

*République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique*

*Université Mouloud Mammeri, Tizi-Ouzou  
Faculté des Sciences  
Département de Mathématiques*



*Spécialité : Mathématiques  
Option : Recherche Opérationnelle  
Mémoire de fin de cycle intitulé :*

---

## *Optimisation locale et globale*

---

*Réalisé par :*  
**MOKRANE BOUGRAB**  
*et*  
**SALAH BEN MOUSSA**

*Encadré par :*  
**MOHAND OUANES**

*Soutenu devant le jury d'examen composé de :*

M <sup>F</sup> HOCINE KOURAT,	Président.
M <sup>F</sup> OUANES MOHAND,	Rapporteur.
M <sup>F</sup> CHEBBAH MOHAMMED,	Examineur.

*Année Universitaire 2024-2025*

# Remerciements

*Avant tout, je remercie Dieu de m'avoir donné le courage pour mener à bien ce travail, malgré tous les obstacles.*

*Je remercie mon encadreur monsieur **OUANES Mohand**, professeur à l'université Mouloud Mammeri de Tizi-Ouzou, pour m'avoir encadré et orienté durant mon travail.*

*Je  
lui témoigne ma respectueuse gratitude.*

*J'adresse mes vifs remerciements aux membres du jury pour avoir pris le temps de lire et d'examiner ce travail.*

*Je ne peux pas oublier de remercier mes parents, ma famille et tous mes proches, pour  
leurs  
soutiens et leurs encouragements.*

# *DÉDICACES BOUGRAB*

*Je dédie ce modeste travail :*

À ma très chère mère, source inépuisable d'amour, de patience et de prières.

À la mémoire de mon père, que Dieu lui accorde Sa miséricorde.

À mon frère Mohamed et à ma sœur Sylia, pour leur soutien, leurs encouragements  
et leur présence à mes côtés.

Et à

Tous mes amis, pour leur aide précieuse, leur bienveillance  
et leur accompagnement tout au long de ce parcours.

# Table des Matières

<b>Introduction Générale</b>	<b>6</b>
<b>1 Optimisation convexe et non convexe</b>	<b>7</b>
1.1 introduction . . . . .	7
1.2 Généralités sur les matrices . . . . .	8
1.2.1 Mineurs principaux diagonaux d'une matrice . . . . .	10
1.2.2 Le gradient . . . . .	10
1.2.3 Matrice hessienne . . . . .	11
1.2.4 Matrice jacobienne . . . . .	11
1.3 Matrices (semi) définies positives, matrice (semi) définies négatives . . . . .	11
1.3.1 Matrice définie positive . . . . .	11
1.3.2 Matrice semi-définie positive . . . . .	11
1.3.3 Matrice définie négative . . . . .	12
1.3.4 Matrice semi-définie négative . . . . .	12
1.3.5 Matrice indéfinie . . . . .	12
1.4 Ensembles convexes, Polyèdres , Combinaison et Enveloppe convexes . . . . .	13
1.4.1 Ensembles convexes . . . . .	13
1.4.2 Polyèdres convexes . . . . .	13
1.4.3 Combinaison convexe . . . . .	13
1.4.4 Enveloppe convexe . . . . .	16
1.5 fonction convexe et non convexe . . . . .	16
1.5.1 Interprétation géométrique . . . . .	17
1.5.2 Propriétés des fonctions convexes et concaves . . . . .	18
1.6 Fonctions convexes différentiables . . . . .	20
1.7 Fonctions convexes deux fois différentiables . . . . .	21
1.8 Problèmes d'optimisation . . . . .	21
1.8.1 Cadre et vocabulaire . . . . .	21
1.8.2 Définitions : . . . . .	22
1.9 Problèmes convexes et non convexes . . . . .	23
1.9.1 Problèmes convexes . . . . .	23
1.9.2 Problèmes non convexes . . . . .	23
1.9.3 Conditions d'existence et d'unicité de la solution . . . . .	24
1.10 Problèmes d'optimisation sans contraintes . . . . .	24
1.10.1 Condition d'optimalité du premier ordre . . . . .	24
1.10.2 Condition d'optimalité du second ordre . . . . .	25
1.11 Problèmes d'optimisation avec contraintes . . . . .	26
1.11.1 Condition d'optimalité du premier ordre . . . . .	26
1.11.2 Conditions d'optimalité du second ordre . . . . .	27

1.12	Conclusion . . . . .	28
<b>2</b>	<b>Optimisation globale</b>	<b>29</b>
2.1	Introduction . . . . .	29
2.2	Cadre mathématique de l'optimisation globale . . . . .	30
2.2.1	Formulation Générale du Problème . . . . .	30
2.2.2	Définitions : . . . . .	30
2.3	Théorème d'existence d'un minimum global . . . . .	31
2.4	Méthodes déterministes d'optimisation globale . . . . .	31
2.4.1	Méthode de Branch and Bound . . . . .	32
2.4.2	Méthode d'approximation extérieure . . . . .	36
2.4.3	Méthode D.C : différence de fonctions convexes . . . . .	38
2.4.4	Méthode de Piyavskii . . . . .	41
2.5	Méthodes basées sur l'algorithme Branch and Bound . . . . .	48
2.5.1	Méthode $\alpha$ BB [6] ,[3], [2] . . . . .	48
2.5.2	Méthode de borne inférieure quadratique [26] : . . . . .	51
2.6	conclusion : . . . . .	53
<b>3</b>	<b>Exemples numériques</b>	<b>54</b>
3.1	Exemples d'application sur la méthode de branch and bound [12] . . . . .	54
3.2	Exemples d'application sur la Méthode de Piyavskii [12] . . . . .	57
3.2.1	Application numérique de l'algorithme (Cas sans contraintes) . . . . .	57
3.2.2	Application numérique de l'algorithme (Cas avec contraintes) . . . . .	58
3.3	Exemple d'application sur la méthode D.C : . . . . .	58
3.4	Exemple d'application d'approximation extérieure : . . . . .	63
3.5	Méthode de borne inférieure quadratique (QLB) . . . . .	65
	<b>Conclusion générale</b>	<b>67</b>
	<b>Bibliographie</b>	<b>68</b>

# Liste des figures

1.1	Définition des ensembles convexes et non convexes. . . . .	13
1.2	Enveloppe Convexe. . . . .	16
1.3	Fonction convexe. . . . .	17
1.4	Fonction concave. . . . .	17
1.5	Fonction non convexe et non concave. . . . .	17
1.6	Épigraphe de $f$ . . . . .	19
1.7	Extremum global est aussi un extremum local. . . . .	23
2.1	Minimum local et minimum global. . . . .	31
2.2	Exemple d'une arborescence de la subdivision des sous-ensembles. . . . .	35
2.3	Illustration de la méthode de Piyavskii–Shubert pour l'optimisation globale de fonctions lipschitziennes. . . . .	44
3.1	Exemple 02 , D.C et D.C.A ,F1 . . . . .	60
3.2	Exemple 02 , D.C et D.C.A ,F2 . . . . .	60
3.3	Exemple 02 , D.C et D.C.A ,F3 . . . . .	61
3.4	Exemple 02 , D.C et D.C.A ,F4 . . . . .	61
3.5	Exemple 02 , D.C et D.C.A ,F5 . . . . .	62
3.6	Exemple 02 , D.C et D.C.A ,F6 . . . . .	63
3.7	Comparaison de deux bornes inférieures $LB_q(x)$ et $LB_\alpha(x)$ , exemple 1. . .	65
3.8	Comparaison de deux bornes inférieures $LB_q(x)$ et $LB_\alpha(x)$ , exemple 2. . .	66

# Introduction Générale

L'optimisation est une branche des mathématiques cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou à maximiser une fonction sur un ensemble.

L'optimisation joue un rôle important en recherche opérationnelle (domaine à la frontière entre l'informatique, les mathématiques et l'économie), dans les mathématiques appliquées (fondamentales pour l'industrie et l'ingénierie), en analyse et en analyse numérique, en statistique pour l'estimation du maximum de vraisemblance d'une distribution, pour la recherche de stratégies dans le cadre de la théorie des jeux, ou encore en théorie du contrôle et de la commande.

De nombreux systèmes sensibles peuvent être modélisés mathématiquement et optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle choisi, de l'efficacité des algorithmes utilisés et des moyens de traitement numérique mis en œuvre. Dans ce contexte, il est essentiel de distinguer deux grandes catégories d'optimisation : **l'optimisation locale** et **l'optimisation globale**.

Ce mémoire se concentre principalement sur l'optimisation statique, qui utilise des méthodes indépendantes du temps. Il est structuré autour de trois chapitres principaux : **Dans le premier chapitre**, nous rappelons les généralités sur l'algèbre et quelques notions et propriétés d'analyse convexe. Et nous essayons d'étudier les problèmes d'optimisation convexes et non convexes (sans contraintes et avec contraintes) et les conditions d'existence et d'unicité ainsi que l'optimalité de chaque type de problème.

**le deuxième chapitre** est consacré à l'optimisation globale.

**Le troisième chapitre** est consacré à présenter quelques exemples numériques. Puis enfin **une conclusion générale** sur le travail effectué.

# Chapitre 1

## Optimisation convexe et non convexe

### 1.1 introduction

L'optimisation mathématique vise à déterminer un optimum (maximum ou minimum) parmi un ensemble de solutions admissibles. Elle repose sur deux approches principales :

- **L'optimisation sans contraintes**, où l'on cherche à maximiser ou minimiser une fonction sur  $\mathbb{R}$  ou plus généralement sur  $\mathbb{R}^n$ .
- **L'optimisation avec contraintes**, où la solution doit satisfaire des conditions spécifiques imposées par un domaine donné.

Dans ce cadre, les matrices jouent un rôle fondamental en permettant notamment :

- La modélisation des systèmes d'équations sous-jacents,
- L'expression des conditions d'optimalité,
- L'analyse et la caractérisation des propriétés de convexité.

Ces deux approches de l'optimisation s'appuient sur des concepts essentiels tels que **la convexité et la non-convexité**. Ces notions feront l'objet du premier chapitre, qui abordera les bases théoriques nécessaires. En particulier, la compréhension approfondie de ces principes, à travers l'algèbre matricielle, est un élément clé pour maîtriser les méthodes avancées d'optimisation.

## 1.2 Généralités sur les matrices

**Définition 1.2.1** (Matrice). Une matrice  $A$  à  $m$  lignes et  $n$  colonnes est un tableau rectangulaire de  $m \times n$  nombres, rangés ligne par ligne. Il y a  $m$  lignes, et dans chaque ligne  $n$  nombres.

On dit que la matrice  $A$  a  $m$  lignes et  $n$  colonnes, ou qu'elle est de **dimension** ou de **taille**  $(m, n)$ . En notant  $a_{ij}$  les éléments de  $A$ , la matrice peut alors être notée

$$A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$$

La disposition générale des coefficients d'une matrice  $A$  de taille  $(m, n)$  est donc la suivante

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

Les coefficients  $a_{ij}$  avec  $i = j$  sont dits **diagonaux**.

**Exemple :** est représentée ci-dessous une matrice  $A$ , à coefficients entiers, et de dimension  $(3, 4)$

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \end{bmatrix}$$

Une matrice pour laquelle le nombre  $m$  de lignes est égal au nombre  $n$  de colonnes sera dite *matrice carrée* de taille  $n$ . Une matrice ne comportant qu'une seule ligne et  $n$  colonnes est appelée *matrice ligne* de taille  $n$ . Une matrice comportant  $m$  lignes et une seule colonne est appelée *matrice colonne* de taille  $m$ .

**Définition 1.2.2** (Matrice transposée). On appelle une matrice transposée de  $A$  la matrice  $A^t$  de dimension  $(n, m)$

$$A^t = (a_{ji})_{1 \leq j \leq n, 1 \leq i \leq m}$$

**Exemple :** Avec la matrice  $A$  de l'exemple précédent, on a

$$A^t = \begin{pmatrix} 0 & 4 & 8 \\ 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \end{pmatrix}$$

**Définition 1.2.3** (Matrice symétrique). Une matrice  $A$  est symétrique si  $A^t = A$

**Définition 1.2.4** (Matrice identité). Soit  $n \in \mathbb{N}^*$ . La **matrice identité** d'ordre  $n$ , notée  $\mathbf{I}_n$ , est la matrice carrée de taille  $n \times n$  définie par :

$$\mathbf{I}_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix},$$

où les éléments de la diagonale principale valent 1 et tous les autres éléments valent 0. Formellement, ses coefficients sont donnés par le *symbole de Kronecker* :

$$(\mathbf{I}_n)_{ij} = \delta_{ij} = \begin{cases} 1 & \text{si } i = j, \\ 0 & \text{sinon.} \end{cases}$$

**Définition 1.2.5** (Déterminant d'une matrice). Soit  $A$  une matrice carrée de taille  $n \times n$  à coefficients dans un corps commutatif  $\mathbb{K}$  (par exemple  $\mathbb{R}$  ou  $\mathbb{C}$ ) :

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Le **déterminant** de  $A$ , noté  $\det(A)$  ou  $|A|$ , est défini **récurivement** par :

1. **Cas de base** ( $n = 1$ ) :

$$\det(A) = a_{11}.$$

2. **Cas général** ( $n \geq 2$ ) :

On peut développer selon la  **$i$ -ème ligne** (pour un  $i$  fixé) :

$$\det(A) = \sum_{j=1}^n (-1)^{i+j} a_{ij} \det(M_{ij}),$$

où :

- $M_{ij}$  est la **sous-matrice** obtenue en supprimant la  $i$ -ème ligne et la  $j$ -ème colonne de  $A$ ,
- $\det(M_{ij})$  est le **mineur** de  $a_{ij}$ ,
- $(-1)^{i+j} \det(M_{ij})$  est le **cofacteur** de  $a_{ij}$ .

On peut aussi développer selon la  **$j$ -ème colonne** (pour un  $j$  fixé) :

$$\det(A) = \sum_{i=1}^n (-1)^{i+j} a_{ij} \det(M_{ij}).$$

**Exemple** : Soit  $A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$ . Le déterminant de la matrice  $A$  s'écrit :

$$\det(A) = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}.$$

$$= a(ei - hf) - b(di - gf) + c(dh - ge).$$

**Définition 1.2.6** (Rang d'une matrice).

on définit le rang de  $A$  par le nombre max de lignes ou de colonnes qui sont linéairement indépendantes et aussi le plus grand ordre d'un déterminant non nul que l'on peut extraire de  $A$

### 1.2.1 Mineurs principaux diagonaux d'une matrice

Soit  $A$  une matrice carrée symétrique de dimension  $(n, n)$ . Le mineur principal diagonal d'ordre  $k$  de la matrice  $A$  est le déterminant de la matrice de taille  $(k, k)$  obtenue en éliminant les  $n - k$  dernières lignes et  $n - k$  dernières colonnes de la matrice  $A$ . Une matrice carrée d'ordre  $n$  admet  $n$  mineurs principaux diagonaux.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Le mineur principal diagonal d'ordre 1 de  $A$  est :  $a_{11}$ .

Le mineur principal diagonal d'ordre 2 de  $A$  est :

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}.$$

Le mineur principal diagonal d'ordre 3 de  $A$  est :

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}.$$

### 1.2.2 Le gradient

Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction différentiable. Le gradient de  $f$  par rapport à la matrice  $\mathbf{x} = (x_{ij}) \in \mathbb{R}^{m \times n}$  est la matrice :

$$\nabla_{\mathbf{x}} f = \begin{pmatrix} \frac{\partial f}{\partial X_{11}} & \frac{\partial f}{\partial X_{12}} & \cdots & \frac{\partial f}{\partial X_{1n}} \\ \frac{\partial f}{\partial X_{21}} & \frac{\partial f}{\partial X_{22}} & \cdots & \frac{\partial f}{\partial X_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial X_{m1}} & \frac{\partial f}{\partial X_{m2}} & \cdots & \frac{\partial f}{\partial X_{mn}} \end{pmatrix} \in \mathbb{R}^{m \times n},$$

où chaque élément  $(i, j)$  est la dérivée partielle de  $f$  par rapport à l'élément  $x_{ij}$  de  $\mathbf{x}$ .

### 1.2.3 Matrice hessienne

Soit  $f$  définie de  $\mathbb{R}^n$  à  $\mathbb{R}$ . On appelle matrice hessienne  $H(x)$  de  $f$  la matrice des dérivées secondes de  $f$  évaluées au point  $x = (x_1, x_2, \dots, x_n)^t$ .

$$H(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix}$$

La matrice hessienne de  $f$  est une matrice symétrique d'ordre  $n$ .

### 1.2.4 Matrice jacobienne

Soit  $G = (g_1, g_2, \dots, g_m)$  une fonction définie de  $\mathbb{R}^n$  dans  $\mathbb{R}^m$ . À tout vecteur  $x^* = (x_1, x_2, \dots, x_n)$ , la fonction  $G$  associe le vecteur de fonctions  $(g_1(x^*), g_2(x^*), \dots, g_m(x^*))$ . On appelle matrice jacobienne de  $G$  la matrice de dimension  $(m, n)$ ,  $J_G(x)$  des dérivées partielles d'ordre 1 des  $m$  fonctions qui composent  $G$  :

$$J_G(x) = \begin{pmatrix} \frac{\partial g_1}{\partial x_1}(x^*) & \frac{\partial g_1}{\partial x_2}(x^*) & \cdots & \frac{\partial g_1}{\partial x_n}(x^*) \\ \frac{\partial g_2}{\partial x_1}(x^*) & \frac{\partial g_2}{\partial x_2}(x^*) & \cdots & \frac{\partial g_2}{\partial x_n}(x^*) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x^*) & \frac{\partial g_m}{\partial x_2}(x^*) & \cdots & \frac{\partial g_m}{\partial x_n}(x^*) \end{pmatrix}$$

## 1.3 Matrices (semi) définies positives, matrice (semi) définies négatives

### 1.3.1 Matrice définie positive

Soit  $A$  une matrice carrée symétrique d'ordre  $n$ . Soit  $X$  un vecteur colonne de  $\mathbb{R}^n$ . On note  $X^t$  sa transposée.  $A$  est dite définie positive si et seulement si :

$$X^t A X > 0 \quad \forall X \neq 0$$

Les éléments diagonaux  $a_{ii}$  d'une matrice définie positive sont tous  $> 0$ .

### 1.3.2 Matrice semi-définie positive

Soit  $A$  une matrice carrée symétrique d'ordre  $n$ . Soit  $X$  un vecteur colonne de  $\mathbb{R}^n$ . On note  $X^t$  sa transposée. Une matrice  $A$  est dite semi-définie positive si et seulement si :

$$X^t A X \geq 0 \quad \forall X \neq 0$$

Les éléments diagonaux  $a_{ii}$  d'une matrice semi-définie positive sont tous  $\geq 0$

### 1.3.3 Matrice définie négative

Soit  $A$  une matrice carrée symétrique d'ordre  $n$ . Soit  $X$  un vecteur colonne de  $\mathbb{R}^n$ . On note  $X^t$  sa transposée.  $A$  est dite définie négative si et seulement si :

$$X^t A X < 0 \quad \forall X \neq 0$$

Les éléments diagonaux  $a_{ii}$  d'une matrice définie négative sont tous  $< 0$ .

### 1.3.4 Matrice semi-définie négative

Soit  $A$  une matrice carrée symétrique d'ordre  $n$ . Soit  $X$  un vecteur colonne de  $\mathbb{R}^n$ . On note  $X^t$  sa transposée. Une matrice  $A$  est dite semi-définie négative si et seulement si :

$$X^t A X \leq 0 \quad \forall X \neq 0$$

Les éléments diagonaux  $a_{ii}$  d'une matrice semi-définie négative sont tous  $\leq 0$ .

### 1.3.5 Matrice indéfinie

Soit  $A$  une matrice carrée symétrique d'ordre  $n$ . On dit que  $A$  est indéfinie s'il existe  $X$  et  $Y$  deux vecteurs colonne quelconques de  $\mathbb{R}^n$  tel que :

$$X^t A X > 0 \quad \text{et} \quad Y^t A Y < 0$$

#### Caractérisation :

Soit  $A$  une matrice carrée symétrique d'ordre  $n$ .

- $A$  définie positive  $\iff$  ses  $n$  mineurs principaux  $D_k$  d'ordre successifs sont  $> 0$ .
- $A$  semi-définie positive  $\iff$  tous ses mineurs principaux  $D_k$  d'ordre successifs sont  $\geq 0$ .
- $A$  définie négative  $\iff (-1)^k D_k > 0$ .
- $A$  semi-définie négative  $\iff (-1)^k D_k \geq 0$ .

#### **Théorème 1.2.1** [14]

*Soit  $A$  une matrice carrée symétrique*

- $A$  est définie positive si toutes ses valeurs propres sont strictement positives.
- $A$  est semi-définie positive si toutes ses valeurs propres sont positives ou nulles, dont l'une au moins est nulle.
- $A$  est définie négative si toutes ses valeurs propres sont strictement négatives.
- $A$  est semi-définie négative si toutes ses valeurs propres sont négatives ou nulles, dont l'une au moins est nulle.

