

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

Université Akli Mohand Oulhadj de Bouira
Faculté des Sciences et des Sciences Appliquées
Département du Mathématiques



Mémoire de Master

Filière : Mathématiques

Spécialité : Recherche Opérationnelle

Thème

Optimisation locale et introduction à
l'optimisation globale

Présenté par : Djamel ACHOUR

Devant le jury composé de :

Président	<i>M^r</i> Nacer DEMMOUCHE	MAA	U. A/M/O Bouira.
Encadreur	<i>M^r</i> Abderrahmane AKKOUICHE	MCB	U. A/M/O Bouira.
Examinatrice	<i>M^{me}</i> Daya OUIDJA	MAA	U. A/M/O Bouira.

2017/2018

Remerciements

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

Un grand merci à mon promoteur monsieur *Abderrahmane AKKOUCHE* pour ses conseils et son aide et qui a mis à notre disposition tout le nécessaire pour réaliser ce travail.

Je remercie les membres de jury pour avoir accepté de juger ce travail.

- Monsieur *Nacer DEMMOUCHE* , qui me fait l'honneur de précider ce jury

- Madame *Daya OUIDJA*, pour avoir acceptée d'être examinatrice de mon mémoire

J'aimerais remercier mes parents pour leurs encouragements et leur soutien

Sans oublier les bons collègues, Je les remercie 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

À les plus beaux créatures que Dieu a créé sur terre ,,,

À ces sources de tendresse, de patience et de générosité,,,

À mes parents l'Oxygène de ma vie

À mes frères Ziane, Mahmoud, Saad, A.essalem, Nour eddine

Ibrahim, A.elkader, Ameer

À mes sœurs Zohra, Dawadia, Latifa, Fatima

ainsi que les enfants A.erraouf, A.errahmane, Oussama,

Naila et Mohamed djawad

Aux femmes de mes frères Naima, Merzaka, Rokaya, Somia

À tous les membres de notre grande famille Ahmed, Makhluof,

Ali, Omar, Lakhdar, ...

À Mon cousin monsieur Khalifa

À tous ceux qui m'ont appris et tous mes disciples

À tous mes amis et mes proches sans exception

À tous les étudiants de la promotion 2017/2018

Option : Recherche Opérationnelle

À tous ceux qui, par un mot, m'ont donné la force de continuer...

Je dédie ce travail

Table des matières

Introduction générale	3
1 Résultats fondamentaux	4
1.1 Généralités sur les matrices	4
1.1.1 Déterminant d'une matrice	5
1.1.2 Mineurs principaux d'une matrice	6
1.1.3 Mineurs principaux diagonaux d'une matrice	6
1.1.4 Matrice hessienne	6
1.1.5 Matrice jacobienne	6
1.2 Matrices (semi) définies positives, matrice (semi) définies négatives	7
1.2.1 Matrice définie positive	7
1.2.2 Matrice semi-définie positive	7
1.2.3 Matrice définie négative	7
1.2.4 Matrice semi-définie négative	7
1.2.5 Matrice indéfinie	7
1.2.6 Application à l'étude des points critiques	8
1.3 Ensembles convexes, Polyèdres, Combinaison et Enveloppe convexes	8
1.3.1 Ensembles convexes	8
1.3.2 Polyèdres convexes	9
1.3.3 Combinaison convexe	9
1.3.4 Enveloppe convexe	9
1.3.5 Propriétés algébriques	10
1.3.6 Propriétés topologiques	10
1.4 Fonctions convexes	10
1.4.1 Interprétation géométrique	11
1.5 Fonctions convexes différentiables	12
1.6 Fonctions convexes deux fois différentiables	12
1.7 Problèmes d'optimisation	13
1.7.1 Cadre et vocabulaire	13
1.7.2 Définitions	13
2 Problèmes d'optimisation convexes	15
2.1 Problèmes d'optimisation sans contraintes	15
2.1.1 Conditions d'existence et d'unicité de la solution	15
2.1.2 Conditions d'optimalité	17
2.1.3 Conditions d'optimalité globale	17
2.2 Problèmes d'optimisation sous contraintes	18
2.2.1 Problèmes d'optimisation sous contraintes égalités	18
2.3 Espace tangent et espace normal	19
2.3.1 Problème d'optimisation avec contraintes inégalités	23

