en machine et contenant ce réel. Ainsi toutes les erreurs d'arrondi machine, peuvent être automatiquement prises en compte.

Exemple 1.3.

$$\frac{1}{3} = [0.33333333; 0.33333334]$$

$$log(2) = [0.69314718055; 0.69314718056]$$

Tout comme une arithmétique classique, le calcul par intervalles manipule des opérations arithmétiques.

1.10.2 Opération Arithmétiques

Une opération mathématique quelconque o entre deux sous-ensembles A et B de est R^n définie comme suit :

$$AoB = \{xoy | x \in A \wedge y \in B\}$$

Les opérations arithmétiques considérées ici sont stables. Autrement dit, les résultats de ces opérations sont de même type que leurs arguments. Par exemple, le résultat d'une addition ou du produit de deux intervalles est aussi un intervalle. Pour $o \in \{+, -, \cdot, /\}$, le récapitulatif des opérations sur les intervalles [x] et [y] est donné par :

1. $[x] + [y] = [x^L + y^L, x^U + y^U]$
2. $[x] - [y] = [x^L - y^L, x^U - y^U]$
3. - $[y] = [-y^U, -y^L]$
4. $[x] \cdot [y] = [min(x^L \cdot y^L, x^U \cdot y^U, x^L \cdot y^L; x^U \cdot y^L); min(x^L \cdot y^L, x^U \cdot y^U, x^L \cdot y^L; x^U \cdot y^L)]$

   Dans le cas de la division, Nous Avons :

5. $\frac{1}{y}$ =$[\frac{1}{y}|y \in [y]]$

qui représente le plus petit intervalle qui contient lénsemble des réels $\frac{1}{y}$ *pour tout y appartenant à [y].Plus concrétement, il vient :*

$$\frac{1}{[y]} = \begin{cases} R \qquad [y] = [0,0] \\ [\ \frac{1}{y^U},(\frac{1}{y^L}\ ] \quad si \quad 0 \notin [y] \\ [(\frac{1}{y^U},+\infty] \quad si \quad y^L < 0 \quad et \quad y^U > 0 \\ [-\infty,\frac{1}{y^L}\ ] \quad si \quad y^L < 0 \quad et \quad y^U = 0 \\ R\ si\ y^L < 0 \quad et \quad y^U > 0 \end{cases}$$

1.11 Fonctions Usuelles:

On peut également définir des fonctions usuelles prenant X un intervalle de R , nous avons :

1.11.1 Puissance:

$$X^n = \begin{cases} [1,1] \qquad n=1 \\ [(x^L)^n,(x^U)^n] \quad x^L \geq 0 ou 0 \in X\ et\ n\ impair \\ [(x^U)^n,(x^L)^n] \quad x^L \leq 0 \\ [0,max[(x^L)^n,(x^U)^n]] \quad si 0 \in X\ et\ n\ pair \end{cases}$$

Ceci pour tout n appartenant à N.

1.11.2 Logarithme :

Le logarithme étant une fonction strictement croissante sur l'intervalle [0;+1[, nous obtenons :

$$log(x) = (log(x^L); log(x^U)) \quad avec \quad X \subset [0;+1[$$

1.11.3 Racine carrée :

Laracine carée étant une fonction strictement croissante sur l'itervalle]0;+1[, nous obtenouns :

$$\sqrt{x} = [\sqrt{x^L}, \sqrt{x^U}] \quad avec \quad X \subset [0, +\infty[$$

1.11.4 Exponentielle :

L'exponentielle étant une fonction strictement croissante sur R on obtient :

$$\forall X \subset R : exp(X) = \{expx \mid x \in X\} = [exp(x^L); exp(x^U)].$$

1.11.5 valeur absolue :

$$\mid X \mid = \begin{cases} [\ 0, max\{\mid x^L \mid , \mid x^U) \mid\}] & 0 \in X \\ [\ x^U, x^L] & x^L \geq 0 \\ [\mid x^L \mid , \mid x^U \mid] & x^U \leq 0 \end{cases}$$

1.11.6 Extension naturelle et propriétés

Il existe des definitions plus précises d'une fonction explicite (factorable fonctions en anglais) et d'un problème explicite (factorable program en anglais). Mais, par souci de concision, nous avons préféré utiliser des définitions plus intuitives.

Définition 1.19. Soit $f : R^n \to R$,f possède une expression explicite si l'expression analytique de f est connue de façon explicite, c'est-à-dire qu'elle peut s'écrire en n'utilisant que des variables, des fonctions et des opérateurs élémentaires tels que +,-,×,/,$\sqrt{}$, log , exp ,abs, cos , sin , arccos. On dit que f est une fonction explicite.

Définition 1.20. Un problème explicite est un problème dans lequel toutes les fonctions possèdent des expressions explicites et les contraintes n'utilisent que les relations standards ($\leq$, $<$, $\geq$, $>$, $=$).

Définition 1.21. Soit $f : X \subset Rn \rightarrow R$. Une extension naturelle aux intervalles d'une fonction est la réécriture de f en remplaçant toutes les occurrences d'une variable par l'intervalle correspondant et les opérateurs classiques par leur équivalent en arithmétique d'intervalles, elle est notée ENf (x).

Toutes les fonctions explicites possèdent une extension naturelle. Mais, l'évaluation de celleci n'est pas unique. En effet, certaines fonctions peuvent s'écrire sous forme développée ou factorisée.

Il existe donc une extension naturelle pour chaque façon d'écrire la fonction.

Bien entendu, les fonctions explicites ne sont pas les seules à posséder une fonction d'inclusion.

Il est par exemple possible d'écrire des fonctions d'inclusion de fonction comportantdes if..then...else.

Proposition. *L'extension naturelle aux intervalles d'une fonction est une fonction d'inclusion*

Exemple 1.4. *Voici un exemple d'application de l'extension naturelle.*

Les occurrences de x_1 *sont remplacées par [1,2] et les occurrences de* x_2*par [2,6].*

Les calculs sont ensuite effectués en utilisant les formules de l'arithmétique d'intervalles. $\forall x \in X = [1,2] \times [2,6], f(x) = x_1 \times x_2^2 - exp(x_1 + x_2)$

$EN_f(x) = [1,2] \times [2,6]^2 - exp([1,2] + [2,6])$.

$EN_f(x) = [-2976.9579870417284, 51446307861234]$

Ainsi, -2976.9579870417284 *est un minorant et* 51.91446307681234 *un majorant de* f *sur* $[1,2] \times [2,6]$*. L'analyse d'intervalles peut donc se munir de tous les outils de l'arithmétique classique pour calculer des minorants et des majorants d'une fonction explicite sur un intervalle. Malheureusement, les définitions naturelles proposées n'offrent pas les propriètés d'inversibilité et de distributivité.*

Notons que l'inverse de l'addition dans I n'est pas la soustraction et l'inverse de la multiplication n'est pas la division.

De plus, l'ensemble I muni de la loi $\times$ *est sous-distributive.*

Ainsi,$\forall x,y,z \in I, z \times (x+y) \leq z \times x + z \times y$.

Définition 1.22. La matrice Hessienne $H(f(x))$ de la fonction $f(x)$ deux fois differentiable est définie sur X peut être transformée en une matrice d'intervalle $[H(f(x))]$. De même cas pour les polynômes caractéristiques l'ensemble de matrices d´ecrit par $H(f(x))$ contient ceux-là définis par $[H(f(x))]$

$$H(f(x)) = \begin{pmatrix} a_{11} & . & . & . & a_{1n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ a_{n1} & . & . & . & a_{nn} \end{pmatrix} \subseteq [H(f(x))] = \begin{pmatrix} [a_{11}^L, a_{11}^U] & . & . & . & [a_{1n}^L, a_{1n}^U] \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ [a_{n1}^L, a_{n1}^U] & . & . & . & [a_{nn}^L, a_{nn}^U] \end{pmatrix}$$

1.12 La méthode Branch and Bound:

La méthode Branch and Bound est une méthode qui devise le probléme non linéaire donné en plusieurs sous problèmes et qui exploite les informations qu'elle produit durant la résolution de chacun de ces sous problèmes. De nombreuses améliorations ont été proposées pour cette méthode notamment à partir de différentes structures des problèmes à résoudre,et ça en prenant appui sur les outils classiques de la recherche opérationnelle.

Branch and Bound (séparation et évaluation) c'est une technique qui est très utilisée dans le domaine de la recherche opérationnelle pour résoudre les problèmes de la classe NP.

séparation:

La séparation consiste à diviser le problème en un certain nombre de sous-problèmes qui ont chacun leur ensemble de solutions réalisables de telle sorte que tous ces ensembles forment une partition de l'ensemble H. Ainsi, en résolvant tous les sous-problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial. Ce principe de séparation peut être appliqué d'une manière récursive à chacun des sous-ensembles de solutions obtenus, et ceci tant qu'il y a des ensembles

contenant plusieurs solutions. Les ensembles de solutions (et leurs sous-problèmes associés) ainsi construits ont une hiérarchie naturelle en arbre, souvent appelée arbre de recherche ou arbre de décision.

Évaluation:

L'évaluation d'un noeud de l'arbre de recherche a pour but de déterminer l'optimum de l'ensemble des solutions réalisables associé au noeud en question ou, au contraire, de prouver mathématiquement que cet ensemble ne contient pas de solution intéressante pour la résolution du problème. Lorsqu'un tel noeud est identifié dans l'arbre de recherche, il est donc inutile d'effectuer la séparation de son espace de solutions.

1.12.1 Le principe de la méthode de Branch and Bound :

Soit (p) le probl'eme d'optimisation globale

$$(p)\begin{cases} minf(x) \\ x \in H \end{cases}$$

Où :$H \subseteq R^n$.

$f : K \rightarrow R(H \subseteq K \subseteq R^n)f$ continue et non convexe . L'algorithme Branch and Bound consiste à engendrer deux suites convergentes UB_KetLB_K des bornes supérieure et inférieure respectivement de la valeur minimale de la fonction du problème (p).

- UB_K:Upper bound.
- LB_K:Lower bound.

Une relaxation initial R de l'ensemble réalisable H sera définie telle que:$H \subseteq R$. R est convexe ,il peut être un simplex,un rectangle ,un cône,...

A chaque itération K les problèmes de bornes inférieure et supérieure seront résolus sur un nombre fini de sous ensembles de R.On notera ces sous ensembles $R_{K_i} \in I_k où I_k$ est l'ensemble des sous ensembles actifs à l'itération K .Sur chaque sous ensemble R_{K_i}

les bornes inférieures et supérieures LB_K et UB_Kseront calculées par la relaxation de f sur R_{K_i} et la relaxation de min f localement sur le sous ensemble réalisable $R_{K_i} \bigcap H$. Cette méthode utilise la stratégie "le meilleur d'abord".

En effet ,les bornes inférieures supérieures finales pour l'itération K seront données par:

$$\begin{cases} LB_K = minLB_K \\ UB_K = minUB_K \end{cases}$$

respectivement ,et tout sous ensemble sur lequel la borne inférieure dépasse UB_K sera éliminée,car min f ne peut être atteint sur un tel sous ensemble .

Au fait ,cette méthode peut se représenter schématiquement par une arborescence qui a pour racine l'ensemble R ,et pour sommet des sous ensembles RKi qui s'obtiennent par les subdivisions successives ,et deux sommets seront reliés si et seulement si le deuxième sous ensemble est obtenu par la partition directe du premier,et à chaque niveau de l'arborescence crée la borne inférieure et supérieure seront obtenues par l'application d'une recherche locale.

Notons x^* la solution optimal du problème (p)pour ce qui suit .

1.12.2 L'algorithme de base de la méthode Branch and Bound:

Algorithme général:

1- Construire l'ensemble R tel que :$H \subset R$.

2- Posons :$K = 1$,$I_K = R$,fixer $\epsilon > 0$.

3-Construire les problèmes des bornes inférieure et supérieure de min f(x) sur R .

Soient LB_K,UB_K des solution obtenues respectivement .

4-Si: $UB_K - LB_K \leq \epsilon$ donc on s'arrête et on pose:

min $f(x) = UB_K$ et$x^* = x^K \in x : f(x) = UB_K$,$x \in H \cap R$.

5-Sinon: subdiviser I_K en deux sous ensembles R_{K1} et R_{K2} tel que:

$$\bigsqcup_{i=1}^{i=2} R_{Ki} = R \quad et \quad \overline{R}_{K1} \cap \overline{R}_{K2} = 0$$

Où :$\overline{R}$ est l'intérieur de R.

6-construire les problèmes des bornes inférieures et supérieures de min f(x) sur $H \cap R_{Ki}$,$i = 1,2$.Soient :

$$LB_{K_1},UB_{K_1} \;\; et \;\; LB_{K_2} \;\; et \;\; UB_{K_2}$$

les solutions obtenues .

7- Posons :

$$\begin{cases} UB_{K+1} = minUB_{K1},UB_{K2},UB_K \\ LB_{K+1} = minLB_{K1},LB_{K2} \end{cases}$$

8-Posons: $I_K = R_{K1},R_{K2}$.

9-Éliminer de I_K tous sous ensembles R_{K_j},$j = 1,2$ tel que $LB_{K_j} > UB_{K+1} où H \cap R_{K_j} = \emptyset$.

et posons : $I_{K+1} = R_{K_i}*$.

10-Posons : $K = k + 1$ et revenonsà 4.

Notation

R_K: le sous ensemble actuel ; LB_K : la borne inférieure (à la K^{eme} itération); UB_K : la borne supérieure (à la K^{eme} itération) ; x^K : la solution trouvée (à la K^{eme} itération).

1.12.3 La convergence de la méthode

Évidemment si l'algorithme précédent se termine à l'itération j ,alors x^j est la solution optimale et UB_j est la valeur optimale de la fonction objectif, mais en général on ne peut pas garantir ça, c'est-à-dire le fait de s'arrêter après un nombre fini d'itérations, et si l'algorithme est infini, alors il engendre au moins une suite R_K infinie des sous ensembles des subdivisions successives telle que $R_{K+1} \subset R_K$,$K \in N$.