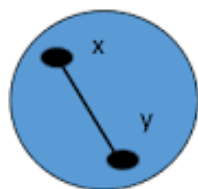
## 1.4 Ensembles convexes, Polyèdres , Combinaison et Enveloppe convexes

### 1.4.1 Ensembles convexes

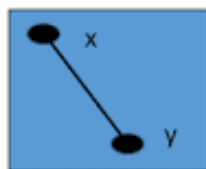
**Définition 1.4.1.1.** Un ensemble  $E \in \mathbb{R}^n$  est dit convexe si pour tout couple  $(x, y) \in E$  et  $\lambda \in [0, 1]$ , on a :

$$\lambda x + (1 - \lambda)y \in E$$

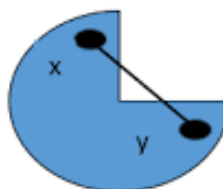
*Géométriquement, cette notion s'interprète comme suit : "Pour tout segment reliant deux points quelconques  $x$  et  $y$  de  $E$ , le segment  $[x; y]$  doit être aussi inclus dans  $E$ "*



Ensemble convexe



Ensemble convexe



Ensemble non convexe

FIGURE 1.1 – Définition des ensembles convexes et non convexes.

### 1.4.2 Polyèdres convexes

On rappelle qu'un *polyèdre convexe* dans  $\mathbb{R}^n$  est un ensemble  $P$  de la forme :

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}, \text{ avec } Am \times n \text{ et } b \in \mathbb{R}^m.$$

et l'inégalité  $Ax \leq b$  se lit composante par composante dans  $\mathbb{R}^m$  :  $(Ax)_i \leq b_i$ , pour tout  $i \in \{1, \dots, m\}$ .

*Géométriquement, un polyèdre convexe est donc l'intersection d'un nombre fini de demi-espaces de  $\mathbb{R}^n$ .*

### 1.4.3 Combinaison convexe

**Définition 1.4.3.1** Soit  $\lambda_i \geq 0$ ,  $\sum_{i=1}^m \lambda_i = 1$ . Alors  $\sum_{i=1}^m \lambda_i X_i$  est appelée **combinaison convexe** des points  $X_i$ ,  $i = 1, \dots, m$ .

**Théorème 1.4.3.1**[25] *Un ensemble  $K \subset \mathbb{R}^n$  est convexe si et seulement si toute combinaison convexe de points de  $K$  appartient à  $K$ .*

$$K \text{ convexe} \iff (\forall X_i \in K, \forall \lambda_i \geq 0 \text{ avec } \sum \lambda_i = 1, \sum \lambda_i X_i \in K)$$

Preuve :

$$* K \text{ convexe} \Leftrightarrow \left( \begin{array}{l} \forall X_i \in K, \\ \forall \lambda_i \geq 0 \text{ tel que } \sum \lambda_i = 1, \\ \sum \lambda_i X_i \in K \end{array} \right)$$

$\forall n \in \mathbb{N}^*$ , en particulier c'est vrai pour  $n = 2$ ,

$$\begin{aligned} \lambda_1 x_1 + \lambda_2 x_2 &\in K, \lambda_1, \lambda_2 \geq 0, \\ \lambda_1 x_1 + (1 - \lambda_2) x_2 &\in K, \lambda_1 + \lambda_2 = 1, \\ &\Rightarrow K \text{ est convexe.} \end{aligned}$$

$$** K \text{ convexe} \Rightarrow (\forall X_i \in K, \forall \lambda_i \geq 0 \text{ avec } \sum \lambda_i = 1, \sum \lambda_i X_i \in K)$$

on utilise un raisonnement par récurrence :

Pour  $n = 2$  et vrai car  $K$  est convexe.

On suppose Vrai à l'ordre  $n$ .

$$\text{C.à-d : } \sum \lambda_i x_i \in K, \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1$$

et on montre que c'est vrai à l'ordre  $n + 1$

$$\text{c.à-d : } \sum_{i=1}^{n+1} \lambda_i x_i \in K, \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1$$

On a :

$$\sum_{i=1}^{n+1} \lambda_i x_i = \sum_{i=1}^n \lambda_i x_i + \lambda_{n+1} x_{n+1}$$

Alors on a deux cas :

**1er cas :**

$$\lambda_{n+1} = 0$$

on a :

$$\lambda_{n+1} = 0 \implies \sum_{i=1}^{n+1} \lambda_i x_i = \sum_{i=1}^n \lambda_i x_i \in K$$

d'après l'hypothèse de récurrence.

**2ème cas :**

$$\lambda_{n+1} \neq 0, \text{ et } \lambda_{n+1} \neq 1, \text{ car } n > 2$$

on a :

$$\begin{aligned} \sum_{i=1}^{n+1} \lambda_i &= \sum_{i=1}^n \lambda_i X_i + \lambda_{n+1} X_{n+1} \\ &= \sum_{j=1}^n \lambda_j \cdot \frac{\sum \lambda_i X_i}{\sum_{j=1}^n \lambda_j} + \lambda_{n+1} X_{n+1} \\ &= \sum_{j=1}^n \lambda_j \cdot \sum_{i=1}^n \left( \frac{\lambda_i}{\sum_{j=1}^n \lambda_j} \right) X_i + \lambda_{n+1} X_{n+1} \end{aligned}$$

on pose  $\mu_i = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j}$ ,  $\mu_i \geq 0$ ,  $\forall i = 1, \dots, n$

$$\sum_{i=1}^n \mu_i = \sum_{i=1}^n \frac{\lambda_i}{\sum_{j=1}^n \lambda_j} = \frac{\sum_{i=1}^n \lambda_i}{\sum_{j=1}^n \lambda_j} = \frac{\lambda_1 + \lambda_2 + \dots + \lambda_n}{\lambda_1 + \lambda_2 + \dots + \lambda_n} = 1.$$

$\sum_{i=1}^n \mu_i x_i$  : est une combinaison linéaire convexe des points  $X_1, \dots, X_n$

Par hypothèse de récurrence,  $\sum_{i=1}^n \mu_i x_i \in K$

$$\sum_{i=1}^n \mu_i x_i = y \in K$$

$$\begin{aligned} \sum_{i=1}^n \lambda_i x_i &= \sum_{j=1}^n \lambda_j \sum_{i=1}^n \left( \frac{\lambda_i}{\sum_{j=1}^n \lambda_j} \right) x_i + \lambda_{n+1} x_{n+1} \\ &= \sum_{j=1}^n \lambda_j \sum_{i=1}^n \mu_i x_i + \lambda_{n+1} x_{n+1} \end{aligned}$$

$$= \left( \sum_{j=1}^n \lambda_j \right) y + \lambda_{n+1} x_{n+1} \in K, \text{ car } y \text{ et } x_{n+1} \in k, \text{ et}$$

$$\sum_{j=1}^n \lambda_j + \lambda_{n+1} = 1, \sum_{j=1}^n \lambda_j \geq 0, \lambda_{n+1} \geq 0.$$

**Théorème 1.4.3.2**[28] *L'intersection d'un nombre fini ou infini d'ensembles convexes est convexe.*

**Remarque :** la réunion d'ensembles convexes n'est pas toujours convexe

**Théorème 1.4.3.3**[28] *Soit  $E$  un ensemble convexe. Alors le produit  $\alpha E$ , où  $\alpha$  est un nombre réel, est un ensemble convexe.*

**Théorème 1.4.3.4** *La somme  $E = E_1 + E_2$  de deux ensembles convexes  $E_1, E_2 \subset \mathbb{R}^n$  est convexe.*

## 1.4.4 Enveloppe convexe

**Définition 1.4.4.1** L'enveloppe convexe d'un ensemble arbitraire  $A$  de  $\mathbb{R}^n$  est le plus petit convexe (au sens de l'inclusion) qui contient  $A$ . Elle est notée  $\text{conv}(A)$ .

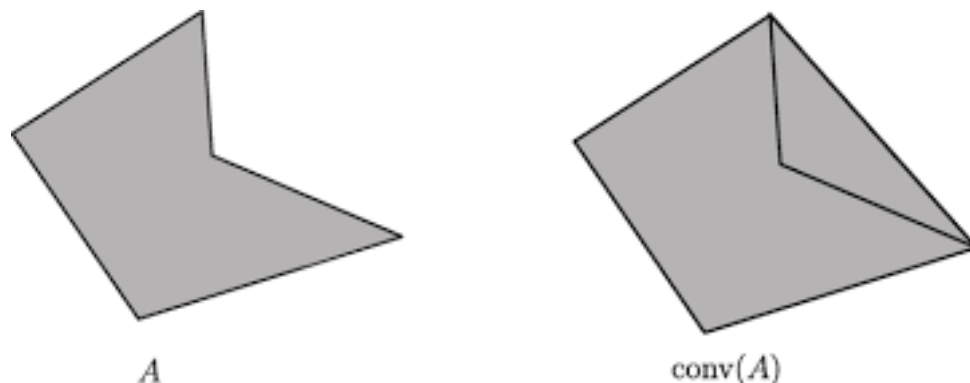


FIGURE 1.2 – Enveloppe Convexe.

**Théorème 1.4.4.1** [25] L'enveloppe convexe  $\text{conv}(A)$  d'un ensemble  $A$  dans  $\mathbb{R}^n$  est l'ensemble de toutes les combinaisons convexes des points de  $A$ .

## 1.5 fonction convexe et non convexe

**Définition 1.5.1** (Fonction convexe). Soit  $E \subset \mathbb{R}^n$  un ensemble convexe. Soit  $f : E \rightarrow \mathbb{R}$  une fonction à valeurs réelles.

—  $f$  est **convexe** si :

$$\forall x, y \in E, \forall \lambda \in [0, 1], \quad f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (1.1)$$

—  $f$  est **strictement convexe** si :

$$\forall x, y \in E, x \neq y, \forall \lambda \in ]0, 1[, \quad f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y). \quad (1.2)$$

**Définition 1.5.2** (Fonction concave). Soit  $E \subset \mathbb{R}^n$  un ensemble convexe. Soit  $f : E \rightarrow \mathbb{R}$  une fonction à valeurs réelles.

—  $f$  est **concave** si :

$$\forall x, y \in E, \forall \lambda \in [0, 1], \quad f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y). \quad (1.3)$$

—  $f$  est **strictement concave** si :

$$\forall x, y \in E, x \neq y, \forall \lambda \in ]0, 1[, \quad f(\lambda x + (1 - \lambda)y) > \lambda f(x) + (1 - \lambda)f(y). \quad (1.4)$$

**Remarque 1.5.1** : Il existe des fonctions qui sont ni convexes ni concaves

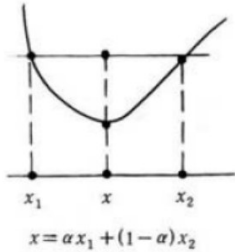


FIGURE 1.3 – Fonction convexe.

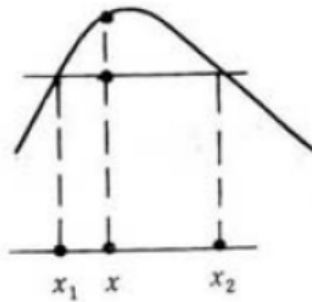


FIGURE 1.4 – Fonction concave.

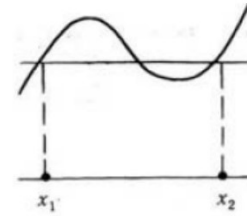


FIGURE 1.5 – Fonction non convexe et non concave.

**Remarque :** Par convention, la fonction identiquement égale à  $+\infty$  ou  $-\infty$  est convexe et concave au même temps.

**Exemples :**

- Sur  $\mathbb{R}$ , la fonction  $x \mapsto x^2$  est strictement convexe.
- Sur  $\mathbb{R}$ , la fonction  $x \mapsto |x|$  est convexe mais pas strictement convexe.

## 1.5.1 Interprétation géométrique

### 1.5.1.1 Interprétation géométrique des fonctions convexes

Soit  $S = [a, b]$  un intervalle. On a l'équation de la droite qui passe par

$$(X_1, f(X_1)) \text{ et } (X_2, f(X_2)) \quad \frac{y - f(X_1)}{f(X_2) - f(X_1)} = \frac{X - X_1}{X_2 - X_1}$$

$$Y = \frac{X - X_1}{X_2 - X_1}(f(X_2) - f(X_1)) + f(X_1)$$

Soit le point  $\lambda X_1 + (1 - \lambda)X_2 \in ]X_1, X_2[$

$$Y_{\lambda X_1 + (1-\lambda)X_2} = \frac{\lambda X_1 + (1 - \lambda)X_2 - X_1}{X_2 - X_1}(f(X_2) - f(X_1)) + f(X_1)$$

$$= \frac{(\lambda - 1)X_1 + (1 - \lambda)X_2}{X_2 - X_1}(f(X_2) - f(X_1)) + f(X_1)$$

$$= \frac{(1 - \lambda)(X_2 - X_1)}{X_2 - X_1}(f(X_2) - f(X_1)) + f(X_1)$$

$$= (1 - \lambda)(f(X_2) - f(X_1)) + f(X_1)$$

$$= (1 - \lambda)f(X_2) - (1 - \lambda)(f(X_1) + f(X_1))$$

$$\begin{aligned}
&= (1 - \lambda)f(X_2) + \lambda f(X_1) \\
&\geq f(\lambda X_1 + (1 - \lambda)X_2)
\end{aligned}$$

La corde joignant les points  $(X_1, f(X_1))$  et  $(X_2, f(X_2))$  se trouve au-dessus du graphe de  $f$ .

### 1.5.1.2 Interprétation géométrique des fonctions concaves

L'interprétation géométrique des fonctions concaves se fera de la même façon que celle des fonctions convexes ; et on trouve que la corde joignant les deux points  $(X_1, f(X_1))$  et  $(X_2, f(X_2))$  est toujours au-dessous du graphe de  $f$ .

Dans tout ce qui suit minimiser une fonction convexe  $f \Leftrightarrow$  maximiser une fonction concave  $-f$

## 1.5.2 Propriétés des fonctions convexes et concaves

### Propositions 1.5.2.1 [25]

Soit  $E$  un ensemble convexe de  $\mathbb{R}^n$ , et soient  $f_1$  et  $f_2$  deux fonctions convexes sur  $E$ , alors :

- $f_1 + f_2$  est convexe.
- $f_1 - f_2$  n'est pas nécessairement convexe.
- $\max\{f_1, f_2\}$  est convexe.
- $\min\{f_1, f_2\}$  n'est pas nécessairement convexe.

**Démonstration 1.5.2.1** Ces propositions sont démontrées en détail en [25].

**Proposition 1.5.2.2** [1] Soit  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction. Les propriétés suivantes sont équivalentes :

1.  $f$  est convexe.
2.  $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$ ,  $\forall x, y \in \mathbb{R}^n$ ,  $\forall \lambda \in [0, 1]$
3.  $f(\sum_{i=1}^n \lambda_i x_i) \leq \sum_{i=1}^n \lambda_i f(x_i)$ ,  $\forall \lambda_i \geq 0$ ,  $\sum_{i=1}^n \lambda_i = 1$ ,  $\forall x_i \in \mathbb{R}^n$ ,  $\forall n \in \mathbb{N}$

**Définition 1.5.2.1** (Épigraphe). Soit  $E$  un sous-ensemble de  $\mathbb{R}^n$  et soit  $f : E \rightarrow \mathbb{R}$ . On appelle épigraphe de  $f$ , notée  $\text{epi}(f)$ , le sous-ensemble de  $\mathbb{R}^{n+1}$  donné par :

$$\text{epi}(f) = \{(x, r); x \in E, r \in \mathbb{R} : f(x) \leq r\}.$$

**Théorème 1.5.2.1** [1] Soit  $E$  un ensemble convexe non vide de  $\mathbb{R}^n$  et  $f : E \rightarrow \mathbb{R}$ . Alors  $f$  est convexe si et seulement si son épigraphe  $\text{epi}(f)$  est convexe.

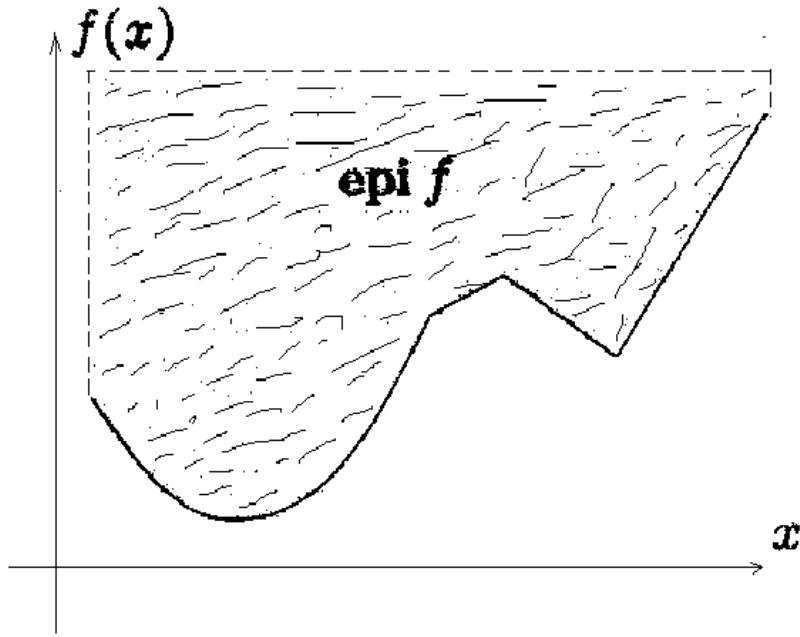


FIGURE 1.6 – Épigraphe de  $f$ .

**Démonstration 1.5.2.1** On doit montrer la double implication ((i)  $f$  est convexe  $\Leftrightarrow$  (ii)  $\text{épi}(f)$  est convexe.)

(i)  $\Rightarrow$   $f$  est convexe  $\Rightarrow \text{épi}(f)$  est un ensemble convexe. Soient  $(X_1, r_1)$  et  $(X_2, r_2) \in \text{épi}(f)$ ,  $\lambda \in [0, 1]$ , on montre que :

$$\lambda(X_1, r_1) + (1 - \lambda)(X_2, r_2) \in \text{épi}(f).$$

$$\begin{aligned} \lambda(X_1, r_1) + (1 - \lambda)(X_2, r_2) &= (\lambda X_1, \lambda r_1) + ((1 - \lambda)X_2, (1 - \lambda)r_2) \\ &= \lambda X_1 + (1 - \lambda)X_2, \lambda r_1 + (1 - \lambda)r_2 \in \text{épi}(f)(?) \end{aligned}$$

(Convexité de  $f$ )

$$f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2) \leq \lambda r_1 + (1 - \lambda)r_2$$

$$\Rightarrow (\lambda X_1 + (1 - \lambda)X_2, \lambda r_1 + (1 - \lambda)r_2) \in \text{épi}(f)$$

Donc  $\text{épi}(f)$  est convexe.

(ii)  $\Leftarrow$   $\text{épi}(f)$  convexe  $\Rightarrow f$  est convexe.

Soient  $X_1, X_2 \in S$ ,  $\lambda \in [0, 1]$ . On montre que :

$$f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2).$$

Soient  $(X_1, f(X_1))$  et  $(X_2, f(X_2)) \in \text{épi}(f)$ .

$\text{épi}(f)$  est convexe  $\Rightarrow (\lambda X_1, f(X_1)) + (1 - \lambda)(X_2, f(X_2)) \in \text{épi}(f)$ .

$$(\lambda X_1, f(X_1)) + ((1 - \lambda)(X_2, f(X_2))) \in \text{épi}(f)$$

$$(\lambda X_1 + (1 - \lambda)X_2, \lambda f(X_1) + (1 - \lambda)f(X_2)) \in \text{épi}(f).$$

$$\Rightarrow f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2).$$

Donc  $f$  est convexe.

**Définition 1.5.2.2 (Ensemble de niveau).** Soit  $E$  un ensemble convexe de  $\mathbb{R}^n$ , et soit  $f$  une fonction convexe sur  $E$ . On appelle **ensemble de niveau**  $\alpha$  l'ensemble :

$$E_\alpha = \{x \in E \mid f(x) \leq \alpha\}.$$

**Lemme 1.5.2.1** Si  $E \subset \mathbb{R}^n$  est un ensemble convexe et  $f : E \rightarrow \mathbb{R}$  une fonction convexe, alors l'ensemble de niveau  $E_\alpha = \{x \in E \mid f(x) \leq \alpha\}$  est un ensemble convexe.