2.3.2	Problème d'optimisation avec contraintes égalités et inégalités	23
2.3.3	Problèmes d'optimisation linéaires	26
2.3.4	Conditions d'existence et d'unicité de solution	28
2.3.5	Conditions d'optimalité	28
2.4	Problèmes d'optimisation quadratiques	29
2.4.1	Formulation du problème	29
2.4.2	Conditions d'existence et d'unicité de solution	30
2.4.3	Conditions d'optimalité	31
3	Méthodes de résolution	32
3.1	Problèmes d'optimisation sans contraintes	33
3.1.1	Méthode du Gradient	33
3.1.2	Méthode de Newton	35
3.1.3	Méthode du Gradient conjugué	35
3.1.4	Méthode de Relaxation	36
3.2	Problèmes d'optimisation avec contraintes	37
3.2.1	Rappel	37
3.2.2	Méthode du Simplexe	38
3.2.3	Méthode du gradient projeté	42
3.2.4	Méthode SQP (Sequential Quadratic Programming)	44
3.2.5	La méthode SQP pour les problèmes sous contraintes de type égalité	44
3.2.6	Algorithme de la méthode SQP pour les problèmes sous contraintes de type égalités	45
3.2.7	La méthode SQP pour les contraintes inégalités	50
3.2.8	Algorithme de la méthode des contraintes actives	50
4	Introduction à l'optimisation globale	55
4.1	Optimisation globale	56
4.1.1	Problème optimisation globale	56
4.1.2	Optimisation DC	56
4.1.3	Optimisation Lipshitzienne	56
4.2	Les méthodes de résolution	57
4.2.1	La méthode de branch-and-bound	57
4.2.2	Principe de la méthode	57
4.2.3	Algorithme de Branch-and-bound	58
4.3	Conclusion	61
5	Application	63
	Conclusion générale	68
	Bibliographie	69

Introduction générale

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

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

Beaucoup de systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'algorithme et des moyens pour le traitement numérique. Dans ce cadre d'étude, il nous faut d'abord distinguer deux filières d'optimisation :

- L'optimisation statique
- L'optimisation dynamique

Dans ce mémoire, on s'intéresse à l'optimisation statique ; qu'est la filière d'optimisation qui utilise les méthodes de type indépendant de temps (méthodes de programmation linéaire,...).

Ce mémoire s'articule autour de quatre chapitres principaux :

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

Et dans le deuxième chapitre nous essayons d'étudier les problèmes d'optimisation convexes (sans contraintes et avec contraintes) et les conditions d'existence et d'unicité avec l'optimalité de chaque type de problème.

Le troisième chapitre est consacré à présenter quelques méthodes (algorithmes) qui permet de calculer (de manière approchée) la ou les solutions du problème d'optimisation : sans contraintes (méthode de gradient, Newton, gradient conjugué,...), avec contraintes linéaire(méthode de simplexe) et non linéaire (méthode SQP), le problème quadratique (Méthode du gradient projeté).

Le quatrième chapitre est consacré à l'optimisation globale.

Après avoir terminé le quatrième chapitre on passe à une petite application.

Puis enfin une conclusion générale sur le travail effectué.

Chapitre 1

Résultats fondamentaux

Afin de rendre facile la compréhension de ce mémoire, nous avons besoin de rappeler quelques notions et propriétés d'analyse convexe, qui sont utiles pour la suite et elles sont données sous une forme beaucoup plus adéquate que possible.

1.1 Généralités sur les matrices

Définition 1.1 (Matrice). Une matrice à m lignes et n colonnes est un tableau rectangulaire de $m \times n$ nombres, rangés ligne par ligne. Il y a m lignes, et dans chaque ligne n nombres.

On dit que la matrice a m lignes et n colonnes, ou qu'elle est de **dimension** ou de **taille** (m, n) . En notant a_{ij} l'image d'un couple (i, j) par l'application A , la matrice peut alors être notée

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

Ou plus simplement (a_{ij}) si le contexte s'y prête.

On représente généralement une matrice sous la forme d'un tableau rectangulaire. Par exemple, est représentée ci-dessous une matrice A , à coefficients entiers, et de dimension $(3, 4)$

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

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

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

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

Les coefficients a_{ij} avec $i = j$ sont dits **diagonaux**. La diagonale de A est le vecteur $\begin{pmatrix} a_{11} \\ a_{22} \\ \vdots \\ a_{pp} \end{pmatrix}$

Définition 1.2 (Matrice transposée). On appelle une matrice transposée de A la matrice A^t de dimension (n, m)

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

Par exemple, avec la matrice A de l'exemple précédent, on a

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

Définition 1.3 (Matrice symétrique). Une matrice A est symétrique si $A^t = A$

Définition 1.4 (Matrice identité). Pour chaque nombre entier n , on note I_n la matrice carrée de taille n dont les coefficients diagonaux sont égaux à 1 et dont les autres coefficients sont nuls; elle est appelée matrice **identité** de taille n . $I_n = (\delta_{ij})_{1 \leq i \leq n, 1 \leq j \leq n}$

Définition 1.5 (Matrice inversible). Soit A une matrice de dimension (m, n) . On dit que A est inversible à droite (respectivement à gauche) s'il existe une matrice B de taille (n, m) telle que $AB = I_m$ (respectivement $BA = I_n$). Elle est simplement dite inversible si elle l'est à la fois à droite et à gauche.

