

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE DE BOUIRA



FACULTE DES SCIENCES ET DES SCIENCES APPLIQUEES

DEPARTEMENT DE MATHEMATIQUES

MEMOIRE POUR L'OBTENTION DU DIPLOME

DE MASTER EN MATHEMATIQUES

OPTION

Recherche opérationnelle

THEME

**Optimisation non linéaire et Introduction à
l'optimisation globale**

Présenté par : Saoudi Moussa et Ahmed Belaidi

Soutiendra-le :

Devant le jury composé de :

| | | | |
|-------------|-----------------------|-----|------------|
| President: | Boudref Mohamed Ahmed | MCA | UMO Bouira |
| Examineur : | Alem lala Maghnia | MCB | UMO Bouira |
| Examineur : | Boughani Lhadi | MAA | UMO Bouira |
| Promoteur : | Bekri Houria | MAA | UMO Bouira |

Année universitaire: 2019/2020

Remerciements

Un grand merci revient à Dieu le tout puissant que lui seul nous a donné la volonté de réaliser ce modeste travail.

Un grand merci à notre promotrice, *M^{me}. Bekri Houria*, pour ses conseils et son aide et qui a mis à notre disposition tout le nécessaire pour réaliser ce travail.

Nous remercions Monsieur *M^r. Boughani Lhadi* et *M^{me}. Alem lala Maghnia* pour avoir accepté de juger ce travail.

Nous remercions Monsieur *M^r. Mohamed Ahmed Boudref* pour avoir accepté de présider le jury de ce mémoire.

Nous remercions tous les membres de département des mathématiques pour leurs aides, leurs orientations et leurs participations pour atteindre le but recherché.

Nous tenons à remercier nos parents pour leurs encouragements et leurs soutiens et les enseignants pour l'aide et la patience dont ils ont fait preuve tout au long de ce travail, ainsi que leurs orientations afin d'accéder à l'objectif tracé.

Sans oublier les bons collègues, nous les remercions pour toutes les belles journées qu'on a passées ensemble.

Et tous les autres sans exception.

Ainsi, à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail, nous leur disons :

Merci pour tout.

Dédicaces

Tout d'abord je remercie Dieu le tout puissant, de m'avoir aidé dans les moments les plus difficiles, de m'avoir aidé à accomplir ce travail.
Qu'il soit toujours dans mon cœur et ma tête.

Je dédie ce modeste travail :

** A Mon cher père : je te souhaite une longue vie
et Dieu vous protège pour moi.*

** A mes sœurs et mes beaux frères.*

** A mes collègues du service national (R21), en particulier
Radouan (khabisa).*

** A mon binôme et amie : Moussa .*

** A tous mes amis et toute la famille Belaidi.*

** A toute personne qui mérite l'appréciation et le respect de ma part.*

AHMED.

Dédicaces

Je dédie ce modeste travail :

- * *A ma chère mère Reguia..*
- * *A mon père Ahmed dont le mérite, ces qualités humaines m'ont permis de vivre ce jour.*
- * *A mon épouse et mes enfants khadidja et fatima ezahra.*
- * *A mon frère et mes sœurs.*
- * *A mes enseignants.*

MOUSSA.

Table des matières

| | |
|--|-----------|
| Introduction Générale | 2 |
| 1 Notions Fondamentales | 5 |
| 1.1 Introduction | 5 |
| 1.2 Généralités | 5 |
| 1.3 Rappels sur les matrices | 5 |
| 1.3.1 Vecteurs et matrices | 5 |
| 1.3.2 Matrices et vecteurs partitionnés | 6 |
| 1.3.3 Signe d'une matrice | 7 |
| 1.3.4 Mineurs d'une matrice | 7 |
| 1.3.5 Caractérisations | 8 |
| 1.3.6 Gradient | 8 |
| 1.3.7 Matrice Jacobienne et matrice Hessienne | 9 |
| 1.4 Convexité | 9 |
| 1.4.1 Ensembles convexes | 10 |
| 1.4.2 Fonctions convexes | 10 |
| 1.4.3 Problèmes convexes | 11 |
| 2 Optimisation non linéaire sans contraintes | 13 |
| 2.1 Introduction | 13 |
| 2.2 Formulation mathématique d'un problème d'optimisation | 13 |
| 2.3 Minima locaux et globaux | 14 |
| 2.4 Conditions d'optimalité | 17 |
| 2.4.1 Pourquoi avons-nous besoin de conditions d'optimalité? | 17 |
| 2.4.2 Condition nécessaire d'optimalité du premier ordre | 17 |
| 2.4.3 Condition nécessaire d'optimalité du second ordre | 18 |
| 2.4.4 Condition suffisante d'optimalité du second ordre | 18 |
| 2.5 Les Méthodes d'optimalité | 20 |
| 2.5.1 Introduction | 20 |
| 2.5.2 La méthode de descente basée sur le gradient | 20 |
| 2.5.3 Méthode de Newton | 21 |
| 2.5.4 Étude comparative des méthodes d'optimisation sans contraintes . . | 23 |

| | | |
|----------|--|-----------|
| 3 | Optimisation non linéaire avec contraintes | 25 |
| 3.1 | Introduction | 25 |
| 3.2 | Théorèmes généraux d'existence | 26 |
| 3.3 | Conditions d'optimalité | 28 |
| 3.3.1 | Cas de contraintes quelconques | 28 |
| 3.3.2 | Cas de contraintes linéaires de type égalités | 28 |
| 3.3.3 | Cas de contraintes linéaires de type inégalités | 31 |
| 3.3.4 | Cas de contraintes égalités non linéaires | 33 |
| 3.3.5 | Cas de contraintes inégalités non linéaires | 35 |
| 3.4 | Les Méthodes d'optimalité | 37 |
| 3.4.1 | Les méthodes de pénalité intérieure | 37 |
| 3.4.2 | Les méthodes de pénalité extérieure | 39 |
| 3.4.3 | Étude comparative des méthodes d'optimisation avec contraintes | 42 |
| 4 | Introduction à l'Optimisation Globale | 44 |
| 4.1 | Introduction | 44 |
| 4.2 | La méthode de Branch and Bound | 44 |
| 4.2.1 | Le principe de la méthode de Branch and Bound | 45 |
| 4.2.2 | L'algorithme de base de Branch and Bound | 46 |
| 4.2.3 | La convergence de la méthode branch and bound | 47 |
| 4.2.4 | Application numérique de Branch-and-Bound | 48 |
| 4.2.5 | Exemples d'applications | 49 |
| 4.3 | La méthode de Piyavskii | 52 |
| 4.3.1 | Introduction | 52 |
| 4.3.2 | La méthode de Piyavskii-Shubert | 53 |
| 4.3.3 | Algorithme de Piyavskii généralisé | 54 |
| 4.3.4 | Convergence de l'algorithme | 57 |
| 4.3.5 | Application numérique de l'algorithme | 58 |
| | Conclusion Générale | 59 |
| | Bibliographie | 60 |

Introduction générale

La recherche opérationnelle (RO), aussi appelée science du management ou science de la décision, est une discipline dont l'objet est d'aider les gestionnaires à prendre des décisions en utilisant des modèles et des méthodes scientifiques adaptées. Une des parties essentielles de la recherche opérationnelle est la programmation non linéaire, qui étudie l'optimisation d'une fonction objectif non linéaire .

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

L'optimisation est devenue une discipline incontournable du monde moderne dans lequel nous vivons, car celui-ci est sujet à une compétition internationale excessive et croissante.

Dés lors, il devient nécessaire, voir vital pour les entreprises comme pour les gouvernements, de maximiser ou de minimiser toutes sortes de choses ; par exemple maximiser les profits tout en minimisant les pertes, améliorer si possible de façon optimale certains processus de fabrication ou les fonctionnalités, de certains processus objets ou produits.

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

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

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

Dans ce présent document, nous allons intéressé à l'étude des problèmes d'optimisation non linéaire et d'optimisation globale, pour ce faire, nous avons opté pour le plan du travail suivant :

Dans le premier chapitre nous rappelons les généralités sur les matrices et quelques notions et propriétés d'analyse convexe .

Dans le deuxième chapitre nous abordons les conditions d'optimalité de l'optimisation sans contraintes ainsi que la description des méthodes de résolution de ce type de problèmes d'optimisation.

Parmi lesquelles on peut citer :

- Méthode d'optimisation basée sur le gradient.
- Méthode de Newton.

Dans le troisième chapitre nous abordons les conditions d'optimalité sous contraintes et les méthodes de résolution pour cette optimisation, à savoir :

- Les méthodes de pénalité intérieure.
- Les méthodes de pénalité extérieure.

Le quatrième chapitre sera consacré à l'optimisation globale. Après avoir présenté la méthode de branch-and-bound pour la résolution d'un problème non convexe et non linéaire, dont la fonction objectif est une fonction lipschitzienne (un cas particulier des fonctions de Holder), nous présentons aussi la méthode de piyavskii dans \mathbb{R} avec la construction de la fonction borne inférieure, ainsi que son extension dans \mathbb{R}^n .

Nous terminerons notre travail par une conclusion générale.

Chapitre 1

Notions Fondamentales

1.1 Introduction

En général, les modèles non linéaires sont beaucoup plus difficiles à résoudre que les modèles linéaires en raison de leurs structures. Dans ce chapitre, on donnera brièvement quelques notions de base qui nous seront utiles pour la suite de ce mémoire.

1.2 Généralités

De façon générale, un problème d'optimisation mathématique peut s'écrire sous la forme suivante :

$$(P) \left\{ \begin{array}{l} \min \text{ ou } (\max) f(x) \\ h(x) \leq 0 \\ g(x) \geq 0 \\ k(x) = 0 \\ x \in \mathbb{R}^n \end{array} \right.$$

Où le vecteur $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ représente les inconnus du problème (P) , $f(x)$ est la fonction objectif optimisée (minimiser ou maximiser).

Soit $S = \{x \in \mathbb{R}^n, h(x) \leq 0, g(x) \geq 0, k(x) = 0\}$ une partie de \mathbb{R}^n qu'on appelle le domaine (ensemble) réalisable du problème (P) .

1.3 Rappels sur les matrices

1.3.1 Vecteurs et matrices

Définition 1.1 Soient $n, m \in \mathbb{N}^*$. Une matrice d'ordre $n \times m$ à coefficients dans \mathbb{R} est un tableau à deux dimensions, ayant m lignes et n colonnes, représenté sous la forme suivante :

$$A = A(I, J) = (a_{i,j}, i \in I, j \in J) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

où $I = \{1, 2, \dots, m\}$ et $J = \{1, 2, \dots, n\}$ représentent respectivement l'ensemble des indices des lignes et des colonnes de A .

Pour des calculs pratiques, la matrice A se note aussi :

$$A = (a_1, a_2, \dots, a_j, \dots, a_n) = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_m \end{pmatrix}$$

où $a_j = A(I, j) = \begin{pmatrix} A_{1j} \\ A_{2j} \\ \vdots \\ A_{mj} \end{pmatrix}$ est un vecteur colonne de dimension m ;

$A_i = A(i, J) = (a_{i1}, a_{i2}, \dots, a_{in})$ est un vecteur ligne de dimension n .
Chaque vecteur, noté $x = x(J) = (x_j, j \in J)$, sera ainsi considéré comme un vecteur-colonne tandis que le vecteur-ligne sera noté x^T . La matrice transposée de A sera notée :

$$x^T = A^T(J, I) = (a_{j,i}, j \in J, i \in I)$$

Notons qu'un vecteur-colonne de dimension n peut être considéré comme une matrice d'ordre $(n \times 1)$, tandis qu'un vecteur ligne de dimension n peut être considéré comme une matrice d'ordre $(1 \times n)$.

La matrice A est dite carrée si on a $n = m$; de plus, si $A = A^T$, la matrice est dite symétrique.

La matrice identité d'ordre n sera notée I_n .

1.3.2 Matrices et vecteurs partitionnés

On peut effectuer le produit d'une matrice A et d'un vecteur x , après les avoir partitionnés judicieusement. On dit alors qu'on a effectué le produit par blocs. En effet, si l'on a :

$$A = [A_1/A_2], \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

alors on peut écrire :

$$Ax = [A_1/A_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = A_1x_1 + A_2x_2$$

De même pour

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

L'équation $Ax = b$ peut alors s'écrire :

$$\begin{cases} A_{11}x_1 + A_{12}x_2 = b_1 \\ A_{21}x_1 + A_{22}x_2 = b_2 \end{cases}$$

1.3.3 Signe d'une matrice

Soit A une matrice carrée symétrique d'ordre n . Soit X un vecteur colonne de \mathbb{R}^n . On note X^T sa transposé. A est dite :

- **Semi-définie positive** si $X^TAX \geq 0$ pour tout $X \in \mathbb{R}^n$.
- **Définie positive** si $X^TAX > 0$ pour tout $X \in \mathbb{R}^n \neq 0$.
- **Semi-définie négative** si $X^TAX \leq 0$ pour tout $X \in \mathbb{R}^n$.
- **Définie négative** si $X^TAX < 0$ pour tout $X \in \mathbb{R}^n \neq 0$.
- **Indéfinie** sinon.

En pratique, on peut vérifier le signe d'une matrice en examinant ses valeurs propres ou ses mineurs principaux dominants.

1.3.4 Mineurs d'une matrice

Mineurs principaux d'une matrice

Soit A une matrice carré symétrique de dimension $(n \times n)$. Un mineur principal d'ordre k est le déterminant de la sous-matrice de A d'ordre k obtenu en supprimant $(n - k)$ lignes dernières et les $(n - k)$ colonnes correspondantes dans A .

Exemple 1.1 Soit la matrice A définie par :

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

- Les trois mineurs principaux d'ordre 1 de A sont a_{11}, a_{22}, a_{33} .
- Les trois mineurs principaux d'ordre 2 de A sont : $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}, \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}$.
- Le mineur principal d'ordre 3 de A est : $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$.

On note Δ_i .

Mineurs principaux diagonaux d'une matrice

Le mineur principal diagonal d'ordre k de la matrice A est le déterminant de la matrice de taille $(k \times k)$ 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.

Exemple 1.2 Soit $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$

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

1.3.5 Caractérisations

A une matrice carrée symétrique d'ordre n .

- A **définie positive** \iff ses n mineurs principaux diagonaux D_k sont > 0 .
- A **semi-définie positive** \iff tous ses mineurs principaux diagonaux D_k sont ≥ 0 .
- A **définie négative** \iff ses n mineurs principaux diagonaux D_k sont alternativement < 0 (k impaire) et > 0 (k paire).
- A **semi-définie négative** \iff tous ses mineurs principaux diagonaux D_k sont alternativement ≤ 0 (k impaire) et ≥ 0 (k paire).