**Démonstration de lemme 1.5.2.1** 1)  $f$  est convexe  $\Rightarrow \forall X_1, X_2 \in E, \forall \lambda \in [0, 1] :$

$$f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2).$$

Montrons que  $L^\alpha f$  est convexe  $\Rightarrow \forall X_1, X_2 \in L^\alpha f, \forall \lambda \in [0, 1]$

$$\lambda X_1 + (1 - \lambda)X_2 \in L^\alpha f, \quad f(\lambda X_1 + (1 - \lambda)X_2) \leq \alpha(?)$$

$$f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2) \quad (\text{Par la convexité de } f)$$

$$X_1 \in L^\alpha f \Rightarrow f(X_1) \leq \alpha$$

$$X_2 \in L^\alpha f \Rightarrow f(X_2) \leq \alpha$$

$$f(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda f(X_1) + (1 - \lambda)f(X_2) \leq \lambda \alpha + (1 - \lambda)\alpha \leq \alpha$$

$$\Rightarrow \lambda X_1 + (1 - \lambda)X_2 \in L^\alpha f. \text{ Donc } L^\alpha f \text{ est convexe.}$$

2)  $L^\alpha f$  convexe  $\Rightarrow f$  convexe (?)

*Cette implication n'est pas vraie ; car on peut avoir l'ensemble de niveau ( $L^\alpha$ ) convexe sans que la fonction  $f$  soit convexe.*

## 1.6 Fonctions convexes différentiables

**Théorème 1.6.1**[13] Si  $f$  est une fonction différentiable sur un ensemble convexe ouvert  $E \subset \mathbb{R}^n$ , alors  $f$  est convexe si et seulement si pour chaque  $x_1, x_2 \in E :$

$$f(x_2) - f(x_1) \geq \nabla f(x_1)^t (x_2 - x_1).$$

**Définition 1.6.1** Soit  $E$  un ensemble non vide de  $\mathbb{R}^n$ , et soit  $f : E \rightarrow \mathbb{R}$ . Alors  $f$  est dite différentiable au point  $\bar{x} \in \text{int}(E)$  s'il existe un vecteur  $\nabla f(\bar{x})$ , appelé **vecteur gradient**, et une fonction  $\gamma : \mathbb{R}^n \rightarrow \mathbb{R}$  telles que :

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^t(x - \bar{x}) + \|x - \bar{x}\|\gamma(x, \bar{x}),$$

pour chaque  $x \in E$ , où

$$\lim_{x \rightarrow \bar{x}} \gamma(x, \bar{x}) = 0.$$

## 1.7 Fonctions convexes deux fois différentiables

**Théorème 1.7.1**[13] Soit  $E$  un ensemble convexe non vide de  $\mathbb{R}^n$ , et  $f : E \rightarrow \mathbb{R}$  une fonction deux fois différentiable sur  $E$ . Alors  $f$  est convexe si et seulement si la matrice Hessienne  $H(x)$  est semi définie positive en chaque point de  $E$  :

$$y^t \nabla^2 f(x) y \geq 0, \quad \forall y.$$

**Définition 1.7.1** Soit  $E$  un ensemble non vide de  $\mathbb{R}^n$ , et soit  $f : E \rightarrow \mathbb{R}$ . Alors  $f$  est dit **deux fois différentiable** au point  $\bar{x} \in \text{int}(E)$  s'il existe un vecteur  $\nabla f(\bar{x})$ , et une matrice symétrique  $H(x)$   $n \times n$ , appelée la **matrice Hessienne**, et une fonction  $\gamma : \mathbb{R}^n \rightarrow \mathbb{R}$  telles que :

$$f(x) = f(\bar{x}) + \nabla f(\bar{x})^t(x - \bar{x}) + \frac{1}{2}(x - \bar{x})^t H(\bar{x})(x - \bar{x}) + \|x - \bar{x}\|^2 \gamma(\bar{x}, x - \bar{x}),$$

pour chaque  $x \in E$ , où

$$\lim_{x \rightarrow \bar{x}} \gamma(\bar{x}, x - \bar{x}) = 0.$$

## 1.8 Problèmes d'optimisation

### 1.8.1 Cadre et vocabulaire

On appelle *problème d'optimisation* tout problème de la forme :

$$\begin{cases} \text{Trouver } x^* \text{ tel que} \\ x^* \in U \text{ et } f(x^*) = \min_{x \in U} f(x) \end{cases} \quad (1.5)$$

Où  $U$  est une partie donnée de  $\mathbb{R}^n$  et  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction donnée que l'on appelle *fonctionnelle* du problème d'optimisation.

Un programme d'optimisation s'écrit typiquement sous la forme (avec s.c. c'est-à-dire *sous contraintes*) :

$$(\mathcal{P}) \quad \begin{cases} \max_{x \in \mathbb{R}^n} f(x) \\ \text{s.c. } g_j \quad \forall j = 1, \dots, m \end{cases}$$

s'il s'agit d'un programme de **maximisation** sous contraintes et sous la forme :

$$(\mathcal{P}') \quad \begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.c.} \quad g_j \quad \forall j = 1, \dots, m \end{cases}$$

s'il s'agit d'un programme de **minimisation** sous contraintes.

La fonction  $f$  est appelée aussi *fonction objectif*. Le programme consiste à chercher les valeurs  $(x_1^*, x_2^*, \dots, x_n^*)$  pour lesquelles la valeur de cette fonction est maximale (ou minimale) sous les contraintes. On appelle **optimum** la solution d'un programme d'optimisation : il s'agit soit d'un **maximum**, soit d'un **minimum**.

Les contraintes peuvent prendre plusieurs formes distinctes :

- **contraintes en équations** :  $g_j(x_1, x_2, \dots, x_n) = 0 \quad \forall j = 1, \dots, m$ .
- **Contraintes en inéquations** :  $-g_j(x_1, x_2, \dots, x_n) \geq 0 \quad \forall j = 1, \dots, m$ .

Le but de l'optimisation est de proposer des algorithmes permettant d'approcher les solutions  $x^*$  au sens où, partant d'un vecteur initial  $x^{(0)}$  quelconque, on construit explicitement une suite de vecteurs  $(x^{(k)})_{k \geq 0}$  convergeant vers la solution  $x^*$ .

Le problème d'optimisation est **dit sans contraintes** si  $U = \mathbb{R}^n$  et **sous contraintes** sinon.

Les méthodes de résolution développées permettent également de trouver les valeurs maximales de fonctions  $f$ . Pour cela,, il suffit de remplacer  $f$  par  $-f$  puisque :

$$\max_{x \in U} f(x) = - \min_{x \in U} (-f(x))$$

## 1.8.2 Définitions :

**Maximum, minimum :**

- La valeur  $x^*$  qui résout le programme  $\mathcal{P}$  est un **maximum** de la fonction  $f$  sous les contraintes du programme.
- La valeur  $x^*$  qui résout le programme  $\mathcal{P}'$  est un **minimum** de la fonction  $f$  sous les contraintes du programme.

**Optimum local :**

- La variable  $x^*$  est un **maximum local** d'une fonction  $f$  définie sur l'ensemble convexe  $E$

$$\iff \exists \epsilon > 0 \quad \text{tel que} \quad f(x) \leq f(x^*) \quad \forall x \in E \quad \text{et} \quad |x - x^*| \leq \epsilon.$$

- La variable  $x^*$  est un **minimum local** d'une fonction  $f$  définie sur l'ensemble convexe  $E$

$$\iff \exists \epsilon > 0 \quad \text{tel que} \quad f(x) \geq f(x^*) \quad \forall x \in E \quad \text{et} \quad |x - x^*| \leq \epsilon.$$

**Optimum global :**

- La variable  $x^*$  est un **maximum global** d'une fonction  $f$  définie sur l'ensemble convexe  $E$

$$\iff f(x) \leq f(x^*) \quad \forall x \in E.$$

- La variable  $x^*$  est un **minimum global** d'une fonction  $f$  définie sur l'ensemble convexe  $E$

$$\iff f(x) \geq f(x^*) \quad \forall x \in E.$$

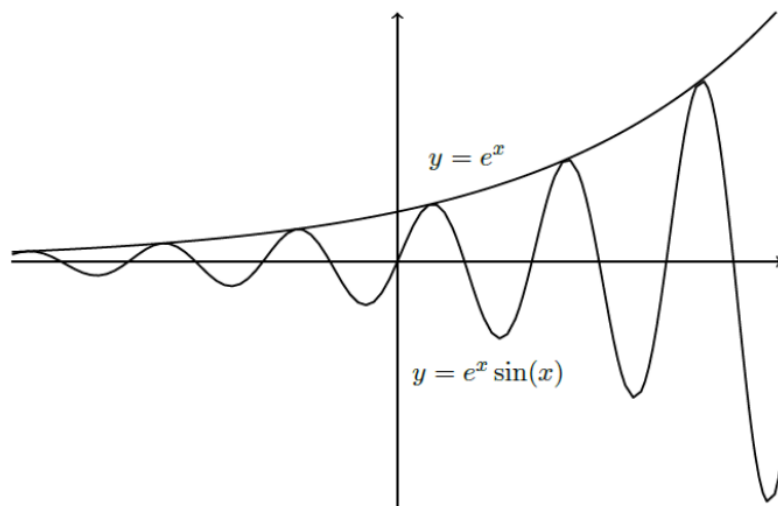


FIGURE 1.7 – Extremum global est aussi un extremum local.

Clairement **tout extremum global est aussi un extremum local**. La réciproque n'est pas toujours vraie, comme le montre l'exemple de la figure 1.7.

– L'application  $x \rightarrow \exp(x) \sin(x)$  a une infinité de minima locaux (en  $-\frac{\pi}{4}[2\pi]$ ) et de maxima locaux (en  $-\frac{3\pi}{4}[2\pi]$ ) mais aucun extremum global.

## 1.9 Problèmes convexes et non convexes

### 1.9.1 Problèmes convexes

**Définition 1.9.1.1** (Problème convexe) Un problème de la programmation mathématique est dit problème convexe s'il consiste à minimiser une fonction convexe (resp : maximiser une fonction concave) sur un domaine convexe fermé.

**Exemple :**

$$\min_{x \in \mathbb{R}} f(x) = x^2 \quad \text{sous la contrainte } x \geq 1.$$

Ici,  $f(x) = x^2$  est une fonction convexe, et la contrainte  $x \geq 1$  définit un ensemble convexe, Le problème donc est convexe.

### 1.9.2 Problèmes non convexes

**Définition 1.9.2.1** (Problème non convexe) Un problème de la programmation mathématique est dit problème non convexe si sa fonction objectif ou l'une des contraintes n'est pas convexe.

**Exemple :**

$$\min_{x \in \mathbb{R}} f(x) = \sin(x) \quad \text{sous la contrainte } x \in [0, 4\pi].$$

Ici,  $f(x) = \sin(x)$  est une fonction non convexe, et la contrainte  $x \in [0, 4\pi]$  définit un ensemble convexe, mais le problème reste non convexe en raison de la fonction objectif.

### 1.9.3 Conditions d'existence et d'unicité de la solution

La première question concernant un problème d'optimisation est celle de l'existence d'une solution. Si on cherche à minimiser une fonction  $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$  continue sur  $U$ , alors il est bien connu que si  $U$  est compact (*i.e.* fermé et borné) la fonction  $f$  est bornée et atteint ses bornes sur  $U$ . Elle admet donc au moins un minimum global  $x^* \in U$ . La notion de fonction *coercive* permet d'étendre ce type de raisonnement pour des fonctions définies sur des domaines non bornés.

**Définition 1.9.3.1 (Fonctions coercives).** Une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est dite *coercive* si

$$\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$$

ici  $\|\cdot\|$  désigne la norme de l'espace de Hilbert  $\mathbb{H}$ . Dans le cas où  $\mathbb{H} = \mathbb{R}^n$  les normes sont toutes équivalentes et  $\|\cdot\|$  désigne une norme quelconque de  $\mathbb{R}^n$ . On notera  $\|\cdot\|_p$  ( $p \in \mathbb{N}$ ) par

$$\forall x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n, \quad \|x\|_p = \left[ \sum_{i=1}^n |x_i|^p \right]^{\frac{1}{p}}$$

La norme infinie de  $\mathbb{R}^n$  est

$$\forall x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n, \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

## 1.10 Problèmes d'optimisation sans contraintes

Considérons le problème ( $\mathcal{P}$ ) d'optimisation sans contraintes sous la forme :

$$(\mathcal{P}) \quad \begin{cases} \min f(x) \\ x \in \mathbb{R}^n \end{cases}$$

- La fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est appelée fonction coût, objectif ou critère.
- Tout point  $x \in \mathbb{R}^n$  est appelé point admissible du problème ( $\mathcal{P}$ ).

### 1.10.1 Condition d'optimalité du premier ordre

On considère une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  que l'on souhaite minimiser (ou maximiser) sans aucune contrainte sur les variables. On cherche donc à trouver un point  $x^* \in \mathbb{R}^n$  tel que  $f(x^*)$  soit un minimum local. Si  $f$  est **différentiable** au point  $x^*$ , alors une condition nécessaire pour que  $x^*$  soit un minimum local est que le **gradient** de  $f$  en  $x^*$  soit nul. Mathématiquement, cela s'exprime par :

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

où  $\nabla f(x^*)$  est le vecteur gradient de  $f$  au point  $x^*$ , défini par :

$$\nabla f(x^*) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x^*) \\ \frac{\partial f}{\partial x_2}(x^*) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x^*) \end{pmatrix}$$

- Cette condition signifie qu’au point optimal  $x^*$ , la fonction  $f$  ne varie pas au premier ordre dans aucune direction. En d’autres termes,  $x^*$  est un point stationnaire.
- Cependant, cette condition n’est pas suffisante pour garantir que  $x^*$  est un minimum local. Elle peut également correspondre à un maximum local ou à un point selle. Pour déterminer la nature du point critique, il faut utiliser des conditions d’optimalité du second ordre (basées sur la matrice hessienne).

### Remarque 1.10.1.1

- Si la fonction  $f$  n’est pas différentiable en  $x^*$ , cette condition ne s’applique pas. Il faut alors utiliser d’autres outils (comme les sous-gradients pour les fonctions convexes non différentiables).

## 1.10.2 Condition d’optimalité du second ordre

On considère une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  deux fois différentiable. On suppose que  $x^*$  est un point critique, c’est-à-dire qu’il satisfait la condition d’optimalité du premier ordre :

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

Pour déterminer si  $x^*$  est un minimum local, un maximum local ou un point selle, on utilise la **matrice hessienne** de  $f$  en  $x^*$ , notée  $Hf(x^*)$ .

1. **Condition nécessaire pour un minimum local** : Si  $x^*$  est un minimum local de  $f$ , alors la matrice hessienne  $Hf(x^*)$  est **semi-définie positive**, c’est-à-dire :

$$\forall d \in \mathbb{R}^n, \quad d^\top Hf(x^*) d \geq 0$$

2. **Condition suffisante pour un minimum local** : Si la matrice hessienne  $Hf(x^*)$  est **définie positive**, c’est-à-dire :

$$\forall d \in \mathbb{R}^n \setminus \{0\}, \quad d^\top Hf(x^*) d > 0$$

alors  $x^*$  est un minimum local strict de  $f$ .

3. **Condition pour un maximum local** :

- Si  $Hf(x^*)$  est **semi-définie négative**, alors  $x^*$  peut être un maximum local.
- Si  $Hf(x^*)$  est **définie négative**, alors  $x^*$  est un maximum local strict.

4. **Point selle** : Si  $Hf(x^*)$  a des valeurs propres strictement positives et strictement négatives, alors  $x^*$  est un **point selle** (ni un minimum, ni un maximum).

**Exemple 1.10.2.1** : Considérons la fonction  $f(x, y) = x^2 + y^2$ . Le gradient est :

$$\nabla f(x, y) = \begin{pmatrix} 2x \\ 2y \end{pmatrix}$$

Le point critique est  $(0, 0)$ . La matrice hessienne est :

$$\nabla^2 f(x, y) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Cette matrice est définie positive (ses valeurs propres sont  $2 > 0$ ), donc  $(0, 0)$  est un minimum local strict.

## 1.11 Problèmes d'optimisation avec contraintes

### 1.11.1 Condition d'optimalité du premier ordre

On considère un problème d'optimisation avec contraintes de la forme suivante :

$$\min_{x \in \mathbb{R}^n} f(x)$$

sous les contraintes :

$$\begin{cases} g_i(x) \leq 0, & i = 1, \dots, m \\ h_j(x) = 0, & j = 1, \dots, p \end{cases}$$

où :

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est la fonction objectif à minimiser,
- $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$  sont les fonctions de contraintes d'inégalité,
- $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$  sont les fonctions de contraintes d'égalité.

**Conditions de Karush-Kuhn-Tucker (KKT)** Pour qu'un point  $x^* \in \mathbb{R}^n$  soit un minimum local, il doit satisfaire les conditions suivantes :

1. **Stationnarité** :

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) + \sum_{j=1}^p \mu_j \nabla h_j(x^*) = 0$$

où  $\lambda_i \in \mathbb{R}$  et  $\mu_j \in \mathbb{R}$  sont les multiplicateurs de Lagrange.

2. **Admissibilité primale** :

$$\begin{cases} g_i(x^*) \leq 0, & i = 1, \dots, m \\ h_j(x^*) = 0, & j = 1, \dots, p \end{cases}$$

3. **Admissibilité duale** :

$$\lambda_i \geq 0, \quad i = 1, \dots, m$$

4. **Condition de complémentarité** :

$$\lambda_i g_i(x^*) = 0, \quad i = 1, \dots, m$$

**Exemple 1.11.1.1** : Considérons le problème suivant :

$$\min_{x \in \mathbb{R}^2} f(x, y) = x^2 + y^2$$

sous la contrainte :

$$g(x, y) = x + y - 1 \leq 0$$

1. Le gradient de  $f$  est :

$$\nabla f(x, y) = \begin{pmatrix} 2x \\ 2y \end{pmatrix}$$

2. Le gradient de  $g$  est :

$$\nabla g(x, y) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

3. La condition de stationnarité donne :

$$\begin{pmatrix} 2x \\ 2y \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Ce qui implique  $x = y$ .

4. La condition de complémentarité impose  $\lambda(x + y - 1) = 0$ .

— Si  $\lambda = 0$ , alors  $x = y = 0$ , mais cela ne satisfait pas  $x + y \leq 1$ .

— Si  $x + y - 1 = 0$ , alors  $x = y = \frac{1}{2}$ .

Le point optimal est  $(\frac{1}{2}, \frac{1}{2})$ .

### 1.11.2 Conditions d'optimalité du second ordre

1. **Matrice hessienne du lagrangien** : Le lagrangien  $\mathcal{L}$  est défini par :

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \mu_j h_j(x).$$

La matrice hessienne du lagrangien par rapport à  $x$  est :

$$\nabla^2 \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 g_i(x^*) + \sum_{j=1}^p \mu_j^* \nabla^2 h_j(x^*).$$

2. **Condition nécessaire du second ordre** : Si  $x^*$  est un minimum local, alors pour tout vecteur  $d \in \mathbb{R}^n$  tel que :

$$\nabla g_i(x^*)^\top d = 0 \quad \text{pour les contraintes actives} \quad (g_i(x^*) = 0),$$

et

$$\nabla h_j(x^*)^\top d = 0 \quad \text{pour toutes les contraintes d'égalité},$$

on a :

$$d^\top \nabla^2 \mathcal{L}(x^*, \lambda^*, \mu^*) d \geq 0.$$

3. **Condition suffisante du second ordre** : Si pour tout vecteur  $d \in \mathbb{R}^n \setminus \{0\}$  tel que :

$$\nabla g_i(x^*)^\top d = 0 \quad \text{pour les contraintes actives} \quad (g_i(x^*) = 0),$$

et

$$\nabla h_j(x^*)^\top d = 0 \quad \text{pour toutes les contraintes d'égalité},$$

on a :

$$d^\top \nabla^2 \mathcal{L}(x^*, \lambda^*, \mu^*) d > 0,$$

alors  $x^*$  est un minimum local strict.

**Exemple 1.11.2.1 :** Considérons le problème suivant :

$$\min_{x \in \mathbb{R}^2} f(x, y) = x^2 + y^2$$

sous la contrainte :

$$h(x, y) = x + y - 1 = 0.$$

On suppose que  $x^*$  est un point critique satisfaisant les conditions KKT du premier ordre, et on souhaite déterminer si  $x^*$  est un minimum local.

1. Le lagrangien est :

$$\mathcal{L}(x, y, \mu) = x^2 + y^2 + \mu(x + y - 1).$$

2. Les conditions KKT du premier ordre donnent :

$$\nabla \mathcal{L} = \begin{pmatrix} 2x + \mu \\ 2y + \mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

et  $x + y = 1$ . On trouve  $x = y = \frac{1}{2}$  et  $\mu = -1$ .

3. La matrice hessienne du lagrangien est :

$$\nabla^2 \mathcal{L} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

4. Pour tout vecteur  $d = (d_1, d_2)$  tel que  $\nabla h(x, y)^\top d = d_1 + d_2 = 0$ , on a :

$$d^\top \nabla^2 \mathcal{L} d = 2d_1^2 + 2d_2^2 > 0.$$

Ainsi,  $(\frac{1}{2}, \frac{1}{2})$  est un minimum local strict.