Donc on doit montrer que chaque point d'accumulation de la suite des solutions x^K correspondante est une solution optimale du problème donné.

Le théorème suivant démontre la convergence de l'algorithme de Branch and Bound.

Théorème. *Si pour chaque suite infinie* R_K,$R_{K+1} \subset R_K$,$K \in N$ *des ensembles des partitions successives , les bornes inférieure et supérieure vérifient :*

$$lim_{K\to\infty}(UB_K - LB_K) = lim_{K\to\infty}(UB_K - LB(R_K)) \quad (1)$$

Alors

$$UB = lim_{K\to\infty}UB_K = lim_{K\to\infty}f(x_k) = lim_{K\to\infty}LB_K = LB \quad (2)$$

Et chaque point d'accumulation X^* *de la suite* (X_k) *est une solution optimale de min f(X),*
$x \in H$.

preuve

A l'itération K le sous ensemble R_K *sera choisi à partir de la règle suivante :*

$$LB_{K+1} = minLB_{K1},LB_{K2}$$

de l'algorithme ci-dessus ,à la fin de l'itération $(K-1)$ *et donc :*

$$LB_K = LB(R_K)$$

Soit (X_K) *la suite des solutions optimales engendrées par l'algorithme , comme H est compact , alors* (X_K) *a des points d'accumulations . Soit* X^* *un point d'accumulation de la suite* (X_K) *, donc il existe une sous suite infinie de* (X_K) *qui converge vers* X^* *, et comme f est continue alors :*

$$lim_{K\to\infty}f(X_K) = f(X^*)$$

Posons: $f^* = minf(X),X \in H$, *la suite* (LB_K) *des bornes inférieures est croissante monotone,majorée par* f^*,*donc la suite* $LB = lim_{K\to 1}LB_K$ *existe .*
D'autre part ,la suite (UB_K) *des bornes supérieures est décroissante , minorée par* f^* ,

donc sa limite : $UB = lim_{K\to 1} UB_K$ existe , et on a : $UB_K = f(X_K) \geq f^*$, et ça implique que :

$$LB \leq f^* \leq lim_{K\to\infty} f(X_K) = f(X^*) = UB$$

Et du fait de(1) on peut déduire directement(2) .

Chaque réalisation de l'algorithme Branch and Bound doit donc spécifier :

1. L'ensemble R tel que :$H \subset R$.

2. Les procédures qui donneront les bornes inférieures et supérieures sur les sous-ensembles engendrés par l'algorithme.

3. Les subdivisions successives de R En sous-ensembles. Bien entendu, ça va dépendre de la structure du problème (P).pour cela nous allons étudier chaque cas a part .en se basant sur les trois points précédents, et en montrant la convergence de l'algorithme a chaque fois.

Chapitre 2

La méthode QBB de l'Optimisation Globale

2.1 Introduction

L'optimisation globale est une branche des mathématiques en plein expansion depuis quelques années. Ce domaine étudie dans un cadre générale la question de trouver l'optimum (maximum ou minimum) d'une fonction sous diverses contraintes. L'intérêt de l'étude de cette classe de problème d'optimisation est qu'elle englobe un vaste éventail d'applications réelles.

Le retour au premier plan de l'optimisation globale correspond à un besoin industriel.De nombreuses applications que se soit au niveau de la conception ou de l'exploitation, se ramène à la recherche d'optima qui n'entre pas dans le cadre des hypothèses simplificatrices (convexité, differentiabilité . . . .). Pour cela, il existe plusieurs méthodes algorithmiques qui mènent à la solution sûrement, mais la vitesse de convergence varie selon la méthode utilisée.

En particulier parmi maintes méthodes proposées pour la résolution de tels problèmes nous exposons dans ce chapitre la méthode d'optimisation globale Branch and Bound(QBB).

La méthode ,QBB,d'optimisation globale,pour les problèmes PNL(Programmation Non Linéaire)deux fois différentiable est développée pour opérer avec la structure branch-and-bound et éxige la construction du problème convexe relaxé en se basant

sur les fonctions quadratiques de borne inférieure pour les structures nonconvexes .Avec une division simpliciale éxhaustive pour le domaine S,la fonction régoureuse de sous-estimation est construite pour la fonction générique nonconvexe en vertue de l'analyse de la valeur propre maximale de la matrice Hessienne d'intervalle .
Chaque borne inférieure du problème PNL avec la progression de la division est calculée en utilisant la programmation convexe du problème relaxé obtenu en préservant les termes convexes ou linéaires et en remplaçant les termes concaves par leurs enveloppe convexe,et les termes nonconvexes par leurs fonctions de borne inférieure .
Les proprietés de convergence standard de l'algorithme QBB pour les problèmes nonconvexes d'optimisation globale sont garanties.

2.2 l'Algorithme QBB de l'optimisation Globale:

Les problèmes de l'optimisation nonconvexe peuvent être formulés comme suit:

$$(P)\begin{cases} minf(x) \\ g_i(x) \leq 0 \quad i = 1,2,...,m \\ X \in S^0 \subset \mathbb{R}^n \end{cases}$$

où,f et g_i appartiennent à C^2,l'ensemble des fonctions deux fois differentiables ,et S^0 est un simplex définit par:

$$S^0 = \{X \in R^n : X = \sum_{i=1}^{n+1} \lambda_i V^i, \lambda \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1\}$$

où $V^i \in V \subset R^n$,i=1,2,...,n+1 sont les (n+1) sommets du simplex S^0, et V est l'ensemble de ces sommets . les fonctions f et g_i peuvent prendre des formes simples comme lineaire.
et si une contrainte d'égalité apparait ,elle peut être transformée à deux contraintes d'inégalité équivalentes comme montré au premier chapitre.
Soit D_g un sous-ensemble de R^n définit par:

$$D_g = \{X \in R^n : g_i(x) \leq 0, i = 1,2,...,m\}$$

En général,l'ensemble D_g est nonconvexe et même déconnecté(détaché).nous supposons partout dans ce travail que le problème (P) à une solution optimale.
Pour n'importe quel problème d'optimisation nonconvexe (P) l'algorithme QBB proposé dans ce travail appartient à la class branch and bound.
Durant chaque itération de cet algorithme ,l'étape séparation et l'ètape évaluation doivent terminer simultaniement.
Nous commençons à développer cet algorithme avec les opérations de bese dont on aura besoin dans ce travail tel que:

2.2.1 La partition simpliciale:

Pour la procedure de séparation ,le simplex S^0 va être divisé en sous ensembles distincts en utilisant la partition simpliciale souvent utilisée dans les algorithmes de l'optimisation Globale.Pour ce genre de séparation ,il est simple à verifier que pour tout $i \in I$, où I est l'ensemble des sommets de S^0, les points V^1,...,V^{i-1},U,V^{i+1},...,V^{n+1} sont des sommets du simplex $S_i \subset S$, S est le courant simplex, et que:

$$S_i \bigcap S_j = \emptyset, \forall j \neq i; \bigcup_{i \in I} S_i = S$$

Alors les simplexes S_i,$i \in I$,forment une subdivision du simplex S par U.
Notons que le sous-simplex S_i fait référence à S.
Cette partition est adéquate lorsqu'elle est appliquée en au moin deux membres si et seulement si U ne coïncide pas avec les V^i.
Un cas spécial est la bissection où U est un point du plus long bord(distance)du simplex S,par exemple $U \in [V^m, V^n]$:

$$\| V^m - V^n \| = \max_{i<j} \{ \| V^i - V^j \| \}, i,j = 1,...,n+1$$

où $\| \cdot \|$ représente n'importe quelle norme donnée de R^n,et $U = aV^m + (1-a)V^n$ avec$0 < a \leq \frac{1}{2}$.

il faut noter ici que le simplex V est divisé en deux sous-simplex tel que la proportion du volume du plus petit sous-simplex de S est égale à a.

***Z**hu et **I**noue(2001) ont utilisé une méthode de bissection éxacte en fixant a égal à* $\frac{1}{2}$*.Evidemment on aura* $S_1 \supset S_2 \supset ... \supset S_k \supset ...$*,le diamètre du simplex* S_k *noté* $\delta(S_k)$*,la longueur du plus long bord de* S_k*,va diminuer d'une manière monotone.*
Pour la preuve de la convergence de l'algorithme branch and bound ,le concept le plus utilisé est l'exhaustivité du processus partition (Horst et al.,1995).
Une séquence d'ensembles S^j *engendrées par la partition ,tel que* $S^j \supset S^{j+1}$,$\forall j$*est dite exhaustive si* S^j *rétrécé pour devenir un point c-à-d :*

$$\bigcap_{j=1}^{\infty} S^j = \{X\}$$

Un processus de partition dans l'algorithme branch and bound est appelé exhaustif si chaque séquence d'ensemble de partition générée partout dans l'algorithme est exhaustive.
***K**unno et al.(1997) ont prouvé que la bissection simplicialle exacte montionnée en haut est exhaustive quand* $\delta(S_k) \to 0$ *avec* $k \to +\infty$

2.2.2 La fonction de sous-estimation quadratique pour les structures Nonconvexes en général:

Dans l'étape évaluation de l'algorithme branch and bound la borne inférieure est toujours obtenue par la construction du problème de sous-estimation convexe pour le problème original (P),et résoudre le programme non linéaire convexe pour l'optimalité globale.
Pour le courant simplex donné par:

$$S = \{x \in R^n : x = \sum_{i=1}^{n+1} \lambda_i V^i, \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1\} \qquad (1)$$

où $V^i \in V \subset R^n$*, i=1,2,...,n+1 sont les (n+1) sommets du courant simplex S,et V est l'ensemble de ces sommets.*

Alors nous projetons de calculer la borne inferieure $\mu(S)$ pour la fonction objectif f sur $S \cap D_g$.Autrement dit, nous calculons la borne inférieure pour la valeur optimale pour le problème:

$$(P(S)) \begin{cases} \min_x f(x) \\ g_i(x) \leq 0 \quad i = 1,2,...,m \\ X \in S \subset \mathbb{R}^n \end{cases}$$

Comme montionné au dessus ,f et g_i sont des fonctions nonconvexes appartenant à C^2,alors l'idée principale pour le calcul de la borne inférieure $\mu(S)$ est de construir pour le problème (P(S)) un problème convexe par le remplacement de toutes ses fonctions nonconvexes par leurs fonctions de sous-estimation respectives ,et ensuite résoudre le problème convexe relaxé qui en découle. Et pour arriver à ce but,nous voyons la définition suivante:

Définition 2.1. pour toute fonction nonconvexe donnée $f(X) : S \to R, X \in S \subseteq R^n$ appartenant à C^2 ,la fonction quadratique suivante est définie par:

$$F(X) = \sum_{i=1}^{n} a_i X_i^2 + \sum_{i=1}^{n} b_i X_i + c \qquad (2)$$

où ,$X \in S \subseteq R^n$ et $F(X) = f(X)$ soumise à tout les sommets de S.les a_i sont des scalaires positifs ou nulls et assez grand tel que

$$F(X) \leq f(X), \forall X \in S$$

Le théorème suivant(**Z**hu et **I**noue,2001) peut être utilisé pour assurer que F(X) est un sous-estimateur régoureux pour f(X) c-à-d

$$F(X) \leq f(X), \forall X \in S$$

Théorème. ***2.1.****F(X) définie dans la Définition 2.1 est un sous-estimateur convexe pour $f(X)$ si la difference entre elles c-à-d : $D(X) = F(X) - f(X)$ est une fonction convexe.*

Preuve:

Supposons que X^1 et X^2 deux points quelconques du simplex S défini par l'équation

(1) ,alors il existe $2(n+1)$ *valeurs réelle.,*α_i,$\beta_i \in R$ *qui verifient* $0 \leq \alpha_i$,$\beta_i \leq 1$
et

$$\sum_{i=1}^{n+1} \alpha_i = 1, \sum_{i=1}^{n+1} \beta_i = 1$$

tel que: $X^1 = \sum_{i=1}^{n+1} \alpha_i V^i$,$X^2 = \sum_{i=1}^{n+1} \beta_i V^i$.*quand* $D(X) = F(X) - f(X)$ *est une fonction convexe ,nous avons l'inégalité suivante d'après la définition de la fonction convexe:*

$$D(\lambda X^1 + (1-\lambda)X^2) \leq \lambda D(X^1) + (1-\lambda)D(X^2)$$

où,$\lambda \in R$,*et*$0 \leq \lambda \leq 1$. *et d'après l'équation précédente et en vertue d'inégalité de* ***J****ensen(****R****ockfellar,1972)nous avons:*

$$D(\lambda X^1 + (1-\lambda)X^2) \leq \lambda D(\sum_{i=1}^{n+1} \alpha_i V^i) + (1-\lambda) D(\sum_{i=1}^{n+1} \beta_i V^i)$$

$$\leq \lambda \sum_{i=1}^{n+1} \alpha_i D(V^i) + (1-\lambda) \sum_{i=1}^{n+1} \beta_i D(V^i) \qquad (*)$$

pour les sommets de S d'après la définition 2.1
On sait que $F(x) = f(x)$ *c-à-d* $F(V^i) = f(V^i)$.*alors* $D(V^i) = 0$,*i=1,...,n+1.et d'après l'inégalité* $(*)$ *nous avons:*

$$D(\lambda X^1 + (1-\lambda)X^2) \leq 0$$

et puisqueX^1 *et*X^2 *sont deux point quelconque du simplex S, alors*
$X = (\lambda X^1 + (1-\lambda)X^2)$ *est aussi un point de ce simplex.*
donc $D(X) \leq 0$ *ce qui veut dire* $F(x) \leq f(x)$,$\forall X \in S$. *cela signifie que* $F(X)$ *et un sous-estimateur rigoureux de la fonction nonconvexe* $f(X)$, $\forall X \in S$.
D(X) est convexe si et seulement si sa matrice Hessienne $H_D(X)$ *est semi-définie positive dans S. une condition de convexité utile est tirée en ayant* $H_D(X)$ *reliée directement a la matrice Hessienne* $H_f(X)$ *de f(X),*$X \in S$ *par l'équation suivante:*

$$H_D(X) = 2\triangle - H_f(X)$$

Où $\triangle$ est la matrice diagonale dont les éléments diagonaux sont les a_i définis dans la définition 2.1.