1.3.6 Gradient

Le gradient joue un rôle essentiel dans le développement et l'analyse des algorithmes d'optimisation.

Définition 1.2 [1] La fonction notée : $\nabla f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ est appelée Gradient de f au point $x = (x_1, \dots, x_n)^t$ si toutes les dérivées partielles existent, on définit le gradient par la formule suivante :

$$\nabla f(x) = \frac{\partial f(x)}{\partial x_i}_{i=1, \dots, n} = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}$$

On le note dans \mathbb{R}^n par : $\nabla f(x) = f''(x)$.

1.3.7 Matrice Jacobienne et matrice Hessienne

Matrice Jacobienne

Soit $F = f_1, \dots, f_m$ une fonction définie de \mathbb{R}^n dans \mathbb{R}^m . A tout vecteur $x^* = (x^1, \dots, x^n)$, la fonction F associe le vecteur des fonctions $(f_1(x^*), \dots, f_m(x^*))$. On appelle matrice jacobienne de F la matrice de dimension (m, n) , $J_F(x)$ des dérivées partielles d'ordre 1 des m fonctions qui composent F : [2]

$$J_F(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x^*) & \dots & \frac{\partial f_1}{\partial x_n}(x^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x^*) & \dots & \frac{\partial f_m}{\partial x_n}(x^*) \end{pmatrix}$$

Matrice Hessienne

On appelle Hessienne de f la matrice carrée définie par :

$$H(x) = \nabla^2 f(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}_{j=1, \dots, n}$$

Alors

$$H(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial^2 x_1} & \dots & \frac{\partial f_1(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1(x)}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f(x)}{\partial^2 x_n} \end{pmatrix}$$

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

On note dans $\mathbb{R}^n \times \mathbb{R}^n$: $H(x) = f''(x)$ [2]

1.4 Convexité

Pour étudier les problèmes d'optimisation, il est nécessaire de recourir à des outils scientifiques dont l'étude est basée sur l'analyse convexe. En effet, l'hypothèse de convexité va jouer un rôle très important pour la plupart des algorithmes que nous décrirons, la convergence vers l'optimum ne pourra être démontrée qu'avec cette hypothèse.

Nous allons ici rappeler quelques notions de convexité importantes aux quelles nous ferons appel par la suite, ainsi que quelques propriétés.

1.4.1 Ensembles convexes

Définition 1.3 Un ensemble $S \subset \mathbb{R}^n$ est dit convexe si et seulement si :

$$\forall x_1, x_2 \in S : \alpha x_1 + (1 - \alpha)x_2 \in S, \forall \alpha \in [0, 1]$$

Cette définition peut s'interpréter en disant que S est convexe si et seulement si pour deux points quelconques x_1 et x_2 pris dans S , le segment $[x_1, x_2]$ tout entier est contenu dans S .

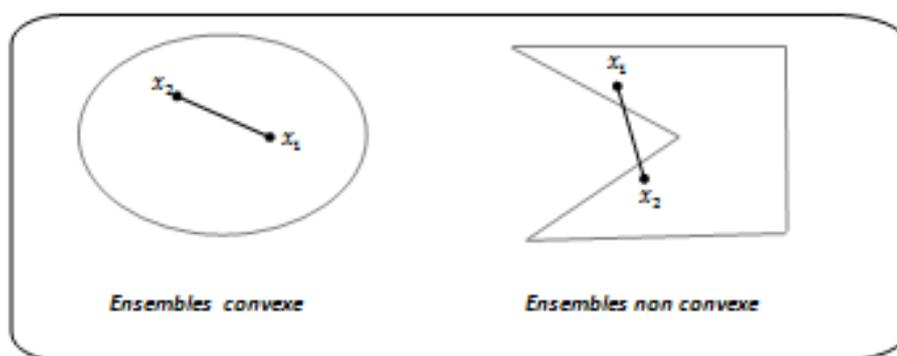


FIG. 1.1 – Interprétation géométrique d'ensembles convexe et non convexe

Proposition 1.1 Soit S_1, S_2, \dots, S_p des ensembles convexes de \mathbb{R}^n alors :

$$S = \bigcup_{i=1}^{i=p} S_i$$

est un ensemble convexe.

Théorème 1.1 Soient S_1 et S_2 deux ensembles convexes de \mathbb{R}^n . Alors

1. $S_1 \cap S_2$ est convexe.
2. $S_1 \pm S_2 = \{x_1 \pm x_2 \mid x_1 \in S_1, x_2 \in S_2\}$ est convexe.
3. L'intersection d'un nombre fini ou infini d'ensembles convexes est convexe.
4. S ensemble convexe, Alors αS , où $\alpha \in \mathbb{R}$ est un ensemble convexe.

1.4.2 Fonctions convexes

Définition 1.4

1. On dit qu'une fonction $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$, définie sur un ensemble convexe S , est convexe, si elle vérifie :

$$\forall x_1 \in S, \forall x_2 \in S, \forall \gamma \in [0, 1], \text{ on a : } f(\gamma x_1 + (1 - \gamma)x_2) \leq \gamma f(x_1) + (1 - \gamma)f(x_2).$$

2. Une fonction f est dite strictement convexe si l'inégalité ci-dessus est stricte pour $x_1 \neq x_2$ et $\gamma \in]0, 1[$.
3. Une fonction f est dite concave si $(-f)$ est convexe.

L'interprétation géométrique de cette définition est que le graphe d'une fonction convexe est toujours en dessous du segment reliant les points $(x_1, f(x_1))$ et $(x_2, f(x_2))$.

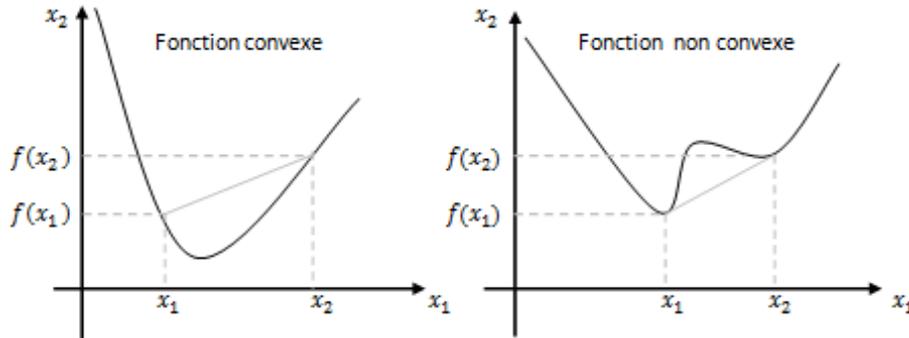


FIG. 1.2 – Interprétation géométrique d'une fonction convexe et non convexe

Exemple 1.3 *Un des exemples basiques de fonctions convexes est la fonction indicatrice. Soit $S \subset \mathbb{R}^n$ un sous-ensemble non vide ; la fonction indicatrice $I_s : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ est définie par :*

$$I_s(x) = \begin{cases} 0, & \text{si } x \in S \\ +\infty, & \text{ailleurs.} \end{cases}$$

La fonction indicatrice I_s est convexe si et seulement si S est convexe.

Théorème 1.2

1. Soit f une fonction convexe sur un ensemble convexe $S \subset \mathbb{R}$ et un nombre réel $\alpha \geq 0$, alors αf est aussi une fonction convexe sur S .
2. Soient f_1, f_2 deux fonctions convexes sur un ensemble convexe S , alors $f_1 + f_2$ est aussi une fonction convexe sur S .
3. Soient f_1, f_2, \dots, f_m des fonctions convexes sur un ensemble convexe S et $\alpha_1, \alpha_2, \dots, \alpha_m$ des nombres réels, alors $\sum_{i=1}^m \alpha_i f_i$ est aussi une fonction convexe sur S .

1.4.3 Problèmes convexes

Maintenant que nous avons défini les notions d'ensemble convexe et de fonction convexe, nous pouvons définir la notion de problème convexe.

Soit (P) le problème défini par :

$$(P) \begin{cases} \text{minimiser } f(x) \\ x \in S \end{cases}$$

On dit que (P) est un problème convexe si S est convexe et $f(x)$ est une fonction convexe sur S .

Théorème 1.3

1. Si (P) est convexe alors toute solution locale de ce problème est une solution globale, c'est-à-dire les notions d'optimums global et local coïncident.
2. Dans les problèmes convexes certaines conditions nécessaires d'optimalité deviennent des conditions suffisantes.

Chapitre 2

Optimisation non linéaire sans contraintes

2.1 Introduction

Dans ce chapitre, nous rappelons les concepts mathématiques que nous jugeons nécessaires pour la présentation de ce projet. Par la suite, nous présentons les conditions nécessaires et suffisantes pour la minimisation d'une fonction non linéaire sans contraintes.

2.2 Formulation mathématique d'un problème d'optimisation

Il existe différents problèmes d'optimisation. Certaines caractéristiques permettent de les distinguer : comportent-ils des contraintes ? les fonctions objectifs sont-elles linéaires ? sont-elles quadratiques ? sont-elles convexes ? les domaines de définition des fonctions sont-ils continus ou discrets ? le problème contient-il une seule ou plusieurs fonctions objectifs ? Tous ces problèmes possèdent des structures différentes et ne peuvent être traités de la même façon.

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, on appelle problème de minimisation (maximisation) sans contraintes le problème (p_1) suivant :

$$(p_1) \iff \begin{cases} f(x) \rightarrow \min \text{ (max)} \\ x \in \mathbb{R}^n \end{cases}$$

- Le vecteur $x = (x_1, \dots, x_n)$ est appelé vecteur des variables de décisions du problème.
- La fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est la fonction objectif.

Définition 2.1 *Le problème d'optimisation (p_1) est dit programme linéaire si la fonction f est linéaire. Si la fonction objectif n'est pas linéaire, le problème est dit problème non-linéaire.*

Définition 2.2 (Fonction linéaire) Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite linéaire si elle s'écrit comme suit :

$$f(x) = C^T x = \sum_{i=1}^{i=n} c_i x_i$$

C : un vecteur de \mathbb{R}^n constant, indépendant de x . $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est dite linéaire si chaque'une de ses composantes $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $i = 1, \dots, m$ est linéaire $f(x) = Ax$ ou $A \in \mathbb{R}^{n \times m}$ est une matrice constante.

Définition 2.3 (Fonction affine) Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite affine si elle s'écrit :

$$f(x) = C^T x + d$$

C : un vecteur constant de \mathbb{R}^n . d : une constant de \mathbb{R} .

Définition 2.4 (Fonction non linéaire) Toute fonction qui n'est pas affine est dite non linéaire.

2.3 Minima locaux et globaux

Dans un problème d'optimisation non linéaire on distingue deux types de solutions :

Le minimum local et le minimum global.

Les minima locaux et globaux de f sont définis de la manière suivante :

Définition 2.5 Un vecteur $x^0 \in \mathbb{R}^n$ est un minimum local de f sur \mathbb{R}^n si $\exists \varepsilon > 0$ tel que $f(x^0) \leq f(x)$, $\forall x \in \mathbb{R}^n$, avec $\|x - x^0\| < \varepsilon$.

Définition 2.6 Un vecteur $x^0 \in \mathbb{R}^n$ est un minimum global de f sur \mathbb{R}^n si

$$f(x^0) \leq f(x), \forall x \in \mathbb{R}^n$$

Remarque 2.1 Le problème (p_1) peut ne pas admettre des solutions optimales, comme dans la figure 2.1 :

Remarque 2.2 Le problème (p_1) peut aussi admettre une infinité de solutions optimales, comme dans la figure 2.2 .

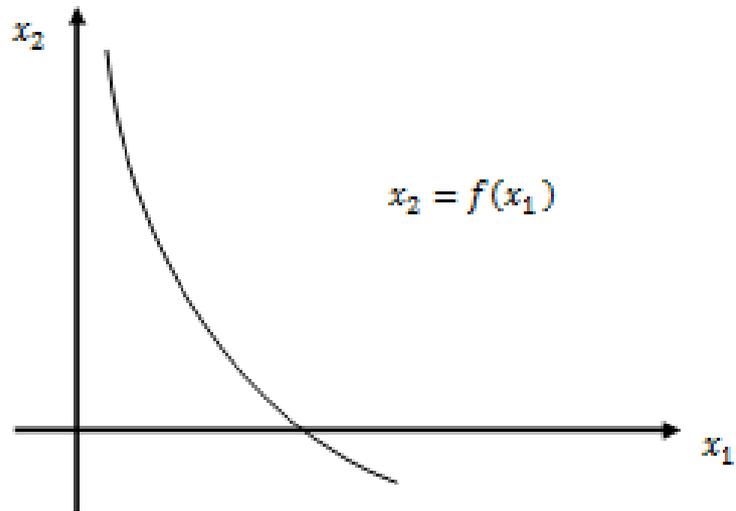


FIG. 2.1 – un problème n'admet pas de solutions

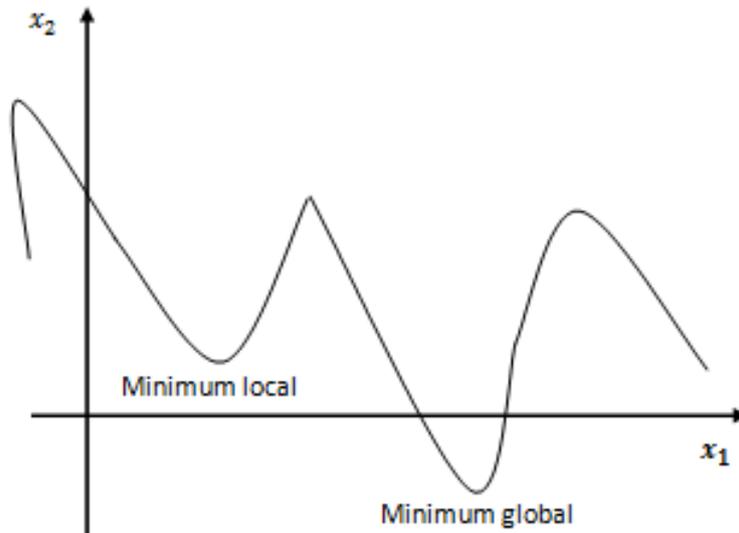


FIG. 2.2 – un problème admet une infinité de solutions

Remarque 2.3 Les maxima locaux et globaux sont définis de manière similaire.

Notons que x^0 est un maximum local (respectivement global) de la fonction f sur l'ensemble \mathbb{R}^n , si x^0 est un minimum local (respectivement global) de la fonction $(-f)$ sur \mathbb{R}^n .

Il découle de cette observation que tout problème de maximisation peut être réduit immédiatement à un problème de minimisation (et inversement) en multipliant la fonction objectif par -1 .