1.1.1 Déterminant d'une matrice

Le déterminant d'une matrice est défini par induction :

- Une matrice de taille $(1, 1)$ est juste un scalaire a . Le déterminant de cette matrice est égal à a .
- Matrice de taille $(2, 2)$: soit $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$. Le déterminant de M s'écrit $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$.
- Matrice de taille (n, n) : soit $M = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$, Pour calculer le déterminant de

M , il suffit d'utiliser la méthode dite du *développement selon la première ligne* :

1. On associe à chaque coefficient a_{1i} de la première ligne de la matrice M un signe $+$ si i est impair et un signe $-$ si i est pair.
2. Le déterminant de M peut s'écrire comme la somme des n déterminants d'ordre $n - 1$ obtenus en éliminant de la matrice M la ligne et la colonne contenant le coefficient a_{1i} . Chacun de ces déterminants est multiplié par $(-1)^{i+1}a_{1i}$.

Exemple 1.1. Soit $M = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$. Le déterminant de la matrice M s'écrit :

$$\begin{aligned} \det(M) &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ &= a(ei - fh) - b(di - fg) + c(dh - eg). \end{aligned}$$

1.1.2 Mineurs principaux d'une matrice

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

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

Les trois mineurs principaux d'ordre 1 de M sont a_{11} , a_{22} et a_{33} .

Les trois mineurs principaux d'ordre 2 de M sont : $\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$, $\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}$ et $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$.

Le mineur principal d'ordre 3 de M 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

1.1.3 Mineurs principaux diagonaux d'une matrice

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

Soit ; $M = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$. Le mineur principal diagonal d'ordre 1 de M est : a_{11} .

Le mineur principal diagonal d'ordre 2 de M est : $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$.

Le mineur principal diagonal d'ordre 3 de M est : $\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$.

1.1.4 Matrice hessienne

Soit f définie de \mathbb{R}^n à \mathbb{R}

On appelle matrice hessienne $H(x)$ de f la matrice des dérivées secondes de f évaluées au point $x = (x_1, x_2, \dots, x_n)$.

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

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

1.1.5 Matrice jacobienne

Soit $G = (g_1, g_2, \dots, g_m)$ une fonction définie de \mathbb{R}^n dans \mathbb{R}^m . A tout vecteur $x^* = (x_1, x_2, \dots, x_n)$, la fonction G associe le vecteur de fonctions $(g_1(x^*), g_2(x^*), \dots, g_m(x^*))$.

On appelle matrice jacobienne de G la matrice de dimension (m, n) , $J_G(x)$ des dérivées partielles d'ordre 1 des m fonctions qui composent G :

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

1.2 Matrices (semi) définies positives, matrice (semi) définies négatives

1.2.1 Matrice définie positive

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

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

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

1.2.2 Matrice semi-définie positive

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

$$X^t M X \geq 0 \quad \forall X$$

Les éléments diagonaux a_{ii} d'une matrice semi-définie positive sont tous ≥ 0 .

1.2.3 Matrice définie négative

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

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

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

1.2.4 Matrice semi-définie négative

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

$$X^t M X \leq 0 \quad \forall X$$

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

1.2.5 Matrice indéfinie

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

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

Caractérisation :

Soit M une matrice carrée symétrique d'ordre n .

- M définie positive \iff ses n mineurs principaux diagonaux D_k sont > 0 .
- M semi-définie positive \iff tous ses mineurs diagonaux (et pas seulement diagonaux !) D_k sont ≥ 0 .
- M définie négative \iff ses n mineurs principaux diagonaux D_k sont alternativement < 0 (k impair) et > 0 (k pair).
- M semi-définie négative \iff tous ses mineurs principaux D_k (et pas seulement diagonaux !) sont alternativement ≤ 0 (k impair) et ≥ 0 (k pair).