*Analogue à "diagonal shift matrix" définie par **A**djiman et al.(1998),$\triangle$ est jugée ici comme la matrice de sous-estimation diagonale, tant que ses paramètres garantissent que $F(X)$ est un sous-estimateur rigoureux pour $f(X)$.*

Le théorème suivant va aider à garantir que $D(X)$ définie dans le théorème précedent est convexe.

Théorème. ***2.2**: D(X),comme définie dans le théorème 2.1,est convexe si et seulement si $2\triangle - H_f(x) = 2diag(a_i) - H_f(X)$ et semi-définie positive pour tout $X \in S$*

pour simplifier le calcul du paramètre, le sous-estimateur F(X) et reformulé en utilisant une seule valeur positive a,et on obtient:

$$F(X) = a\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} b_i x_i + c \qquad (3)$$

Alors, tout les éléments diagonaux de la matrice de sous-estimation diagonale $\triangle$ sont par conséquant égaux au coefficient quadratique a défini par l'équation (3).

*En basant sur le théorème 2.2,le théorème suivant (**Z**hu et **I**none,2001) similaire a celui fait par (**M**aranas et **F**loudas,1992) et (**A**djiman et al,1998) peut alors être utilisé pour assurer que $F(X)$ définie par les équations (2)ou (3) est en effet un sous-estimateur rigoureux pour f(X).*

Théorème. ***.2.3**:$F(X)$ définie par l'équation (2) est un sous-estimateur rigoureux convexe de $f(X)$ ssi:*

$$a_i \geq max\{0, \frac{1}{2}max\{H_{ii}^f + \sum_{j\neq i} \mid H_{ij}^f(X) \mid\}\} \qquad (4)$$

où,si $F(X)$ est celle définie par l'équation (3),nous avons:

$$a \geq max\{0, \frac{1}{2}max_{i,X\in S}\lambda_i(X)\} \qquad (5)$$

où les $\lambda_i(X)$ sont les valeurs propres de $H_f(X)$,la matrice Hessienne de la fonction nonconvexe f(X),$X \in S$.

preuve:comme la matrice Hessienne $H_f(X)$ de f(X) est symétrique, et ses valeurs propres, sont des valeurs réelles. d'après les théorèmes 2,1 et 2,2 , F(X) définie dans l'équation (2) est un sous-estimateur convexe (ou linéaire) de f(X) si et seulement si D(X) définie dans le théorèmes 2,1 est convexe.

D(X) est convexe si pour tout $X \in S$, toutes les valeurs propres $\lambda_i^D(X)$ de D(X) sont positives ou nulles.

Dans le deuxieme cas, quand on utilise le coeficient quadratique uniforme a, les valeurs propres de D(X) peuvent être relliées directement à celles de f(X),la précédente condition c-à-d l'équation (5), et équivalente à éxiger que le minimum des valeurs propres de D(X) soit positive ou nulle c-à-d:

$$min_{i,X\in S}\lambda_i^D \geq 0$$

après substitution $\lambda_i^D(X) = 2a - \lambda_i(X)$ et a

$$max\{0,\frac{1}{2}max_{i,X\in S}\lambda_i(X)\}$$

nous avons:

$min_{i,X\in S}\lambda_i^D(X) = min_{i,X\in S}(2a - \lambda_i(X))$

$$\geq min_{i,X\in S}\{max\{0,max_{i,X\in S}\lambda_i(X)\} - \lambda_i(X)\}$$

$$\geq min_{i,X\in S}\{max_{i,X\in S}\{0,\lambda_i(X)\} - \lambda_i(X)\}$$

*Evidemment , $max_{i,X\in S}\{0,\lambda_i(X)\} - \lambda_i(X) \geq 0$ en considèrant les deux cas pour le signe de $\lambda_i(X)$,donc $min_{X\in S}\lambda_i^D(X) \geq 0$ c'est D(X) est convexe pour $X \in S$.Cela signifie que F(X) définie par l'équation (3) est un sons-estimateur rigoureux de f(X) dans le deuxième cas,en vertue du théorème de **G**erschgorin (**G**erschgorin,1931), la borne inférieure des valeurs propres de la matrice symétrique $A = (a_{ij})$ est donnée par:*

$$min\lambda_i \geq min(a_{ii} - \sum_{i\neq j} \mid a_{ij} \mid)$$

Après avoir substituée l'équation (4) à $H_D(X) = 2\Delta - H_f(X)$ *sa bornne inférieure peut être donnée comme suit:*

$$min_{X\in S}\lambda_i^D(X) \geq min_{X\in S}(2max\{0,\frac{1}{2}max_{X\in S}\{H_{ii}^f(X)+\sum_{i\neq j} \mid H_{ij}^f(X) \mid\}\}-H_{ii}^f(X)-\sum_{i\neq j} \mid H_{ij}^f(X) \mid)$$

$$\geq min_{X\in S}(max\{0,\frac{1}{2}max_{X\in S}\{H_{ii}^f(X)+\sum_{i\neq j} \mid H_{ij}^f(X) \mid\}\}-\{H_{ii}^f(X)+\sum_{i\neq j} \mid H_{ij}^f(X) \mid\})$$

$$\geq min_{X\in S}(max_{X\in S}\{0,\{H_{ii}^f(X)+\sum_{i\neq j} \mid H_{ij}^f(X) \mid\}\}-\{H_{ii}^f(X)+\sum_{i\neq j} \mid H_{ij}^f(f) \mid\}).$$

Evidemment $min_{X\in S}\lambda_i^D(X) \geq 0$ *en cosidèrant les deux cas pour le singe de* $H_{ii}^f(X) + \sum_{i\neq j} \mid H_{ij}^f(X) \mid$*.donc D(X) est convexe pour* $X \in S$ *par conséquant,*$F(X)$ *définie par (2) est un sous-estimateur rigoureux de f(X).*

la proposition suivante affirme le rapport entre les coefficients linéaires et constant de F(X) et ces coefficients quadratiques, et ces derniers peuveut être déterminés uniquement par le dernier et tout les sommets du courant simplex S.

Proposition. ***2.1:****les coefficients linéaires et constant de* $F(X)$ *définis par l'équation (2) ou (3),c-à-d les* b_i *et c peuvent êtres donnés par les coefficients quadratique* a_i *connus dans le théorème et le simplex courant.*

preuve: vue la définition 2.1, on sait que $F(X) = f(X)$*sur tout les sommets de S,alors:*

$$V^{kT}\Delta V^k + b^T V^k + c = f(V^k),k = 1,....,n+1$$

où $\Delta \in R^{n\times n}$ *est la matrice diagonale de sous-estimation dont les éléments diagonaux sont les* a_i *définis dans l'équation (2) ou (1).*$b \in R^n$ *est le vecteur des coefficients linéaires qui est composé par les* b_i *définis aussi dans (2) ou (1),et c et un scalaire.*

$$b^T V^k + c = f(V^k) - V^{kT}\Delta V,k = 1,....,n+1$$

le vecteur $b \in R^n$ *et augmenté à* $(b,c) \in R^{n+1}$ *,pour inclur c. de la même façon, la matrice* $V \in R^{(n+1)\times n}$*est augmentée à* $(V,1) \in R^{(n+1)(n+1)}$*, ou 1 est la matrice colonne unité de* R^N*.* $(V,1) \in R^{(n+1)(n+1)}$*est une matrice carrée riégulière. Alors on a:*

$$(b,c)^T = (V,1)^{-1}[f(V) - V^T\Delta V]$$

où, $[f(V) - V^T \Delta V] \in R^{n+1}$*,est un vecteur colonne pour les (n+1) sommets du courant simplex.*

En vertue de cette équation, c'est évident que les coefficients linéaires et constants sont déterminés uniquement par les coefficients quadratique et le courant simplex.

En remplaçant toutes les fonctions nonconvexes dans le problème (P(S)) par leurs fonctions quadratiques de sous-estimation correspondente définies par l'équation (3), nous avons le problème (QP(S))relaxé de programmation convexe suivant:

$$(QP(S)) = \begin{cases} min\ F(X) \\ G_i(X) \leq 0 i = 1,2...,m \\ x \in S \subset R^n \end{cases}$$

où

$$F(X) = \sum_{i=1}^{n} a_i^f x_i^2 + \sum_{i=1}^{n} b_i^f x_i + c^f$$

$$G_i(X) = \sum_{i=1}^{n} a_i^{g_i} x_i^2 + \sum_{i=1}^{n} b_i^{g_i} x_i + c^{g_i}, j = 1,..,n$$

soit D_G *l'ensemble de* R^n *défini par:*

$$D_G = \{x \in R^n : G_i(X) \leq 0, i = 1,2,...,m\}.$$

Evidemment, l'ensemble D_G *est convexe et compact.*

Alors, le problème (QP(S)) à une solution optimale d'après le théorème de ***W****eiestrass.*

le thèorème suivant confirme que la solution optimale F^* *du problème de programmation convexe est une borne inférieure du problème primal (P(S)).*

Théorème. ***2.4:****pour chaque simplex* $S=\{x \in R^n : x = \sum_{i=1}^{n+1} \lambda_i V^i, \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1\} \subseteq S^0$*,la borne inférieure* $\mu(S)$ *de f sur* $S \cap D_g$ *peut être calculée par* $\mu(S) = F^*$*, où* F^* *est la solution optimale de F sur* $S \cap D_G$*.*

preuve:

(a) nous montrons $S \in D_g \subseteq S \in D_G$*. quand* $G_i(X)$ *est un sous-estimateur convexe*

de $g_i(X)$,c-à-d $G_i(X) \leq g_i(X)$, nous avons $G_i(X) \leq g_i(X) \leq 0$ pour toutX $\in$ D_g, alors $X \in D_G$. finalement nous avons $S \cap D_g \subseteq S \cap D_G$ en notant $D_G \subseteq D_G$.

(b) en vertue de $F(X) \leq f(X)$ pour tout $X \in S \cap D_g \subseteq S \cap D_G$ nous avons:

$$F^* = min\{F(X), X \in S \cap D_G\} \leq F(X) pour X \in S \cap D_G \leq f(X) pour X \in S \cap D_g.$$

et cele montre que $\mu(S) = F^*$ est une borne inférieure de f sur $S \cap D_g$.

La proposition suivante montre que la borne inférieure obtenue par le théorème 2,4 est toujours bornée de dessous.

Proposition. ***2.2:***

(a) soit S^1 et S^2 deux simplexes tel que $S^2 \subset S^1$.

Alors $\mu(S^2) \geq \mu(S^1)$.

(b) si le problème (P) à une solution réalisable, alors $\mu(S) > -\infty$ pour tout $S \subseteq S^0$

preuve:

(a) soient $F^1(X)$ et $F^2(X)$ des fonction quadratique de sous estimation pour f(X) générées dans S^1 et S^2,tel que $S^2 \subset S^1$,respectivement.

Alors nous allons montres $F^1(X) \leq F^2(X)$ pour $X \in S^2$.

avec l'équation (2), nous avons :

$$F^1(X) = \sum_{i=1}^{n} a_i^1 x_i^2 + \sum_{i=1}^{n} b_i^1 x_i + c^1$$

$$F^2(X) = \sum_{i=1}^{n} a_i^2 x_i^2 + \sum_{i=1}^{n} b_i^2 x_i + c^2$$

Alors,

$$F^1(X) - F^2(X) = \sum_{i=1}^{n} (a_i^1 - a_i^2) x_i^2 + \sum_{i=1}^{n} (b_i^1 - b_i^2) x_i + c^1 - c^2$$

quand $S^2 \subset S^1$, nous avons $max_{X \in S^1} \lambda_i(X) \geq max_{X \in S^2} \lambda_i(X)$

et

$$max_{X \in S^1}\{H_{ii}^f(X) + \sum_{j \neq i} \mid H_{ij}^f(X) \mid\} \geq max_{X \in S^2}\{H_{ii}^f(X) + \sum_{j \neq i} \mid H_{ij}^f(X) \mid\}$$

En vertue du théorème 2.3, nous avons $a_i^1 \geq a_i^2$.*Alors la fonction* $D_F(X) = F^1(X) - F^2(X)$ *est convexe.*
quand $F^1(X)$ *est la fonction de sous-estimation de f(X)*
Alors nous avons $F^1(X) \leq F(X)$*pour tout* $X \in S^1$.
D'après la définition 2.1,on sait que $F^2(V_i^2) = f(V_i^2)$ *pour tout les sommets du simplex* S^2, *c-a-d* V_i^2,*pour i=1,2,...,n+1.*
quand $S^2 \subset S^1$, *on a* $F^1(V_i^2) \leq F^2(V_i^2)$ *pour i=1,2,...,n+1.*
cela signifie:

$$D(V_i^2) = F^1(V_i^2) - F^2(V_i^2) \leq 0,i = 1,2,...,n+1.$$

pour tout $X \in S^2$, *et* $X = \sum_{i=1}^n \lambda_i V_i^2$ *avec* $\lambda_i \geq 0 \forall i$ *et* $\sum_{i=1}^n \lambda_i = 1$, *par les caracteristiques de la fonction convexe de D(x), nous avons:*

$$D(X) = D(\sum_{i=1}^{n} \lambda_i V_i^2) \leq \sum_{i=1}^{n} D\lambda_i(V_i^2) \leq 0.$$

Alors, on obtient $F^1(X) \leq F^2(X)$ *pour* $X \in S^2$.
Avec la même manière, pour $X \in S^2$ *nous avons*

$$G_i^1 \leq G_i^2, \quad i = 1,2...,m$$

Alors nous avons: $D_G^1 = \{X \in R^n : G_i^1(X) \leq 0,i = 1,2...,m\} \cap S^1 \supseteq D_G^2 = \{X \in R^n : G_i^2(X) \leq 0,i = 1,2...,m\} \cap S^2$ *quand* $D_G^2 \cap S^2 \subset D_G^1 \cap S^1$, *finalement nous avons:*
$\mu(S^2) = min\{F^2(X) : X \in D_G^2 \cap S^2\} \geq min\{F^1(X) : X \in D_G^1 \cap S^1\} = \mu(S^1)$

(b) *quand le problème est supposé réalisable, alors par (a), nous avons besoin seulement de montrer que* $\mu(S^0) > +\infty$. *alors le problème relaxé du problème (P(S)) sur le simplex initial* S^0 *c-à-d* $(QP(S^0))$ *est convexe.*
Alors ce problème à une solution optimale, ce qui implique que $\mu(S^0) > -\infty$

2.2.3 la borne supérieure:

pour le simplex S, si la valeur de la fonction du problème (P(S)) n'est pas bornée en dessus,c-à-d: $\mu(S) = +\infty$, alors $f(X) = +\infty$,pour tout $x \in S$
dans ce cas,l'ensemble de partition S peut être enlevé de considèration suplémentaire.Autrement, on essaie de trouver un ensemble F(S) de solution réalisable dans S et les utilisées pour calculer une borne supérieure de la valeur optimale de problème (QP(X)). durant l'algorithme, de plus en plus en trouvas des solution réalisable, alors la borne supérieure de la valeur optimale peut être améliorée itérativement.