## 1.12 Conclusion

Dans ce premier chapitre, nous avons exploré les concepts fondamentaux de l'optimisation convexe et non convexe. Nous avons introduit les notions de convexité des ensembles et des fonctions, ainsi que leurs implications sur la résolution des problèmes d'optimisation. La convexité joue un rôle clé en garantissant l'unicité et la stabilité des solutions optimales, tandis que la non-convexité introduit des défis supplémentaires nécessitant des approches plus complexes.

Nous avons également examiné les propriétés mathématiques des matrices définies positives et semi-définies positives, qui sont essentielles pour caractériser la convexité des fonctions différentiables. Enfin, nous avons posé les bases de l'optimisation en définissant les problèmes convexes et non convexes, ce qui nous permettra d'approfondir l'étude des méthodes de résolution adaptées dans les chapitres suivants.

# Chapitre 2

## Optimisation globale

### 2.1 Introduction

Durant les 20 dernières années, le domaine de la recherche sur « l'optimisation globale » s'est considérablement enrichi grâce, notamment, à l'accroissement de la puissance de calcul des ordinateurs. Ces progrès ont permis de résoudre des problèmes auparavant insolubles.

L'optimisation globale recherche la meilleure solution du domaine en entier, c'est-à-dire que dans tout le domaine il n'existe aucune solution qui lui soit meilleure tout en respectant les contraintes. Cette solution est appelée l'optimum global.

Il n'y a pas si longtemps, mettre au point des méthodes numériques permettant de déterminer la solution d'un problème d'optimisation non linéaire et non convexe pouvant répondre à un ensemble de contraintes, elles aussi non linéaires et non convexes, paraissait très difficile. Bien que la théorie classique de l'optimisation ne peut pas être appliquée directement dans les problèmes d'optimisation globale, les outils traditionnels tels que l'analyse convexe, sont largement utilisés dans la construction des méthodes d'optimisation globale. Cette approche constitue une partie importante de l'optimisation globale déterministe. Par exemple, un remarquable progrès a été accompli dans la construction des algorithmes de minimisation des fonctions concaves dans des régions convexes.

Ces dernières années, de nombreux travaux ont été réalisés. On peut les classer en deux grandes familles :

Les Méthodes déterministes, comme leur nom l'indique, nous offrent la certitude d'obtenir l'optimum global et ne laissent aucune place au hasard et conduiront pour un contexte initial donné à une même solution finale. Pour ces méthodes, l'exploration de l'espace de solutions se fait grâce à des procédures de recherche qui sont élaborées à partir de la constante de Lipschitz, des dérivées ou d'autres informations locales et globales concernant la fonction objectif. Les modèles déterministes ne traitent pas adéquatement les informations disponibles sur la fonction objectif.

Les Méthodes stochastiques peuvent souvent faire face à ce genre de problèmes plus facilement et plus efficacement que les algorithmes déterministes. Les algorithmes stochastiques explorent l'espace de solutions grâce en partie à des procédures de transitions aléatoires. Ainsi, plusieurs exécutions successives de ces algorithmes pourront conduire à des résultats différents (pour un même point initial). L'avantage des méthodes stochastiques est leur simplicité et leur pertinence pour les problèmes où les évaluations de la fonction

objectif sont corrompues par un bruit aléatoire, ainsi que leur robustesse à l'égard de la croissance de la dimension. L'inconvénient majeur de ces méthodes est qu'elles peuvent diverger, et passer plusieurs fois à côté de la solution. L'obtention de l'optimum global n'est pas garantie, il est seulement repéré avec une probabilité proche de 1. De nombreux algorithmes, où l'aléatoire et l'argument statistique sont impliqués ont été proposés. Ces algorithmes sont basés sur des analogies avec les processus naturels. Les exemples les plus connus de tels algorithmes sont l'algorithme génétique, le recuit simulé, et la recherche tabou. Les algorithmes stochastiques d'optimisation globale sont très populaires dans les applications.

Dans ce chapitre, on s'intéresse à quelques méthodes d'optimisation globale déterministes. Nous nous sommes concentrés sur la résolution de problèmes explicites, c'est-à-dire des problèmes dans lesquels toutes les expressions des équations de la fonction objectif sont connues de façon explicite, nous présenterons quelques méthodes déterministes d'optimisation globale basées sur l'utilisation des techniques de partition et d'élimination (Branch-and-Bound). Notre attention sera portée sur les méthodes de recouvrement ou lipschitziennes qui ont la réputation d'être efficaces en dimension 1. Parmi ces méthodes, il y a celles qui sont basées sur la construction des fonctions sous-estimateurs de la fonction objectif sur le domaine faisable.

## 2.2 Cadre mathématique de l'optimisation globale

### 2.2.1 Formulation Générale du Problème

Un problème d'optimisation globale consiste à rechercher la valeur minimale d'une fonction objectif  $f$  sur un domaine admissible  $\mathcal{D}$ . Ce problème s'écrit généralement sous la forme :

$$(P) \begin{cases} \min & f(x) \\ & x \in \mathcal{D} \end{cases}$$

où :

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction potentiellement non convexe, non linéaire, ou même non différentiable.
- $\mathcal{D} \subseteq \mathbb{R}^n$  est l'ensemble admissible, défini par des contraintes.

### 2.2.2 Définitions :

**Définition :** (Minimum global)

$x_0 \in \mathbb{R}^n$  est un minimum global si et seulement si  $x_0 \in \mathbb{R}^n$  est un minimum global de  $f$  sur  $\mathcal{D}$  si et seulement si  $x_0 \in \mathcal{D}$  et  $\forall x \in \mathcal{D}, f(x) \geq f(x_0)$

**Définition :** (Minimum local)

$x_0 \in \mathbb{R}^n$  est un minimum local de  $f$  sur  $\mathcal{D} \subset \mathbb{R}^n$  si et seulement si  $x_0 \in \mathcal{D}$  et  $\exists \nu_0$  un voisinage de  $x_0$  tel que  $\forall x \in \nu_0 \cap \mathcal{D}, f(x_0) \leq f(x)$ .

C'est clair qu'un minimum global est un minimum local, et le contraire n'est pas vrai, sauf pour les problèmes convexes.

## 2.3 Théorème d'existence d'un minimum global

**Théorème 2.3.1** (Weierstrass [22]). *Si  $f : E \mapsto \mathbb{R}$  est une fonction continue sur un ensemble non vide compact  $E$ , alors le problème (P) admet au moins une solution (c'est à dire la fonction  $f$  admet au moins un minimum global dans l'ensemble  $E$ ).*

**Théorème 2.3.2** [15]. *Soit  $f : \mathbb{R}^n \mapsto \mathbb{R}$  une fonction continue. Si  $\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty$ , alors le problème (P) admet au moins une solution (c'est à dire la fonction  $f$  admet au moins un minimum global dans  $\mathbb{R}^n$ ).*

**Théorème 2.3.3** [7]. *Si  $f : X \subset \mathbb{R}^n \mapsto \mathbb{R}$  est une fonction convexe sur un ensemble convexe  $X$ , alors tout minimum local est un minimum global de  $f$ .*

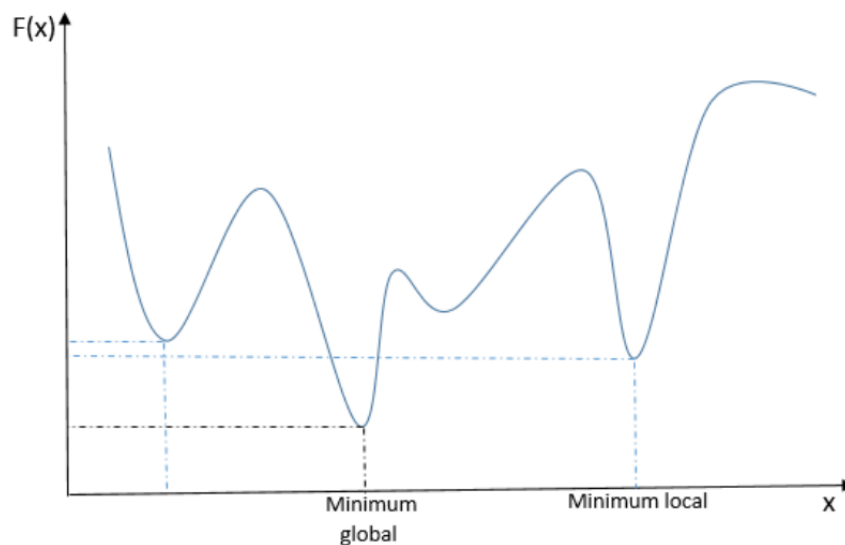


FIGURE 2.1 – Minimum local et minimum global.

## 2.4 Méthodes déterministes d'optimisation globale

Dans cette section, nous présentons plusieurs familles majeures de méthodes déterministes utilisées en optimisation globale :

- **La méthode de Branch and Bound**, l'une des plus classiques, repose sur une exploration arborescente intelligente de l'espace de recherche. Nous en détaillons l'historique, le principe, l'algorithme de base ainsi que les résultats de convergence.
- **La méthode d'approximation extérieure**, qui repose sur une succession d'optimisations convexes extérieures à la région admissible, est particulièrement efficace pour les problèmes non convexes à contraintes.

- **La programmation D.C. (Différence de Fonctions Convexes)**, qui repose sur la reformulation des fonctions non convexes comme différences de fonctions convexes, offre une approche élégante et puissante grâce à sa théorie de la dualité et à l'algorithme DCA.
- **La méthode de Piyavskii**, conçue initialement pour les fonctions univariées lipschitziennes, a été généralisée à des cas plus complexes. Elle permet une recherche adaptative et rigoureuse avec garanties de convergence.
- Enfin, nous explorons les **méthodes basées sur l'algorithme Branch and Bound**, notamment :
  - la **méthode  $\alpha$ BB**, qui combine des bornes inférieures basées sur une relaxation convexe avec une stratégie d'exploration hiérarchique ;
  - la **méthode de borne inférieure quadratique**, qui s'appuie sur des estimations quadratiques de la fonction objectif pour renforcer l'encadrement.

L'ensemble de ces méthodes constitue une base théorique et algorithmique robuste pour résoudre des problèmes complexes en optimisation globale, notamment en génie chimique, en mathématiques appliquées et en intelligence artificielle. Chacune sera détaillée dans les sections suivantes à travers ses principes, ses formulations, ses algorithmes et ses résultats théoriques de convergence.

## 2.4.1 Méthode de Branch and Bound

### ► Historique :

Les algorithmes Branch and Bound ont pris une place non négligeable dans le domaine de l'optimisation globale. Les raisons de cet engouement sont simples : il s'agit d'un principe à la fois basique et général, sur lequel peuvent venir se greffer de nombreuses idées pour améliorer la convergence.

Dans notre étude, nous nous intéresserons uniquement à la méthode Branch and Bound, une méthode permettant de résoudre une classe de problèmes d'optimisation globale ; il s'agit d'une approche basée sur la décomposition du domaine en sous-domaines.

Cette technique repose sur l'utilisation des intervalles, puisqu'à chaque étape, le problème courant est traité de manière globale, c'est-à-dire que le domaine (intervalle ou pavé) est considéré comme un élément du calcul pour l'évaluation.

### ► Introduction :

Branch and Bound est un algorithme assez général qui joue un rôle très important dans la théorie de l'optimisation globale. L'idée principale de cette méthode est de décomposer le problème initial en sous-problèmes parallèles (branching), qui peuvent être progressivement plus faciles à résoudre, puis d'évaluer les bornes inférieure et supérieure (bounding) des valeurs des solutions optimales associées à ces sous-problèmes.

Par conséquent, le procédé Branch and Bound peut être représenté sous la forme d'un arbre : le problème initial constitue la racine de cet arbre, et les branches représentent les sous-problèmes construits hiérarchiquement par l'algorithme. Cette construction est guidée par une stratégie de recherche qui détermine l'ordre dans lequel les sous-problèmes sont explorés.

La décomposition et la recherche de la solution se poursuivent tant qu'il est possible d'examiner de nouvelles branches susceptibles de conduire à une meilleure solution. Dès

qu'il est vérifié qu'une branche donnée ne peut produire une solution meilleure que celle déjà trouvée (appelée solution candidate), cette branche est élaguée.

Il est à noter que l'algorithme Branch and Bound a été initialement développé pour résoudre des problèmes de programmation entière. Par la suite, il a été appliqué avec succès à des problèmes beaucoup plus complexes en optimisation globale.

► **Le principe de la méthode branch and bound :**

Soit (P) le problème d'optimisation globale :

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

Où :  $H$  est un compact de  $\mathbb{R}^n$ .

$f : K \mapsto \mathbb{R}$  (avec  $H \subseteq K \subseteq \mathbb{R}^n$ ),  $f$  continue et non convexe.

L'algorithme de B&B consiste à engendrer deux suites convergentes  $\{UB_k\}$  et  $\{LB_k\}$  des bornes supérieure et inférieure respectivement de la valeur minimale de la fonction du problème (P).

$$\begin{cases} UB : \text{Upper bound} \\ LB : \text{Lower bound} \end{cases}$$

Une relaxation initiale  $R$  de l'ensemble réalisable  $H$  sera définie telle que :  $H \subset R$ .  $R$  est convexe, il peut être un simplexe, un rectangle, un cône, ... À chaque itération  $k$ , les problèmes des 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 inférieure et supérieure  $LB_{k_i}$  et  $UB_{k_i}$  seront calculées par la relaxation de  $f$  sur  $R_{k_i}$  et la relaxation de  $f$  localement sur le sous-ensemble réalisable  $R_{k_i} \cap H$  respectivement. Cette méthode utilise la stratégie "le meilleur d'abord".

En effet, les bornes inférieure et supérieure finales pour l'itération  $k$  seront données par :

$$\begin{cases} LB_k = \min LB_{k_i} \\ UB_k = \min UB_{k_i} \end{cases}$$

respectivement, et tout sous-ensemble sur lequel la borne inférieure dépasse  $UB_k$  sera éliminé, car  $\min f$  ne peut être atteint sur un tel sous-ensemble, on peut représenter cette méthode par une arborescence de racine  $R$  et de sommets les sous-ensembles  $R_k$  obtenus par subdivisions successives.

Au fait, cette méthode peut se représenter schématiquement par une arborescence qui a pour racine l'ensemble  $R$ , et pour sommets les sous-ensembles  $R_{k_i}$  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 nœud de l'arborescence créée, les bornes inférieure et supérieure seront obtenues par l'application d'une recherche locale.

Notons  $x^*$  la solution optimale du problème (P) pour ce qui suit.

**Remarque :** On peut subdiviser tout ensemble  $R_k$  en plusieurs sous-ensembles et non pas forcément en deux sous-ensembles.

► L'algorithme de base de la méthode branch and bound :

**Algorithme général**

On peut résumer la procédure précédente par les étapes suivantes :

- i. Construire l'ensemble  $R$  tel que :  $H \subset R$ .
- ii. Posons :  $k = 1$ ,  $I_k = \{R\}$ , fixer  $\varepsilon > 0$ .
- iii. Construire les problèmes des bornes inférieure et supérieure de  $\min f(x)$  sur  $R$ . Soient  $LB_k$ ,  $UB_k$  les solutions obtenues respectivement.
- iv. Si :  $UB_k - LB_k \leq \varepsilon$ , donc on s'arrête et on pose :

$$\min f(x) = UB_k \quad \text{et} \quad x^* = x^k \in \{x : f(x) = UB_k, \quad x \in H \cap R\}$$

- v. Sinon, subdiviser  $I_k$  en deux sous-ensembles (ou en un nombre fini de sous-ensembles)  $R_{k_1}$  et  $R_{k_2}$  tels que :

$$\bigcup_{i=1}^2 R_{k_i} = R \quad \text{et} \quad \overset{\circ}{R}_{k_1} \cap \overset{\circ}{R}_{k_2} = \emptyset$$

où  $\overset{\circ}{R}$  est l'intérieur de  $R$ .

- vi. Construire les problèmes des bornes inférieure et supérieure de  $\min f(x)$  sur  $H \cap R_{k_i}$ ,  $i = 1, 2$ . Soient  $LB_{k_1}$ ,  $UB_{k_1}$  et  $LB_{k_2}$ ,  $UB_{k_2}$  les solutions obtenues.
- vii. Posons :

$$\begin{cases} UB_{k+1} = \min\{UB_{k_1}, UB_{k_2}, UB_k\} \\ LB_{k+1} = \min\{LB_{k_1}, LB_{k_2}\} = LB_k \end{cases}$$

- viii. Posons :  $I_{k+1} = \{R_{k_1}, R_{k_2}\}$ .

- ix. Éliminer de  $I_k$  tout sous-ensemble  $R_{k_j}$ ,  $j = 1, 2$ , tel que :

$$LB_{k_j} > UB_{k+1} \quad \text{où} \quad H \cap R_{k_j} = \emptyset$$

et posons :  $I_{k+1} = R_{k_i}$ .

- x. Posons :  $k = k + 1$  et revenir à la quatrième étape de l'algorithme.

**Notation :**

- $R_k$  : Le sous-ensemble actuel.
- $LB_k$  : La borne inférieure (à la  $k^{\text{ième}}$  itération).
- $UB_k$  : La borne supérieure (à la  $k^{\text{ième}}$  itération).
- $x^k$  : La solution trouvée (à la  $k^{\text{ième}}$  itération).

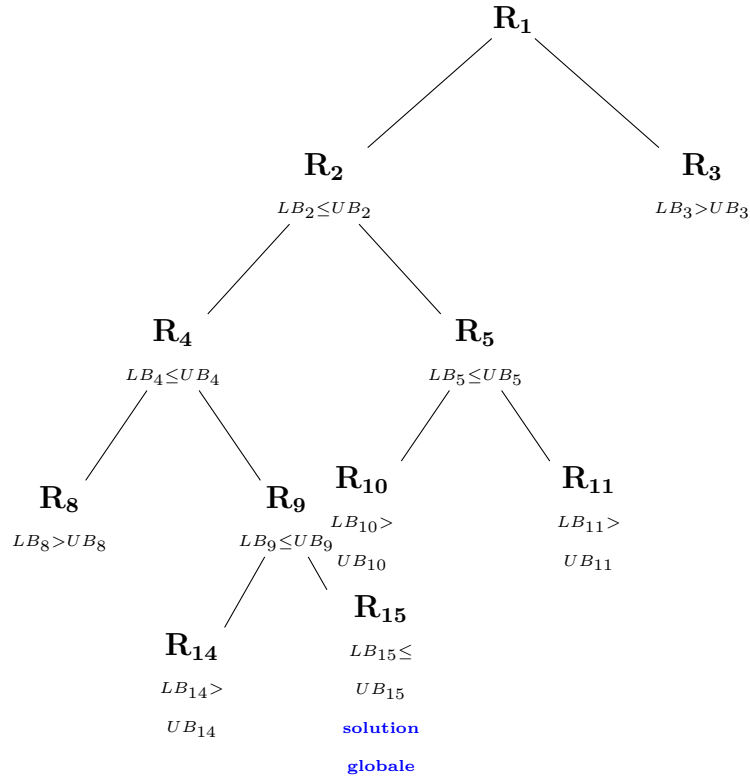


FIGURE 2.2 – Exemple d’une arborescence de la subdivision des sous-ensembles.

► **La convergence de la méthode branch and bound :**

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 :  $R_{k+1} \subset R_k$ ,  $k \in \mathbb{N}$ .

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

$$\lim_{k \rightarrow \infty} (UB_k - LB_k) = \lim_{k \rightarrow \infty} (UB_k - LB(R_k)) = 0, \quad (2.1)$$

alors

$$UB = \lim UB_k = \lim f(X_k) = \lim LB_k = LB. \quad (2.2)$$

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

## Preuve :

À l'itération  $k$  le sous-ensemble  $R_k$  sera choisi à partir de la règle suivante :

$$LB_{k+1} = \min\{LB_k, LB_{R_k}\}$$

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'accumulation.

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 :

$$\lim_{k \rightarrow \infty} f(X_k) = f(X^*).$$

Posons :  $f^* = \min\{f(X), X \in H\}$ , la suite  $\{LB_k\}$  des bornes inférieures est croissante monotone, majorée par  $f^*$ , donc la suite  $LB = \lim LB_k$  existe.

D'autre part, la suite  $\{UB_k\}$  des bornes supérieures est décroissante, minorée par  $f^*$ , donc sa limite :  $UB = \lim UB_k$  existe, et on a :  $UB = f(X_k) \geq f^*$ , ce qui implique que :

$$LB \leq f^* \leq \lim f(X_k) = f(X^*) = UB.$$

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

- i. L'ensemble  $R$  tel que :  $H \subset R$ .
- ii. Les procédures qui donneront les bornes inférieure et supérieure sur les sous-ensembles engendrés par l'algorithme.
- iii. 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 à part, en se basant sur les trois points précédents, et en montrant la convergence de l'algorithme à chaque fois.