Définition 2.7 Un vecteur x^0 est dit solution optimale du problème (p_1) si pour tous les vecteurs x , $x \in \mathbb{R}^n$, on a :

$$f(x) \geq f(x^0).$$

Remarque 2.4 Dans le cas d'une fonction objectif convexe, il n'y a pas de distinction entre minimum local et global.

Tout minimum local est également global, comme l'établit le théorème suivant :

Théorème 2.1 Soit $f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe définie sur un ensemble convexe S . Alors, tout minimum local est également un minimum global. Si f est strictement convexe, alors il existe au plus un minimum global de f .

Théorème 2.2

1. Si x^0 est un minimum global, alors c'est un minimum local [5].
2. Si on connaît tous les minima locaux alors le plus petit est le minimum global [5].

Théorème 2.3 (Weierstrass) Soit $S \subseteq \mathbb{R}^n$ un ensemble non vide. Si $f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction semi-continue inférieurement sur S compact (fermé et borné), alors il existe $x^0 \in S$ tel que

$$\min_{x \in S} f(x) = f(x^0)$$

Théorème 2.4 Si $f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue sur \mathbb{R}^n . Si f est coercitive, i.e.

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

alors le problème (p_1) admet une solution optimale $x^0 \in \mathbb{R}^n$.

L'unicité résulte en général des propriétés de convexité de f .

2.4 Conditions d'optimalité

2.4.1 Pourquoi avons-nous besoin de conditions d'optimalité ?

Afin d'analyser ou de résoudre de manière efficace un problème d'optimisation, il est fondamental de pouvoir disposer de conditions d'optimalité. En effet, celles-ci nous servent non seulement à vérifier la validité des solutions obtenues, mais souvent l'étude de ces conditions aboutit au développement des algorithmes de résolution eux-mêmes.

On distingue deux types de conditions d'optimalité :

1. Les conditions nécessaires d'optimalité.
2. Les conditions suffisantes d'optimalité.

2.4.2 Condition nécessaire d'optimalité du premier ordre

Théorème 2.5 *Si x^0 est un minimum local du problème (p_1) et supposons que la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est continûment différentiable (f est continue et ses dérivées partielles $\partial f / \partial x_i$ sont continues) sur un ensemble ouvert $S \subset \mathbb{R}^n$ contenant x^0 , alors*

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

Démonstration. Soit $d \in \mathbb{R}^n$ un vecteur quelconque. On a :

$$g(\alpha) = f(x^0 + \alpha d) \geq f(x^0), \quad \forall \alpha \in \mathbb{R}, \quad (x^0 + \alpha d) \in S$$

En utilisant l'hypothèse que f est différentiable, on obtient :

$$0 \leq \lim_{\alpha \rightarrow 0^+} f(x^0 + \alpha d) - f(x^0)\alpha = dg(0)d\alpha = \lim_{\alpha \rightarrow 0^-} f(x^0 + \alpha d) - f(x^0)\alpha \leq 0$$

D'où

$$dg(0)d\alpha = d^T \nabla f(x) = 0$$

Comme d est choisi arbitrairement, alors

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

Les points x^0 tels que $\nabla f(x^0) = 0$ sont appelés points critiques (ou stationnaires) de f .

Remarque 2.5 *Si f est convexe, la condition nécessaire du premier ordre est également suffisante pour que x^0 soit un minimum global.*

2.4.3 Condition nécessaire d'optimalité du second ordre

Théorème 2.6 [5] Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois continûment différentiable et $x^0 \in \mathbb{R}^n$. Si x^0 est un minimum local (ou global) de (p_1) , alors $\nabla f(x^0) = 0$ et la matrice Hessienne de f , noté par $H(x) = \nabla^2 f(x^0)$, est semi définie positive ($H \geq 0$).

Démonstration. Supposons que f est deux fois continûment différentiable et $d \in \mathbb{R}^n$ un vecteur quelconque. Pour tout $\alpha \in \mathbb{R}$, le développement de Taylor d'ordre 2 s'écrit :

$$f(x^0 + \alpha d) - f(x^0) = \alpha[\nabla f(x^0)]^T d + \alpha^2 d^T \nabla^2 f(x^0) d + \sigma(\alpha^2)$$

Le passage à la limite quand $\alpha \rightarrow 0$ et l'utilisation de la relation :

$$\lim_{\alpha \rightarrow 0} \sigma(\alpha^2) \alpha^2 = 0$$

Entraînent

$$d^T \nabla^2 f(x^0) d \geq 0$$

Ce qui démontre que $\nabla^2 f(x^0)$ est une matrice semi-définie positive.

2.4.4 Condition suffisante d'optimalité du second ordre

Les conditions données précédemment sont nécessaires, c'est à-dire qu'elles doivent être satisfaites pour tout minimum local (ou global), cependant, tout vecteur vérifiant ces conditions n'est pas nécessairement un minimum.

Exemple 2.1 Soit la fonction définie par :

$$f(x) = -x^4, \quad x \in \mathbb{R}$$

Le point $x^0 = 0$ satisfait les conditions nécessaires du premier et du second ordre

$$\begin{cases} f'(x) = -4x^3 \\ \text{et} \\ f''(x) = -12x^2 \end{cases} \implies \begin{cases} f'(x^0) = f'(0) = 0 \\ \text{et} \\ f''(x^0) = f''(0) = 0 \geq 0 \end{cases}$$

le point $x^0 = 0$ ne constitue pas un minimum de la fonction f . Il est au contraire un maximum global.

$$\forall x \in \mathbb{R}, \quad f(x) = -x^4 \leq 0 = f(0).$$

Le théorème qui suit établit une condition suffisante pour qu'un vecteur soit un minimum local, si f est deux fois continûment différentiable.

Théorème 2.7 (Condition suffisante du second ordre)

Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois continûment différentiable et $x^0 \in \mathbb{R}^n$, Si $\nabla f(x^0) = 0$ (x^0 est un point stationnaire de f) et $H(x)$ est définie positive ($H > 0$), alors x^0 est un minimum local de (p_1).

Démonstration. Le développement de Taylor de la fonction f à l'ordre 2, au voisinage de x^0 s'écrit :

$$f(x^0 + l) - f(x^0) = \frac{1}{2} l^T \nabla^2 f(x^0) l + \sigma(\|l\|^2).$$

Toute direction $l \in \mathbb{R}^n$, tendant vers 0, peut être écrite sous la formule

$$l = \theta \text{ d'où } \|d\| = 1 \text{ et } \theta > 0 \text{ (assez petit)}$$

On aura alors

$$f(x^0 + l) - f(x^0) = f(x^0 + \theta d) - f(x^0) = \theta^2 \left[\frac{1}{2} d^T \nabla^2 f(x^0) d + \sigma(\alpha^2) \alpha^2 \right]$$

Comme

$$d^T \nabla^2 f(x^0) d > 0$$

et

$$\lim_{\theta \rightarrow 0} \frac{\sigma(\theta^2)}{\theta^2} = 0$$

Alors pour tout nombre positif $\theta > 0$ assez petit, on aura

$$f(x) = f(x^0 + \theta d) \geq f(x^0)$$

Ce qui prouve que x^0 est un minimum local de f .

Exemple 2.2 Soit $f(x, y) = mx^2 + y^2$ définie sur \mathbb{R}^n dans \mathbb{R} , $m \neq 0$ pour déterminer les extrêmes de f , il faut d'abord déterminé les points stationnaire de f et la matrice Hessienne va déterminer la nature du point stationnaire.

– Calcul de $\nabla f(x, y) = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right) = (2mx, 2y)$

$$(2mx, 2y) = (0, 0) \Rightarrow \begin{cases} 2mx = 0 \\ 2y = 0 \end{cases} \Rightarrow \begin{cases} x = 0 \\ y = 0 \end{cases}$$

$(x, y) = (0, 0)$ est une point stationnaire.

– Nature du point stationnaire $(0, 0)$

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

– Si $m > 0$, alors $H(x^*, y^*)$ est définie positive ($H > 0$) donc par la condition suffisante $(x^*, y^*) = (0, 0)$ est minimum local strict.

– Si $m < 0$, alors les valeurs propre n'ont pas le même signe donc par la condition nécessaire, $(x^*, y^*) = (0, 0)$ ne peut pas être un minimum local.

2.5 Les Méthodes d'optimalité

2.5.1 Introduction

Parfois les conditions nécessaires et suffisantes d'optimalité permettent de déterminer les solutions des problèmes d'optimisation. Mais dans la plus part des cas on est obligé de faire appelle aux méthodes d'optimalité ou bien aux méthodes numériques.

Dans cette partie, on s'intéresse à la description plus spécifique des algorithmes itératifs (ou méthodes itératives) qui permettent la résolution des problèmes d'optimisation non linéaire.

2.5.2 La méthode de descente basée sur le gradient

Les méthodes basées sur le gradient de la fonction objectif sont des procédures parmi les plus fondamentales pour minimiser une fonction différentiable de \mathbb{R}^n dans \mathbb{R} . Comme la plupart des autres méthodes développées pour ce problème, elles reposent sur la propriété dite de descente itérative. Rappelons qu'un algorithme itératif part d'un vecteur $x^0 \in \mathbb{R}^n$ et génère une suite de vecteur x^1, x^2, \dots de \mathbb{R}^n , la propriété de descente itérative impliquant que coût des vecteurs ainsi générés décroisse à chaque itération :

$$f(x^{k+1}) < f(x^k), \quad \forall k \in \mathbb{N}$$

Et les vecteurs $x^1, x^2, \dots \in \mathbb{R}^n$ sont générés de la manière suivante :

$$x^{k+1} = x^k - \alpha^k d^k$$

Ainsi, pour assurer la propriété de descente itérative, la direction d^k choisie dans l'équation ci-dessus doit être une direction de descente. Les algorithmes de ce type sont appelés méthodes du gradient.

Certains auteurs réservent cette appellation au cas particulier où

$$d^k = -\nabla f(x^k).$$

L'appellation "Méthode du gradient" sera utilisée uniquement au cas où d est choisi ainsi et α^k est déterminé suivant la politique dite de minimisation qui revient à calculer de façons a minimiser le problème unidimensionnelle.

$$q(\alpha) = x^k - \nabla f(x^k)$$

tels que q est la demi-droite définie par le point x^k et la direction $-\nabla f(x^k)$.

$$x^{k+1} = x^k + \alpha^k d^k, \quad \alpha^k > 0$$

Le problème de recherche de cette direction consiste à trouver la direction d qui minimise $-\nabla f(x).d$ sous la contrainte $\|d\| = 1$. La proposition ci-dessous stipule que la direction $d = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$ est la solution optimale de ce problème.

Proposition 2.1 Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuellement différentiable et $x \in \mathbb{R}^n$. Supposons que $\nabla f(x) \neq 0$. Alors le problème qui consiste à minimiser $f(x, d)$ sous la contrainte $\|d\| = 1$ a pour solution optimale

$$d^* = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$$

La direction de descente choisie sera donc, à chaque itération $d^* = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$. Les points sont ainsi successivement générés par la méthode du gradient de la manière suivante : Remarquons que la méthode s'arrête lorsque $\nabla f(x^k) = 0$, car dans ce cas $x^{k+1} = x^k$.

Algorithme du Gradient

1. Initialisation $k = 0$:

On se donne une fonction f différentiable, un point initial x^0 et $\epsilon > 0$.

2. Itération $k = 1, 2, \dots$

Poser $d^* = \frac{-\nabla f(x)}{\|\nabla f(x)\|}$

Calculer : $\alpha_k = \arg \min_{\alpha > 0} f(x^k + \alpha d^k)$

$x^{k+1} = x^k + \alpha d^k$

3. Critère d'arrêt :

Si $\|x^{k+1} - x^k\| < \epsilon$ Stop

Si non on pose $k = k + 1$ et aller à 2.

Convergence de la méthode

Théorème 2.8 (Convergence de la méthode du gradient) [7] Soit x^k une séquence générée par la méthode du gradient. Alors tout point limite de $\{x^k\}$ est un point stationnaire.

Il peut arriver que la méthode du gradient converge de manière finie, mais ce n'est en général pas le cas. Il est donc nécessaire d'utiliser un critère permettant d'arrêter l'exécution lorsque $\{x^k\}$ est suffisamment proche d'un point stationnaire.

2.5.3 Méthode de Newton

L'idée de la méthode de Newton est de minimiser à chaque itération l'approximation de f au point courant x^k donnée par le développement de Taylor d'ordre 2 :

$$q^k(x) = f(x^k) + \nabla f(x^k)(x - x^k) + \frac{1}{2}(x - x^k)\nabla^2 f(x^k)(x - x^k).$$

Une condition nécessaire pour que le minimum de $q^k(x)$ soit atteint est

$$\nabla q^k(x) = 0$$

Soit

$$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0$$

Le vecteur généré à l'itération $k + 1$ est le vecteur minimisant $q^k(x)$, c'est-à-dire le vecteur satisfaisant l'équation précédente, soit

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k)$$

La méthode nécessitant l'évaluation de la matrice Hessienne de f , elle ne peut être utilisée que si f est deux fois continuellement différentiable.

Algorithme de Newton

1. *Initialisation* $k = 0$:

On se donne une fonction f différentiable, un point initial x^0 et $\varepsilon > 0$.

2. *Itération* $k = 1, 2, \dots$

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k)$$

3. *Critère d'arrêt* :

Si $\|x^{k+1} - x^k\| < \varepsilon$ Stop

Si non on pose $k = k + 1$ et aller à 2.

Convergence de la méthode

La méthode de Newton décrite ci-dessus présente plusieurs inconvénients :

- a) L'inverse de la matrice hessienne $\nabla^2 f(x^k)^{-1}$ peut ne pas exister, auquel cas la méthode échoue. Cela intervient typiquement lorsque la méthode atteint une région, où f est linéaire (ses secondes dérivées partielles valent zéro).
- b) La méthode de Newton n'est pas une méthode de descente : il est possible que $f(x^{k+1})$ soit supérieure à $f(x^k)$.
- c) Elle est attirée aussi bien par les minima que par les maxima locaux. En effet, la méthode, à chaque itération, recherche uniquement un point tel que le gradient de l'approximation quadratique soit égal au vecteur nul, que ce point soit un maximum, un minimum ou un point stationnaire.

La méthode ne converge donc pas en général, notamment si elle est démarrée loin d'un minimum local, pour les première et troisième raisons. Cependant, elle converge sous certaines restrictions : si elle est exécutée à partir d'un point suffisamment proche d'un minimum local et que $\nabla^2 f(x^k)$ n'est pas singulière, alors la méthode de Newton convergera vers ce minimum (mais pas de manière finie, de sorte qu'une condition d'arrêt soit requise de façon analogue à la méthode du gradient).