2.3 le calcul des coefficients quadratique en utilisant l'analyse d'intervalles:

*Pour les fonctions nonconvexes, les éléments de leurs matrices Hessienne $H_f(X)$ peuvent être nonlineaires , alors la dirivation de la matrice de de sous-estimation diagonale, c-à-d Δ soit valable sur le simplex entier est une tache assez difficile, cependant la satisfaction de la condition de convexité du théorème 2.2 est essentielle pour préserver la garantie que F(X) définie par l'équation (2) est un sous-estimateur rigoureux pour la fonction non convexe f(X), la complexité qui survient de la présence des variables dans la condition de convexité peut devenir facile en utilissant la transformation éxacte de la matrice Hessienne X- dependent c-à-d $H_f(X)$, vers une matrice Hessienne d'intervalle $[H_f(X)]$ (**N**eumaier,1990,1996;**H**ansen,1992; **K**earfott,1996; **A**djiman et al;1998 a,b), tel que $H_f(X) \subseteq [H_f]$, $\forall X \in S$.le courant simplex S peut être remplacé par une boite (Intervalle), décrite par $[X^L,X^U]$. X^L et X^U sont la borne inférieure et la borne supérieure respectevement du simplex S.*
Evidemment, $S \subseteq [X^L,X^U]$. Alors la matrice Hessienne d'intervalle peut être calculé dans la boite précédente c-à-d $[X^L,X^U]$sans affécter la rigoureusité de l'estimation de la matrice Hessienne, $H_f(X)$ dans le courant simplex.
Les éléments de la matrice Hessienne originale $H_f(X)$ sont traités comme

indépendents en calculant leurs intervalles d'après l'arithmitique d'intervalle.
*le théorème suivant est semblable à celui d'**A**djiman et al.(1998). ce dernier nos dira comment utiliser la matrice Hessienne d'intervalle pour calculer les paramètres quadratiques a_i définis par l'èquation (2).*

Théorème. ***4.1:****considirous la fonction nonconvexe f(X) continue dirivable deux fois et sa matrice Hessienne $H_f(X)$. soit $D(X) = F(X) - f(X)$ définie dans le théorème 2.1 et F(X) définie par l'équation (2). soit $[H_f]$ une matrice symétrique d'intervalle tel que $H_f(X) \subseteq [H_f]$, $\forall X \in S$, si la matrice $[H_D]$ définie par $[H_D] = 2\Delta - [H_f] = 2diag(a_i) - [H_f]$ est semi-définie positive, alors D(X) est convexe sur le simplex courant qui est dans $[X^L, X^U]$*

On a $[H_f] \supseteq H_f(X)$, alors la borne supérieure de la valeur propre maximale de $[H_f(X)]$ peut être calculée plus facilement en utilisant l'arithmitique d'intervalle. Alors, l'équation (5) vue dans le théorème 2.3 peut être remplacée par la fonction d'intevalle suivante, pour générer une seule valeur a qui stisfait la condition suffisante afin que F(X) reste un sous-estimateur rigoureux de f(X):

$$a \geq max\{0, \frac{1}{2}\lambda_{max}([H_f])\} \qquad (6)$$

où, $\lambda_{max}([H_f])$ est la valeur propre maximale de la matrice d'intervalle $[H_f(X)]$. Pour le cas non-constant, l'équation (4) peut être transformée vers l'équation suivante en remplaçant la matrice Hessienne avec sa matrice d'intervalle:

$$a_i \geq max\{0, \frac{1}{2}\{\overline{H}^f_{ii} + \sum_{j \neq i} \mid H^f \mid_{ij}\}\} \qquad (7)$$

où

$$\mid H^f \mid_{ij} = max\{\mid \underline{H}^f_{ij} \mid , \mid \overline{H}^f_{ij}\}$$

Evidemment nous avons

$$\overline{H}^f_{ii} + \sum_{j \neq i} \mid H^f \mid_{ij} \geq [H^f_{ii}] + \sum_{j \neq i} H^f_{ij}].$$

2.3.1 Extention de théorème de Gerschgorin pour le cas uniforme

pour une matrice symétrique des réels $A = (a_{ij})$, *le théorème précident de* ***G****erschgorin (1931), montre que ses valeurs propres sont bornées, tel que* λ_{max}, *par tout ses éléments:*

$$\lambda_{max} = max_i(a_{ii} + \sum_{j \neq i} \mid a_{ij} \mid).$$

dans ce travial, une extension de ce théorème est présentée pour matrices d'intervalle, semblable a celui fait par ***A****djiman et al.(1998) pour l'analyse de valeur propre minimale, dans ce théorème qui suit:*

Théorème. ***4.2:****pour une matrice d'intervalle* $[A] = (\underline{a}_{ij}, \overline{a}_{ij})$,*une borne supérieure pour la valeur propre maximale est donnée par:*

$$\lambda_{max} = max_i[\overline{a}_{ii} + \sum_{j \neq i} max(\mid \underline{a}_{ij} \mid, \mid \overline{a}_{ij} \mid)].$$

preuve: par définition de la matrice d'intervalle, $\lambda_{max}([A]) \geq max_{A \in [A]} \lambda_{max}(A)$,*par conséquant*

$$\begin{aligned} \lambda_{max}([A]) &= max_{A \in [A]} max_i(a_{ii} + \sum_{j \neq i} \mid a_{ij} \mid) \\ &= max_i[max_{A \in [A]}(a_{ii}) + max_{A \in [A]}(\sum_{j \neq i} \mid a_{ij} \mid)] \\ &= max_i[\overline{a}_{ii} + \sum_{j \neq i} max(\mid \underline{a}_{ij} \mid, \mid \overline{a}_{ij} \mid)]. \end{aligned}$$

semblable à cela fait par ***A****djiman et al.(1998) dans leurs algorithme* αBB *pour l'estimation de de la valeur propre minimale d'une matrice d'intervalle. la méthode d'estimation de la borne inférieure de la valeur propre minimale d'une matrice d'intervalle proposée par* ***A****djiman et al. est donnée par:*

$$\lambda_{min} = min_i[\underline{a}_{ii} - \sum_{j \neq i} max(\mid \underline{a}_{ij} \mid, \mid \overline{a}_{ij} \mid)].$$

Si le λ_{min} *calculé de la matrice Hessienne d'intervalle* $[H_f]$ *de la fonction nonconvexe* $f^{NC}(X)$ *est positif ou nul,alors c'est certain que f(X) est convexe sur le courant*

simplex. Donc le plus sérré sous-estimateur ou enveloppe convexe de f(X) est elle même, et elle est utilisée comme sous-estimateur de f(X) sur le courant simplex. Auteremnt,λ_{max} de la matrice Hessienne d'intervalle $[H_f]$ de la fonction nonconvexe $f^{NC}(X)$ est calculé suivant le théorème 4.2, et la fonction quadratique convexe nomée sous-estimateur est construite suivant le théorème 2.3 et proposition 2.2.

2.3.2 sous-estimateur pour les fonctions linéaires convexes:

pour les fonction convexes, notées par $f^C(X)$ ou $f^L(X)$, c'est évident que leurs enveloppes sont elles même,elles vont présérver leurs forme originale dans le sous estimateur final que ce soit la fonction objectif ou bien les contraintes.

2.3.3 sous-estimateur pour les fonctions concaves:

Pour les fonctions concaves, notées $f^{CA}(X)$, dont les valeurs propres sont toutes négatives ou nulles c-à-d $\lambda_{i,X\in S}(X)$. Alors le coefficient quadratique de son sous-estimateur défini dans l'équation (2) est zéro d'après le théorème 2.3, donc la borne inférieure de la fonction concave sur le simplex courant est une fonction linéaire dont les ceofficients linéaires et constant sont donnés par proposition 2.1. cette conclusion est aussi conforme avec celle présentée par ***H****orst et al.(1995). Cela signifie que la borne construite par l'équation (2) et équivalente à l'enveloppe convexe de la fonction concave sur le simplex, qui peut être construite comme une fonction donnée dans la proposition suivante:*

Proposition. ***4.3****: soit S le simplex généré par les sommets V^1,V^2,...,V^{n+1}, c-à-d:*

$$S = \{X \in R^n : X = \sum_{i=1}^{n+1} \lambda_i V^i, \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1\}$$

et soit $f^{CA}(X)$ une fonction concave définie sur S.
Alors l'enveloppe convexe de $f^{CA}(X)$ sur S et la fonction $L^{CA}(X) = b^T X + c$ qui est détérminée uniquement par le système d'équations linéaires
$f^{CA}(V^i) = b^T V^i + c$ pour $i = 1,...,n+1$.

2.3.4 sous-estimateur pour les fonctions Quadratique en général:

*Pour une fonction bilinéaire, qui à le terme X_iX_j, tel que $i \neq j$,**McCormick(1976)** et **AlKhayyal** et **Falk** (1983) ont présentée la borne inférieure convexe la plus serrée,c-à-d l'enveloppe convexe, sur le domaine rèctangulaire $[X_i^L,X_i^U] \times [X_j^L,X_j^U]$. Ici, une fonction convexe de sous estimation est derivée facilement pour n'importe quelle fonction quadratique, quand les valeurs propres de sa matrice Hessienne sont connues.*
la fonction quadratique est présentée comme:

$$f(X) = X^TQX + q^TX$$

Evidemment, la fonction bilinéaire précedente est un cas particulier pour ce genre de fonction. quand $H_f(X) = Q$, nous avons la matrice diagonale de sous-estimation, Δ, construite en se basant sur le théorème 2.3, tel que:

$$a = max_i\{0,\frac{1}{2}\lambda_i^Q\}$$

pour le cas uniforme, et pour le cas nonuniforme on obtient:

$$a_i = max\{0,\frac{1}{2}(Q_{ii} + \sum_{j\neq i} \mid Q_{ij} \mid)\}$$

Alors, nous avons la fonction quadratique de sous-estimation

$$F(X) = X^T\Delta X + b^TX + c$$

où, les coefficients lineaire et constant, c-à-d (b,c), peuvent être détérminés uniquement par la proposition 2.2.1

2.3.5 Le sous-estimateur de QBB Généralisé

Il faut noter ici, que le problème relaxé de programmation convexe (QP(S)) contient non seulement les fonctions quadratique de sous-estimation pour les termes nonconvexe, mais aussi les fonctions convexes qui ne sont pas necessairement transformées

en sous-estimateurs quadratiques. Alors, la stratégie de sous-estimation finale pour le problème relaxé (QP(S)) peut être revisée légèrement vers la fonction de programmation convexe suivante tel:

$$(QP(S)^{'}) \begin{cases} min\ F^{'}(X) \\ G_i^{'}(X) \leq 0 \qquad\qquad i = 1,2,...,m \\ x \in S \subset R^n \end{cases}$$

où,

$$F^{'}(X) = f^L(X) + f^C(X) + L_f^{CA}(X) + F^{NC}(X)$$

$$G_i^{'}(X) = g_i^L(X) + g_i^C(X) + L_{g_i}^{CA}(X) + G_i^{NC}(X) \qquad i = 1,2,...,m$$

Et $f^L(X)$, $f^C(X)$, $L_f^{CA}(X)$,$g_i^L(X)$,$g_i^C(X)$,$L_{g_i}^{CA}(X)$ représentent les termes linéaires, les termes convexes, et les fonctions linéaires de sous-estimation pour les termes concaves dans la fonction objectif et les contraintes, respectivement.
Quand à $F^{NC}(X)$ et $G_i^{NC}(X)$ représentent les fonctions quadratiques convexes de sous-estimation pour les termes nonconvexes. Comparé avec le problème relaxé (QP(S)), le problème relaxé $(QP(S)^{'})$ contient non seulement les fonctions quadratiques, mais aussi les termes convexes du problème original. mais il faut noter ici, qu'un tel genre de relaxation n'affecte pas la monotonicite du sous-estimateur convexe donné dans proposition 2.2, donc il gardera aussi les convergence algorithmique présentées dans la section suivantes.