### 2.4.2 Méthode d'approximation extérieure

L'idée de cette méthode, introduite dans la programmation convexe à la fin des années cinquante par Cheney et Goldstein [9], ensuite par Hoffman [19] et Tuy [33], a été développée sous le nom d'*approximation extérieure*, pour la minimisation d'une fonction concave sous des contraintes convexes, puis plus généralement pour les problèmes d'optimisation non convexe par Mayane et Polak [23] et Tuy [34]. Le grand principe de la méthode d'approximation extérieure est de faire des coupes linéaires sur un ensemble qui contient l'ensemble réalisable du problème original, de façon qu'on ne coupe jamais l'ensemble réalisable tout en se rapprochant de plus en plus de la solution optimale du problème original.

► **Principe de la méthode d'approximation extérieure :**

On considère le problème d'optimisation non convexe suivant :

$$(P) \begin{cases} \min & f(x) \\ & h_i(x) \leq 0 \quad i = 1, \dots, m. \\ & g(x) \geq 0. \end{cases}$$

où  $f$ ,  $h_i$  et  $g$  sont des fonction convexes de  $\mathbb{R}^n \mapsto \mathbb{R}$ . Posons

$$h(x) = \max_{i=1, \dots, m} h_i(x),$$

$$E = \{x \in \mathbb{R}^n : h_i(x) \leq 0\}, \quad G = \{x \in \mathbb{R}^n : g(x) \geq 0\}.$$

Le principe de cette méthode est de relaxer et remplacer le problème original ( $P$ ) par une suite de problèmes ( $Q_k$ ) qui sont facile à résoudre et dont les solutions obtenues convergent vers la solution optimale du problème original ( $P$ ). Pour la réalisation d'un algorithme d'approximation extérieure, on a les deux phases suivantes :

1. Construire des sous problèmes ( $Q_k$ ) qui sont faciles à résoudre.
2. Construire  $l_k(x)$  pour que la suite des solutions  $\bar{x}_k$  converge.

► **Algorithme de la méthode d'approximation extérieure :**

Avant de citer l'algorithme de cette méthode, on note que :

$w$  est la solution optimale du problème d'optimisation non convexe ( $P$ ), sans la contrainte anti-convexe.

$\partial G$  représente la frontière de l'ensemble  $G$ .

**Algorithme d'approximation extérieure**

1. **Initialisation** : Soit  $\bar{x}_1 \in E \cap \partial G$ , poser  $y_1 = f(\bar{x}_1)$ , former un polytope  $P_1$  où le compact convexe  $\{x \in E : f(x) \leq y_1\}$  est contenu dans  $P_1$ , fixer  $k = 1$ .

2. **Faire**

- a) Résoudre le sous-problème ( $Q_k$ ) tel que :

$$(Q_k) : \max(g(x), x \in P_k).$$

La solution est notée  $z_k$ .

- b) **Si**  $g(z_k) = 0$ , **alors Stop** (Solution optimale).
- c) **Sinon** chercher  $x_k$  dans le segment  $[w, z_k]$  tel que  $\max\{f(x_k) - y_k, g(x_k)\} = 0$ .
- d) **Si**  $x_k \in E$  alors  $v_k \in \partial f(x_k)$

$$l_k(x) = \langle v_k, (x - x_k) \rangle.$$

- e) **Si**  $x_k \notin E$  alors  $v_k \in \partial h(x_k)$

$$l_k(x) = \langle v_k, (x - x_k) \rangle + h(x_k).$$

f)

$$P_{k+1} = P_k \cap \{x \in \mathbb{R}^n : l_k(x) \leq 0\}$$

$$\bar{x}_{k+1} = \begin{cases} x_k & \text{si } x_k \in E \text{ et } g(x_k) = 0 \\ \bar{x}_k & \text{Sinon} \end{cases}$$

g)  $y_{k+1} = f(\bar{x}_{k+1})$ .

3. **Jusqu'à** ( $g(z_k) = 0$ )

### 2.4.3 Méthode D.C : différence de fonctions convexes

Les fonctions DC possèdent de nombreuses propriétés qui ont été établies à partir des années 50 par Landis [20] et Hartman [17]. La classe des fonctions DC a été introduite en optimisation au milieu des années 80. Distinguons deux grandes approches DC :

1. L'approche combinatoire en optimisation globale continue.
2. L'approche de l'analyse convexe en optimisation non convexe.

Cette méthode a été proposée par Pham Dinh et Le Thi Hoai an [29], et utilise les techniques d'optimisation globale (méthode Branch and Bound, technique de coupe ...) pour la résolution des problèmes d'optimisation globale.

**La programmation D.C :**

**Définition 2.4.3.1.** Une fonction  $f : \mathbb{R}^n \mapsto \mathbb{R}$  est dite DC si elle peut s'écrire comme suit :

$$f(x) = g(x) - h(x),$$

avec  $g$  et  $h$  des fonctions convexes de  $\mathbb{R}^n$  dans  $\mathbb{R}$ .

La programmation DC consiste à résoudre des programmes mathématiques dont la fonction objectif est une fonction DC. Le problème de la programmation DC est donnée comme suit :

$$(P) \begin{cases} \min f(x) \\ x \in \mathbb{R}^n. \end{cases}$$

où  $f : \mathbb{R}^n \mapsto \mathbb{R}$  est une fonction DC.

**Définition 2.4.3.2** Soit une fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . La fonction  $f$  est dite semi continue inférieure au point  $x_0$ , s'il existe un voisinage  $V$  de  $x_0$  tel que :

$$\forall \epsilon > 0, \forall x \in V : f(x) \geq f(x_0) - \epsilon.$$

La fonction  $f$  est dite propre si

$$\forall x_0 \in \mathbb{R}^n : \lim_{x \rightarrow x_0} f(x) \neq -\infty \quad \text{et} \quad \lim_{x \rightarrow x_0} f(x) \neq +\infty.$$

On note par  $\Gamma_0$  l'ensemble des fonctions convexes, semi continues inférieure et propres dans  $\mathbb{R}^n$ .

**Définition 2.4.3.3** Soit une fonction  $f : \mathbb{R}^n \mapsto \mathbb{R}$ , convexe et propre. On dit que  $v_0$  est le *sous-gradient* de la fonction  $f$  au point  $x_0$ , si :

$$\forall x \in \mathbb{R}^n : \langle v_0, x - x_0 \rangle + f(x_0) \leq f(x).$$

On note par  $\partial f(x_0)$ , l'ensemble des sous-gradients de la fonction  $f$  au point  $x_0$ , appelé *sous-différentiel*.

On dit que  $v_0$  est le  $\epsilon$ -*sous-gradient* de la fonction  $f$  au point  $x_0$ , si :

$$\forall x \in \mathbb{R}^n : \langle v_0, x - x_0 \rangle + f(x_0) \leq f(x) + \epsilon.$$

On note par  $\partial_\epsilon f(x_0)$ , l'ensemble des  $\epsilon$ -sous-gradients de la fonction  $f$  au point  $x_0$ , appelé  $\epsilon$ -*sous-différentiel*.

**Définition 2.4.3.4** Soit une fonction  $f : \mathbb{R}^n \mapsto \mathbb{R}$ . On note par  $f^*$  la fonction conjuguée de la fonction  $f$ , et définie comme suit :

$$f^*(y) = \sup\{\langle x, y \rangle - f(x), x \in \mathbb{R}^n\}, \quad y \in \mathbb{R}^n.$$

La fonction conjuguée  $f^*$  est l'enveloppe supérieure des fonctions affines continues  $\langle x, y \rangle - f(x)$  sur  $\mathbb{R}^n$ .

**propriétés 2.4.3.1** Si  $f \in \Gamma_0$ , alors

- $f \in \Gamma_0 \iff f^* \in \Gamma_0$ , et  $f = f^{**}$ .
- $y \in \partial f(x) \iff f(x) + f^*(y) = \langle x, y \rangle$  et  $y \in \partial f(x) \iff x \in \partial f^*(y)$ .
- $\partial f(x)$  est un ensemble convexe fermé.
- Si  $\partial f(x) = \{y\}$  alors  $f$  est différentiable en  $x$  et  $\nabla f(x) = y$ .

**Théorème 2.4.3.1** (Condition nécessaire d'optimalité locale)[24]. Soit  $x^*$  un *minimum local* de la fonction  $f(x) = g(x) - h(x)$ , alors

$$\partial h(x^*) \subset \partial g(x^*).$$

**Théorème 2.4.3.2** (Condition suffisante d'optimalité locale)[30] [31].

Si un point  $x^*$  admet un voisinage  $V$  tel que :

$$\partial g(x^*) \cap \partial h(x^*) \neq \emptyset, \quad \forall x \in V \cap \text{dom}(g),$$

alors  $x^*$  est un minimum local de la fonction  $f = g - h$ .

$$\text{dom}(g) = \{x \in \mathbb{R}^n \mid g(x) < +\infty\}$$

**Théorème 2.4.3.3** (Optimalité globale) [18].

Soit une fonction  $f = g - h$  avec  $g, h \in \Gamma_0$ . Un point  $x^*$  est un minimum global de la fonction  $f(x)$  sur  $\mathbb{R}^n$ , si et seulement si

$$\partial_\epsilon h(x^*) \subset \partial_\epsilon g(x^*), \quad \forall \epsilon > 0.$$

### Dualité en optimisation DC :

En 1978, la dualité DC introduite par Toland [32] considérée comme généralisation des travaux de Pham Dinh [10] en 1975, concernant la maximisation convexe. On va présenter quelques résultats principaux sur la dualité DC. Pour plus de détails voir [32][31].

Soient deux fonctions  $g(x)$  et  $h(x)$  convexes dont  $g, h \in \Gamma_0$ . Le problème primal ( $P$ ) et son dual ( $D$ ) sont donnés comme suit :

$$(P) \begin{cases} \inf f(x) = g(x) - h(x) \\ x \in \mathbb{R}^n. \end{cases}$$

$$(D) \begin{cases} \inf -f^*(y) = h^*(y) - g^*(y) \\ y \in \mathbb{R}^n. \end{cases}$$

où  $f^*$  est la fonction conjuguée de la fonction  $f$ .

**Théorème 2.4.3.4** [32] Soient  $g$  et  $h$  deux fonctions telles que  $g, h \in \Gamma_0$ . Alors

1.

$$\inf_{x \in \mathbb{R}^n} (g(x) - h(x)) = \inf_{y \in \mathbb{R}^n} (h^*(y) - g^*(y)).$$

2. Si  $y_0$  est un minimum de la fonction  $h^* - g^*$  sur  $\mathbb{R}^n$ , alors tout  $x_0 \in \partial g^*(y_0)$  est un minimum de  $g - h$  sur  $\mathbb{R}^n$ .

3. Si  $x_0$  est un minimum de la fonction  $g - h$  sur  $\mathbb{R}^n$ , alors tout  $y_0 \in \partial h(x_0)$  est un minimum de  $h^* - g^*$  sur  $\mathbb{R}^n$ .

Parfois, pour résoudre un problème d'optimisation DC, il est plus facile de résoudre le problème dual ( $D$ ) que le problème primal ( $P$ ). Le théorème précédent montre que la résolution du problème primal ( $P$ ) implique la résolution du problème dual ( $D$ ) et vice-versa.

**Théorème 2.4.3.5 :** [30] [31] Soit  $x_0 \in \text{dom}(\partial h)$  un minimum local de la fonction  $f = g - h$ .

Soient  $y_0 \in \partial h(x_0)$  et  $V$  un voisinage de  $x_0$  tel que :

$$g(x) - h(x) \geq g(x_0) - h(x_0), \quad \forall x \in V \cap \text{dom}(g).$$

Si

$$x_0 \in \text{int}(\text{dom}(g^*)) \text{ et } \partial g^*(y_0) \subset V,$$

alors  $y_0$  est un minimum local de la fonction  $-f^* = h^* - g^*$ .

### Algorithme DCA : [11]

La construction de l'algorithme *DCA*, proposé par Pham Dinh Tao en 1986, repose sur la caractérisation des solutions locales des problèmes :

- Primal ( $P$ )
- Dual ( $D$ )

en optimisation DC.

#### Algorithme DCA :

1. **Initialisation :** Soit  $x_0$  une solution initiale. Poser  $k = 0$  et  $\epsilon > 0$ .
2. Calculer  $y_k \in \partial h(x_k)$ .
3. Calculer  $x_{k+1} \in \partial g^*(y_k)$ .
4. **Critère d'arrêt :** Si  $\|x_{k+1} - x_k\| < \epsilon$ , **STOP**.
5. **Itération :**  $k \leftarrow k + 1$ , retour à l'étape 2.

L'algorithme DCA consiste à construire deux suites  $\{x_k\}$  et  $\{y_k\}$ , où la première est la solution du problème primal ( $P$ ) et la deuxième est la solution du problème dual ( $D$ ). Les deux suites sont liées par dualité et vérifient les propriétés suivantes :

- Les deux suites  $\{g(x_k) - h(x_k)\}$  et  $\{h^*(y_k) - g^*(y_k)\}$  sont décroissantes.
- L'algorithme DCA s'arrête à l'itération  $(k + 1)^{\text{ième}}$ , lorsque  $(g - h)(x_{k+1}) = (g - h)(x_k)$ , et le point  $x_k$  est un point critique de  $g - h$ , de même pour le point  $y_k$  qui est un point critique de  $h^* - g^*$ .

### 2.4.4 Méthode de Piyavskii

#### ► Introduction :

Considérant ( $P$ ) le problème de minimisation globale :

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

Où :  $X$  est un compact de  $\mathbb{R}^n$ , et  $f : X \mapsto \mathbb{R}$  une fonction continue et non convexe. Dans cette section, on considère le problème de minimiser une fonction  $f$  lipschitzienne

sur un compact  $X$ , c-à-d une fonction qui satisfait la condition de Lipschitz. Des papiers et livres proposent plusieurs approches pour la résolution numérique de ce problème et donnent une classification des problèmes et de leurs méthodes de résolution. Dans cette section, on va présenter l'algorithme de Piyavskii qui est un algorithme pour optimiser une fonction d'une seule variable sur un intervalle  $[a,b]$

**Définition 2.4.2.1 :** Une fonction  $f : P \subset \mathbb{R}^n \rightarrow \mathbb{R}$  est dite lipschitzienne sur  $P$ , s'il existe une constante réelle  $L = L(f, P)$  telle que :

$$\forall x, y \in P, \quad |f(x) - f(y)| \leq L \|x - y\| \quad (*) \quad (2.3)$$

où  $L$  est une constante appelée la constante de Lipschitz

**Définition 2.4.2.2 :** On dit qu'une fonction  $g$  est un sous-estimateur de la fonction  $f$  sur  $P$  si :

$$\forall x \in P, \quad g(x) \leq f(x)$$

D'après l'inéquation (2.3) on a :

$$f(y) - L \|x - y\| \leq f(x) \quad (2.4)$$

Fixons  $y \in P$ , il résulte de l'inéquation (2.4) que la fonction concave :  $F(x) = f(y) - L \|x - y\|$  est un sous-estimateur de  $f$  sur  $P$ . Nous allons construire une suite de points  $\{x_k\}$  et une famille d'ensembles  $\{E_k\}$  vérifiant  $P \subset \cup_k E_k$  de la manière suivante :

Considérons la fonction

$$F_1(x) = f(x_1) - L \|x - x_1\| \quad (2.5)$$

où  $x_1$  est un point quelconque de  $P$ . Le point suivant de l'itération est défini par :

$$x_2 = \arg \min_{x \in P} f(x)$$

En remplaçant  $x_1$  par  $x_2$  dans l'inéquation (2.5), on obtient un autre sous-estimateur de  $f(x)$ , qui est :

$$F_2(x) = f(x_2) - L \|x - x_2\| \quad (2.6)$$

mais encore un nouveau sous-estimateur qui est donné par :

$$\max_{1 \leq i \leq 2} (f(x_i) - L \|x - x_i\|)$$

et il est plus proche de  $f(x)$  voir la Figure 2.1

On prend donc

$$F_2(x) = \max_{1 \leq i \leq 2} (f(x_i) - L \|x - x_i\|) \quad \text{et} \quad x_3 = \arg \min_{x \in P} F_2(x)$$

À l'étape  $k$ , on prend

$$F_k(x) = \max_{1 \leq i \leq k} (f(x_i) - L \|x - x_i\|) \quad (2.7)$$

et

$$x_{k+1} = \arg \min_{x \in P} F_k(x) \quad (2.8)$$

Dans la méthode de Piyavskii, on prend comme recouvrement de  $P$  la famille d'ensemble suivant :

$$E_k = \left\{ x \in \mathbb{R}^n : \max_{1 \leq i \leq k} (f(x_i) - L \|x - x_i\|) \geq f_k^* - \varepsilon \right\} \quad (2.9)$$

$$= \{ x \in \mathbb{R}^n : F_k(x) \geq f_k^* - \varepsilon \}$$

où  $f_k^*$  est une approximation du minimum global de  $f$ .

Dans le cas unidimensionnel, la méthode de Piyavskii-Shubert consiste à construire des recouvrements de  $P = [a, b]$  à partir des fonctions  $F_k(x)$ . Elle utilise la suite de points  $\{x_k\}$  définie par le terme général :

$$x_{k+1} = \arg \min_{x \in [a, b]} F_k(x)$$

avec

$$x_1 = \frac{a + b}{2}$$

et

$$F_k(x) = \max_{1 \leq i \leq k} (f(x_i) - L \|x - x_i\|).$$

Dans ce cas,  $F_k(x)$  est un sous-estimateur linéaire par morceaux.

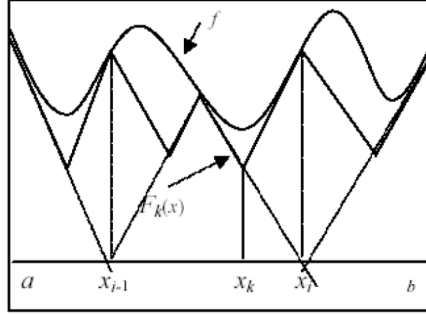


FIGURE 2.3 – Illustration de la méthode de Piyavskii–Shubert pour l’optimisation globale de fonctions lipschitziennes.

► **La méthode de Piyavskii-Shubert :**

Le premier algorithme proposé pour résoudre les problèmes d’optimisation des fonctions lipschitziennes à une seule variable ( $n = 1$ ) a été indépendamment donné par Piyavskii et Shubert. La méthode de Piyavskii-Shubert [16;18] consiste à construire des recouvrements de plus en plus raffinés de la fonction objectif sur le domaine d’optimisation  $P \subset \mathbb{R}^n$ . Cette méthode nécessite la connaissance de la constante de Lipschitz  $L$ .

Soit  $f: [a, b] \rightarrow \mathbb{R}$  une fonction lipschitzienne de constante  $L > 0$  sur  $\mathbb{R}$  c’est-à-dire :

$$|f(x) - f(y)| \leq L|x - y|, \quad \forall x, y \in [a, b]. \quad (**) \quad (2.10)$$

La méthode de Piyavskii minimise la fonction lipschitzienne  $f$  définie sur l’intervalle  $[a; b]$ , en construisant une fonction borne inférieure de la fonction objectif sur l’intervalle  $[a; b]$ . L’idée est de poser  $y = a$  et  $y = b$  dans l’inégalité (2.10) pour obtenir les deux inégalités suivantes :

$$\begin{aligned} f(x) &\leq f(a) - L(x - a), \quad \forall x \in [a, b], \\ f(x) &\leq f(b) + L(x - b), \quad \forall x \in [a, b]. \end{aligned}$$

Utilisant ces deux inégalités, on peut construire une fonction borne inférieure linéaire telle que :

$$f(x) \leq \underline{f}(x), \quad \forall x \in [a, b],$$

où  $\underline{f}$  est donnée par :

$$\underline{f}(x) = \begin{cases} f(a) - L(x - a), & \forall x \in [a, x(a, b)], \\ f(b) + L(x - b), & \forall x \in [x(a, b), b], \end{cases}$$

où

$$x(a, b) = \frac{f(a) - f(b)}{2L} + \frac{a + b}{2}. \quad (2.11)$$

Le lemme suivant montre que la fonction  $\underline{f}$  est toujours définie. Pour ce faire, nous devons seulement nous assurer que  $x(a, b) \in [a, b]$ .