2.5.4 Étude comparative des méthodes d'optimisation sans contraintes

Dans cette partie, nous présentons une étude comparative entre la méthode du gradient et la méthode de Newton. Les problèmes résolus sont :

$$P_1 \begin{cases} 3x_1^2 + 3x_2^2 - 6x_2 - 3x_1x_2 \\ x^0 = (3, 4) \end{cases}, \text{ avec } |\nabla f(x^k)| < 0.01.$$

et

$$P_2 \begin{cases} 4x_1^2 + 4x_2^2 - 12x_2 - 4x_1x_2 \\ x^0 = (-20, 15) \end{cases}, \text{ avec } |\nabla f(x^k)| < 0.01.$$

Pour le premier problème P_1

Pour résoudre le problème P_1 on prend comme point de départ $x^0 = (3, 4)$ et la condition d'arrêt $|\nabla f(x^k)| < 0.01$.

A) La méthode du gradient :

Le résultat est donné dans le tableau suivant :

| k | x^k | $f(x^k)$ | $\nabla f(x^k)$ | d^k | α^k | x^{k+1} |
|-----|--------------|----------|-----------------|---------------|------------|--------------|
| 1 | (3.00, 4.00) | 15.00 | (6.0, 9.0) | (6.00, 9.00) | 0.31 | (1.14, 1.12) |
| 2 | (1.14, 1.21) | -3.11 | (3.2, -2.16) | (3.21, -2.14) | 0.31 | (0.78, 1.46) |
| 3 | (0.78, 1.46) | -3.96 | (0.3, -5.55) | (0.28, 0.42) | 0.11 | (0.69, 1.33) |
| 4 | (0.69, 1.33) | -4.00 | (1.77, -5.85) | (0.15, -0.10) | 0.31 | (0.67, 1.34) |
| 5 | (0.67, 1.34) | -4.00 | (0.0, 0.0) | (0.01, 0.02) | 0.11 | (0.67, 1.33) |
| 6 | (0.67, 1.33) | -4.00 | (0.0, 0.0) | (0.01, -0.00) | 0.31 | - |

TAB. 2.1 – Résultats de la méthode du gradient de P_1

B) La méthode de Newton

Et ils ont résolu les mêmes problèmes avec la méthode de Newton.

L'exécution de la méthode de newton donne les résultats suivants :

| k | x^k | $\nabla f(x^k)$ | $\nabla^2 f(x^k)$ | $\nabla^2 f(x^k)^{-1}$ | d^k | x^{k+1} |
|-----|--------------|-----------------|--------------------|----------------------------|------------|--------------|
| 1 | (3.00, 4.00) | (6.0, 9.0) | [(6, -3), (-3, 6)] | [(0.2, 0.11), (0.11, 0.2)] | (6, 9) | (0.67, 1.33) |
| 2 | (0.67, 1.33) | (0.0, 0.0) | - | - | (0.01, -0) | - |

TAB. 2.2 – Résultats de la méthode de Newton de P_1

Pour le deuxième problème P_2

A) La méthode du gradient :

Pour résoudre le problème P_2 on prend comme point de départ $(-20, 15)$ et la condition d'arrêt $|\nabla f(x^k)| < 0.01$.

Le résultat est donné dans le tableau suivant :

| k | x^k | $f(x^k)$ | $\nabla f(x^k)$ | d^k | α^k | x^{k+1} |
|----------|------------------|----------|-------------------|-------------------|------------|------------------|
| 1 | $(-20.0, 15.0)$ | 3520 | $(-220.0, 188.0)$ | $(220.0, -188.0)$ | 0.08 | $(-1.59, -0.73)$ |
| 2 | $(-1.59, -0.73)$ | 16.39 | $(-9.81, -11.48)$ | $(9.81, 11.48)$ | 0.25 | $(0.83, 2.1)$ |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| 5 | $(1.0, 2.0)$ | -12 | $(-0.01, 0.01)$ | $(0.01, -0.01)$ | 0.08 | $(1.0, 2.0)$ |
| 6 | $(1.0, 2.0)$ | -12 | $(0.0, 0.0)$ | — | — | — |

TAB. 2.3 – Résultats de la méthode du gradient de P_2

B) La méthode de Newton

Et ils ont résolu les mêmes problèmes avec la méthode de Newton.

L'exécution de la méthode de newton donne les résultats suivants :

| k | x^k | $\nabla f(x^k)$ | $\nabla^2 f(x^k)$ | $\nabla^2 f(x^k)^{-1}$ | d^k | x^{k+1} |
|-----|--------------|-------------------|----------------------|------------------------------|-------------|-----------|
| 1 | $(-20, 15)$ | $(-220.0, 188.0)$ | $[(8, -4), (-4, 8)]$ | $[(0.2, 0.11), (0.11, 0.2)]$ | $(21, -13)$ | $(1, 2)$ |
| 2 | $(1.0, 2.0)$ | $(0.0, 0.0)$ | $[(8, -4), (-4, 8)]$ | — | — | — |

TAB. 2.4 – Résultats de la méthode de Newton de P_2

Commentaires :

- Un premier constat est que la méthode du gradient n'a pas eu besoin de beaucoup d'itérations pour trouver la solution optimale de ce problème en partant de ce point initial.
- La méthode de newton converge de manière finie en une seule itération.
- Pour les deux problèmes : La méthode du gradient à eu besoin de 6 itérations pour trouver une approximation du minimum, alors que la méthode de Newton n'en a nécessita que 2.
- La méthode de Newton, elle se rapproche plus lentement du minimum durant les premières itérations que la méthode du gradient, cependant elle converge ensuite beaucoup plus vite.

Chapitre 3

Optimisation non linéaire avec contraintes

3.1 Introduction

Un problème de minimisation d'une fonction non-linéaire f sous contraintes $x \in S \subset \mathbb{R}^n$ ou S est défini par une collection de m inégalités et r égalités, se présente sous la forme suivante :

$$(P_2) \begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0, \quad 1 \leq i \leq m \\ h_j(x) = 0, \quad 1 \leq j \leq r \end{cases}$$

Les fonctions g_i , $i = 1, \dots, m$ et h_j , $j = 1, \dots, r$ sont des fonctions réelles de classe C^1

Remarque 3.1 *Si les fonctions f , g_i , h_j sont linéaires, alors le problème (P_2) est appelé problème de programmation linéaire.*

Remarque 3.2 *Dans un problème de programmation linéaire, l'optimum est toujours atteint en un point extrême du polyèdre S . En programmation non linéaire (ou optimisation non linéaire), même si le domaine admissible S est un polyèdre, le minimum peut être atteint en un point quelconque de S (en un point extrême, point frontière ou encore point intérieur).*

Exemple 3.1 . Soit le problème suivant :

$$\begin{cases} f(x) = (x_1 - 4)^2 + (x_2 - 6)^2 \rightarrow \min \\ x_1 + x_2 \geq 1 \\ 2x_1 + 3x_2 \leq 12 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

L'ensemble $S = \{x_1 + x_2 \geq 1, 2x_1 + 3x_2 \leq 12, x_1 \geq 0, x_2 \geq 0\}$ est un polytope et il est représenté sur la figure suivante :

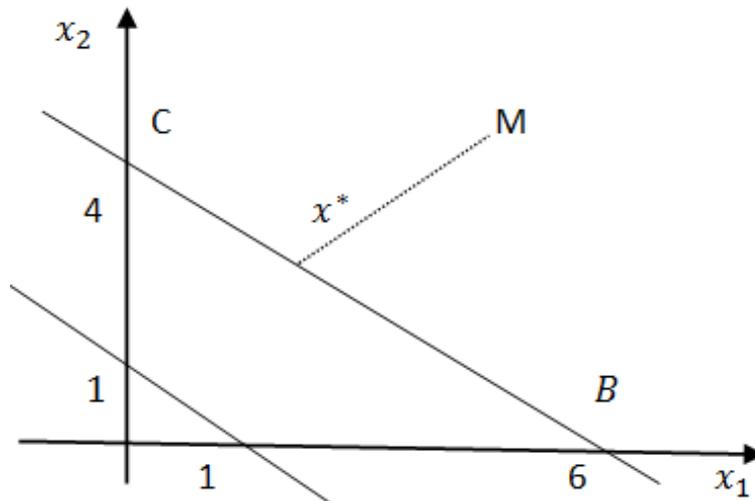


FIG. 3.1 – un problème admet une infinité de solutions

L'équation $f(x) = \alpha$ constante > 0 représente un cercle de centre $M(4,6)$ et de rayon $R = \sqrt{\alpha}$. En faisant diminuer (ou augmenter) le nombre α , la valeur de $f(x)$ diminue (ou augmente). Il s'ensuit alors que le point x^* , qui est l'intersection des droites perpendiculaires BC et Mx^* , réalise le minimum de la fonction f sur l'ensemble S . De plus, le point x^* se trouve sur l'arête du polyèdre S .

Ici, on trouve

$$x^* = \left(\frac{24}{13}, \frac{36}{13} \right), \quad f(x^*) = \frac{196}{13}.$$

Exemple 3.2 Considérons la fonction :

$$f(x) = (x_1 - 4)^2 + (x_2 - 1)^2$$

et cherchons son minimum sur le même polyèdre précédent S . Cette fois-ci, le minimum sera atteint au point intérieur $x^* = (4, 1)$, $f(x^*) = 0$.

Exemple 3.3 Soit la fonction

$$f(x) = (x_1 - 8)^2 + x_2^2$$

et cherchons son minimum sur le même polyèdre précédent S . Cette fois-ci, le minimum sera atteint au point extrême

$$x^* = (6, 0), \quad f(x^*) = 4.$$

3.2 Théorèmes généraux d'existence

Considérons le problème d'optimisation suivant :

$$(P_{2.1}) \begin{cases} \min f(x) \\ x \in S \end{cases}$$

Nous allons donner deux résultats très généraux d'existence d'une solution optimale au problème $(P_{2.1})$.

Théorème 3.1 (Existence) Soit $S \subset \mathbb{R}^n$ (S non vide) et $f : S \rightarrow \mathbb{R}$ semi-continue inférieurement.

- Si S est un ensemble non vide fermé et borné de \mathbb{R}^n , alors il existe une solution optimale $x^0 \in S$ qui vérifie donc

$$f(x^0) \leq f(x), \quad \forall x \in S.$$

- Si S est un ensemble non vide fermé de \mathbb{R}^n et si f est coercitive, alors il existe une solution $x^0 \in S$ qui vérifie

$$f(x^0) \leq f(x), \quad \forall x \in S.$$

Démonstration.

- Si $S \neq \emptyset$ est un sous-ensemble fermé borné de \mathbb{R}^n , comme f est continue, elle atteint ses bornes sur S , d'où l'existence de x^0 .
- Si f est croissante à l'infini, alors il existe $R > 0$ tel que si $\|x\| > R$, alors $f(x) > f(x^0)$, donc

$$\min_S f = \min_{S \cap B_R} f$$

où B_R désigne la boule fermée de centre 0 et de rayon R . L'ensemble $S \cap B_R$ est compact, car c'est l'intersection d'un fermé et d'un compact. Donc, par ce qui précède, il existe $x^0 \in S$ tel que

$$f(x^0) = \min_{S \cap B_R} f = \min_{B_R} f$$

Théorème 3.2 (Unicité) Soit $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$. On suppose que f est strictement convexe et que S est convexe. Alors il existe au plus un élément x^0 de S tel que

$$f(x^0) = \min_S f$$

Démonstration. Supposons que x^0 et x^* soient deux solutions du problème $(P_{2.1})$, avec $x^0 \neq x^*$. Alors

$$\min_S f \leq f\left(\frac{1}{2}x^0 + \frac{1}{2}x^*\right) < \frac{1}{2}f(x^0) + \frac{1}{2}f(x^*) = \min_S f$$

On aboutit donc à une contradiction.

Théorème 3.3 (Existence et Unicité) Soit $f : S \subset \mathbb{R}^n$ une fonction continue. On suppose que f est strictement convexe et que S est non vide et convexe fermé sur \mathbb{R}^n

Si S est borné ou f est croissante à l'infini, c'est-à-dire que $f(x) \rightarrow \infty$ quand $\|x\| \rightarrow \infty$ alors il existe un unique élément x^0 de S solution du problème de minimisation $(P_{2.1})$, tel que $f(x) = \min_S f$

3.3 Conditions d'optimalité

3.3.1 Cas de contraintes quelconques

Soit f une fonction non linéaire définie de \mathbb{R}^n dans \mathbb{R} et de classe C^1 . Soit S un sous-ensemble non vide de \mathbb{R}^n et considérons le problème de minimisation suivant :

$$(P_{2.2}) \begin{cases} \min f(x) \\ x \in S \end{cases}$$

Définition 3.1

- Un vecteur x de \mathbb{R}^n est dit solution réalisable ou plan du problème $(P_{2.2})$, si x appartient à l'ensemble des contraintes S .
- Un plan x^* est appelé solution optimale du problème $(P_{2.2})$, si

$$f(x^*) = \min_S f$$

On dit aussi que x^* constitue un point de minimum relatif global.