2.4 les étapes de l'algorithme QBB:

Au debut de cette section, le problème (P) est formulé sur un simplex initial S^0. Cependant, le problème pratique ne fournit pas nécessairement un tel simplex, alors une méthode d'approximation externe commode pour obtenir ce simplex est présentée ici sur une base plus générale, à condition que les contraintes linéaires peuveut être separées de celles avec les termes noncovexes, et les bornes inférieures et supérieures des variables indépendantes sont connues a priori d'une manière physique, comme:

$$(P') \begin{cases} min\ f(X) & \\ g_i(X) \leq 0 & i = 1,2,...,m \\ AX\text{-}\ b \leq 0 & \\ \underline{X} \leq X \leq \overline{X} & \end{cases}$$

Où, $\underline{X}$ et $\overline{X}$ sont la borne inférieure et supérieure de X. Le polyèdre défini par les contrainte linéaires est donné par:

$$P = \{X \in R^n, AX - b \leq 0\}$$

Pour incorporer les bornes inférieure et supérieure des variables dans ce polytope, les matrices A et b sont étendues comme suit:

$$\widetilde{A} = \begin{pmatrix} A \\ 1 \\ -1 \end{pmatrix} \quad et \quad \widetilde{b} = \begin{pmatrix} b \\ \overline{X} \\ -\underline{X} \end{pmatrix}$$

où, 1 et -1 sont les matrices diagonales avec 1 et -1 comme éléments diagonaux réspectivement. Alors, on obtient un polytope décrit comme suit:

$$\widetilde{P} = \{X \in R^n, \widetilde{A}X - \widetilde{b} \leq 0\}.$$

Les problèmes de programmation linéaire suivants aideront pour produire un simplex initiale S^0 aussi petit que possible:

$$\mu_0 = max\{\sum_{i=1}^{n} X_i, X \in \widetilde{P}\}$$

$$\mu_i = min\{X_i, X \in \widetilde{P}\} \quad i = 1,...,n.$$

Alors, tout les n+1 sommets du simpex initial peuvent être calculés par

$$V^i = (\mu_1,...,\mu_{i-1},\mu_0 - \sum_{j=i,j\neq 1}^{n} \mu_j,\mu_{i+1},...,\mu_n) \quad i = 1,..,n \qquad (8)$$

$$V^{n+1} = (\mu_1,...,\mu_n). \qquad (9)$$

Evidemment nous avons $\widetilde{P} \subseteq S^0$. Maintenant, nous sommes dans une position pour presénter l'algoritheme proposé pour résoudre le problème (P) en utilisant les

opérations de base décrites dans les séctions précédentes.

Etape 1-Initialisation: une tolérance de la convergence, ε_c , une tolérance de la faisabilité, ε_f, sont séléctionnées et le compteur d'intération K est fixé à zéro. le simplex initial S^0 est calculé par les équations (8) et (9), comme $S^0 = (V^1,V^2,..,V^n)$, et les courantes bornes des variables $\underline{X}$ et $\overline{X}$ pour la première itération sont posées égales aux solution du problème de programmation linéaire, c-à-d $\underline{X}_i = min\{X_i,X \in S^0\}$ et $\overline{X}_i = max\{X_i,X \in S^0\}$ pour i= 1,...,n.
Les bornes inférieure et supérieure globales μ_0 et γ_0 du minimum globale du problème (P) sont initialisées, un point initial $X^{k,c}$ est séléctionné.

Etape 2- Solution locale du problème (P) et mise à jour de la borne supérieure:
Le probléme de l'optimisation nonconvexe et non lénéaire (P) est résolu localement dans le simplex courant S.
Si la solution f^K_{local} du problème (P) est ε_f-faisable, la borne supérieure γ_k est mise à jour comme $\gamma_k = min(\gamma_k,f^K_{local})$.

Etape 3- La partition du simplex courant: Le simplex courant S^k, est partitionné (découpé) en deux simplexes (r= 1,2):

$$S^{k,1} = (V^{k,0},...,V^{k,m},...,\frac{V^{k,m}+V^{k,l}}{2},V^{k,n})$$
$$S^{k,2} = (V^{k,0},...,\frac{V^{k,m}+V^{k,l}}{2},...,V^{k,l},V^{k,n})$$

où, (k,m) et (k,l) correspondent aux sommets du plus longue distance dans le simplex courant c-à-d

$$(k,m)et(k,l) = argmax_{i<j}\{\|V^{k,j} - V^{k,i}\|\}.$$

Etape 4- Mise à jour de $a^r_{k,f}$, $b^r_{k,f}$, $c^r_{k,f}$ et a^r_{k,g_i}, b^r_{k,g_i}, c^r_{k,g_i} à l'intérieur des deux sous sim
r= 1,2.

Les paramètres positif ou nuls $a^r_{k,f}$ et a^r_{k,g_i} des termes nonconvexes dans la fonction objectif et les contraintes sont mise à jour a l'itérieur des deux simplexes r=1,2, d'après la méthode présentée dans la section 2,4,et coefficients linéaires et constants correspondants c-à-d $b^r_{k,f}$,$c^r_{k,f}$ et b^r_{k,g_i}, c^r_{k,g_i}, sont renouvelés suivant la proposition 2.2.1

Etape 5- Solution a l'intérieure des deux sous-simplexes r=1,2. Le problème de programmation convexe $(QP(S)')$ est résolu a l'itérieure des deux sous-simplexes (r= 1,2) en utilisant une résolution non linéaire convexe si la solution $F^{k,r}_{sol}$ est faisable (réalisable) et plus petite que la borne supérieure courante γ_k,alors elle est entreposée avec le point de la solution $X^{k,r}_{sol}$

Etape 6- Mise à jour du compteur d'itération K et la borne inférieure μ_k:
Le compteur d'itération est augmenté d'un (incrémenté).
$K \longleftarrow K+1$
et la borne inférieure est mise à jour pour être la solution minimum partout sur celles trouvées dans les itération précédentes. En outre la solution séléctionnée est éffacée de l'ensemble de solution trouvé:
$\mu_k = F^{k',r'}_{sol}$
où,

$$F^{k',r'}_{sol} = min_{r,I}\{F^{I,r}_{sol}, r = 1,2, I = 1,...,k-1\}.$$

Si l'ensemble I est vide, poser $\mu_k = \gamma_k$ et aller à l'étape 8.

Etape 7- Mise à jour du point courant $X^{k,c}$ et le simplex courant S^k.
Le point courant est séléctionné pour être le point solution minimum trouvé dans l'étape 6
$X^{k,c} = X^{I',r'}_{sol}$
et le simplex courant devient le sous-simplex qui contient la précédente solution

trouvée.

$$S^k = (V^{k^{'},0},...,V^{k^{'},m},...,\frac{V^{k^{'},m}+V^{k^{'},l}}{2},...,V^{k^{'},n}) \quad si \quad r^{'}=1$$

$$S^k = (V^{k^{'},0},...,\frac{V^{k^{'},m}+V^{k^{'},l}}{2},...,V^{k^{'},l},...,V^{k^{'},n}) \quad sinon$$

Etape 8- teste de convergence. Si ($\gamma_k - \mu_k > \varepsilon_c$,

alors aller a l'étape 2.

sinon, ε_c-convergence est atteinte. La solution minimum globale et le point solution sont donnés par:

$$f^* \longleftarrow f^{c,k^{''}}$$

$$X^* \longleftarrow X^{c,k^{''}}$$

où,

$$K^{''} = arg_I\{f^{c,I} = \gamma_k\}, \quad I = 1,...,k.$$

Il faut noter que le simplex courant peut être éléminé dans l'étape 5 quand le problème $(QP(S)^{'})$ est irréalisable ou sa solution est plus grande que la borne supérieure courante. c'est évident que le problème (P) est irrèalisable quand le problème $(QP(S)^{'})$ est irréalisable.

La preuve mathématique de la convergence de l'algorithme QBB de l'optimisation globale vers le minimum globale est présentée dans la section suivante.

2.5 Preuve de convergence vers le minimum globale:

Si l'algorithme QBB présenté dans la séction précédente se termine à l'itération K, alors le point X^k est une solution optimale pour le problème (P). Dans le cas ou l'algorithme est infini, il génère (produit) au moin une séquence infinie de simplexes $\{S^j\}$tel que $S^{j+1} \subset S^j$, pour tout j. la convergence de l'algorithme QBB est affirmée

avec les résultat suivants

Proposition. ***2.5.1:*** *supposons que la problème (P) possède une solution réalisable (faisable), supposons que l'algorithme QBB produit (génère) une séquence infinie de simplexes* $\{S^j\}$ *tel que* $S^{j+1} \subset S^j$.
pour tout j,et

$$lim_{j\longrightarrow\infty} S^j = \bigcap_{j=1}^{\infty} S^j = \{X^*\}$$

Alors,X est une solution optimale du problème (P).*

Preuve: premièrement, nous montrons que le point X* est un point réalisable pour le problème (P).

Pour faire ça, pour tout j, soit V^j un sommet de simplex S^j. pour tout j, soit (X^j) une solution optimale pour le problème relaxé de programmation convexe (QP(S)) avec $S = S^j$. Il faut noter que (X^j) éxiste pour n'importe quel j comme montré dans proposition 2.2.2 (b). Quand les bords du simplex S^j sont bornés et

$$lim_{j\longrightarrow\infty} S^j = \bigcap_{j=1}^{\infty} S^j = \{X^*\}$$

Alors, nous avons aussi:

$$lim_{j\longrightarrow\infty} V^j = \{X^*\}.$$

nous pouvons supposer que $X^j \rightarrow X^*$,quand $j \rightarrow \infty$.

De ceci ,nous avons

$G_i(X^j) \rightarrow G_i(X^*) \leq 0$ pour i = 1,...,m et $j \rightarrow \infty$.

Supposons que X^* n'est pas une solution réalisable du problème (P);cela veut dire qu'il éxiste un nombre $\varepsilon > 0$ et pour quelques contraintes K tel que:

$$g_K(X^*) \geq \varepsilon > 0.$$

Quand X^* est un sommet a la limite du simplex , alors d'après la Définition 2.2.1, nous avons :

$$g_K(X^*) = G_K(X^*) \geq \varepsilon > 0.$$

Ce qui implique que X^* n'est pas un point réalisable pour le problème (QP(S)).Cette contradiction implique que X^* est un point réalisable(faisable) pour le problème (P). Deuxièmement ,quand $\mu(S^{j+1}) \geq \mu(S^j) > -\infty$,pour tout j,par la proposition 2.2.2,il existe une limite μ^* de $\{\mu(S^j)\}$ bornée par la valeur optimale du problème (P). De plus ,dans l'algorithme QBB ,nous avons

$$lim_{j\to\infty}\mu(S^j) = lim_{j\to\infty}\gamma(S^j) \geq f(X^*)$$

Ce qui implique que X^* est une solution optimale pour le problème (P).
Il faut noter que la $\varepsilon-convergence$ finie pour ce cas peut être obtenue en soumetant la séquence infinie précédente à la $\varepsilon-tolérance$ entre les bornes inférieure et supérieure dans l'arbre énumèré(**M**aranas et **F**loudas,1994).on observe l'éxistance d'un point d'accumulation des bornes supérieures à cause de la compacticité du simplex S^0,et que ce point est une solution optimale pour le problème (P), ce qui est précisé dans la proposition qui suit.

Proposition. ***2.5.2.*** *(a) supposons que le problème (P) a une solution réalisable , et que le processus de partition simpliciale de l'algorithme QBB présenté dans la séction 2.1 est éxhaustive , alors l'algorithme QBB a les proprietés de convergence suivantes:*
Si l'algorithme QBB génére (produit) une séquence infinie de simplexes $\{S^j\}$ tel que l'ensemble des bornes supérieures $F(S^j) \neq \emptyset$ pour tout j,alors tout point d'accumulation $\{X^j\}$ de la séquence correspondente est une solution optimale pour le problème (P).
(b) l'algorithme QBB se termine après plusieurs itérations finies quand l'ensemble réalisable du problème (P) est vide.

Pour la preuve de (a) ,nous pouvons q'un ensemble de séquence de borne supérieure existe dont la limite est la solution optimale du problème(P).
Plus encore , si l'algorithme QBB ne se término pas après plusieurs itérations finies , il doit générer une séquence de points qui convergent à la solution optimale du problème (P) d'après la proposition 2.5.1.

Cette contradiction implique que l'algorithme QBB se términe d'une manière finie. Le problème général d'optimisation nonconvexe est NP-Difficil (**V**avasis,1991). Alors , on peut s'attendre a ce que quelques grands problèmes soient difficiles pour l'algorithme QBB .Cependant ,cela ne veut pas dire précisément que l'algorithme QBB est incapable de résoudre les grands problèmes dans une durée résonable de temps.Comme nous l'avons décrit dans les étapes de l'algorithme QBB, il est possible d'obtenir de bonnes solutions réalisables et aussi de montrer avec une tolérance spécifiée leurs optimalité,surtout pour les problèmes qui ont quelques structures favorables.

Résultats préliminaires d'un éxemple numérique:

$$
\begin{cases}
\text{Min f(x)} = 2\ 37.293239\mathrm{x}_1 + 0.8356891\mathrm{x}_1\mathrm{x}_5 + 5.3578547\mathrm{x}_3^2 - 40792.141 \\
sc \\
\text{-}0.0022053\ \mathrm{x}_3\mathrm{x}_5 + 0.0056858\mathrm{x}_2\mathrm{x}_5 + 0.0006262\mathrm{x}_1\mathrm{x}_4 - 6.665593 \leq 0 \\
0.0022053\ \mathrm{x}_3\mathrm{x}_5 - 0.0056858\mathrm{x}_2\mathrm{x}_5 - 0.0006262\mathrm{x}_1\mathrm{x}_4 - 85.334407 \leq 0 \\
0.0071317\ \mathrm{x}_2\mathrm{x}_5 + 0.00218133\mathrm{x}_3^2 + 0.0029955\mathrm{x}_1\mathrm{x}_2 - 29.48751 \leq 0 \\
\text{-}0.0071317\ \mathrm{x}_2\mathrm{x}_5 - 0.00218133\mathrm{x}_3^2 - 0.0029955\mathrm{x}_1\mathrm{x}_2 + 9.48751 \leq 0 \\
0.0047026\ \mathrm{x}_3\mathrm{x}_5 + 0.0019085\mathrm{x}_3\mathrm{x}_4 + 0.0012547\mathrm{x}_1\mathrm{x}_3 - 15.699039 \leq 0 \\
\text{-}0.0047026\ \mathrm{x}_3\mathrm{x}_5 - 0.0019085\mathrm{x}_3\mathrm{x}_4 - 0.0012547\mathrm{x}_1\mathrm{x}_3 + 10.699039 \leq 0 \\
78 \leq \mathrm{x}_1 \leq 102 \\
33 \leq \mathrm{x}_2 \leq 45 \\
27 \leq \mathrm{x}_3 \leq 45 \\
27 \leq \mathrm{x}_4 \leq 45 \\
27 \leq \mathrm{x}_5 \leq 45
\end{cases}
$$

La nonlinéaireté de ce problème apparait dans les termes bilinéres $\pm X_i X_j$ pour $i \neq j$,et $-X_i^2$,dans la fonction coût et les fonctions contraintes.Pour ces derniers termes ,quand ils appartiennent aux structures des fonctions concaves ,alors leurs enveloppes convexes décrites dans Proposition 4.3 c-à-d une fonction affine sur le simplex ,peut être construite facilement.Pour les autres on peut voire que leurs matrices Hessiennes sont constantes tel:

$$H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad ou \quad H = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

dont ,deux valeurs propres sont 1 et -1 respectivement.