► **Algorithme de Piyavskii unidimensionnel (a,b,L,ε) :**

$k = 1$ ,  $sample = 1$ ,  $l_{sample} = a$ ,  $u_{sample} = b$   
 Calculer  $B_1 = B(a, b)$ ,  $x_1 = x(a, b)$   
 Soit  $f_{opt} = \min \{f(a), f(b)\}$ ,  $x_{opt} = \arg \min \{f(a), f(b)\}$ ,  $B_{opt} = B_1$ ,  
 Tant que  $f_{opt} - B_{opt} > \varepsilon$  faire  
 Choisissez un nouvel intervalle d'échantillonnage, cet intervalle a 'sample' pour indice.  
 $l_{k+1} = x_k$ ,  $u_k = u_{sample}$ ,  $u_{sample} = x_k$ ,  $f_k = f(x_k)$   
 Calculer  $B_{k+1} = B(l_{k+1}; u_{k+1})$ ;  $x_{k+1} = x(l_{k+1}; u_{k+1})$ ;  
 $B_{sample} = B(l_{sample}; u_{sample})$ ;  $x_{sample} = x(l_{sample}; u_{sample})$   
 Mise à jour  
 $f_{opt} = \min_{1 \leq i \leq k+1} \{f(a); f(u_i)\}$ ,  
 $x_{opt} = \arg \min_{1 \leq i \leq k+1} \{f(a), f(u_i)\}$   
 $B_{opt} = \min_{1 \leq i \leq k+1} B_i$   
 $k = k + 1$   
 Fin Tant que

► **Algorithme de Piyavskii généralisé**

**Cas sans contraintes :**

Considérons le problème de minimisation globale

$$\min_{x \in P} f(x)$$

que l'on veut résoudre avec une précision exigée  $\varepsilon > 0$ , avec  $f$  une fonction lipschitzienne de constante  $L > 0$ , définie sur un pave  $P = \prod_{i=1}^n [a_i, b_i]$  de  $\mathbb{R}^n$  et a valeurs dans  $\mathbb{R}$ . C'est-à-dire, une fonction telle que :

$$\forall x, y \in P, \quad |f(x) - f(y)| \leq L \|x - y\|$$

ou  $\|\cdot\|$  est la norme euclidienne.

Le principe de cette méthode est de diviser le pave  $P$  en plusieurs sous paves et de construire sur chacun d'eux une fonction borne inférieure constante de la fonction objectif.

Considérons un point  $x_i$  du pave  $P$  dont la fonction objectif  $f$  est évaluée. La meilleure fonction borne inférieure de  $f$  est  $f_i(x) = f(x_i) - L \|x - x_i\|$ , et donc la meilleure fonction borne inférieure constante de  $f$  est le minimum de  $f_i(x)$  sur le pave  $P$ .

Si  $c = \frac{a+b}{2}$  est le centre du pave  $P$ , alors la constante borne inférieure de  $f$  est :

$$F = \min_{x \in P} f_i(x) = f(c) - L \frac{\|b - a\|}{2}.$$

En effet, le minimum de  $f_i(x)$  est évalué en un point  $\bar{x}$ , où la distance entre  $\bar{x}$  et le centre  $c$  est maximale. Comme  $c$  est le centre du pave  $P$ , alors

$$\|\bar{x} - c\| = \frac{\|b - a\|}{2},$$

où  $\|b - a\|$  est la longueur de la diagonale du pavé  $P$ .

L'algorithme divise le pavé  $P$  en  $p$  sous pavés  $(P_j)_{1 \leq j \leq p}$ . Soit  $P_j = \prod_{i=1}^n [a_i^j, b_i^j] = [a^j, b^j]$  un de ces sous pavé et  $c_j$  le centre de  $P_j$ , la valeur de la fonction constante borne inférieure sur  $P_j$  est définie par :

$$F_j = f(c_j) - L \frac{\|b^j - a^j\|}{2}$$

On note par  $l_i^j = b_i^j - a_i^j$ ,  $i = 1, \dots, n$ . Soit  $D = \|b - a\| = \left( \sum_{i=1}^n l_i^2 \right)^{1/2}$  où  $l_i = \|b_i - a_i\|$  la longueur de la diagonale du pavé  $P$ ,  $\mathbf{L}$  la liste des sous problèmes  $P_i$  pour  $i = 1, \dots, n$ .

### Algorithmes

```

 $k \leftarrow 1$ ;
 $x^1 \leftarrow (a_1 - \frac{l_1}{2}, a_2 - \frac{l_2}{2}, \dots, a_n - \frac{l_n}{2})$ ;
 $x_{opt} \leftarrow x^1$ .
 $f_{opt} \leftarrow f(x_{opt})$ ;
 $F_{opt} \leftarrow f_{opt} - \frac{LD}{2}$ .
 $F_1 \leftarrow F_{opt}$ ;
 $P_1 \leftarrow (F_1(a_1, \dots, a_n), (l_1, \dots, l_n))$ ;
 $\mathbf{L} \leftarrow \{P_1\}$ ,

Tant que  $f_{opt} - F_{opt} > \varepsilon$  faire
 $P_i \leftarrow (F_i(a_1^j, \dots, a_n^j), (l_1^j, \dots, l_n^j))$ ;
 $l_{i_0}^j \leftarrow \max l_i^j$ ;

 $D \leftarrow \left( \sum_{i=1, i \neq i_0}^n (l_i^j)^2 + (\frac{l_{i_0}^j}{p})^2 \right)^{1/2}$ 
Supprimer  $P_i$  de  $\mathbf{L}$ ;
Pour  $k : 1$  a  $p$  faire
 $A_k \leftarrow (a_1^j, \dots, a_{i_0-1}^j, a_{i_0}^j + \frac{k-1}{p} l_{i_0}^j, \dots, a_n^j)$ ;

 $L_k \leftarrow (l_1^j, \dots, l_{i_0-1}^j, \frac{l_{i_0}^j}{p}, \dots, l_n^j)$ ;
 $F_k \leftarrow f(S_k - \frac{L_k}{2}) - \frac{LD}{2}$ ;
Si  $f(S_k - \frac{L_k}{2}) < f_{opt}$ ;
 $f_{opt} \leftarrow f(S_k - \frac{L_k}{2})$ ;
 $x_{opt} \leftarrow S_k - \frac{L_k}{2}$ ;
Fin Si;
Ajouter le sous probleme a  $\mathbf{L}$ 
Fin Pour;
Éliminer de  $\mathbf{L}$  tous les sous problèmes ayant une borne inférieure supérieure à  $f_{opt}$ 
Fin Tant que

```

**Remarque 2.4.2.1** Le test d'élimination n'est pas nécessaire pour la convergence de l'algorithme.

### Cas avec contraintes

Soit le problème suivant :

$$\begin{cases} \min_{x \in P} f(x) \\ g_i(x) \leq 0 \quad i = 1, \dots, k \\ P = \prod_{i=1}^n [a_i, b_i] \end{cases}$$

Supposons que les fonctions contraintes sont lipschitziennes :

$$\forall x, y \in P \times P \quad |g_i(x) - g_i(y)| \leq L_i \|x - y\| \quad i = 1, \dots, k$$

Pour les problèmes avec contraintes on ajoute deux tests :

#### Test d'élimination

L'algorithme calcule une borne inférieure pour chaque contrainte dans le n-rectangle  $P_j$  :

$$\forall x \in P_i \quad g_i(x) \geq g_i(c_j) - \frac{L_i D}{2} = LB_{ij}$$

où : -  $c_j$  est le point milieu de  $P_j$  -  $D$  est la diagonale  $P_j$

Et l'algorithme élimine les sous-problèmes qui correspondent à une surface non admissible :

Si  $LB_{ij} > 0 \forall i$  alors éliminer le sous-problèmes correspondant.

#### Test de mise à jour

L'algorithme vérifie la nouvelle valeur de  $x_{opt}$  et la met à jour si elle satisfait les contraintes.

### ► Convergence de l'algorithme :

On cite ici un théorème de convergence établi par [27]

**Théorème 2.4.2.1** : L'algorithme de Piyavskii généralisé est convergent, c'est-à-dire se termine après un nombre fini d'itérations, ou bien on a

$$\lim_{k \rightarrow \infty} F_{opt}^k = \lim_{k \rightarrow \infty} f_{opt}^k = f^* = \min_{x \in P} f(x).$$

**Démonstration** Soit  $\varepsilon > 0$ , pour montrer la convergence de l'algorithme, on doit trouver une suite  $(r_1, \dots, r_n) \in \mathbb{N}$  telle que :

$$D = \frac{L}{2} \left( \sum_{i=1}^n \left( \frac{l_i}{p_i^{r_i}} \right)^2 \right)^{1/2} \leq \varepsilon.$$
$$\frac{LD}{2} \leq \varepsilon \Leftrightarrow D \leq \frac{2\varepsilon}{L} \Leftrightarrow \sum_{i=1}^n \left( \frac{l_i}{p_i^{r_i}} \right)^2 \leq \frac{4\varepsilon^2}{L^2}.$$

Il suffit de choisir :

$$\forall i \quad r_i \geq \log \left( \frac{L l_i \sqrt{n}}{2\varepsilon} \right).$$

Donc  $\exists m \in P, \exists k, \exists (r_1, \dots, r_n)$  tels que

$$|f^k(m) - F_{\text{opt}}^k| \leq \varepsilon$$

où  $f^k(m) = F_{\text{opt}}^k + \frac{LD}{2}$ .

Par définition on a  $F_{\text{opt}}^k \leq f^* \leq f^k(m)$  d'où  $\exists k \in \mathbb{N}$  tel que  $|f^* - F_{\text{opt}}^k| \leq \varepsilon$ .

Puisque  $\varepsilon$  est arbitraire, on a :

$$\lim_{k \rightarrow \infty} F_{\text{opt}}^k = f^*$$

De plus  $f^* \leq f_{\text{opt}}^k \leq f^k(m)$ , ce qui implique que

$$|f^* - f_{\text{opt}}^k| \leq \varepsilon.$$

## 2.5 Méthodes basées sur l'algorithme Branch and Bound

### 2.5.1 Méthode $\alpha$ BB [6], [3], [2]

La méthode  $\alpha$ BB est développée par Androulakis, Maranas, Floudas et d'autres [6],[3], [2],[21] Cette méthode est basée sur l'algorithme de *branch and bound* pour la résolution des problèmes d'optimisation globale avec contraintes.

L'importance centrale de cette méthode est le calcul du paramètre  $\alpha$  qui garantit la convexité de la fonction borne inférieure utilisée par l'algorithme  $\alpha$ BB durant la résolution du problème d'optimisation globale. Le problème d'optimisation globale considéré est le suivant :

$$(P) \begin{cases} \min & f(x) \\ \text{s.c.} & h_j(x) = 0, j = 1, \dots, M \\ & g_k(x) \leq 0, k = 1, \dots, K \\ & Ax \leq b \\ & \underline{x} \leq x \leq \bar{x} \end{cases}$$

où : Les fonctions  $f$ ,  $h$  et  $g$  sont des fonctions non linéaires, et au moins une de ces fonctions est non convexe.

$$A \in \mathbb{R}^{n \times n} \text{ est une matrice carrée, } b \in \mathbb{R}^n, \text{ et } \underline{x}, \bar{x} \in \mathbb{R}^n \text{ avec } \underline{x} < \bar{x}.$$

Avec :

- $x \in \mathbb{R}^n$  est le vecteur des variables de décision
- $f(x)$  est la fonction objectif à minimiser
- $h_j(x)$  représentent  $M$  contraintes d'égalité
- $g_k(x)$  représentent  $K$  contraintes d'inégalité
- $Ax \leq b$  sont des contraintes linéaires
- $\underline{x}$  et  $\bar{x}$  sont les bornes inférieures et supérieures des variables

### La fonction borne inférieure :

La fonction borne inférieure (sous-estimateur) d'une fonction non convexe  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est donnée comme suit :

$$LB_\alpha(x) = f(x) - \frac{1}{2} \sum_{i=1}^n \alpha_i (x_i - \underline{x}_i)(\bar{x}_i - x_i), \quad (2.12)$$

avec les paramètres  $\alpha_i$  définis par :

$$\alpha_i \geq \max \left\{ 0, - \min_{\underline{x} \leq x \leq \bar{x}} \lambda_i(x) \right\}, \quad (2.13)$$

où  $\lambda_i(x)$  désigne les valeurs propres de la matrice hessienne de  $f$ .

**Théorème 2.5.1.1** ([6]). La fonction  $LB_\alpha(x)$  coïncide avec la fonction  $f(x)$  aux extrémités du domaine  $[\underline{x}, \bar{x}]$ .

**Démonstration.** Soit  $x^0$  un sommet du domaine  $[\underline{x}, \bar{x}]$ , alors pour chaque  $i = 1, \dots, n$  on a

$$(x_i^0 - \underline{x}_i)(\bar{x}_i - x_i^0) = 0,$$

ce qui implique que

$$LB_\alpha(x^0) = f(x^0).$$

**Théorème 2.5.1.2** ([6]).  $LB_\alpha(x) \leq f(x)$  pour tout point  $x$  sur le domaine  $[\underline{x}, \bar{x}]$ .

**Démonstration.** Par construction de la fonction  $LB_\alpha(x)$ , on a :

$$\frac{1}{2} \sum_{i=1}^n \alpha_i (x_i - \underline{x}_i)(\bar{x}_i - x_i) \geq 0,$$

alors,  $LB_\alpha(x) \leq f(x)$ ,  $\forall x \in [\underline{x}, \bar{x}]$ .

**Théorème 2.5.1.3** ([6]). La fonction  $LB_\alpha(x)$  est convexe sur le domaine  $[\underline{x}, \bar{x}]$ .

**Démonstration.** C'est une conséquence directe de la définition des paramètres  $\alpha_i$ .

**Théorème 2.5.1.4** ([6]). La distance maximale de séparation entre la fonction  $f(x)$  et la fonction borne inférieure  $LB_\alpha(x)$  sur le domaine  $[\underline{x}, \bar{x}]$ , est donnée comme suit :

$$\max_{\underline{x} \leq x \leq \bar{x}} f(x) - LB_\alpha(x) = \frac{1}{4} \sum_{i=1}^n \alpha_i (\bar{x}_i - \underline{x}_i)^2.$$

**Démonstration.**

$$\begin{aligned} \max_{\underline{x} \leq x \leq \bar{x}} f(x) - LB_\alpha(x) &= \max_{\underline{x} \leq x \leq \bar{x}} f(x) - f(x) + \frac{1}{2} \sum_{i=1}^n \alpha_i (x_i - \underline{x}_i)(\bar{x}_i - x_i) \\ &= \max_{\underline{x} \leq x \leq \bar{x}} \frac{1}{2} \sum_{i=1}^n \alpha_i (x_i - \underline{x}_i)(\bar{x}_i - x_i) = \frac{1}{4} \sum_{i=1}^n \alpha_i (\bar{x}_i - \underline{x}_i)^2. \end{aligned}$$

## La fonction borne inférieure améliorée :

Une fonction borne inférieure améliorée [4] [5] utilisée dans la méthode  $\alpha$ BB, est donnée comme suit :

$$LB_\gamma(x) = f(x) - \sum_{i=1}^n \left(1 - e^{\gamma_i(x_i - \underline{x}_i)}\right) \left(1 - e^{\gamma_i(\bar{x}_i - x_i)}\right),$$

où  $\gamma = (\gamma_1, \dots, \gamma_n)$  est un vecteur de valeurs non négatives, et  $\underline{x}$  et  $\bar{x}$  représentent respectivement les bornes inférieure et supérieure du domaine.

### Théorème 2.5.1.5 ([5]).

1. La fonction  $LB_\gamma(x)$  coïncide avec la fonction  $f(x)$  aux extrémités du domaine  $[\underline{x}, \bar{x}]$ .
2.  $LB_\gamma(x) \leq f(x)$  pour tout  $x$  sur le domaine  $[\underline{x}, \bar{x}]$ .

**Théorème 2.5.1.6 ([5]).** Il existe une valeur  $\gamma \in [\underline{\gamma}, \bar{\gamma}]$  telle que  $LB_\gamma(x)$  est convexe et  $LB_\gamma(x) \geq LB_\alpha(x)$ , sur le domaine  $[\underline{x}, \bar{x}]$ .

## Principe général de la méthode $\alpha$ BB :

La méthode  $\alpha$ BB a été proposée pour localiser la solution optimale du problème d'optimisation globale ( $P$ ). Cette méthode génère une séquence de bornes inférieures et supérieures qui convergent vers la valeur optimale.

Les bornes inférieures sont obtenues par la résolution de problèmes convexes, en remplaçant les fonctions non convexes du problème ( $P$ ) par des fonctions borne inférieure  $LB_\alpha(x)$ . Les bornes supérieures correspondent aux solutions du problème ( $P$ ), obtenues par des méthodes de recherche locale.

Par la suite, l'algorithme  $\alpha$ BB est présenté pour détailler le principe de la méthode  $\alpha$ BB.

## Algorithme de la méthode $\alpha$ BB :

1. **Initialisation :** Fixer  $k = 1$ ,  $\epsilon_c > 0$  et  $\epsilon_f > 0$ . Initialiser  $x^{c,k}$ ,  $UB = +\infty$  et  $LB = -\infty$ . Poser  $[\underline{x}^k, \bar{x}^k] = [\underline{x}, \bar{x}]$ .

$$\Lambda = \{[\underline{x}^k, \bar{x}^k]\}.$$

2. Extraire le premier élément de la liste  $\Lambda$ , noté  $[\underline{x}^k, \bar{x}^k]$ . Résoudre localement le problème ( $P$ ) sur  $[\underline{x}^k, \bar{x}^k]$ ,  $x_{\text{up}}^k$  est la solution locale. Si  $x_{\text{up}}^k$  est  $\epsilon_f$ -réalisable et  $f(x_{\text{up}}^k) < UB$ , alors mettre à jour la borne supérieure  $UB := f(x_{\text{up}}^k)$  et  $x^* = x_{\text{up}}^k$ .
3. Construire le problème borne inférieure ( $P_L$ ) en remplaçant les fonctions non convexes par les fonctions  $LB_\alpha(x)$ . Résoudre ( $P_L$ ) sur  $[\underline{x}^k, \bar{x}^k]$ . Soit  $x_{\text{low}}^k$  la solution optimale de ( $P_L$ ). Si  $LB_\alpha(x_{\text{low}}^k) > LB$ , alors mettre à jour  $LB := LB_\alpha(x_{\text{low}}^k)$ , et  $x^{c,k+1} = x_{\text{low}}^k$ .

4. Si  $(UB - LB_\alpha(x_{\text{low}}^k) \leq \epsilon_c)$ , alors  $[\underline{x}^k, \bar{x}^k]$  peut être supprimé. **Sinon**, partitionner le sous-domaine courant  $[\underline{x}^k, \bar{x}^k]$  en deux sous-domaines selon la plus grande dimension :

$$\left( [\underline{x}_1^k, \bar{x}_1^k], \dots, [\underline{x}_j^k, \frac{\bar{x}_j^k + \underline{x}_j^k}{2}], \dots, [\underline{x}_n^k, \bar{x}_n^k] \right),$$

et

$$\left( [\underline{x}_1^k, \bar{x}_1^k], \dots, [\frac{\bar{x}_j^k + \underline{x}_j^k}{2}, \bar{x}_j^k], \dots, [\underline{x}_n^k, \bar{x}_n^k] \right),$$

où  $[\underline{x}_j^k, \bar{x}_j^k]$  est l'intervalle le plus large. Ajouter ces sous-domaines à  $\Lambda$ , en maintenant un ordre décroissant des bornes inférieures.

5. Si  $\Lambda = \emptyset$ , arrêter. La solution optimale est  $x^*$  avec valeur  $f(x^*)$ . **Sinon**, poser  $k = k + 1$  et retourner à l'étape 2.

## 2.5.2 Méthode de borne inférieure quadratique [26] :

Cette méthode est développée en 2014 par M. Ouanes, H.A. Le Thi, T.P. Nguyen et A. Zidna [26], pour les fonctions multivariées générales en optimisation globale. Le principe de cette méthode est le même que la méthode  $\alpha$ BB. Cependant, cette méthode diffère dans la fonction borne inférieure et l'algorithme de branch and bound pour la résolution des problèmes d'optimisation globale. Le problème considéré ici est le suivant :

Le problème d'optimisation globale  $(P)$  s'écrit :

$$(P) \begin{cases} \min_x & f(x) \\ \text{s.c.} & x \in B \end{cases}$$

où

$$B = \prod_{i=1}^n [x_i^0, x_i^1]$$

est un hyperrectangle dans  $\mathbb{R}^n$  et  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  est une fonction de classe  $\mathcal{C}^2$  (deux fois continûment différentiable).

### La fonction borne inférieure :

La fonction borne inférieure (sous-estimateur) quadratique de la fonction  $f$  est donnée comme suit :

$$LB_q(x) = L_h f(x) - \frac{1}{2} K \sum_{i=1}^n (x_i^1 - x_i)(x_i - x_i^0).$$

où  $L_h f(x)$  est un interpolant multi-linéaire de la fonction  $f(x)$  dans  $B$ , et  $K$  est un nombre réel positif qui satisfait l'inégalité  $K \geq \|H_f(x)\|$  pour tout  $x$  dans  $B$ , avec  $H_f(x)$  la matrice hessienne de  $f$ .