- Un vecteur $x^0 \in S$ est appelé point de minimum relatif local de la fonction f , si pour un certain $\varepsilon > 0$, il satisfait les relations :

$$f(x^0) = \min_{x \in B(x^0, \varepsilon)} f(x)$$

$$B(x^0, \varepsilon) = \{x \in S, \|x - x^0\| \leq \varepsilon\}.$$

Définition 3.2 (Direction admissible) Soient un ensemble $S \subset \mathbb{R}^n$ et $x^0 \in S$ Une direction admissible de S en x^0 est un vecteur d tel que

$$d \neq 0, x^0 + \alpha d \in S, \quad \forall \alpha \in [0, \bar{\alpha}] \quad \text{pour un certain } \bar{\alpha} \geq 0.$$

Théorème 3.4 (Condition nécessaire d'optimalité) Soit f une fonction non linéaire, définie de \mathbb{R}^n dans \mathbb{R} et de classe C^1 . Si x^0 est un minimum local (ou global) du problème $(P_{2.2})$, alors pour toute direction admissible d en x^0 on a

$$\nabla^T f(x^0) d \geq 0$$

3.3.2 Cas de contraintes linéaires de type égalités

A)- Condition nécessaire d'optimalité du premier ordre

Considérons dans \mathbb{R}^n le problème de programmation non linéaire :

$$(P_{2.3}) \begin{cases} f(x) \rightarrow \min \\ g(x) = Ax - b = 0 \end{cases}$$

$b \in \mathbb{R}^m$, A une matrice d'ordre $m \times n$, formée des vecteurs colonnes et des vecteurs lignes suivants :

$$A = (a_1, a_2, \dots, a_n) = \begin{pmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_m^T \end{pmatrix}, \quad g_i(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix}$$

où

$$g_i(x) = A_i^T x - b_i, \quad 1 \leq i \leq m$$

Pour que l'ensemble des solutions réalisables

$$S = \{x \in \mathbb{R}^n, g(x) = 0\} = \{x \in \mathbb{R}^n, A_i^T x + b_i = 0, 1 \leq i \leq m\}$$

ne soit pas vide ou ne soit pas réduit à un point isolé, on considérera que

$$\text{rang} A = m < n.$$

Proposition 3.1 Soit x une solution réalisable du problème $(P_{2.3})$. Un vecteur $d \in \mathbb{R}^n$ est alors une direction admissible en x si et seulement si

$$Ad = 0$$

De plus, pour un tel vecteur, on a

$$x(\alpha) = x + \alpha d \in S; \quad \forall \alpha \in \mathbb{R}$$

En effet,

$$g(x(\alpha)) = Ax(\alpha) - b = Ax + \alpha Ad - b = 0, \quad \forall (\alpha \neq 0) \in \mathbb{R} \text{ si et seulement si } Ad = 0.$$

Théorème 3.5 (Multiplicateurs de Lagrange) Si x^0 est un point de minimum relatif local (ou global) pour le problème $(P_{2.3})$, alors il existe un vecteur unique $\lambda^0 \in \mathbb{R}^m$ appelé vecteur des multiplicateurs de Lagrange, tel que :

$$\nabla L(x^0, \lambda^0) = 0 \Leftrightarrow \begin{cases} \nabla_x L(x^0, \lambda^0) = 0 \\ \text{et} \\ \nabla_\lambda L(x^0, \lambda^0) = 0 \end{cases}$$

Donc si x^0 est un point de minimum relatif du problème $(P_{2.3})$, alors le couple $(x^0; \lambda^0)$ est un point stationnaire de la fonction de Lagrange .

Exemple 3.4 Résoudre le problème de minimisation suivant :

$$\begin{cases} f(x) = 2x_1^2 + x_2^2 + 5x_3^2 \rightarrow \min \\ x_1 - x_2 + x_3 = 1 \\ 2x_1 + x_2 + 5x_3 = 0 \end{cases}$$

On définit la fonction de Lagrange :

$$L(x; \lambda) = 2x_1^2 + x_2^2 + 5x_3^2 + \lambda_1(x_1 - x_2 + x_3 - 1) + \lambda_2(2x_1 + x_2 + 5x_3)$$

La condition nécessaire d'optimalité est donnée comme suit :

$$\begin{cases} \nabla_{x_1} L(x; \lambda) = 0 \\ \nabla_{x_2} L(x; \lambda) = 0 \\ \nabla_{x_3} L(x; \lambda) = 0 \\ \nabla_{\lambda_1} L(x; \lambda) = 0 \\ \nabla_{\lambda_2} L(x; \lambda) = 0 \end{cases} \implies \begin{cases} 4x_1 + \lambda_1 + 2\lambda_2 = 0 & C_1 \\ 2x_2 - \lambda_1 + \lambda_1 = 0 & C_2 \\ 10x_3 + \lambda_1 + 5\lambda_2 = 0 & C_3 \\ x_1 - x_2 + x_3 = 1 & C_4 \\ 2x_1 + x_2 + 5x_3 = 0 & C_5 \end{cases}$$

En faisant la somme des équations C_1 , C_2 et C_3 et en tenant compte de C_5 , on obtient

$$\lambda_1 + 8\lambda_2 = 0 \implies \lambda_1 = -8\lambda_2$$

On aura alors

$$x_1 = \frac{3}{2}\lambda_2, \quad x_2 = \frac{-9}{2}\lambda_2, \quad x_3 = \frac{3}{10}\lambda_2$$

En utilisant C_4 , on déduit la valeur de $\lambda_2 = \frac{10}{63}$. D'où

$$\lambda_1 = \frac{-80}{63}, \quad x_1 = \frac{5}{21}, \quad x_2 = \frac{-5}{7}, \quad x_3 = \frac{1}{21}, \quad f(x) = \frac{40}{63}$$

On trouve donc un seul vecteur $x \in \mathbb{R}^n$ qui réalise le minimum de f sur l'ensemble des contraintes. En effet, la condition nécessaire de Lagrange est aussi suffisante puisque la fonction considérée ici est strictement convexe. Par conséquent

$$x^* = \frac{1}{21}(5, -15, 1), \quad f(x^*) = \frac{40}{63}$$

B)- Condition nécessaire d'optimalité du second ordre

Considérons le problème $(P_{2.3})$ et supposons de plus que la fonction f est de classe C^2 . On a alors la condition nécessaire suivante de second ordre.

Théorème 3.6 Si x^0 est un point de minimum relatif du problème $(P_{2.3})$ et λ^0 le vecteur des multiplicateurs de Lagrange correspondant, alors la forme quadratique

$$\nabla_x^2(x^0, \lambda^0)$$

est semi-définie positive sur l'ensemble des points de la variété

$$Ay = 0$$

C)- Condition suffisante d'optimalité du second ordre

Théorème 3.7 Soit f une fonction de classe C^2 et (x^0, λ^0) un couple de vecteurs vérifiant la condition nécessaire d'optimalité de premier ordre pour le problème $(P_{2.3})$, i.e,

$$\nabla_x L(x^0, \lambda^0) = 0$$

Pour que x^0 soit un point de minimum relatif local du problème $(P_{2.3})$, il est alors suffisant que la forme quadratique

$$y^T \nabla_x^2 L(x^0, \lambda^0) y$$

soit définie positive sur l'ensemble des points y de la variété $Ay = 0$.

D)- Cas d'une fonction convexe

Considérons le problème :

$$(P_{2.4}) \begin{cases} f(x) \rightarrow \min \\ x \in S = \{x \in R^n, g(x) = Ax - b = 0\} \end{cases}$$

où f est une fonction convexe de classe C^1 et A une matrice d'ordre $m \times n$, avec $\text{rang} A = m < n$.

Soit $x^0 \in S$ et λ^0 un m -vecteur tels que le couple (x^0, λ^0) vérifie la condition nécessaire d'optimalité du premier ordre, i.e :

$$\nabla_x L(x^0, \lambda^0) = \nabla f(x^0) + A^T \lambda^0 = 0 \quad (3.1)$$

Cette dernière relation est aussi suffisante pour l'optimalité du vecteur x^0 dans le problème $(P_{2.4})$.

Puisque S est aussi convexe, on peut écrire

$$f(x) - f(x^0) \geq (\nabla f(x^0))^T (x - x^0), \quad \forall x \in S$$

De légalité (3.1), on déduit que

$$f(x) - f(x^0) \geq -(\lambda^0)^T A(x - x^0), \quad \forall x \in S$$

Comme $A(x - x^0) = 0$, on déduit $f(x) \geq f(x^0)$.

Ce qui prouve que x^0 réalise un minimum pour la fonction objectif du problème $(P_{2.4})$.

3.3.3 Cas de contraintes linéaires de type inégalités

Avant d'aborder le problème, voyons un lemme, appelé lemme de Farkas, qui joue un rôle très important en optimisation mathématique.

Lemme 3.1 (Lemme de Farkas) Soient $(m + 1)$ vecteurs de \mathbb{R}^n , C , A_i , $1 \leq i \leq m$, avec $m < n$.

Si pour chaque vecteur x vérifiant $A_i^T x \leq 0$, $1 \leq i \leq m$ on a $C^T x \leq 0$ alors il existe des coefficients $\lambda_i \geq 0$, $1 \leq i \leq m$ tels que

$$C = \sum_{i=1}^m \lambda_i A_i$$

A)- Condition nécessaire d'optimalité du premier ordre

Le problème de minimisation d'une fonction f avec des contraintes linéaires de type inégalités se résume ainsi :

$$(P_{2.5}) \begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0, \quad i = \{1, \dots, m\} \end{cases}$$

Tout vecteur x vérifiant la contrainte du problème $(P_{2.5})$ est appelé solution admissible du problème $(P_{2.5})$.

La contrainte $g_i(x) \leq 0$ sera dite active au point x , si $g_i(x) = 0$ et passive si $g_i(x) < 0$

On notera l'ensemble des indices actifs au point x par

$$I_a = I_a(x) = \{i \in I : g_i(x) = 0\}$$

Théorème 3.8 (Théorème de Karush-Kuhn-Tucker)

Soit x^0 un point de minimum du problème $(P_{2.5})$. Il existe alors un m vecteurs $\lambda^0 \geq 0$ tel que :

i) Pour la fonction de Lagrange $L(x, \lambda) = f(x) + \lambda^T g(x)$ la condition de stationnarité est satisfaite :

$$\nabla_x L(x^0, \lambda^0) = 0$$

ii) La condition de complémentarité est remplie :

$$\lambda_i^0 g_i(x_0) = 0, \quad i \in I$$

B)- Condition nécessaire d'optimalité du second ordre

Théorème 3.9 Soit f une fonction de classe C^2 . Si x^0 un point de minimum du problème $(P_{2.5})$, alors la forme quadratique :

$$y^T \nabla^2 f(x^0) y$$

est définie non négative sur l'ensemble de la variété :

$$A_i^T y = 0, \quad i \in I_a(x^0) \tag{3.2}$$

où $I_a(x^0)$ est l'ensemble des indices actifs en x^0 .

C)- Condition suffisante d'optimalité du second ordre

Théorème 3.10 Soit x^0 un point admissible du problème $(P_{2.5})$ vérifiant les conditions nécessaires de Karush-Kuhn-Tucker. Si la matrice $\nabla^2 f(x^0)$ est définie positive sur l'ensemble (3.2), alors x^0 est un point de minimum du problème $(P_{2.5})$.

D)- Cas de fonction convexe

Considérons le problème suivant :

$$(P_{2.6}) \begin{cases} f(x) \rightarrow \min \\ x \in S = \{x \in \mathbb{R}^n, g(x) = Ax - b \leq 0\} \end{cases}$$

où f est une fonction convexe de classe C^1 . Soit (x^0, λ^0) un couple des vecteurs vérifiant le théorème de Karush-Kuhn-Tucker :

$$\begin{aligned} \nabla_x L(x^0, \lambda^0) = 0 &\iff \nabla f(x^0) = -\sum_{i=1}^m \lambda_i^0 A_i^T, \quad \lambda_i^0 \geq 0, \quad 1 \leq i \leq m. \\ \lambda_i^0 g_i(x^0) &= 0, \quad 1 \leq i \leq m. \end{aligned}$$

Alors le vecteur x^0 constitue un point minimum global du problème $(P_{2.6})$

3.3.4 Cas de contraintes égalités non linéaires

Considérons le problème suivant :

$$(P_{2.7}) \begin{cases} f(x) \rightarrow \min \\ x \in S = \{x \in \mathbb{R}^n, g(x) = 0\} \end{cases}$$

où f est une fonction non linéaire définie de \mathbb{R}^n dans \mathbb{R} et de classe C^1 :

$$g(x) = (g_1(x), g_2(x), \dots, g_m(x))$$

avec $g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $1 \leq i \leq m$ sont de classes C^1 .

Considérons la fonction de Lagrange :

$$L(x, \lambda) = f(x) + \lambda^T g(x).$$

A)- Conditions nécessaires de premier ordre

Théorème 3.11 (Règle des multiplicateurs de Lagrange) Soit x^0 une solution optimale du problème $(P_{2.7})$ et supposons que les vecteurs

$$\nabla g_1(x^0), \nabla g_2(x^0), \dots, \nabla g_m(x^0)$$

sont linéairement indépendants.

Alors il existe un vecteur multiplicateur de Lagrange (unique) λ^0 tel que la paire (x^0, λ^0) vérifié les relations appelées conditions de stationnarité de la fonction de Lagrange :

$$\nabla_x L(x^0, \lambda^0) = 0 \quad \text{et} \quad \nabla_\lambda L(x^0, \lambda^0) = 0$$

Définition 3.3 Un point x^* est appelé point stationnaire du problème $(P_{2.7})$, s'il existe un vecteur λ^* tel que la paire (x^*, λ^*) est un point stationnaire de la fonction de Lagrange :

$$\nabla_x L(x^*, \lambda^*) = 0 \quad \text{et} \quad \nabla_\lambda L(x^*, \lambda^*) = 0$$

La recherche des points stationnaires du problème $(P_{2.7})$ se ramène à la résolution du système précédant de $(m + n)$ équations avec $(m + n)$ inconnus (x, λ) .

Ainsi, d'après la règle des multiplicateurs de Lagrange, la résolution du Problème $(P_{2.7})$ (si la solution existe) se ramène à la recherche de la valeur minimale de la fonction $f(x)$ sur l'ensemble des points stationnaires.

Exemple 3.5 Considérons le problème de minimisation suivant :

$$\begin{cases} f(x_1, x_2) = x_2^3 \rightarrow \min \\ g(x_1, x_2) = x_2 - x_1^2 = 0 \end{cases}$$

La solution optimale étant $x^0 = (0; 0)$.

On a $\frac{\partial g}{\partial x_1}(0, 0) = -2x_1 |_{(0,0)} = 0$, $\frac{\partial g}{\partial x_2}(0, 0) = 1$. Le vecteur $\frac{\partial g}{\partial x}(0, 0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ est un vecteur linéairement indépendant.

On définit la fonction de Lagrange :

$$L(x, \lambda) = f(x) + \lambda g(x) = x_2^3 + \lambda (x_2 - x_1^2).$$

Les conditions de stationnarité sont ainsi données :

$$\begin{cases} \frac{\partial L}{\partial x_1}(x, \lambda) = -2\lambda x_1 = 0 \\ \frac{\partial L}{\partial x_2}(x, \lambda) = 3\lambda x_2^2 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda}(x, \lambda) = x_2 - x_1^2 = 0 \end{cases}$$

Ainsi à la solution optimale $x^0 = (0, 0)$ correspond un multiplicateur de Lagrange $\lambda^0 = 0$.

On a alors $L(x, \lambda^0) = L(x, 0) = x_2^3$, le point $x = (0, 0)$ est un point d'inflexion de la fonction de Lagrange $L(x, 0) = x_2^3$.

B)- Condition nécessaire du second ordre

Considérons le problème $(P_{2.7})$, où on suppose que $f(x)$ et $g(x)$ sont de classe C^2

Théorème 3.12 *Pour qu'un point stationnaire x^0 soit une solution optimale locale du problème $(P_{2.7})$, il est suffisant qu'avec le vecteur multiplicateur de Lagrange correspondant λ^0 , la forme quadratique suivante soit définie positive :*

$$l^T \nabla_x^2 L(x^0, \lambda^0) l > 0$$

sur l'hyperplan :

$$l^T \nabla g_i(x^0) = 0, \quad i = 1, \dots, m$$

avec $l \neq 0$.