D'après l'analyse dans la section 2.3 pour les fonctions quadratiques en général ,on obtient le coefficient uniforme c-à-d **a** de cette manière :

$$a = max_i\{0, \frac{1}{2}\lambda_i^Q\}$$

donc on auras a=0.5. par conséquant ,les coefficients linéaires et constant pour la fonction quadratique de sous estimation peuvent être calculés en utilisant la Proposition 2.1 sur le simplex courant.Après avoir remplacés toutes les fonctions nonconvexes par leurs fonctions quadratiques de sous estimation on obtient un problème relaxé de programmation convexe qu'on peut résoudre par optimisation convexe pour avoir une borne inférieure du problème original.le simplex courant et décrit par

$$X = \sum_{i=1}^{6} \lambda_i V^i$$
$$0 \leq \lambda_i \leq 1 \qquad i = 1,2...,6$$

L'algorithme de cette méthode a ètè implimenté dans le language C,et après 1090 itération éxécutées on obtient les résultats suivants:

Table I. Calculation results of QBB algorithm for Colville's Problem when a=0.5

Variable x_i	Upper bound	Upper bound solution	Lower bound	Lower bound solution	Iteration number	Number of unfathomed subsimplexes	CPU time (s)
1		78.0		78.0			
2		33.0		33.0			
3	−30665.58848	29.99506	−30665.60118	29.99503	1090	0	20.59
4		45.0		45.0			
5		36.77602		36.77601			

et si on pose $a = 1$*:*

Table II. Calculation results of QBB algorithm for Colville's Problem when $a = 1.0$

Variable x_i	Upper bound	Upper bound solution	Lower bound	Lower bound solution	Iteration number	Number of unfathomed subsimplexes	CPU time (s)
1		78.0		78.0			
2		33.0		33.0			
3	−30665.58848	29.99506	−30665.77273	29.99445	2078	0	42.21
4		45.0		45.0			
5		36.77602		36.77628			

on remarque que le choix de $\boldsymbol{a}$ *par cet algorithme est décisif pour gagner du temps et avoir un meilleur résultat.*

Chapitre 3

application sur le logiciel LINGO

Introduction

La programmation est un ensemble d'outils et de technique permettant de résoudre des problèmes mathématiques par ordinateurs, elle sert a trouver une solution optimale de n'importe quel type de problème. Le processus de résoudre un problème mathématique exige un grand nombre de calculs donc il est mieux de léxécuter par machine . Pour cela on a choisit le logiciel LINGO qui est spécifié pour résoudre des problèmes d'optimisations (linéaire,non linéaire,convexe ,non convexe,..etc). Il comporte un éventail de commandes qui peuvent être appelées á tout moment .

3.1 Présentation du logiciel

LINGO nous permet de formuler rapidement et facilement nos problèmes d'optimisation linéaire, non-linéaire ou en nombres entiers. Grâce à ses outils de modélisation, les modèles sont exprimés de manière transparente à l'aide de sommes et de variables indicées. La méthode ne diffère guère de la méthode traditionnelle avec crayon et papier, mais les modéles seront plus faciles à réutiliser et mettre à jour. Il possède quatre solveurs qu'il utilise afin de résoudre les différents types de modèles :

- *Solveur direct.*
- *Solveur linéaire*

- *Solveur non linéaire.*
- *Méthode de type séparation et évaluation.*

De plus, LINGO est :

- *Un moyen pour confirmer que l'optimum trouver est l'optimum global.*
- *Possibilité pour résoudre les problèmes plus rapidement.*
- *Un moyen amélioré pour résoudre beaucoup de type de problèmes.*
- *Un moyen de décomposition si un modèle contient des sous-modèles.*

LINGO est livré avec un jeu de solveurs pour l'optimisation linéaire, non-linéaire convexe ou non convexe), quadratique, sous contraintes, et en nombre entier. Nous avons même pas à nous préoccuper du choix du solveur : en effet, LINGO iterpréte lui-même nos formulations et sélectionne automatiquement le solveur adapté à chaque problème.
Un modèle d'optimisation se compose de trois parties :

1. *La fonction objective : c'est la formule simple qui décrit exactement ce que le modèle devrait optimiser.*
2. *Les variables : ce sont les quantités qui peuvent être changés pour déterminer la valeur optimale de la fonction objective.*
3. *Les contraintes : ce sont les formules qui définissent les limites sur les valeurs des variables.*

Les fonctions utilisées dans un modèle de LINGO sont :

@FOR – utilisée pour produire des contraintes.
@SUM – calcul de la somme.
@MAX - recherche de maximum.
@MIN – recherche de minimum.

Type de variables dans LINGO :

Toutes les variables dans un modèle de LINGO sont considérées non négatives et continus, les fonctions variables d'un modèle de LINGO sont :

@GIN - toute valeur positive de nombre entiers.

@BIN - une valeur binaire (0 ou 1).

@ FREE - toute valeur positive ou négative réelle.

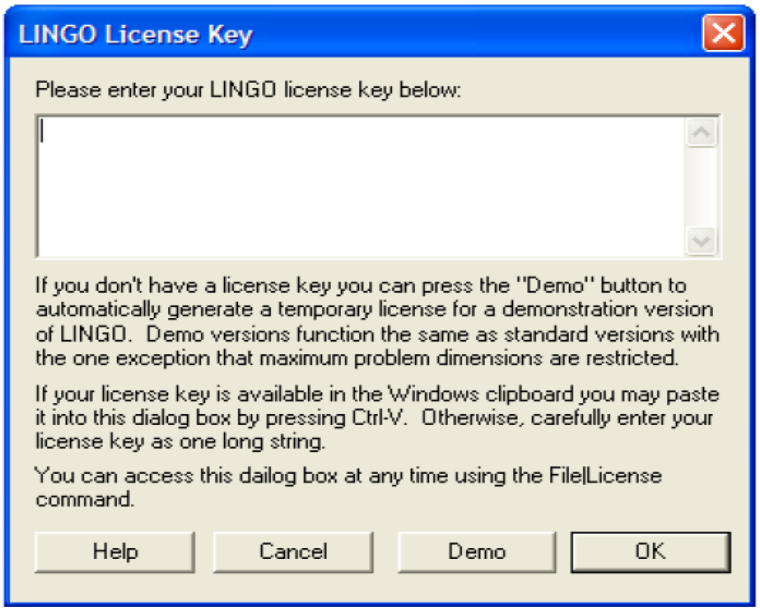
@BND - toute valeur bornée par des limites indiquées.

LA forme générale pour la déclaration d'un variable x, en utilisant les fonctions variables @GIN,@BIN,@FREE est : @function (x) ;

La fonction @BND inclut les limites inférieure et supérieure des variables, sa forme générale est : @BND (limite inférieure, x, limite supérieure).

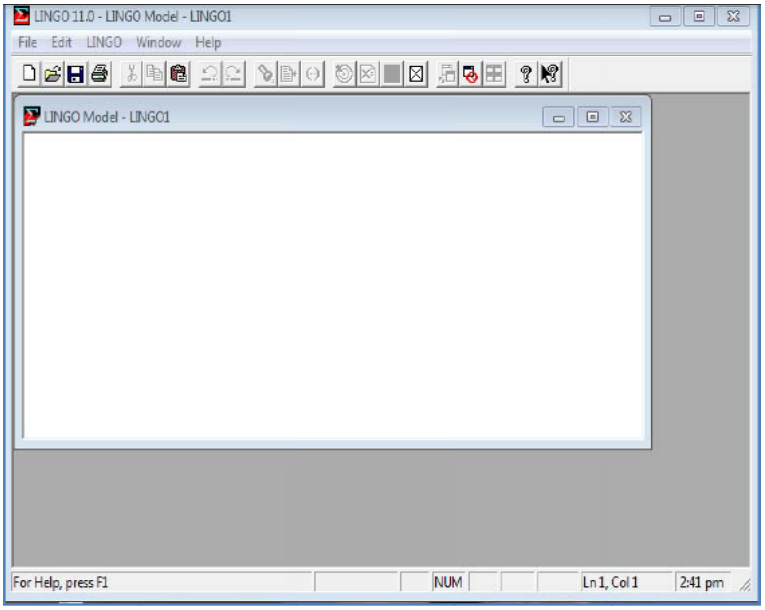
3.1.1 Interface de logiciel :

Lors de lancement de LINGO après l'installation, certaines versions nous demandent déntrer la clé de la licence, donc on voit apparaitre la fenêtre suivante :

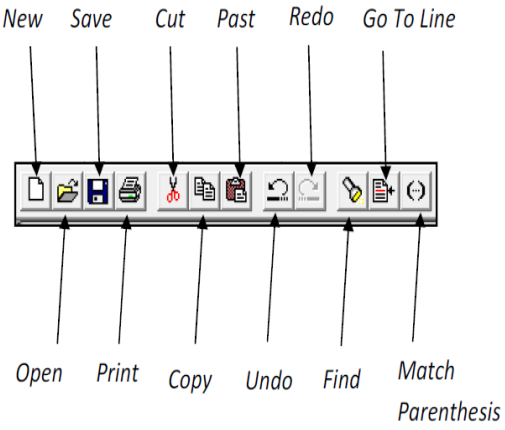


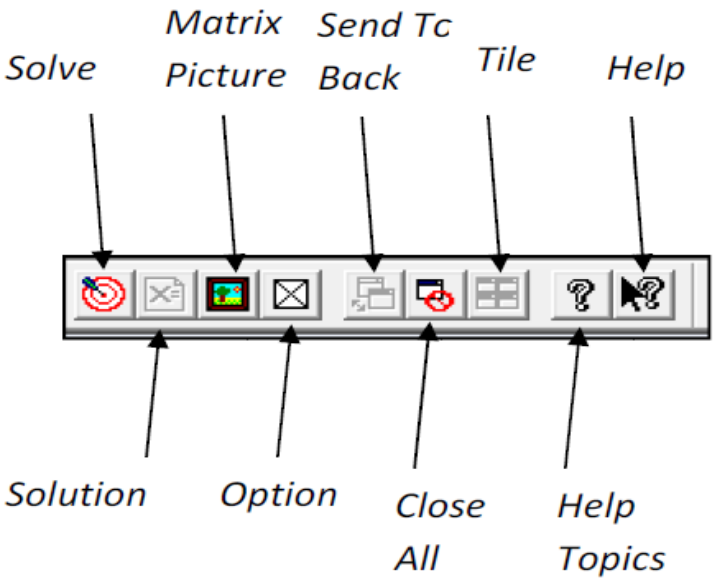
Donc on fait entrer la clé et on click sur « ok », si on n'a pas la clé de la licence on peut utiliser LINGO sous le mode « Demo », mais la licence de ce mode expire dans 30jours.

Juste après la barre des menus LINGO et une fenêtre pour saisir le modèle de problème à résoudre apparait:



La barre des outils de LINGO est la suivante:





Signification :

New: ouvrir une nouvelle fenêtre.

Open: ouvrir un dossier enregistré.

Save: enregistrer un modèle.

Print: imprimer la fenêtre courante.

Cut: copier et supprimer le texte sélectionné.

Copy: copier le texte sélectionné.

Past: coller le contenu sélectionné dans le document.

Undo: refaire l'action précédente.

Redo: revenir à l'action suivante.

Find: chercher un document.

Go To Line: déplacer le curseur à un certain nombre de ligne.

Match Parenthesis: trouvez la parenthèse étroite qui correspond à la parenthèse ouverte que vous avez choisie.

Solve: résoudre le modèle.

Solution: faire la fenêtre de rapport de la solution de modèle courant.

Option: ensemble des options de système.

Send To Back: placer la fenêtre courante derrière les autres fenêtres ouvertes.

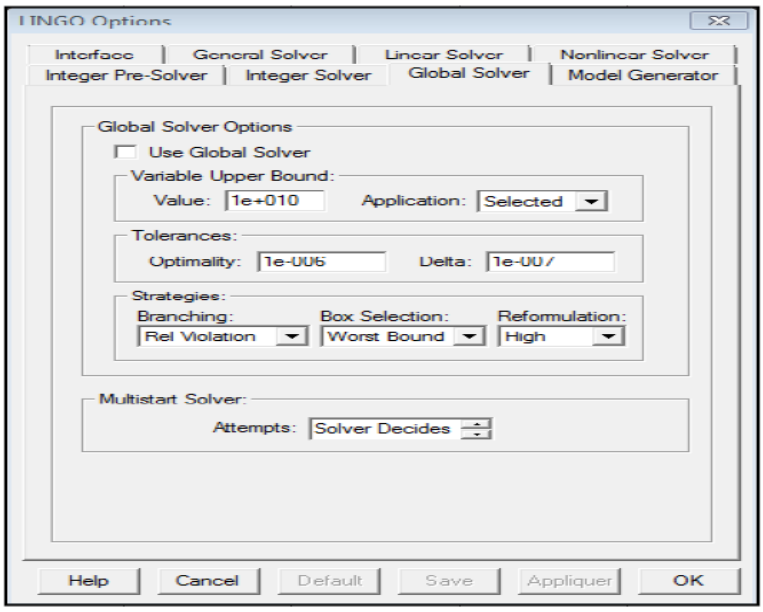
Close All: fermer toutes les fenêtres ouvertes.