Soient  $w_0(x_i)$  et  $w_1(x_i)$  deux réelles valeurs définies par

$$w_0(x_i) = \frac{x_i^1 - x_i}{x_i^1 - x_i^0}, \quad w_1(x_i) = \frac{x_i - x_i^0}{x_i^1 - x_i^0}.$$

La construction de la fonction  $L_h f(x)$  est donnée par :

$$L_h f(x) = \sum_{i_n=0}^1 \left( \cdots \left( \sum_{i_1=0}^1 f(x_1^{i_1}, \dots, x_n^{i_n}) w_{i_1}(x_1) \right) \cdots \right) w_{i_n}(x_n).$$

Dans le cas uni-dimensionnel ( $n = 1$ ), la fonction borne inférieure  $LB_q(x)$  devient :

$$LB_q(x) = f(x^0)w_0(x) + f(x^1)w_1(x) - \frac{1}{2}K(x^1 - x)(x - x^0),$$

avec  $k \geq |f''(x)|$  pour tout  $x$  dans  $B = [x^0, x^1]$ .

**Théorème 2.5.1.7** [26].

- (i) Pour tout sommet  $x$  dans  $V(B)$  on a  $LB_q(x) = f(x)$ .
- (ii) Pour tout  $x$  dans  $B$ , si  $\max_{x \in B} |f''_{x_i x_i}(x)| \leq K$ , alors  $LB_q(x) \leq f(x)$ .

**Théorème 2.5.1.8** [26]. La fonction borne inférieure

$$LB_q(x) = L_h f(x) - \frac{1}{2}K \sum_{i=1}^n (x_i^1 - x_i)(x_i - x_i^0).$$

est convexe si

$$K \geq \max_{x \in B} \max_{i=1, \dots, n} \sum_{\substack{j=1 \\ j \neq i}}^n |f''_{x_i x_j}(x)|.$$

Pour plus de détails sur la preuve des deux théorèmes précédents, voir [26].

**Algorithme de borne inférieure quadratique :**

1. **Initialisation :** Fixer  $\epsilon > 0$ .
2. Calculer  $K$  une borne supérieure de  $\|H_f\|$  sur  $B$ .
3. Mettre  $T^0 = B$ . Pour  $i = 1, \dots, n$ , faire  $h_i = x_i^1 - x_i^0$ .
4. Résoudre le problème convexe suivant pour obtenir une solution optimale  $s_0$  :

$$\min \{LB_q(x) : x \in T^k\} \tag{2.14}$$

5. Mettre  $LB = LB(T^0) := LB_q(s_0)$ . Mettre  $UB_0 = \min \{ \min_{v \in V(B)} f(v), f(s_0) \}$ .
6. **Si**  $UB_0 - LB_0 \leq \epsilon$  **alors**  $s_0$  est une solution  $\epsilon$ -optimale. **STOP**.  
**Sinon** Mettre  $M \leftarrow \{T^0\}$ ,  $k \leftarrow 1$ .

7. **Étape de sélection**

- (a) Soit  $T^k = \prod_{i=1}^n [a_i, b_i] \in M$  un hyper-rectangle tel que  $LB_k = LB(T^k)$ , et  $s_k$  est la solution du problème (2.14)
- (b) Calculer  $K_k$  tel que  $\|H_f(x)\| \leq K_k, \forall x \in T^k$ .

8. Couper  $T^k$  en deux sous-rectangles  $T_1^k$  et  $T_2^k$  par la  $w$ -subdivision par  $s_k$ .
9. **Pour**  $i = 1, 2$  **faire** Résoudre le problème convexe (2.14) avec  $k$  remplacé par  $k_i$  pour obtenir une solution optimale  $s_{ki}$ .
10. **Étape de mise à jour**
  - (a) Mettre à jour la borne supérieure  $UB_k = \min\{UB_{k-1}, f(s_{k1}), f(s_{k2})\}$ .
  - (b) Soit  $s_k$  la meilleure solution actuelle, i.e.  $f(s_k) = UB_k$ .
  - (c) Mettre  $M \leftarrow \{T_i^k : LB(T_i^k) < UB_k - \epsilon, i = 1, 2\} \cup \{T\}$ .
  - (d) Mettre à jour la borne inférieure  $LB_k = \min\{LB(T) : T \in M\}$ .
11. **Si**  $M = \emptyset$  **alors**  $s_k$  est la solution optimale et quitte l'algorithme.  
**Sinon** Mettre  $k \leftarrow k + 1$  et retourner à l'étape 7.

## 2.6 conclusion :

Dans ce chapitre, nous avons présenté quelques méthodes déterministes de résolution en optimisation globale. Nous avons vu des méthodes basées sur **les sous-estimateurs** et l'algorithme de **branch and bound** pour résoudre les problèmes d'optimisation globale.

# Chapitre 3

## Exemples numériques

### 3.1 Exemples d'application sur la méthode de branch and bound [12]

Étant donné le problème de minimisation non convexe suivant :

$$(P) \begin{cases} \min & f(x) \\ \text{s.c.} & x \in S \end{cases}$$

La méthode Branch and Bound consiste à remplacer la fonction objectif par une fonction borne inférieure dans les exemples donnés.

Cette méthode utilise une fonction permettant d'établir une borne sur certaines solutions pour :

- soit les exclure,
- soit les maintenir comme solutions potentielles.

L'idée a déjà été proposée pour des fonctions objectif satisfaisant la propriété de Lipschitz.

En effet, la propriété de Lipschitz fournit un cadre pour construire la fonction borne inférieure.

#### Exemple 1

Soit à résoudre le problème de minimisation suivant :

$$(P1) \begin{cases} \min \left\{ \frac{3}{4} \sin(x) + \frac{1}{4} \cos(x) \right\} \\ \text{avec } 2.7 \leq x \leq 7.5 \end{cases}$$

Avec  $\varepsilon = 2 \times 10^{-3}$  et  $L = 12.5$ .

**Itération 0 :** ( $j = 0$ )

1. Soit  $x \in [a_0, b_0] = [2.7, 7.5]$

2. Calcul :

$$\begin{cases} f_a^0(x) = -0.4352 \\ f_b^0(x) = -3.4794 \end{cases} \Rightarrow \begin{cases} x_0^* = 5.1507 \\ UB_0 = -4.5869 \\ LB_0 = -38.5389 \end{cases}$$

3.  $UB_0 - LB_0 = -4.5869 - (-38.5389) = 33.952 > \varepsilon$

On passe à la 1ère itération : subdivision de  $I_0 = [2.7, 7.5]$  en :

$$I_{11} = [2.7, 5.1507], \quad I_{12} = [5.1507, 7.5]$$

**Itération 1 :** ( $j = 1$ )

**Premier cas :**  $x \in I_{11} = [2.7, 5.1507]$

$$\begin{cases} f_a^{11}(x) = -0.4352 \\ f_b^{11}(x) = -4.5869 \end{cases} \Rightarrow \begin{cases} x_{11}^* = 4.0609 \\ UB_{11} = -4.5869 \\ LB_{11} = -12.0101 \end{cases}$$

**Deuxième cas :**  $x \in I_{12} = [5.1507, 7.5]$

$$\begin{cases} f_a^{12}(x) = -4.5869 \\ f_b^{12}(x) = -3.4794 \end{cases} \Rightarrow \begin{cases} x_{12}^* = 6.3254 \\ UB_{12} = -4.5869 \\ LB_{12} = -13.2107 \end{cases}$$

Calcul des bornes :

$$\begin{cases} UB_1 = \min\{UB_{11}, UB_{12}, UB_0\} = -4.5869 \\ LB_1 = \min\{LB_{11}, LB_{12}\} = -13.2107 \end{cases}$$

$UB_1 - LB_1 = -4.5869 - (-13.2107) = 8.6238 > \varepsilon$

On subdivise  $I_{11} = [2.7, 5.1507]$  en  $I_{31} = [5.1507, 6.3254]$  et  $I_{32} = [6.3254, 7.5]$

**Itération 2 :** ( $j = 2$ )

**Premier cas :**  $x \in I_{21} = [5.1507, 6.3254]$

$$\begin{cases} f_a^{21}(x) = -4.5869 \\ f_b^{21}(x) = -2.6390 \end{cases} \Rightarrow \begin{cases} x_{21}^* = 5.6054 \\ UB_{21} = 5.6054 \\ LB_{21} = -5.8789 \end{cases}$$

**Deuxième cas :**  $x \in I_{22} = [6.3254, 7.5]$

$$\begin{cases} f_a^{22}(x) = -2.6390 \\ f_b^{22}(x) = -3.4794 \end{cases} \Rightarrow \begin{cases} x_{22}^* = 7.0453 \\ UB_{22} = -4.2722 \\ LB_{22} = -5.8789 \end{cases}$$

$$\begin{cases} UB_2 = \min\{UB_{21}, UB_{22}, UB_1\} = -4.5869 \\ LB_2 = \min\{LB_{21}, LB_{22}, LB_{12}\} = -13.2107 \end{cases}$$

$UB_2 - LB_2 = -4.5869 - (-13.2107) = 8.6238 > \varepsilon$

**Remarque :** On a écrit les deux premières itérations pour expliquer la procédure de l'algorithme *Branch and Bound* pour un problème d'une fonction à une seule variable.

**Conclusion :** Après la 7ème itération, on obtient :

$$x^* = 5.199778 \quad \text{et} \quad f(x^*) = -4.601308$$

## Exemple 2

Soit à résoudre le problème de minimisation suivant :

$$(P2) \quad \begin{cases} \min \left\{ \sin(x) + \sin\left(\frac{10x}{3}\right) + \ln(x) - 0.84x \right\} \\ \text{avec } 0 \leq x \leq 1 \end{cases}$$

Avec  $\varepsilon = 2 \times 10^{-3}$  et  $L = 1$ .

**Itération 0 :** ( $j = 0$ )

1. Soit  $x \in [0, 1]$
2. Calcul :

$$\begin{cases} f_a^0(x) = 0.25 \\ f_b^0(x) = 0.7662 \\ x_0^* = -0.0162 \end{cases}$$

Comme  $x_0^* \notin [0, 1]$ , on prend :

$$UB_0 = LB_0 = \min\{f(0), f(1)\} = f(0) = 0.25$$

Donc la solution optimale exacte du problème (P2) est :

$$x^* = 0 \quad \text{et} \quad f(x^*) = 0.25$$

## 3.2 Exemples d'application sur la Méthode de Piyavskii [12]

### 3.2.1 Application numérique de l'algorithme (Cas sans contraintes)

Définition des fonctions pour l'évaluation

Les problèmes de minimisation suivants sont définis dans  $\mathbb{R}^2$  :

Numéro	Fonction $f(x_1, x_2)$	Intervalle $[a, b]^2$	Constante de Lipschitz $L$
1	$4x_1x_2 \sin(4\pi x_2)$	$[0, 1] \times [0, 1]$	75
2	$\sin(x_1) \sin(x_1x_2)$	$[0, 4] \times [0, 4]$	4.299
3	$\frac{\sin(2x_1 + x_2)}{\sin(x_2) + 2}$	$[-5, 5] \times [-5, 5]$	2.237

Résultats numériques obtenus

Les résultats des problèmes ci-dessus sont résumés dans le tableau suivant :

Numéro	$f_{\text{opt}}$	Précision $\varepsilon$	Vecteur optimal trouvé	Itérations	Temps (s)
1	2.5177	$10^{-5}$	(0.992, 0.636)	12901	16.31
2	0.9990	$10^{-5}$	(1.6049, 0.9629)	44275	16.01
3	0.9999	$10^{-2}$	(4.7142, -1.5729)	19120	16.01

### 3.2.2 Application numérique de l'algorithme (Cas avec contraintes)

#### Problème 1

$$P_1 \begin{cases} J(x_1, x_2, x_3) = (x_1^4 + x_2 + x_3) - (x_1 + x_2^2 - x_3) \\ (x_1 - x_2 - 1.2)^2 + x_2 \leq 4.4 \\ x_1 + x_2 + x_3 \leq 6.5 \\ 1.4 \leq x_1 \leq 2 \\ 1.6 \leq x_2 \leq 2 \\ 1.8 \leq x_3 \leq 2 \end{cases}$$

#### Problème 2

$$p_2 \begin{cases} J(x_1, x_2) = -(x_1 - 4.2)^2 - (x_2 - 1.9)^2 \\ -x_1 + x_2 \leq 3 \\ x_1 + x_2 \leq 11 \\ x_1 - x_2 \leq 16 \\ -x_1 - x_2 \leq -1 \\ 0 \leq x_1 \leq 10 \\ 0 \leq x_2 \leq 5 \end{cases}$$

Résultats numériques pour les problèmes contraints

Numéro	$f_{\text{opt}}$	Vecteur optimal trouvé	Itérations	Temps (s)
1	-4.5955	(1.4012, 1.8098, 1.8037)	17533	38.2343
2	28.6419	(9.4444, 0.8333)	123	27.35

### Interprétation des résultats

On remarque que l'algorithme converge vers l'optimum global. Cependant, en pratique, il devient lent à proximité de l'optimum, ce qui explique le temps de calcul élevé malgré la faible dimension des problèmes.

### 3.3 Exemple d'application sur la méthode D.C :

#### Exemple 01 , D.C et D.C.A :[8]

Soit  $f$  la fonction tel que

$$f(x) = \frac{1}{2}x'Qx + b'x \rightarrow \min$$

$$x \in C = \{x \in \mathbb{R}^n : \|x\|_2 \leq \delta\}.$$

D'après la D.C :

$$f(x) = \frac{1}{2}x'Qx + b'x + \chi_C(x) \text{ avec } x \in \mathbb{R}^n, \chi_C \text{ fonction indicatrice.}$$

Posons  $g(x) = \frac{\rho}{2}\|x\|^2 + b'x + \chi_C(x)$ ;  $\rho$  bien choisi.

$$h(x) = \frac{\rho}{2}\|x\|^2 - \frac{1}{2}x'Qx.$$

## Exemple 02 , D.C et D.C.A : [16]

$$P \begin{cases} \text{Max } Z(x, y) = -x + 6y \\ \text{s.c :} \\ -x + y \leq 3 \\ x + y \leq 7 \\ y \leq 4 \\ x, y \geq 0 \end{cases}$$

Les composantes DC sont :

$$R(x) = \frac{\rho}{2}\|x\|^2 + x - 6y + \chi_C(x) \quad \text{et} \quad T(x) = \frac{\rho}{2}\|x\|^2$$

avec  $C = \{x \in \mathbb{R}^2 : Ax \leq b; x, y \geq 0\}$  où :

$$A = \begin{pmatrix} -1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 7 \\ 4 \end{pmatrix}, \quad c = \begin{pmatrix} -1 \\ 6 \end{pmatrix}$$

On applique DCA avec  $\rho = 1$  :

— Pour  $k = 0$  :

$$x^0 = (0, 0), \varepsilon = 10^{-3}, y^0 = \rho x^0 = (0, 0), z^0 = y^0 + c = (-1, 6) \Rightarrow x^1 = (1, 4)$$

Le test d'arrêt n'est pas vérifié puisque  $\|x^1 - x^0\| > \varepsilon$ , donc on pose  $k := k + 1$  et on répète.

— Pour  $k = 1$  :

$$y^1 = (1, 4), z^1 = (0, 10) \Rightarrow x^2 = (1, 4), \|x^2 - x^1\| \leq \varepsilon$$

Donc  $x^2$  est une solution optimale.

(Voir l'interprétation géométrique dans la figure 3.1)

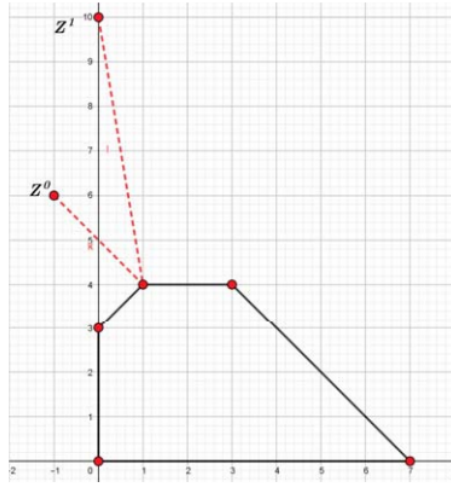


FIGURE 3.1 – Exemple 02 , D.C et D.C.A ,F1

Itérations avec  $\rho = 2$  :

$$\varepsilon = 10^{-3}, \rho = 2$$

$k$	$x^k$	$y^k$	$z^k$	$x^{k+1}$	Test d'arrêt
0	(0, 0)	(0, 0)	(-0.5, 3)	(0, 3)	$\ x^1 - x^0\  > \varepsilon$
1	(0, 3)	(0, 6)	(-0.5, 6)	(1, 4)	$\ x^2 - x^1\  > \varepsilon$
2	(1, 4)	(2, 8)	(0.5, 7)	(1, 4)	$\ x^3 - x^2\  \leq \varepsilon$

D'où  $x^* = (1, 4)$  (voir figure 3.2)

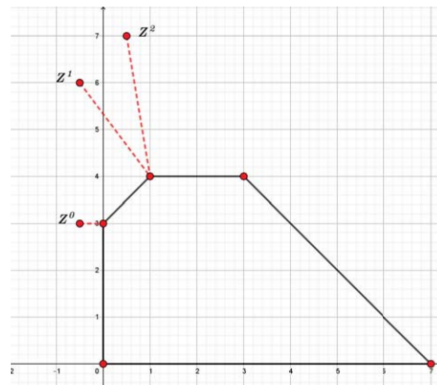


FIGURE 3.2 – Exemple 02 , D.C et D.C.A ,F2

Itérations avec  $\rho = 4$  :

$$\varepsilon = 10^{-3}, \rho = 4$$

$k$	$x^k$	$y^k$	$z^k$	$x^{k+1}$	Test d'arrêt
0	(0, 0)	(0, 0)	(-0.25, 1.5)	(0, 1.5)	$\ x^1 - x^0\  > \varepsilon$
1	(0, 1.5)	(0, 6)	(-0.25, 3)	(0, 3)	$\ x^2 - x^1\  > \varepsilon$
2	(0, 3)	(0, 12)	(-0.25, 4.5)	(0.75, 3.75)	$\ x^3 - x^2\  > \varepsilon$
3	(0.75, 3.75)	(3, 15)	(0.5, 5.25)	(1, 4)	$\ x^4 - x^3\  > \varepsilon$
4	(1, 4)	(4, 16)	(0.75, 8)	(1, 4)	$\ x^5 - x^4\  \leq \varepsilon$

D'où  $x^* = (1, 4)$  (voir figure 3.3)

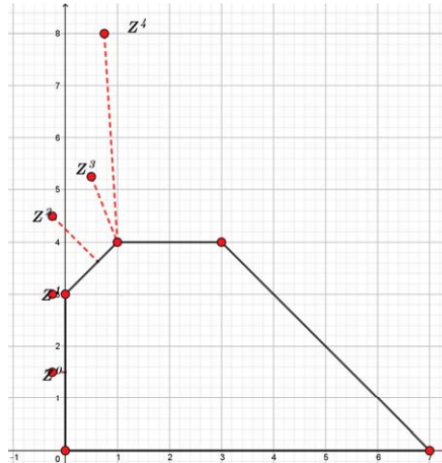


FIGURE 3.3 – Exemple 02 , D.C et D.C.A ,F3

**Remarque 3.2 :** On remarque que le point de départ dans les exemples précédents est un sommet ( $x^0$ ). Dans l'exemple suivant, on choisira un point à l'intérieur de l'ensemble  $C$ .

Point de départ  $x^0 = (1, 1)$  avec  $\rho = 1$  :

$k$	$x^k$	$y^k$	$z^k$	$x^{k+1}$	Test d'arrêt
0	(1, 1)	(1, 1)	(0, 7)	(1, 4)	$\ x^1 - x^0\  > \varepsilon$
1	(1, 4)	(0, 7)	(-1, 13)	(1, 4)	$\ x^2 - x^1\  \leq \varepsilon$

D'où  $x^* = (1, 4)$  (Voir l'interprétation géométrique dans la figure (3.4)).

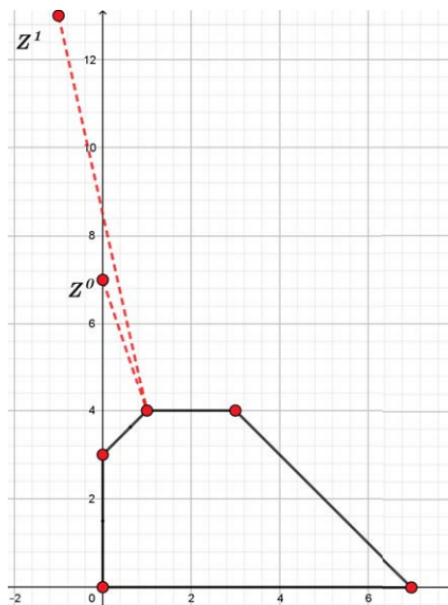


FIGURE 3.4 – Exemple 02 , D.C et D.C.A ,F4

Même point de départ avec  $\rho = 2$ . On aura les résultats suivants :

$k$	$x^k$	$y^k$	$z^k$	$x^{k+1}$	Test d'arrêt
0	(1, 1)	(2, 2)	(0.5, 4)	(0.75, 3.75)	$\ x^1 - x^0\  > \varepsilon$
1	(0.75, 3.75)	(1.5, 7.5)	(0.25, 6.75)	(1, 4)	$\ x^2 - x^1\  > \varepsilon$
2	(1, 4)	(2, 8)	(0.5, 7)	(1, 4)	$\ x^3 - x^2\  \leq \varepsilon$

D'où  $x^* = (1, 4)$  (Voir l'interprétation géométrique dans la figure (3.5)).