3.3.5 Cas de contraintes inégalités non linéaires

Considérons le problème suivant :

$$(P_{2.8}) \begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0 \\ x \in S \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$, et $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ sont des fonctions de classe C^1 , $I = \{1, 2, \dots, m\}$ et $S \subset \mathbb{R}^n$ ouvert non vide.

Définition 3.4 *Tout vecteur $x \in S$ vérifiant les contraintes $g(x) \leq 0$ du problème $(P_{2.8})$ est appelé solution admissible*

- La contrainte $g_i(x) \leq 0$ est dite active au point x si $g_i(x) = 0$. Elle est dite passive si $g_i(x) < 0$.
- On notera l'ensemble des indices actifs au point x par

$$I_a(x) = \{i \in I, \quad g_i(x) = 0\}.$$

A)- Condition nécessaire d'optimalité de premier ordre

Théorème 3.13 *Supposons que f et g_i , $i \in I_a(x^0)$ sont différentiables au point x^0 et les fonctions g_i , $i \notin I_a(x^0)$ sont continues en x^0 . Si x^0 est une solution optimale locale du problème $(P_{2.8})$, alors il n'existe pas de vecteur $l \in \mathbb{R}^n$ vérifiant le système suivant :*

$$\begin{cases} l^T \nabla f(x^0) < 0 \\ l^T \nabla g_i(x^0) < 0, \quad i \in I_a(x^0) \end{cases}$$

Définition 3.5 *Un vecteur l est dit direction admissible au point \tilde{x} par rapport à la contrainte inégalité $g(x) \leq 0$ (direction intérieure) si*

$$l^T \nabla g_i(\tilde{x}) < 0, \quad \text{si } g_i(\tilde{x}) = 0$$

et l un vecteur quelconque si $g_i(\tilde{x}) < 0$.

Un vecteur l est appelé direction admissible au point x par rapport aux contraintes du problème $(P_{2.8})$, si il est une direction admissible par rapport à chacune des contraintes $g_i(x) \leq 0$, $i = 1, \dots, m$ au point \tilde{x} .

Proposition 3.2 Si x^0 est un optimum local du problème $(P_{2.8})$, alors au point x^0 , il n'existe pas de directions admissibles du problème $(P_{2.8})$.

Théorème 3.14 Soit x un point de minimum relatif local du problème $(P_{2.8})$ tel que les vecteurs

$$\nabla g_i(x^0) = 0, i \in I_a(x^0)$$

sont linéairement indépendants.

Il existe alors un unique vecteur multiplicateur de Lagrange vérifiant les relations suivantes :

$$\begin{cases} \lambda^0 \geq 0 \\ \nabla_x L(x^0, \lambda^0) = 0 \\ (\lambda^0)^T g(x^0) = 0 \end{cases}$$

Définition 3.6 On dira qu'un plan x^0 est réguliers si les vecteurs

$$\nabla g_i(x^0), i \in I_a(x^0)$$

sont linéairement indépendants.

B)- Condition nécessaire d'optimalités du second ordre

Définition 3.7

- Une contrainte active $g_i(x) \leq 0$ au point x^0 est dite forte si $\lambda_i^0 > 0$.
L'ensemble des indices des contraintes fortes au point x^0 sera noté :

$$I_a^+(x^0) = \{i, g_i(x) = 0, \lambda_i^0 > 0\}.$$

- Une contrainte active $g_i(x) \leq 0$ au point x^0 est dite faible si $\lambda_i^0 = 0$.
L'ensemble des indices des contraintes faible au point x^0 sera noté :

$$I_a^0(x^0) = \{i \in I, g_i(x) = 0, \lambda_i^0 = 0\}$$

Théorème 3.15 Soient Un plan x^0 optimal régulier du problème $(P_{2.8})$ et λ^0 le vecteur de Lagrange correspondant.

Alors, la forme quadratique est définie non négative :

$$l^T \nabla_x^2 L(x^0, \lambda^0) L \geq 0$$

sur l'ensemble des vecteurs l vérifiant le système suivant :

$$\begin{cases} l^T \nabla g_i(x^0) = 0, & i \in I_a^+(x^0). \\ l^T \nabla g_i(x^0) \leq 0, & i \in I_a^0(x^0). \end{cases}$$

C)- Condition suffisante d'optimalité locale

Supposons que dans le problème $(P_{2.8})$, les fonctions $f(x)$ et $g(x)$ sont de classe C^2 .

Définition 3.8 Un plan x^0 du problème est dit pseudo-stationnaire, s'il existe un m -vecteur λ^0 tel que :

$$\nabla_x L(x^0, \lambda^0) = 0, \quad \lambda_i^0 g_i(x^0) = 0, \quad \lambda_i^0 \geq 0, \quad 1 \leq i \leq m.$$

Théorème 3.16 Pour qu'un plan pseudo-stationnaire x^0 soit localement optimal dans le problème $(P_{2.8})$, il suffit que l'inégalité

$$l^T \nabla^2 L(x^0, \lambda^0) l > 0$$

soit vérifiée pour tout vecteur $l \neq 0$ vérifiant le système :

$$\begin{cases} l^T \nabla g_i(x^0) = 0, & i \in I_a^+(x^0) \\ l^T \nabla g_i(x^0) \leq 0, & i \in I_a^0(x^0) \end{cases}$$

3.4 Les Méthodes d'optimalité

3.4.1 Les méthodes de pénalité intérieure

Le principe de ces méthodes réside dans la transformation d'un problème contraint en une séquence de problèmes sans contraintes, en ajoutant au coût une pénalité en cas de violation de celles-ci. Un tel sous-problème est résolu à chaque itération d'une méthode de pénalité.

Les méthodes de pénalité (aussi appelées méthodes de barrière) s'appliquent aux problèmes dont l'ensemble admissible X est défini uniquement par une collection d'inégalités :

$$(P_{2.6}) \begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0, \quad i = 1, \dots, m \end{cases}$$

où f et les g_i sont des fonctions de \mathbb{R}^n dans \mathbb{R} et sont continues. En effet, ces méthodes utilisent des fonctions dites de barrière, définies uniquement à l'intérieur de X . L'intérieur de l'ensemble X défini par les g_i est le suivant :

$$X_I = \{x / g_i(x) < 0 ; \quad \forall i \in \{1, \dots, m\}\}$$

La fonction de barrière, notée $B(x)$, est ajoutée au coût $f(x)$, elle est continue sur X_I et sa valeur tend vers l'infini lorsque la frontière de X est approchée par l'intérieur, c'est-à-dire lorsque l'un des $g_i(x)$ approche zéro par les valeurs négatives. Une itération de la méthode consiste ensuite à minimiser la fonction $f(x) + tB(x)$ (où t est un paramètre réel strictement positif) à l'aide d'algorithmes de minimisation directe, une fonction $B(x)$ et un t convenablement choisis assurant que cette minimisation ne puisse nous mener à des points situés hors de X_I . La suite du processus consiste à réduire progressivement

t afin de diminuer la pénalité et autoriser les algorithmes de minimisation directe à se rapprocher peu à peu de la frontière de X . Les fonctions de barrière les plus répandues sont les suivantes : Logarithmique :

$$B(x) = -\sum_{i=1}^m \ln(-g_i(x))$$

L'inverse :

$$B(x) = -\sum_{i=1}^m \frac{1}{g_i(x)}$$

Il est important de noter que, si tous les g_i sont convexes, ces deux fonctions de barrière le sont également.

Remarquons qu'une nécessité, pour pouvoir appliquer une telle méthode, est de disposer d'un point initial situé à l'intérieur de X . La méthode de barrière est définie en introduisant la séquence de paramètres $\{t^k\}$; $k = 0, 1, \dots$ avec $0 < t^{k+1} < t^k$ et $t^k \rightarrow 0$ lorsque $k \rightarrow 1$. Une itération de celle-ci consiste à déterminer

$$x^k = \arg \min_{x \in \mathbb{R}^n} \{f(x) + t^k B(x)\}$$

Théorème 3.17 [9] *Tout point limite d'une séquence $\{x^k\}$ générée par une méthode de barrière est un minimum global du problème contraint original.*

Algorithme de la méthode

1. *Initialisation*

Fixer $\varepsilon > 0$ Choisir x^0 admissible, choisir $\eta > 1$ et ($t_1 > 0$) on prend ($t_1 = 10$)

2. *Itération $k = 1, 2, \dots$*

$$\Phi(x, r_x) = f(x) + (t_k \sum_{i=1}^m - \frac{1}{g_i(x)})$$

Calculer $\min \Phi(x, r_x)$

3. *Critère d'arrêt :*

Si $\|x^{k+1} - x^k\| < \varepsilon$ Stop

Si non, on pose $k = k + 1$ et aller à 2 , $t_{k+1} = \eta t_k$.

Convergence de la méthode

Si f et les g_i sont convexe, $f(x) + tB(x)$ l'est également, car elle est la somme de fonctions convexes. Tout point stationnaire d'une telle fonction étant un minimum global et nous avons la garantie que la méthode de barrière convergera vers le minimum global du problème contraint original, et ce même si l'algorithme de minimisation directe utilisé n'est

voué qu'à approcher des minima locaux.

Le comportement de la méthode dépend fortement du choix du paramètre t^0 initial et du facteur, disons β , satisfaisant $0 < \beta < 1$, utilisé pour décroître t^k à chaque itération par la formule $t^{k+1} = \beta t^k$. Il n'existe pas de règle universelle permettant d'obtenir un bon choix de t^0 et de β .

Cela dépend fortement du problème à résoudre ainsi que du point initial x^0 (celui-ci se trouve-t-il ou non à proximité de la frontière du domaine?). L'utilisateur d'une méthode de barrière sera souvent condamné à exécuter la méthode plusieurs fois avec différentes valeurs de ces paramètres jusqu'à obtenir une convergence satisfaisante.

Les faits suivants peuvent néanmoins être relevés : si t^k est trop petit, et à plus forte raison si x^k est proche des bords du domaine, le terme de barrière peut se révéler trop faible ne parvenant pas à empêcher une sortie de X (il faut donc prendre garde, durant l'exécution, et vérifier que l'on ne sorte pas de cet ensemble, et, si cela est tout de même le cas, recommencer avec un t^0 ou un β plus grand). Dans le cas contraire, si t^k est trop grand, l'algorithme de minimisation directe ne pourra s'approcher suffisamment des bords du domaine, conduisant à une convergence globale lente. Ces paramètres doivent donc être soigneusement choisis d'après le problème et le point initial.

Remarque 3.3 *En dehors de x^0 , les points obtenus à l'itération précédente sont, à chaque itération, utilisés comme points de départ de l'algorithme de minimisation directe (cela est valable pour les deux types de méthodes de pénalité).*

3.4.2 Les méthodes de pénalité extérieure

Considérons le problème suivant :

$$(PE) \begin{cases} \text{minimiser } f(x) \\ g_i(x) \leq 0, \quad i = 1, \dots, m \\ x \in \mathbb{R}^n \end{cases}$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}$

Considérons le problème sans contraintes (problème pénalisé) :

$$(PE_r) \begin{cases} \text{minimiser } \Theta_r(x) \\ x \in \mathbb{R}^n \end{cases}$$

où $\Theta_r(x)$ est obtenu en ajoutant à $f(x)$ le terme $rp(x)$

$$\Theta_r(x) = f(x) + rp(x)$$

avec

$$p(x) = \sum_{i=1}^m h(g_i(x)) = \sum_{i=1}^m [(g_i^+(x))]^2$$

où $\forall x; g_i^+(x) = \max\{0, g_i(x)\}$

P est appelée fonction de pénalisation extérieure

Le problème (PE_r) est alors remplacé par le problème d'optimisation sans contraintes :

$$\begin{aligned} \text{Minimiser } \Theta(x, r) &= f(x) + r \sum_{i=1}^m \max\{0, g_i(x)\}^2 \\ x &\in \mathbb{R}^n \end{aligned}$$

où $r > 0$ est appelé le coefficient de pénalité, notons $\bar{x}(r)$ un minimum de $\Theta(x, r)$.

Le choix d'une valeur appropriée du coefficient de pénalité r résulte d'un compromis : d'une part, r doit être suffisamment grand pour que le point $\bar{x}(r)$ obtenu soit proche de l'ensemble des solutions X (autrement dit, que p soit suffisamment faible); d'autre part, si r est choisi trop grand, la fonction Θ peut être mal conditionnée, d'où des difficultés numériques dans la recherche de l'optimum sans contraintes. Ceci explique pourquoi les méthodes de pénalités sont généralement mises en oeuvre sous forme itérative de la façon suivante :

On commence par choisir un coefficient de pénalité r_1 de valeur pas trop élevée (pour éviter les difficultés numériques) puis on résout le problème sans contraintes :

$$\min \Theta(x, r_1) = f(x) + r_1 p(x)$$

Soit $\bar{x}(r_1)$ le point obtenu. Si la quantité $P(\bar{x}(r_1))$ est suffisamment faible, $\bar{x}(r_1)$ est une bonne approximation de l'optimum et les calculs sont terminés. Dans le cas contraire, c'est que la pénalité associée à la violation des contraintes n'est pas assez élevée. On choisira donc un coefficient de pénalité $r_2 > r_1$ et on résoudra le nouveau problème sans contrainte :

$$\min \Theta(x, r_2) = f(x) + r_2 p(x)$$

On obtiendra un nouveau point $\bar{x}(r_2)$ et ainsi de suite.

Remarquons , qu'à chaque étape k du processus précédent, il est avantageux d'utiliser le point $\bar{x}(r_{k-1})$ obtenu à l'étape précédente comme point de départ de l'algorithme d'optimisation sans contrainte choisi.

Algorithme de la méthode

1. Initialisation

Fixer $\varepsilon > 0$ Choisir x^0 non admissible, choisir $\gamma > 1$ et ($t_1 > 0$) on prend ($r_1 \simeq 1$)

2. *Itération* $k = 1, 2, \dots$

$$\Theta(x, r_x) = f(x) + r_k \sum_{i=1}^m \max\{0, g_i(x)\}^2$$

Calculer $\min \Theta(x, r_x)$

3. *Critère d'arrêt* :

Si $\|x^{k+1} - x^k\| < \varepsilon$ Stop

Si non, on pose $k = k + 1$ et aller à 2, $t_{k+1} = \gamma t_k$.

Remarque 3.4

- On commence toujours par un point non admissible.
- Pour le choix du paramètre de pénalité on commence par des valeurs assez petites ($r_1 = 1$) ensuite on l'augmente au fur et à mesure des itérations.