Tile: placer les fenêtres ouvertes dans la même place dans la fenêtre de programme LINGO.

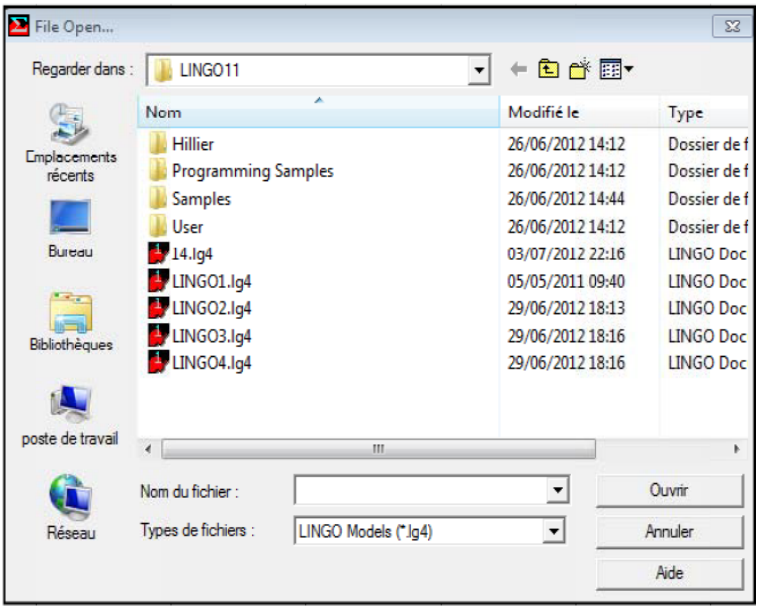
Help Topics: ouvrir le manuel de LINGO.

Help: ouvrir l'aide de LINGO.

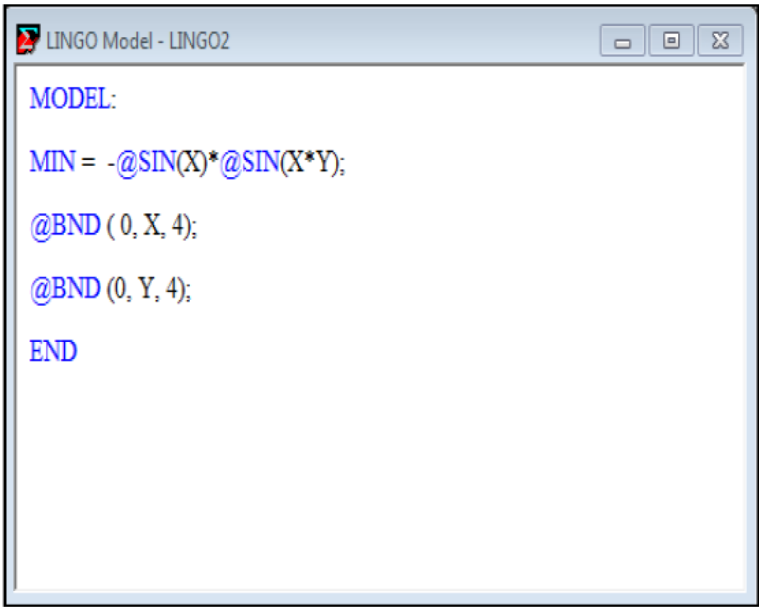
On peut sélectionner les options à suivre pour résoudre un model, en cliquant sur LINGO/options, par exemple dans la fenêtre suivante on sélectionne « global solver»:



Ensuite on fait entrer le modèle de problème à résoudre, pour ouvrir un modèle enregistré on click sur « open » dans la barre des menu, la fenêtre suivante apparait:

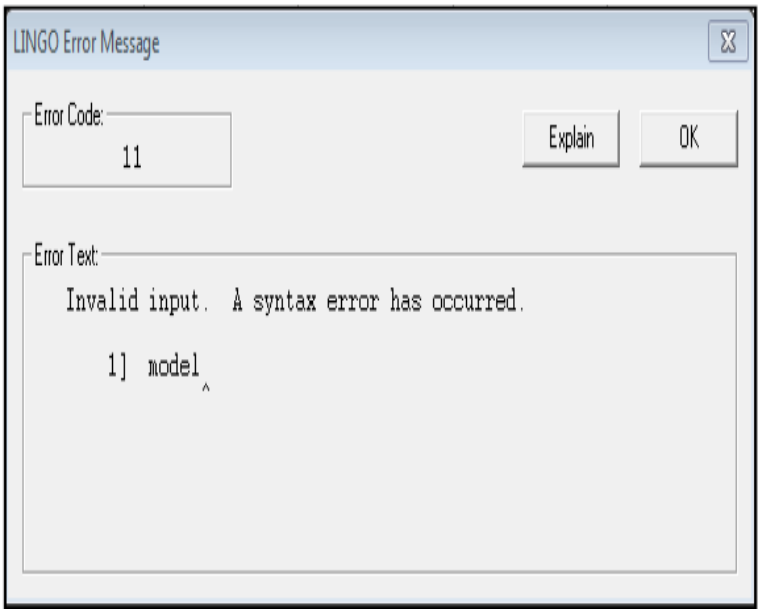


On sélectionne le modèle a ouvrir et on click sur « ouvrir » dans la fenêtre suivante, le modèle sélectionné s'ouvre dans la fenêtre « LINGO model »:

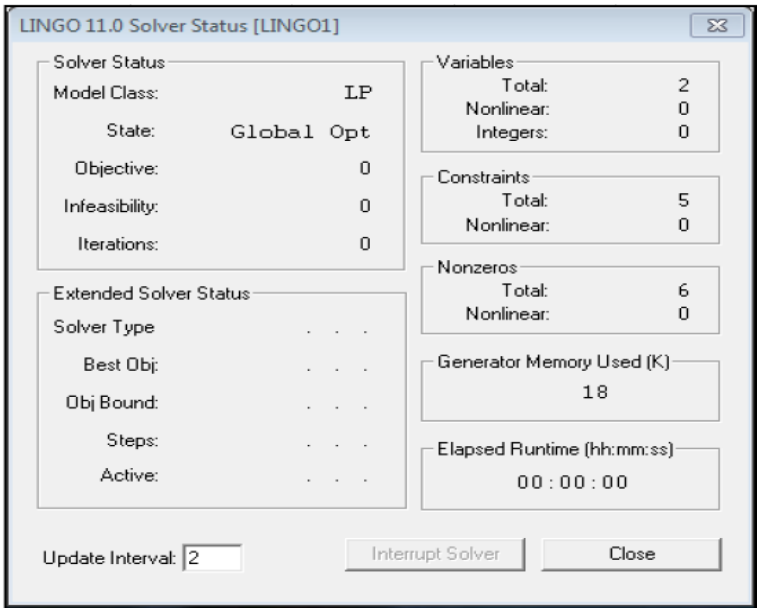


Une fois le modèle de problème est entré, ce modèle peut être résolu en cliquant sur le bouton solve sur la barre des outils, en séléctionnant LINGO/solve de menue, ou par utilisation de raccourci de clavier ctrl+s. LINGO annonce les erreurs rencontrées, la meilleure manière pour avoir des informations sur une erreur c'est de consulter la section « Error Messages ».

Si une erreur est rencontrée la fenêtre suivante apparait:



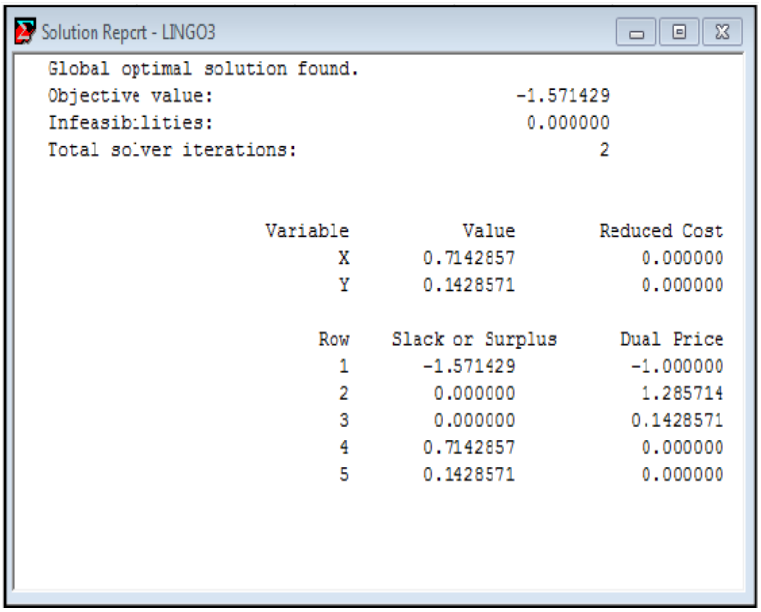
Si aucune erreur n'est signalée alors la fenêtre suivante apparait :



Cette fenêtre fournit des informations sur le nombre des variables non linéaire, entiers, et total dans le modèle, elle fournit aussi le nombre des contraintes non linéaire et total utilisées dans le modèle, et le nombre des coefficients des variables non linéaire et total utilisées et qui sont différentes de zéro.

Cette fenêtre de résolution nous donne des détails sur la classification des modèles (LP, NLP, ILP, QP, IQP, etc.), le type de la solution obtenue (locale ou globale, etc.), la valeur de la fonction objective, le nombre d'itérations requises pour résoudre le modèle.

En fermant cette fenêtre, on peut alors regarder la fenêtre de rapport de solution.



Cette fenêtre nous donne la valeur de l'optimum qui produisent la valeur optimale de la fonction objectif.

3.1.2 Résolution de quelques éxemples ainsi que l'éxemple traité dans le chapitre 2 sur LINGO

3.1.3 Exemple

Soit à résoudre le problème suivant (P):

$$(P)\begin{cases} \min \text{ f(x)} \\ \text{X} \in H \end{cases}$$

Où $f(x) = x^6 - 15x^4 + 27x^2 + 250$

et $x \in [-4,4]$

MODELE d'éxécution sur LINGO est le suivant:

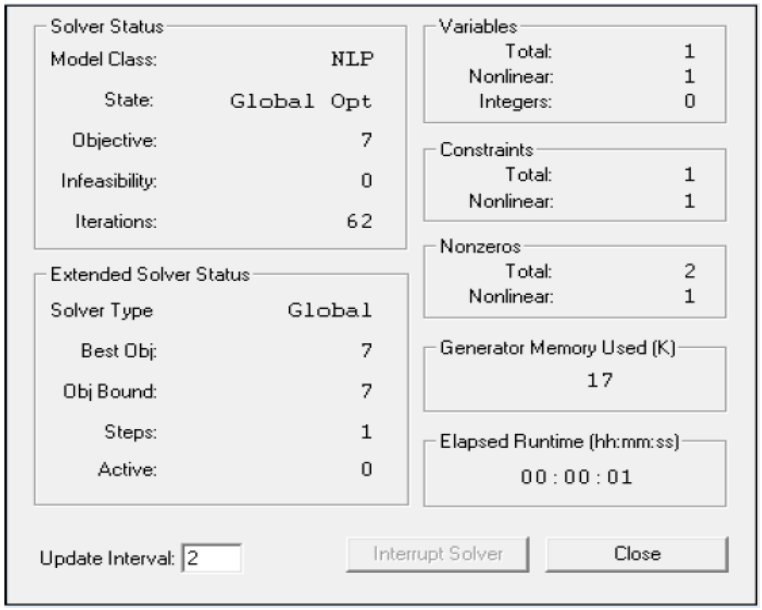
MODEL:

MIN $=x^6 - 15x^4 + 27x^2 + 250$;

@BND(-4,X,4);

END

– Les résultats du modèle après l'éxécution



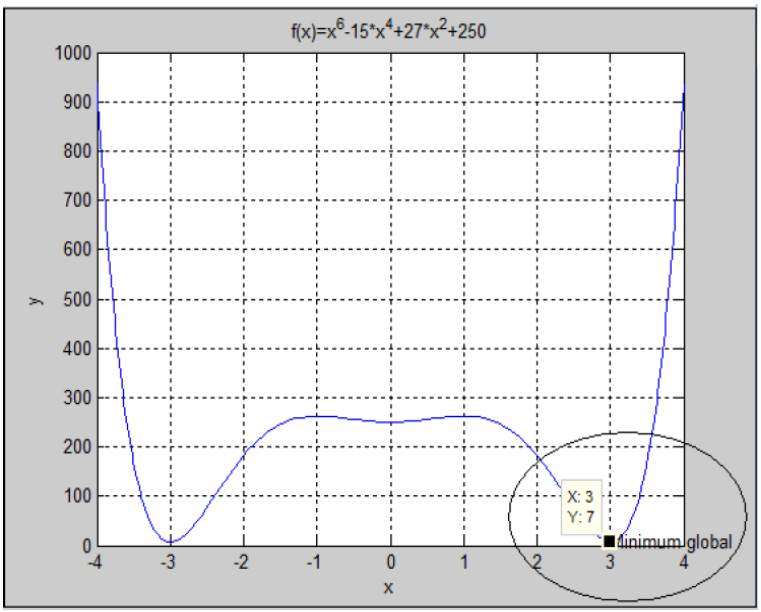
Tab.3.1. – Résultats de l'éxécution du modèle de (P) sur l'intervalle [-4,4]

Global optimal solution found		
Objective value: Objective bound: infeasibilities Extended solver steps : Total solver itérations :		7.000000 7.000000 0.000000 1 62
Variable X Row 1	value 3.000000 Slack or Surplus 7.000000	Reduced Cost 0.000000 Dual Price -1.000000

Tab.3.2. – Résultats finale de l'éxécution

fonction objective f(x)	minimum global x^{opt}	valeur minimale f^{opt}	nombre d'itérations
$f(x) = x^6 - 15x^4 + 27x^2 + 250$	± 3	7	62.