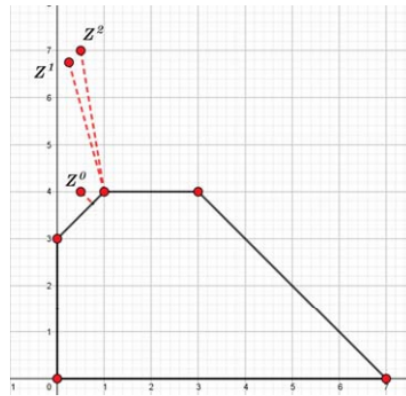


FIGURE 3.5 – Exemple 02 , D.C et D.C.A ,F5

**Commentaire :** On remarque qu'à chaque fois que la valeur de  $\rho$  augmente, le nombre d'itérations augmente.

On essaie d'appliquer l'algorithme DCA avec un point à l'extérieur de l'ensemble C (point n'est pas réalisable), on prend le point  $x^0 = (-1, 0)$ ,  $\rho = 1$  et on aura les résultats suivants :

$k$	$x^k$	$y^k$	$z^k$	$x^{k+1}$	Test d'arrêt
0	(-1, 0)	(-1, 0)	(-2, 6)	(0.5, 3.5)	$\ x^1 - x^0\  > \varepsilon$
1	(0.5, 3.5)	(0.5, 8.5)	(-0.5, 9.5)	(1, 4)	$\ x^2 - x^1\  > \varepsilon$
2	(1, 4)	(1, 4)	(0, 10)	(1, 4)	$\ x^3 - x^2\  \leq \varepsilon$

D'où  $x^* = (1, 4)$  (Voir l'interprétation géométrique dans la figure (3.6)).

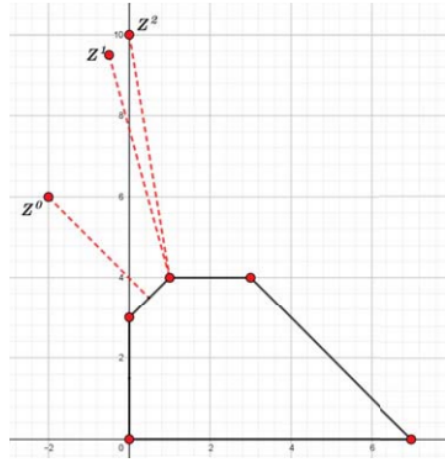


FIGURE 3.6 – Exemple 02 , D.C et D.C.A ,F6

### 3.4 Exemple d'application d'approximation extérieure :

Soit à résoudre le problème (P) suivant :

$$(P) \begin{cases} \min f(x, y) \\ (x, y) \in H \end{cases}$$

Où :  $f(x, y) = x + y$ , et  $H = \{(x, y) \in \mathbb{R}^2 / x^2 + y^2 \leq 1\}$ .

#### Application de l'algorithme de la méthode d'AE :

Choisir un hyperrectangle  $D_1$  telle que  $D_1 \supseteq H$ .

Soit  $D_1$  un carré,  $D_1 = \{(x, y) \in \mathbb{R}^2 / -1 \leq x \leq 1, -1 \leq y \leq 1\}$

Soit à résoudre le problème ( $P_1$ ) suivant :

$$(P_1) \begin{cases} \min f(x, y) \\ (x, y) \in D_1 \end{cases} \iff (P_1) \begin{cases} \min x + y \\ x \leq 1 \\ -x \leq 1 \\ y \leq 1 \\ -y \leq 1 \end{cases}$$

Après la résolution du problème ( $P_1$ ) avec la méthode des points intérieurs, la solution trouvée est  $X^1 = (-1.000, -1.0000)$ , et  $f(X^1) = -2.0000$ .

Comme  $X^1 \notin H$ , alors on construit la coupe linéaire  $L_1(X)$  de la manière suivante :

Soit  $g(x, y) = x^2 + y^2 - 1$ ,  $\nabla g(x, y) = (2x, 2y)$ .

$L_1(X) = g(X^1) + \nabla g(X^1)(X - X^1)$ , avec  $X = (x, y)$ .

$X^1 = (-1, -1) \Rightarrow L_1(X) = -2x - 2y - 3 \leq 0$  On ajoute la contrainte linéaire  $L_1(X)$  au problème ( $P_1$ ), et comme ça on aura le problème ( $P_2$ )

$$(P_2) \begin{cases} \min x + y \\ x \leq 1 \\ -x \leq 1 \\ y \leq 1 \\ -y \leq 1 \\ -2x - 2y \leq 3 \end{cases}$$

à résoudre toujours avec la méthode des points intérieurs, et on obtiendra la solution  $X^2$ .

$$X^2 = (-0.7500, -0.7500), \quad f(X^2) = -1.5000.$$

Si  $X^2 \in H$ , alors  $X^2$  est la solution optimale globale.

Sinon, on construit une autre coupe linéaire  $L_2(X)$ , et ainsi de suite.

$$\text{Comme } X^2 \notin H, \text{ alors } L_2(X) = -\frac{3}{2}x - \frac{3}{2}y - \frac{17}{8} \leq 0.$$

On ajoute la contrainte linéaire  $L_2(X)$  au problème  $(P_2)$ , et on aura le problème  $(P_3)$  suivant :

$$(P_3) \begin{cases} \min x + y \\ x \leq 1 \\ -x \leq 1 \\ y \leq 1 \\ -y \leq 1 \\ -2x - 2y \leq 3 \\ -\frac{3}{2}x - \frac{3}{2}y \leq \frac{17}{8} \end{cases}$$

La résolution du problème  $(P_3)$ , nous donnera la solution suivante :

$$X^3 = (-0.7083, -0.7083), \quad f(X^3) = -1.4167.$$

$$X^3 \notin H \Rightarrow L_3(X) = -1.4166x - 1.4166y - 2.0034 \leq 0.$$

On ajoute la contrainte linéaire  $L_3(X)$  au problème  $(P_3)$ , et on aura le problème  $(P_4)$  suivant :

$$(P_4) \begin{cases} \min x + y \\ x \leq 1 \\ -x \leq 1 \\ y \leq 1 \\ -y \leq 1 \\ -2x - 2y \leq 3 \\ -\frac{3}{2}x - \frac{3}{2}y \leq \frac{17}{8} \\ -1.4166x - 1.4166y \leq 2.0034 \end{cases}$$

La résolution du problème  $(P_4)$ , nous donnera la solution suivante :

$$X^4 = (-0.7071, -0.7071), \quad f(X^4) = -1.4142.$$

$X^4 \in H$ , car  $g(X^4) = 1$  d'où  $X^4$  est la solution optimale globale du problème de départ  $(P)$ .

## 3.5 Méthode de borne inférieure quadratique (QLB)

### Exemple 1

Cette fonction étant choisie, au hasard, on pouvait choisir une autre fonction dans les problèmes Tests par exemples. Soit  $f$  une fonction tel que  $f(x) = -x^3 + x^2, x \in [0, 2]$ . Nous avons  $f''(x) = -6x + 2$  et  $-10 \leq f''(x) \leq 2, K_q = 2, K_\alpha = 10$ . Les bornes inférieures sont données par

$$LB_q(x) = x^2 - 4x, \quad LB_\alpha(x) = -x^3 + 6x^2 - 10x.$$

Le minimum de  $LB_q(x)$  est atteint à  $x = 2$  et alors il est le minimum global de la fonction objectif.

Le minimum de  $LB_\alpha(x)$  est atteint au point  $x = 2 - \frac{\sqrt{6}}{3}$  qui n'est pas le minimum global de la fonction objectif. Pour cet exemple  $LB_q(x)$  est meilleur que  $LB_\alpha(x)$ .

### Remarque :

Cette fonction étant choisie, au hasard, on pouvait choisir une autre fonction dans les problèmes Tests par exemple. Si nous prenons  $f(x) = x^3 - x^2, x \in [0, 2]$ . Nous obtenons  $K_q = 10$  et  $K_\alpha = 2$ .

Par conséquent  $LB_\alpha(x)$  est meilleure que  $LB_q(x)$ .

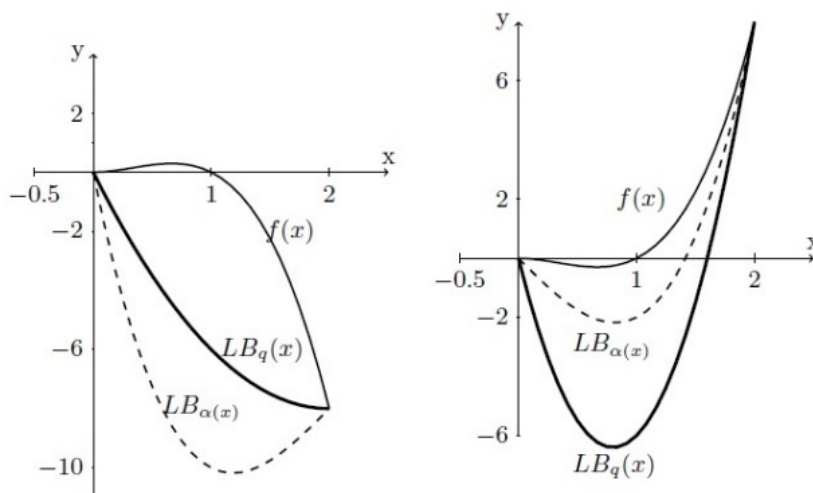


FIGURE 3.7 – Comparaison de deux bornes inférieures  $LB_q(x)$  et  $LB_\alpha(x)$ , exemple 1.

### Exemple 2

Cette fonction étant choisie, au hasard, on pouvait choisir une autre fonction dans les problèmes.

Tests par exemples. Soit  $f$  une fonction  $f(x) = \sin x + \cos x, x \in [0, 2\pi]$ . Nous avons  $f''(x) = -\sin x - \cos x$  et  $-2 \leq f''(x) \leq 2; K_\alpha = K_q = 2$ . Les deux bornes inférieures sont données par

$$LB_q(x) = 1 - x(2\pi - x), \quad LB_\alpha(x) = \sin x + \cos x - x(2\pi - x).$$

Le minimum de  $LB_q(x)$  est atteint au point  $x = \pi$ , et  $LB_q(\pi) = -1 - \pi^2$ . Le minimum de  $LB_\alpha(x)$  est atteint au point  $x = 3.46$ , et  $LB_\alpha(3.46) = -11.02$ .

Pour cet exemple  $LB_q(x)$  est meilleur que  $LB_\alpha(x)$ .

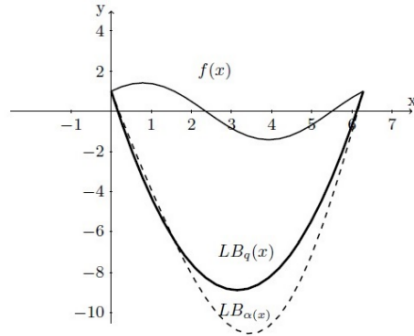


FIGURE 3.8 – Comparaison de deux bornes inférieures  $LB_q(x)$  et  $LB_\alpha(x)$ , exemple 2.

# Conclusion générale

À travers ce mémoire, nous avons cherché à explorer de manière approfondie les deux grandes branches de l'optimisation mathématique : **l'optimisation locale** et **l'optimisation globale**. Nous avons commencé par établir les fondements théoriques nécessaires, en insistant sur les notions de convexité, les propriétés des fonctions et ensembles convexes, ainsi que les conditions d'optimalité du premier et du second ordre.

Dans un second temps, nous nous sommes intéressés à **l'optimisation globale**, en étudiant plusieurs **méthodes déterministes** reconnues pour leur capacité à garantir la recherche de l'optimum global. Parmi elles, nous avons analysé des approches telles que la *méthode de Branch and Bound*, la *méthode d'approximation extérieure*, la *méthode D.C. (différence de fonctions convexes)*, la *méthode de Piyavskii*, ainsi que des variantes comme la *méthode  $\alpha BB$*  et la *méthode de borne inférieure quadratique*.

Enfin, nous avons illustré l'application de ces méthodes à travers une série d'**exemples numériques**, ce qui nous a permis de mieux comprendre leur comportement, leur efficacité, mais aussi leurs limites en pratique.

Ce travail nous a permis de consolider nos connaissances théoriques en optimisation, tout en développant une approche plus concrète et algorithmique des problèmes complexes. Nous espérons que ce mémoire contribuera à éclairer d'autres étudiants ou chercheurs intéressés par ce domaine riche et en constante évolution, en particulier dans le cadre de la *recherche opérationnelle* et de l'*optimisation non convexe*.

## Perspectives

Ce travail sur l'optimisation locale et globale a permis d'explorer les principales méthodes déterministes appliquées à des problèmes sans ou avec contraintes. Plusieurs pistes de recherche peuvent être envisagées pour approfondir et étendre les résultats obtenus.

- **Hybridation des méthodes locales et globales** : Développer des algorithmes hybrides qui exploitent une méthode globale (comme *Branch and Bound* ou  $\alpha BB$ ) pour explorer l'espace de recherche, puis utilisent une méthode locale (par exemple une descente de gradient ou SQP) pour raffiner les solutions candidates.
- **Applications à des problèmes réels de grande dimension** : Étendre les méthodes à des problèmes issus de domaines appliqués comme la finance (optimisation de portefeuille), l'énergie (gestion de réseaux) ou l'intelligence artificielle (entraînement de modèles non convexes).
- **Introduction des méthodes stochastiques** : Étudier des algorithmes tels que les algorithmes génétiques, le recuit simulé, ou l'optimisation par essaim particulaire (PSO), qui permettent de traiter efficacement des fonctions bruitées ou non différentiables.

- **Réduction de la complexité algorithmique :** Améliorer les méthodes existantes pour qu'elles soient plus efficaces en termes de temps de calcul, notamment via la relaxation convexe améliorée, le partitionnement adaptatif ou l'apprentissage automatique pour guider la recherche.
- **Implémentation logicielle avancée :** Développer une bibliothèque logicielle (en Python ou MATLAB) regroupant les méthodes étudiées, avec visualisation graphique des itérations, interface utilisateur et outils de comparaison automatique entre les algorithmes.

# Bibliographie

- [1] N. Abdessemed. Contribution à l'étude de la dualité quasi-convexe en optimisation. Mémoire de magister, Université El Hadj Lakhdar Batna, 2007.
- [2] Claire S. Adjiman, Ioannis P. Androulakis, and Christodoulos A. Floudas. A global optimization method,  $\alpha$ bb, for general twice-differentiable constrained nlp—ii. implementation and computational results. Computers & Chemical Engineering, 22(9) :1159–1179, 1998.
- [3] Claire S. Adjiman, Stefan Dallwig, Christodoulos A. Floudas, and Arnold Neumaier. A global optimization method,  $\alpha$ bb, for general twice-differentiable constrained nlp— i. theoretical advances. Computers & Chemical Engineering, 22(9) :1137–1158, 1998.
- [4] Ioannis G. Akrotirianakis and Christodoulos A. Floudas. Computational experience with a new class of convex underestimators : Box-constrained nlp problems. Journal of Global Optimization, 29 :249–264, 2004.
- [5] Ioannis G. Akrotirianakis and Christodoulos A. Floudas. A new class of improved convex underestimators for twice continuously differentiable constrained nlp. Journal of Global Optimization, 30 :367–390, 2004.
- [6] Ioannis P. Androulakis, Costas D. Maranas, and Christodoulos A. Floudas.  $\alpha$ bb : A global optimization method for general constrained nonconvex problems. Journal of Global Optimization, 7(4) :337–363, 1995.
- [7] Andreas Antoniou and Wu-Sheng Lu. Practical Optimization : Algorithms and Engineering Applications, volume 19. Springer, 2007.
- [8] Mohammed Chebbah. Résolutions et implémentations de problèmes en optimisation globale. Thèse de doctorat es sciences, Université Mouloud Maameri de Tizi-Ouzou, Département de Mathématiques, Faculté des Sciences, 2020.
- [9] Elliot Ward Cheney and Allen A. Goldstein. Newton's method for convex programming and tchebycheff approximation. Numerische Mathematik, 1(1) :253–268, 1959.
- [10] T. Pham Dinh. Calcul du maximum d'une forme quadratique définie positive sur la boule unité de la norme du max. Rapport technique, Technical Report, Grenoble, 1976.
- [11] T. Pham Dinh et al. Algorithms for solving a class of nonconvex optimization problems : Methods of subgradients. In North-Holland Mathematics Studies, volume 129, pages 249–271. Elsevier, 1986.
- [12] Saoudi Moussa et Ahmed Belaidi. Optimisation non linéaire et introduction à l'optimisation globale. Mémoire de master, Université de bouira, Bouira, Algérie, 2020. Mémoire de Master en Mathématiques, option Recherche Opérationnelle. Année universitaire : 2019/2020.

- [13] B. Galerne. Optimisation (mml1e31), 2017. Notes de cours, Master 1 Mathématiques et Modélisation (MM), Université Paris Descartes, 2017-2018.
- [14] J. Grenet. Vade mecum : Optimisation statique. lagrangien et conditions de kuhn et tucker, 2007. TD d'Économie, École Normale Supérieure, Année 2007-2008.
- [15] Djaouida Guettal. Efficacité et fiabilité des méthodes utilisant l'approche branch-and-bound pour l'optimisation globale non convexe. Thèse de doctorat, Université des Sciences et de la Technologie Houari-Boumediene (USTHB), 2018.
- [16] Sid Ali Hamdache. Résolution d'un problème de programmation linéaire avec la méthode dc. Mémoire de fin d'étude, Université de bouira, Bouira, Algérie, 2018. Mémoire de Master en Mathématiques, option Recherche Opérationnelle. Année universitaire : 2017/2018.
- [17] Philip Hartman. On functions representable as a difference of convex functions. 1959. Unpublished manuscript.
- [18] Jean-Baptiste Hiriart-Urruty. Conditions nécessaires et suffisantes d'optimalité globale en optimisation de différences de deux fonctions convexes. Comptes Rendus de l'Académie des Sciences. Série I. Mathématique, 309 :459–462, 1989.
- [19] Karla Leigh Hoffman. A method for globally minimizing concave functions over convex sets. Mathematical Programming, 20 :22–32, 1981.
- [20] E. M. Landis. On functions representable as the difference of two convex functions. Doklady Akad. Nauk SSSR (NS), 80 :9–11, 1951.
- [21] Costas D. Maranas and Christodoulos A. Floudas. A deterministic global optimization approach for molecular structure determination. The Journal of Chemical Physics, 100(2) :1247–1261, 1994.
- [22] Juan Enrique Martínez-Legaz. On weierstrass extreme value theorem. Optimization Letters, 8(1) :391–393, 2014.
- [23] David Q. Mayne and Elijah Polak. Outer approximation algorithm for nondifferentiable optimization problems. Journal of Optimization Theory and Applications, 42(1) :19–30, 1984.
- [24] Mahdi Moeini. La programmation DC et DCA pour l'optimisation de portefeuille. Phd thesis, Université de Metz, Metz, 2008.
- [25] MOHAND OUANES. Cours et td d'onl. optimisation non linéaire.
- [26] MOHAND OUANES, Hoai An Le Thi, Trong Phuc Nguyen, and Ahmed Zidna. New quadratic lower bound for multivariate functions in global optimization. Mathematics and Computers in Simulation, 109 :197–211, 2015.
- [27] S.A. Piyavskii. An algorithm for finding the absolute extremum of a function. USSR Computational Mathematics and Mathematical Physics, 12 :57–67, 1972. (Zh. Vychisl. Mat. Mat. Fiz., 12(4) (1972) 888–896).
- [28] A. Rondepierre. Méthodes numériques pour l'optimisation non linéaire déterministe, 2017. 4ème année, INSA Toulouse, 2017-2018.
- [29] Pham Dinh Tao and Le Thi Hoai An. Convex analysis approach to dc programming : theory, algorithms and applications. Acta Mathematica Vietnamica, 22(1) :289–355, 1997.
- [30] Hoai An Le Thi. Contribution à l'optimisation non convexe et l'optimisation globale : Théorie, algorithmes et applications. Habilitation à diriger des recherches, Institut National des Sciences Appliquées (INSA), Rouen, France, 1997.

- [31] Hoai An Le Thi and Tao Pham Dinh. Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. Journal of Global Optimization, 11 :253–285, 1997.
- [32] John F. Toland. Duality in nonconvex optimization. Journal of Mathematical Analysis and Applications, 66(2) :399–415, 1978.
- [33] Hoang Tuy. On outer approximation methods for solving concave minimization problems. Technical report, Universität Bremen, Fachbereiche Mathematik/Informatik, Elektrotechnik, 1983.
- [34] Hoang Tuy. Convex programs with an additional reverse convex constraint. Journal of Optimization Theory and Applications, 52(3) :463–486, 1987.