Convergence de la méthode :

Le théorème ci-dessous étudie le comportement des solutions \bar{x}^t de (PE_r) lorsque r tend vers l'infini donne des conditions pour que les solutions des problèmes pénalisés convergent vers une solution du problème original.

Théorème 3.18 [10]

Soit $p : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction de pénalisation extérieur vérifiant :

a- $p(x) \geq 0, \forall x$.

b- $p(x) = 0 \iff x \in X = \{x / g_i(x) < 0, i = 1, \dots, m\}$.

c- p continue.

On suppose, d'autre part, que f est une fonction continue, que X est fermé, et que l'une des deux conditions suivantes est vérifiée :

1. $p(x) \geq 0, \forall x$.

2. X est borné et $p(x) \rightarrow +\infty$ quand $\|x\| \rightarrow +\infty$

Alors, lorsque le coefficient de pénalité r tend vers $+\infty$:

- La suite $\{\bar{x}(r)\}$ admet au moins un point d'accumulation, et tout point d'accumulation de cette suite est une solution optimale (globale) du problème (PE) .
- X est borné et $p(x) \rightarrow +\infty$ quand $\|x\| \rightarrow +\infty$.
- $p(\bar{x}(r)) \rightarrow 0$

3.4.3 Étude comparative des méthodes d'optimisation avec contraintes

Dans cette partie, nous présentons une étude comparative entre les méthodes de pénalité intérieure et pénalité extérieure. Ces méthodes ont été exécutées en utilisant le problème suivant :

$$\left\{ \begin{array}{l} \min f(x) = (x_1 - 3)^2 + (x_2 - 4)^2 \\ -x_1 - 2x_2 \leq -5 \\ x_1 - x_2 \leq 3 \\ x_1 + 2x_2 \leq 9 \\ 2x_1 - x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{array} \right.$$

1. La méthode de pénalité extérieure

En démarrant du point non admissible $(3, 4)$, avec un paramètre initial $r = 0.5$, et un facteur d'augmentation associé $\gamma = 2$ et la condition d'arrêt $\|x^{k+1} - x^k\| < 0.01$.

Le résultat est donné dans le tableau suivant :

| k | x^k | $f(x^k)$ | $p(x^k, r^k)$ | r^k | $g(x^k)$ | x^{k+1} |
|-----|--------------|----------|---------------|-------|----------|--------------|
| 1 | (3.00, 4.00) | 0 | 0.57 | 0.50 | 2.00 | (2.67, 3.33) |
| 2 | (2.67, 3.33) | 0.56 | 0.67 | 1.00 | 0.57 | (2.64, 3.27) |
| 3 | (2.64, 3.27) | 0.66 | 0.73 | 2.00 | 0.33 | (2.62, 3.24) |
| 4 | (2.62, 3.24) | 0.73 | 0.76 | 4.00 | 0.18 | (2.61, 3.22) |
| 5 | (2.61, 3.22) | 0.76 | 0.78 | 8.00 | 0.10 | (2.60, 3.21) |
| 6 | (2.60, 3.21) | 0.78 | 0.79 | 16.00 | 0.05 | (2.60, 3.20) |
| 7 | (2.60, 3.20) | 0.79 | 0.80 | 32.00 | 0.02 | — |

TAB. 3.1 – Résultats de la méthode de pénalité extérieure

Un premier constat est que la méthode de pénalité extérieure n'as pas besoin de beaucoup d'itérations pour trouver la solution optimale de ce problème.

Les résultats du tableau nous apprend que les suites $\{t^k\}$, $\{f(x^k)\}$, et $\{P(x^k, t^k)\}$ sont croissantes , par contre la suite $\left\{g(x^k) = \sum_{i=1}^m (\max\{0, g_i(x^k)\})^2\right\}$ est décroissante. d'où la convergence de la méthode vers une solution optimale globale.

2. La méthode de pénalité intérieure :

En démarrant du point admissible $(2, 2)$, avec un paramètre initial $t = 0.5$, et un facteur de réduction associé $\eta = 0.7$ et la condition d'arrêt $\|x^{k+1} - x^k\| < 0.01$.

Le résultat est donné dans le tableau suivant :

| k | x^k | $f(x^k)$ | $p(x^k, t^k)$ | t^k | x^{k+1} |
|----------|----------------|----------|---------------|----------|---------------|
| 1 | (2.00 , 2.00) | 5.00 | 26.67 | 10.00 | (1.57 , 2.92) |
| 2 | (1.57 , 2.92) | 3.23 | 20.17 | 7.00 | (1.72 , 2.90) |
| 3 | (1.72 , 2.90) | 2.85 | 15.62 | 4.90 | (1.86 , 2.89) |
| 4 | (1.86 , 2.89) | 2.52 | 12.43 | 3.43 | (1.98 , 2.90) |
| \vdots | \vdots | \vdots | \vdots | \vdots | \vdots |
| 16 | (2.47,3.26) | 0.82 | 5.10 | 0.05 | (2.47,3.26) |
| 17 | (2.47 , 3.26) | 0.82 | 5.07 | 0.03 | - |

TAB. 3.2 – Résultats de la méthode de pénalité intérieure

Il n'y a pas de technique générale pour le choix du paramètre t et du facteur de réduction, ce qui implique qu'il faille souvent redémarrer la méthode plusieurs fois. La méthode a eu besoin plus d'itération que la méthode de pénalité extérieure, pour trouver une approximation du minimum. Les suites $\{t^k\}$, $\{f(x^k)\}$, et $\{P(x^k, t^k)\}$ sont décroissantes, d'où la convergence de la méthode.

Chapitre 4

Introduction à l'Optimisation Globale

4.1 Introduction

L'optimisation globale est une branche des mathématiques, le domaine de sa recherche est considéré beaucoup plus riche ces dernières années grâce aux ordinateurs développés par la nouvelle technologie qui peuvent résoudre des problèmes d'optimisation de taille plus grande.

Les méthodes d'optimisation globale s'intéressent à la recherche d'un optimum (maximum ou minimum) global d'une fonction, et non à la recherche d'un optimum local que les méthodes classiques permettent de le trouver dans un voisinage réalisable, et c'est ça qui fait la différence entre les deux méthodes.

Dans ce chapitre, nous proposons les différentes méthodes d'optimisation globale déterministes, en donnant leurs principes ainsi que leurs algorithmes, on parlera aussi de la convergence de ces méthodes.

On va aborder les principales techniques fréquemment utilisées dans l'optimisation globale, à savoir :

- Séparation et Evaluation (Branch and Bound).
- Méthode de Piyavskii.

4.2 La méthode de Branch and Bound

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

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

a) Séparation :

La séparation consiste à diviser le problème en sous-problème. Ainsi, en résolvant tous les sous problèmes et en gardant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial.

Cela revient à construire un arbre permettant d'énumérer toutes les solutions. L'ensemble de noeuds de l'arbre qu'il reste encore à parcourir comme étant susceptibles de contenir une solution optimale, c'est-à-dire encore à diviser, est appelé ensemble des noeuds actifs.

b) L'évaluation :

L'évaluation permet de réduire l'espace de recherche en éliminant quelques sous ensembles qui ne contiennent pas la solution optimale. L'objectif est d'essayer d'évaluer l'intérêt de l'exploration d'un sous-ensemble de l'arborescence. La méthode utilise une élimination de branches dans l'arborescence de recherche de la manière suivante : la recherche d'une solution de coût minimal, consiste à mémoriser la solution de plus bas coût rencontré pendant l'exploration, et à comparer le coût de chaque nœud parcouru à celui de la meilleure solution. Si le coût du nœud considéré est supérieur au meilleur coût, on arrête l'exploration de la branche et toutes les solutions de cette branche seront nécessairement de coût plus élevé que la meilleure solution déjà trouvée.

4.2.1 Le principe de la méthode de Branch and Bound

Soit (P) le problème d'optimisation globale :

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

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

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

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

– UB : Upper bound.

– LB : Lower bound.

Une relaxation initiale R de l'ensemble réalisable S sera définie telle que : $S \subset R$.

R est convexe, il peut être un simplexe, un rectangle, un cône, ...etc.

A 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 résolution de la relaxation de f sur R_{k_i} et la relaxation de $\min f$ localement sur le sous ensemble réalisable $R_{k_i} \cap S$ 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.

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 niveau 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.

4.2.2 L'algorithme de base de Branch and Bound

On peut résumer la procédure précédente par les étapes suivantes :

Algorithme général

1. Construire l'ensemble R tel que : $S \subset R$.
2. Posons : $k = 1$, $I_k = R$, fixer $\varepsilon > 0$.
3. Construire les problèmes des bornes inférieure et supérieure de $\min f(x)$ sur R . Soient LB_k , UR_k les solutions obtenues respectivement.
4. Si : $UB_k - LB_k \leq \varepsilon$, on s'arrête et on aura la solution du problème (P) :

$$\begin{cases} x^k = x^* \\ UB_k = f(x^*) = \min f(x) \\ LB_k = f_a^k(x^*) = f_b^k(x^*) \end{cases}$$

5. 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

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

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

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

7. Posons :

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

8. Posons : $I_k = \{R_{k_1}, R_{k_2}\}$.

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

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

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

10. Posons : $k = k + 1$ et revenant à 4.

Notation

Notons R_k , LB_k , UB_k , et x^k : le sous ensemble actuel, La borne inférieure, La borne supérieure, et La solution optimale respectivement à l'itération k .

4.2.3 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 : $R_{k+1} \subset R_k$, $k \in N$. 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 4.1 *Si pour chaque suite infinie R_k , $R_{k+1} \subset R_k$, $k \in N$ des ensembles des partitions successives, les bornes inférieure et supérieure vérifient :*

$$\lim_{k \rightarrow \infty} (UB_k - LB_k) = \lim_{k \rightarrow \infty} (UB_k - LB(R_k)) \quad (4.1)$$

alors

$$UB = \lim_{k \rightarrow \infty} UB_k = \lim_{k \rightarrow \infty} f(X_k) = \lim_{k \rightarrow \infty} LB_k = LB \quad (4.2)$$

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

Démonstration. [8] A l'itération k le sous-ensemble R_k sera choisi à partir de la règle suivante :

$$LB_{k+1} = \min(LB_1, LB_2)$$

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

$$LB_k = LB(R_k)$$

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 : $f^* = \min(f(X), X \in S)$, la suite $\{LB_k\}$ des bornes inférieures est croissante monotone, majorée par f^* , donc la suite $LB = \lim_{k \rightarrow \infty} LB_k$ existe. D'autre part, la suite $\{UB_k\}$ des bornes supérieures est décroissante, minorée par f^* , donc sa limite : $UB = \lim_{k \rightarrow \infty} UB_k$ existe, et on a : $UB_k = f(X_k) \geq f^*$, et ça implique que :

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

Et du fait de (4.1), on peut déduire directement (4.2).

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

1. L'ensemble R tel que : $H \subset R$.
2. Les procédures qui donneront les bornes inférieures et supérieures sur les sous-ensembles engendrés par l'algorithme.
3. Les subdivisions successives de R en sous-ensembles. Bien entendu, ça va dépendre de la structure du problème (P_3). Pour cela nous allons Etudier chaque cas à part, en se basant sur les trois points précédent, et en montrant la convergence de l'algorithme à chaque fois .

4.2.4 Application numérique de Branch-and-Bound

Algorithme

1. Soit $x \in S = [a^j, b^j]$ avec $j = 0, 1, 2 \dots k, k \in N$.
2. Posons $j = k, I^k = [a^k, b^k]$ fixer $\varepsilon > 0, L > 0$.
3. Calcul de $f_a^k(x), f_b^k(x), x = x^k, UB_k$ et LB_k .

$$\begin{cases} f_a^k(x) = f(a^k) - L(x - a^k)^{\frac{1}{\alpha}} \\ f_b^k(x) = f(b^k) - L(b^k - x)^{\frac{1}{\alpha}} \end{cases} \implies \begin{cases} f_a^k(x) = f_b^k(x) \implies x = x^k \\ UB_k = f(x^k) \\ LB_k = f_a^k(x) = f_b^k(x) \end{cases}$$

4. Si : $UB_k - LB_k \leq \varepsilon$, on s'arrête et on aura la solution du problème (P_3) :

$$\begin{cases} x = x^k \\ UB_k = f(x^k) \\ LB_k = f_a^k(x) = f_b^k(x) \end{cases}$$

5. Sinon, subdiviser I_k en deux sous intervalles (ou en un nombre fini de sous intervalles) I_{k_1} et I_{k_2}

$$\bigcup_{i=1}^{i=2} I_{k_i} = I \text{ et } \bar{I}_{k_1} \cap \bar{I}_{k_2} = \emptyset$$

où \bar{I} est l'intérieur de I

6. Construire les problèmes des bornes inférieure et supérieure de $\min f(x)$ sur $S \cap I_{k_i}$, $i = 1, 2$. Soient LB_{k_1} , UB_{k_1} et LB_{k_2} , UB_{k_2} les solutions obtenues.

7. Posons

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

8. Posons : $I_k = R_{k_1}, R_{k_2}$.

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

$$LB_{k_1} > UB_{k+1} \text{ où } S \cap I_{k_j} = \emptyset$$

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

10. Posons : $k = k + 1$ et revenant à 4.

4.2.5 Exemples d'applications

Ètant donné le problème de minimisation non convexe suivant :

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

La méthode branch and bound consiste à remplacer la fonction objectif par la fonction borne inférieure dans les exemples donnés .

Cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour soit les exclure, soit les maintenir comme des solutions potentielles.

L'idée a été déjà proposée pour les fonctions objectifs satisfaisantes la propriété lipchitzienne.

En effet, la propriété lipchitzienne a fourni un chemin pour construire la fonction borne inférieure.

Exemple 4.1 Soit à résoudre le Problème de minimisation suivant :

$$(P_1) \begin{cases} \min \left\{ \frac{3}{4} \sin(x) + \frac{1}{4} \cos(x) \right\} \\ 2.7 \leq x \leq 7.5 \end{cases}$$

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

0 Itération : ($j = 0$)

1. Soit $x \in [a^j, b^j]$
2. Posons $j = 0, I_0 = [a^0, b^0] = [2.7, 7.5]$
3. Calcul de $f_a^0(x), f_b^0(x), x_0^*, UB_0$ et LB_0 .

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

4. $UB_k - LB_k = -4.5869 - (-38.5389) = 33.952 > \varepsilon$, on passe à la 1^{ière} itération, c'est-à-dire : subdiviser l'intervalle $I_0 = [2.7, 7.5]$ en deux sous intervalles $I_{1_1} = [2.7, 5.1507]$ et $I_{1_2} = [5.1507, 7.5]$.