– Le graphe de la fonction avec le minimum global



3.1.4 Exemple

Soit à résoudre le problème suivant (P):

$$(P)\begin{cases} \min \text{f(x)} \\ \text{X} \in H \end{cases} \Longleftrightarrow (P)\begin{cases} \min\ (\text{x}^2 - 5x + 6)/(x^2 + 1) \\ \text{X} \in [-5{,}5] \end{cases}$$

MODELE d'éxécution sur LINGO est le suivant:

MODEL:

MIN = $(x^2 - 5x + 6)/(x^2 + 1)$;

@BND(-5,X,5);

END

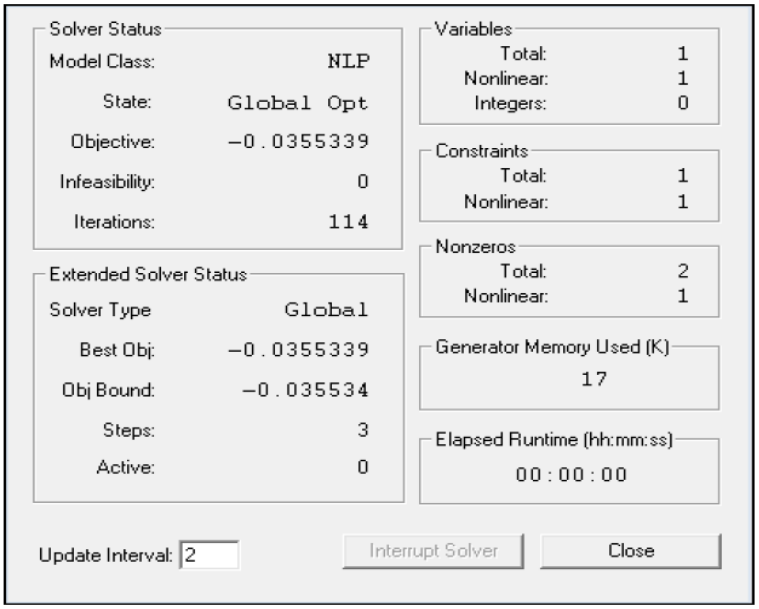
Tab.3.3 – Résultats de l'éxécution du modèle de (P) sur l'intervalle [-5,5]

Global optimal solution found		
Objective value: Objective bound: infeasibilities Extended solver steps : Total solver itérations :		-0.3553391E-01 -0.3553396E-01 0.000000 3 114
Variable X Row 1	value 2.414214 Slack or Surplus -0.3553391E-01	Reduced Cost 0.000000 Dual Price -1.000000

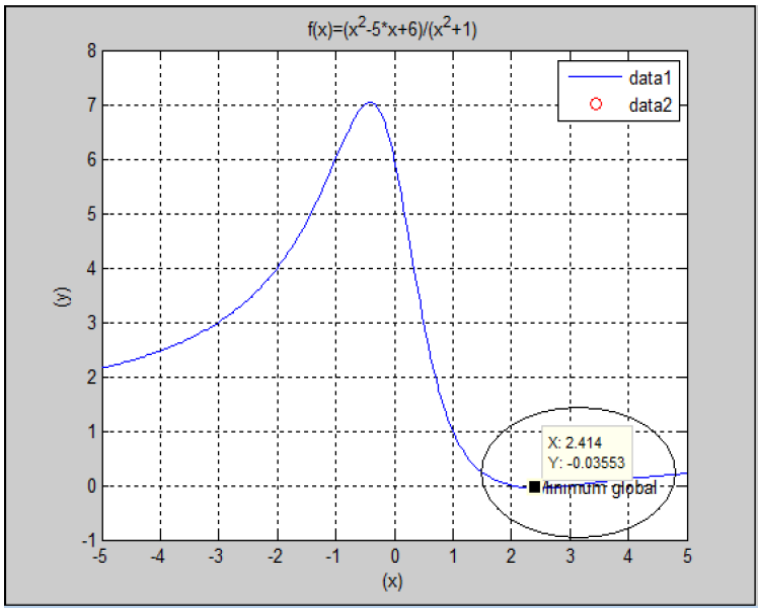
Tab.3.4. – Résultats final de l'éxécution

fonction objective f(x)	minimum global x^{opt}	valeur minimale f^{opt}	nombre d'itérations
$f(x) = (x^2 - 5x + 6)/(x^2 + 1)$	2.414214	0.3553391E-01	114.

– Résultats de l'éxécution du MODELE de (P)



– Le graphe de la fonction avec le minimum global



3.1.5 Exemple

Soit à résoudre le problème Colville suivant:

$$
\begin{cases}
\text{Min f(x)} = 2\ 37.293239x_1 + 0.8356891x_1x_5 + 5.3578547x_3^2 - 40792.141 \\
sc \\
\text{-}0.0022053\ x_3x_5 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 6.665593 \leq 0 \\
0.0022053\ x_3x_5 - 0.0056858x_2x_5 - 0.0006262x_1x_4 - 85.334407 \leq 0 \\
0.0071317\ x_2x_5 + 0.00218133x_3^2 + 0.0029955x_1x_2 - 29.48751 \leq 0 \\
\text{-}0.0071317\ x_2x_5 - 0.00218133x_3^2 - 0.0029955x_1x_2 + 9.48751 \leq 0 \\
0.0047026\ x_3x_5 + 0.0019085x_3x_4 + 0.0012547x_1x_3 - 15.699039 \leq 0 \\
\text{-}0.0047026\ x_3x_5 - 0.0019085x_3x_4 - 0.0012547x_1x_3 + 10.699039 \leq 0 \\
78 \leq x_1 \leq 102 \\
33 \leq x_2 \leq 45 \\
27 \leq x_3 \leq 45 \\
27 \leq x_4 \leq 45 \\
27 \leq x_5 \leq 45
\end{cases}
$$

MODELE d'éxécution sur LINGO est le suivant:

MODEL:

Min= $237.293239x_1 + 0.8356891x_1x_5 + 5.3578547x_3^2 - 40792.141$;

$-0.0022053x_3x_5 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 6.665593 \leq 0$;

$0.0022053x_3x_5 - 0.0056858x_2x_5 - 0.0006262x_1x_4 - 85.334407 \leq 0$;

$0.0071317x_2x_5 + 0.00218133x_3^2 + 0.0029955x_1x_2 - 29.48751 \leq 0$;

$-0.0071317x_2x_5 - 0.00218133x_3^2 - 0.0029955x_1x_2 + 9.48751 \leq 0$;

$0.0047026x_3x_5 + 0.0019085x_3x_4 + 0.0012547x_1x_3 - 15.699039 \leq 0$;

$-0.0047026x_3x_5 - 0.0019085x_3x_4 - 0.0012547x_1x_3 + 10.699039 \leq 0$;

@BND(78 ,x_1, 102);

@BND(33 ,x_2, 45);

@BND(27 ,x_3, 45);

@BND(27 ,x_4, 45);

@BND(27 ,x_5, 45);

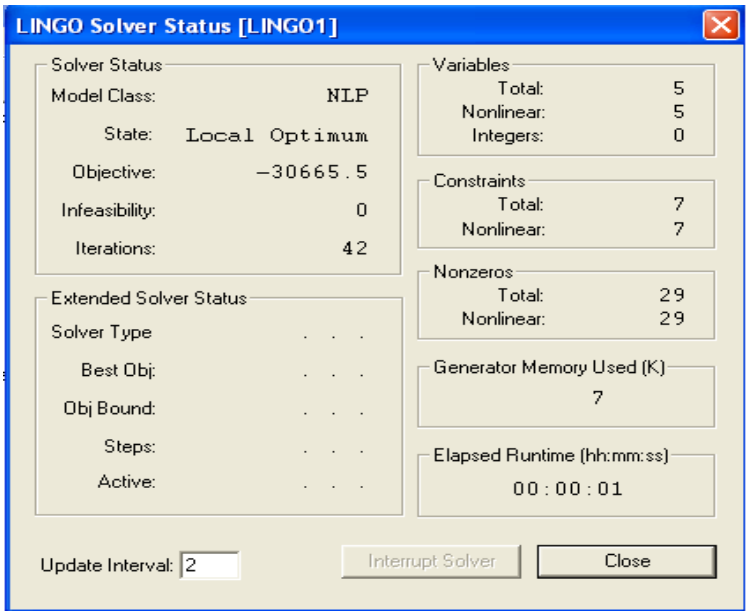
END

Tab.3.5 – Résultats de l'éxécution du modèle de Colville

```
Local optimal solution found at iteration:           42
Objective value:                               -30665.54

                     Variable           Value        Reduced Cost
                           X1        78.00000            48.92735
                           X5        36.77581            0.000000
                           X3        29.99526            0.000000
                           X2        33.00000            84.32353
                           X4        45.00000           -26.63920

                          Row    Slack or Surplus      Dual Price
                            1       -30665.54           -1.000000
                            2        0.000000            403.2691
                            3        92.00000            0.000000
                            4        11.15947            0.000000
                            5        8.840527            0.000000
                            6        5.000000            0.000000
                            7        0.000000            809.4252
```



Conclusion:

On a vu dans ce chapitre comment insérer un modèle d'éxécution d'un problème sur LINGO dans le cas des problèmes traités par la méthode Branch and Bound. Lingo fait appel au "solveur global" après l'éxécution on a directement le minimum global et la valeur optimale de la fonction objectif.

Conclusion Générale

Nous nous sommes intéressé dans notre travail à la résolution des problèmes d'optimisation globale des fonctions deux fois differentiable non linéaires,non convexes sur un intervalle en tenant compte de leurs structures telles que la linéarité et la convexité.
Nous avons présenté une méthode de résolution précisément la méthode déterministe qui est beaucoup plus efficaces dans la détermination des minimums globaux . La méthode Branch-and-Bound a été utilisée dans plusieurs domaines d'optimisation comme l'optimisation combinatoire,l'optimisation semi-infini et l'optimisation quadratique ainsi que l'optimisation globale.
L'algorithme QBB est developpé pour résoudre les problèmes de Programmation Non Linéaires PNL deux fois differentiables .
Pour n'importe quel problème de ce genre,l'abilité de générer progressivement des problèmes de borne inférieure convex dans chaque itération garantit la convergence de cet algorithme.Les différentes méthodes de construction des sous estimateurs pour les structures de fonctions spéciales et les structures de fonctions nonconvexes en général,où l'analyse de la valeur propre maximale de la matrice hessienne d'intervalle fournit la garantie a l'algorithme QBB de converger vers la solution globale.
Les proprietés de convergence de cet algorithme sont obtenues ,et les résultats de l'example numérique montrent l'efficacité de cet algorithme dans l'application pratique.

Bibliographie:

-Adjiman, C.S., Dallwig, S., Floudas, C.A. and Neumaier, A. (1998a), A global optimization method, αBB, for general twice-differentiable constrained NLPs - I. Theoretical

advances, Computers and Chemical Engineering .

-Adjiman, C.S., Androulakis, I.P. and Floudas, C.A. (1998b), A global optimization method, αBB, for general twice-differentiable constrained NLPs - II. Implementation and computational results, Computers Chemical Engineering .

-Al-Khayyal, F.A. and Falk, J.E. (1983), Jointly constrained biconvex programming, Mathematics of Operational and Research.

-Colville, A.R. (1970), A comparative study of nonlinear programming codes. In: Kuhn, H.W. (ed.), Princeton Symposium on Mathematical Programming, Princeton Univ. Press, New Jersy.

-Floudas, C.A., Aggarwal, A. and Ciric, A.R. (1989), Global optimum search for nonconvex NLP and MINLP problems, Computers and Chemical Engineering 13.

-Floudas, C.A. and Visweswaran, V. (1990), A global optimization algorithm (GOP) for certain classes of nonconvex NLPs: I. Theory, Computers and Chemical Engineering.

-Floudas, C.A. and Pardalos, P.M. (1990), A Collection of Test Problems for Constrained Global Optimization Algorithms, Lecture Notes in Computer Science 455, Springer-Verlag, Berlin.

-Floudas, C.A. and Visweswaran, V. (1993), A primal-relaxed dual global optimization approaches, Journal of Optimization Theory and Application.

-Floudas, C.A. (2000), Deterministic Global Optimization: Theory, Methods and Applications, Kluwer Academic Publishers. Dordrecht.

-Gerschgorin, S. (1931), Uber die abgrenzung der eigenwerte einer matrix, Izv.Akademii Nauk SSSR, Ser.fiziku-material.

-Maranas, C.D. and Floudas, C.A. (1992), A global optimization approach for Lennard-Jones microclusters, Journal of Chemical Physics.

-Maranas, C.D. and Floudas, C.A. (1994), Global minimum potential-energy conformations of small molecules, Journal of Global Optimization.

-McCormick, G.P. (1976), Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems, Mathematical Programming.

-Neumaier, A. (1990), Interval Methods for Systems of Equations, Cambridge Univ. Press,

Cambridge, England.

-Neumaier, A. (1996), Second-order sufficient optimality conditions for local and global nonlinear programming, Journal of Global Optimization.

-Rockafellar, R.T. (1972), Convex Analysis, Princeton Univ. Press, Princeton.

-Mr OUANES .Mohand ."Optimisation semi-infinie et Optimisation Globale,Théorie et Algorithmes".Thèse d'état,UMMTO,(2006).

-Melle BELHACENE Saliha. " Résolution de problème de programmation semi-infinie et introduction à l'optimisation globale". Mémoire de magister,UMMTO(2005).