1^{ière} Itération : ($j = 1$)

1. Soit $x \in [a^j, b^j]$
2. Posons $j = 1, I_1 = [a^1, b^1] = [2.7, 7.5]$
premier cas : $x \in [a^1, x_0] = [2.7, 5.1507] = I_{1_1}$
- Calcul de $f_a^{1^1}(x), f_b^{1^1}(x), x_{1_1}^*, UB_{1_1}$ et LB_{1_1} :

$$\begin{cases} f_a^{1^1}(x_{1_1}) = -0.4352 \\ f_b^{1^1}(x_{1_1}) = -4.5869 \end{cases} \implies \begin{cases} x_{1_1}^* = 4.0609 \\ UB_{1_1} = -4.5869 \\ LB_{1_1} = -12.0101 \end{cases}$$

deuxiem cas : $x \in [x_0, b^1] = [5.1507, 7.5] = I_{1_2}$

1. Calcul de $f_a^{1^2}(x), f_b^{1^2}(x), x_{1_2}^*, UB_{1_2}$ et LB_{1_2} :

$$\begin{cases} f_a^{1^2}(x_{1_2}) = -4.5869 \\ f_b^{1^2}(x_{1_2}) = -3.4794 \end{cases} \implies \begin{cases} x_{1_2}^* = 6.3254 \\ UB_{1_2} = -4.5869 \\ LB_{1_2} = -13.2107 \end{cases}$$

2. Calcul UB_1 et LB_1 .

$$\begin{cases} UB_1 = \min \{UB_{1_1}, UB_{1_2}, UB_0\} = -4.5869 \\ LB_1 = \min \{LB_{1_1}, LB_{1_2}\} = -13.2107 \end{cases}$$

3. $UB_1 - LB_1 = -4.5869 - (-13.2107) = 8.6238 > \varepsilon$, donc on passe à la 3^{ieme} itération, c'est-à-dire : subdiviser l'intervalle $I_{1_1} = [2.7, 5.1507]$ en deux sous intervalles $I_{3_1} = [5.1507, 6.3254]$ et $I_{3_2} = [6.3254, 7.5]$. Car $LB_{1_1} > LB_{1_2}$
 - Pas d'élimination d'intervalles car les $LB_{k_j} \not\leq UB_{k+1}$

2^{ieme} **Itération** : ($j = 2$)

1. Soit $x \in [a^2, b^2]$
2. Posons $j = 2, I_2 = [a^2, b^2] = [5.1507, 7.5]$
premier cas : $x \in [a^2, x_1] = [5.1507, 6.3254] = I_{2_1}$
 - Calcul de $f_a^{2_1}(x), f_b^{2_1}(x), x_{2_1}, UB_{2_1}$ et LB_{2_1} :

$$\begin{cases} f_a^{2_1}(x_{1_1}) = -4.5869 \\ f_b^{2_1}(x_{1_1}) = -2.6390 \end{cases} \implies \begin{cases} x_{2_1} = 5.6054 \\ UB_{2_1} = 5.6054 \\ LB_{2_1} = -5.8789 \end{cases}$$

- deuxième cas** : $x \in [x_1, b^2] = [6.3254, 7.5] = I_{2_2}$
 - Calcul de $f_a^{2_2}(x), f_b^{2_2}(x), x_{2_2}^*, UB_{2_2}$ et LB_{2_2} :

$$\begin{cases} f_a^{2_2}(x_{1_2}) = -2.6390 \\ f_b^{2_2}(x_{1_2}) = -3.4794 \end{cases} \implies \begin{cases} x_{2_2}^* = 7.0453 \\ UB_{2_2} = -4.2722 \\ LB_{2_2} = -5.8789 \end{cases}$$

3. Calcul UB_2 et LB_2 .

$$\begin{cases} UB_2 = \min \{UB_{2_1}, UB_{2_2}, UB_1\} = -4.5869 \\ LB_2 = \min \{LB_{2_1}, LB_{2_2}, LB_{1_2}\} = -13.2107 \end{cases}$$

4. $UB_2 - LB_2 = -4.5869 - (-13.2107) = 8.6238 > \varepsilon$, on passe à la 2^{ieme} itération, c'est-à-dire : subdiviser l'intervalle $I_{1_2} = [5.1507, 7.5]$ en deux sous intervalles $I_{3_1} = [5.1507, 3.92535]$ et $I_{3_2} = [3.92535, 7.5]$
 - Pas d'élimination d'intervalles car les $LB_{k_j} \not\leq UB_{k+1}$

Remarque 4.1

On a écrit les deux itérations dans le but d'expliquer la procédure de l'algorithme branch and bound pour un problème d'une fonction à une seule variable d'optimisation globale.

Conclusion

Au beau de la 7^{ieme} itération on aura la solution suivante du problème (P1) :
Donc la solution optimale exacte est $x^* = 5.199778$ et la valeur optimale est -4.601308 .

Exemple 4.2 Soit à résoudre le Problème de minimisation suivant :

$$(P_2) \begin{cases} \min \left\{ \sin(x) + \sin\left(\frac{10x}{3}\right) + \ln(x) - 0.84x \right\} \\ 0 \leq x \leq 1 \end{cases}$$

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

0 Itération : ($j = 0$)

1. Soit $x \in [a^j, b^j]$
2. Posons $j = 0, I_0 = [a^0, b^0] = [0, 1]$
3. Calcul de $f_a^0(x), f_b^0(x), x_0^*, UB_0$ et LB_0 .

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

Calcul UB_0 et LB_0 .

$x_0^* \notin [0, 1]$ Alors $UB_0 = LB_0 = \min \{f(0), f(1)\} = f(0) = 0.25$.

Donc la solution optimale exacte du problème (P2) est $x^* = 0$ et la valeur optimale est 0.25.

4.3 La méthode de Piyavskii

4.3.1 Introduction

On va présenter l'algorithme de Piyavskii [16] qui est un algorithme pour optimiser une fonction lipschitzienne d'une seule variable sur un intervalle $[a; b]$ dans \mathbb{R} , puis on va définir l'extension de cet algorithme au cas multidimensionnel dans \mathbb{R}^n .

Définition 4.1 Une fonction $f : P \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est dite lipschitzienne sur P , s'il existe une constante réelle $L = L(f, P) > 0$ telle que :

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

où L est une constante appelée la constante de Lipschitz

Proposition 4.1 [13] Soit $P \subset \mathbb{R}^n$ un ensemble convexe, et f une fonction continûment différentiable sur un ensemble ouvert contenant X avec un gradient borné sur X alors f est Lipschitzienne sur X avec la constante

$$L = \sup \{ \|\nabla f(x)\|, \quad x \in P \} :$$

4.3.2 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 :

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

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 $(**)$ pour obtenir les deux inégalités suivantes :

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

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

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

où \tilde{f} est donnée par :

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

où

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

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

Lemme 4.1 Soient f une fonction lipschitzienne sur $[a; b]$ de constante de Lipschitz L et $x(a, b)$ donné par l'équation $(***)$, Alors $x(a, b) \in [a, b]$

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 $f_{opt} = \min \{f(a), f(b)\}$, $x_{opt} = \operatorname{argmin} \{f(a), f(b)\}$, $B_{opt} = B_1$,
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
 $f_{opt} = \min_{1 \leq i \leq k+1} \{f(a); f(u_i)\}$,
 $x_{opt} = \operatorname{arg} \min_{1 \leq i \leq k+1} \{f(a), f(u_i)\}$
 $B_{opt} = \min_{1 \leq i \leq k+1} B_i$
 $k=k+1$
Fin Tant que

4.3.3 Algorithme de Piyavskii généralisé

Cas sans contraintes

Considérons le problème de minimisation globale [14]

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

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

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

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

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

Considérons un point x_i du pavé 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 pavé P .

Si $c = \frac{a+b}{2}$ est le centre du pavé 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 pavé 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 $i = 1, \dots, n$.

Algorithme

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

Éliminer de \mathbf{L} tous les sous problèmes ayant une borne inférieure supérieure
à f_{opt}
Fin Tant que

Remarque 4.2 *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.

4.3.4 Convergence de l'algorithme

On cite ici un théorème de convergence établi par [17]

Théorème 4.2 *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 :

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

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

Il suffit de choisir :

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

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

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

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

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

Puisque ε est arbitraire, on a :

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

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

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

4.3.5 Application numérique de l'algorithme

Cas sans contraintes

Dans ce tableau on a les problèmes de minimisation définis dans \mathbb{R}^2 :

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

TAB. 4.1 – Définition des fonctions choisies pour l'évaluation de l'algorithme de piyavskii.

Dans le tableaux suivant on a les résultats numériques des problèmes définis ci-dessus. On a utilisé les notations suivantes :

f_{opt} : Valeur optimale.

$Nbre$: le nombre d'itération de l'algorithme.

t : le temps de calcul.

| Numéro | f_{opt} | Précision ε | Vecteur optimal trouvée | $Nbre$ | t (sec) |
|--------|-----------|-------------------------|-------------------------|--------|-----------|
| 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 |

TAB. 4.2 – Résultats numérique de l'algorithme de piyavskii.

Cas avec contraintes

Soient les problèmes suivants :

Problème 1

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

Problème 2

$$(P_1) \left\{ \begin{array}{l} J(x_1, x_2) = -(x_1 - 4.2)^2 - (x_2 - 1.9)^2 \\ -x_1 + x_2 \leq 3 \\ x_1 + x_2 \leq 11 \\ x_1 - x_2 \leq 16 \\ -x_1 - x_2 \leq -1 \\ 0 \leq x_1 \leq 10 \\ 0 \leq x_2 \leq 5 \end{array} \right.$$

Dans le tableau suivant les résultats numériques des problèmes définis ci-dessus.

On a utilisé les notations suivantes :

f_{opt} : Valeur optimale

$Nbre$: le nombre d'itération de l'algorithme

t : le temps de calcul

| Numéro | f_{opt} | Vecteur optimal trouvée | $Nbre$ | t (sec) |
|--------|-----------|-------------------------|--------|-----------|
| 1 | -4.5955 | (1.4012,1.8098,1.8037) | 17533 | 38.2343 |
| 2 | 28.6419 | (9.4444,0.8333) | 123 | 27.35 |

TAB. 4.3 – Résultats numérique de l'algorithme de piyavskii.

Interprétation des résultats

On remarque que l'algorithme de Piyavskii converge vers l'optimum global, mais en l'implémentant nous avons pu constater qu'il devient lent au voisinage de l'optimum et c'est ce qui explique le temps de calcul élevé par rapport à la dimension des problèmes.

Conclusion générale

Ce travail présente une vision globale sur l'optimisation non linéaire et l'optimisation globale, qui englobe un sujet très intéressant qui est la résolution d'un problème d'optimisation non linéaire sans contraintes et sous contraintes, en utilisant plusieurs méthodes comme la méthode de descente basée sur le gradient, méthode de Newton et les méthodes de pénalité.

On a dressé donc, dans le premier chapitre, un aperçu général sur les notions d'optimisation non linéaire, dont on a besoin tout au long de ce mémoire. Le deuxième et le troisième chapitre sont consacrés à l'optimisation non linéaire sans et avec contraintes, et les différentes méthodes qu'on utilise pour la résolution des problèmes d'optimisation.

Dans le dernier chapitre, on a fourni un aperçu de l'optimisation globale, on a présenté l'algorithme de branch-and-bound et celui de Piyavskii qui sont des algorithmes déterministes de l'optimisation globale, établis pour optimiser une fonction Lipschitzienne (un cas particulier d'une fonction holderienne) d'une seule variable sur un intervalle $[a,b]$ dans \mathbb{R} , aussi on a étendu l'algorithme de Piyavskii au cas multidimensionnel selon l'approche de Shubert [18]. Un théorème de convergence de chacun des deux algorithmes a été présenté et des tests numériques sur des problèmes classiques de la littérature ont montré leurs convergences vers l'optimum global, mais l'algorithme de Piyavskii devient lent au voisinage de l'optimum.

Bibliographie

- [1] L. Pujo-Menjouet. Cours de calcul différentiel. Université Claude Bernard, Lyon, France.
- [2] J. Royer. Calcul Différentiel et intégral. Université Toulouse 3, (2013-2014).
- [3] M.Aiden ,B.Oukacha . Les manuels de l'étudiant Recherche opérationnelle.
- [4] A. Rondepierre. Méthodes numériques pour l'optimisation non linéaires déterministe 4ème année, INSA Toulouse 2017-2018.
- [5] M Marco LOPEZ, George STILL . Semi-Infinite Programming, 2005.
- [6] E. J. Anderson, P. Nas. Linear Programming in Infinite Dimensional Spaces, Wiley, New York, 1987.
- [7] Bertsekas Dimitri.P. Nonlinear programming", Second printing, Athena scientific.
- [8] Belahcene. Saliha. "Résolution de problèmes de programmation Semi-infinie et Introduction à l'optimisation globale". Mémoire de magister, UMMTO(2005).
- [9] Osyezka .A, "Multicriteria optimization for engineering design". Dans J. S. Gero, éditeur, Design optimization ,Academic Press,, pages 193-227.1985 [Osy85].
- [10] Minoux Michel," Programmation Mathématique Théorie et Algorithmes " ; Collection Théchnique et scientifique des Télécommunication, a volume 1.
- [11] Jaumard B. and Ellaia R. Gourdin, E. "global optimization of holder function". Journal of Global Optimization, 8(4) : 323–348, (1996).
- [12] Dr RADJEF. "Optimisation Cours et exercices", 2019.
- [13] E.K.P. CHONG and S.H. ZAK. An Introduction to Optimization, Second Edition, WILEYInterscience 2001.
- [14] GUETTAL Djaouida. Efficacité et fiabilité des méthodes utilisant pour l'optimisation globale non convexe. Doctorat en-sciences, UNIVERSITE FERHAT ABBAS SETIF 1, 2014/2015.
- [15] Houria. BEKRI. Optimisation globale avec application pour les fonctions de hölder. Mémoire de magister, UMMTO (2012).
- [16] S. A. Piyavskii, An algorithm for finding the absolute minimum for a function, Theory of Optimal Solution, No. 2, pp. 13-24. (1967)

- [17] S.A. Piyavskii. An algorithm for finding the absolute extremum of a function, USSR Computational Mathematics and Mathematical physics, 12,1972, pp.57-67 (Zh. vychisl Mat.mat.Fiz. 12(4) (1972) 888-896)
- [18] B. O. Shubert, A sequential method seeking the global maximum of a function, SIAM, Journal of Numerical Analysis, Vol. 3, pp. 379-388,(1972) :
- [19] R. J. Vanderbei, Extension of Piyavskii's Algorithm to Continuous Global Optimization, Journal of Global Optimization, Vol. 14, pp. 205-216.(1999).