Théorème 1.1. [4]

Soit A une matrice carrée symétrique

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

1.2.6 Application à l'étude des points critiques

La matrice hessienne permet de déterminer la nature des points critiques de fonction f , c-à-d un point critique peut être :

- Un minimum si la matrice Hessienne H associée à f est définie positive.
- Un maximum si la matrice Hessienne H associée à f définie négative .
- Un point selle si la matrice Hessienne H associée à f indéfinie.

1.3 Ensembles convexes, Polyèdres, Combination et Enveloppe convexes

1.3.1 Ensembles convexes

Définition 1.6. Un ensemble $E \in \mathbb{R}^n$ est dit convexe si pour tout couple $(x, y) \in E$ et $\lambda \in]0, 1[$, on a : $\lambda x + (1 - \lambda)y \in E$

Géométriquement, cette notion s'interprète comme suit :

"Pour tout segment reliant deux points quelconques x et y de E , le segment $[x; y]$ doit être aussi inclus dans E "

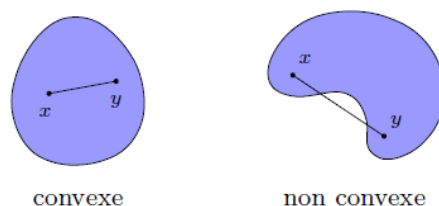


FIGURE 1.1 – Définition des ensembles convexes et non convexes

1.3.2 Polyèdres convexes

On rappelle qu'un *polyèdre convexe* d'un espace vectoriel K est un ensemble E de la forme :

$$E = \{x \in K : Ax \leq b\}$$

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

Géométriquement, un polyèdre convexe est donc l'intersection d'un nombre fini de demi-espaces de K .

Théorème 1.2. [1] Soient E et F deux espaces vectoriels, $f : E \rightarrow F$ une application linéaire et P un polyèdre convexe de E . Alors $f(P)$ est un polyèdre convexe de F .

1.3.3 Combination convexe

Définition 1.7. Un vecteur $X \in \mathbb{R}^n$ est dit *combinaison convexe* des vecteurs $x_1, x_2, \dots, x_m \in \mathbb{R}^n$ s'il existe des nombres réels $\lambda_1, \lambda_2, \dots, \lambda_m$ tel que :

$$X = \sum_{i=1}^m \lambda_i x_i, \lambda_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \lambda_i = 1.$$

Remarque 1.1. Si $m = 2$, on retrouve la définition précédente (définition(1.3.1))

Théorème 1.3. [1] Un ensemble $E \subset \mathbb{R}^n$ est convexe si et seulement si toute combinaison convexe de m points de E appartient à E .

Théorème 1.4. [1] L'intersection d'un nombre fini ou infini d'ensembles convexes est convexe.

Théorème 1.5. [1] Soit E un ensemble convexe. Alors le produit αE , où α est un nombre réel, est un ensemble convexe.

Théorème 1.6. la somme $E = E_1 + E_2$ de deux ensembles convexes $E_1, E_2 \subset \mathbb{R}^n$ est convexe

1.3.4 Enveloppe convexe

Définition 1.8. L'enveloppe convexe d'un ensemble arbitraire E de \mathbb{R}^n est le plus petit convexe (au sens de l'inclusion) qui contient E . Elle est notée $\text{conv}(E)$.

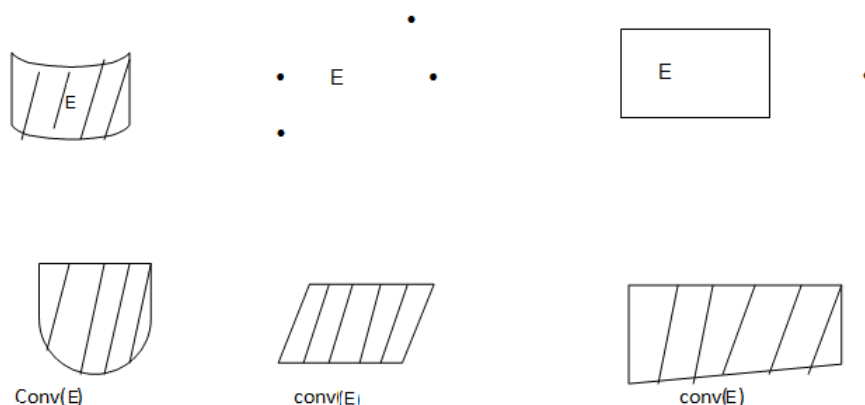


FIGURE 1.2 – Enveloppe convexe

Théorème 1.7. [1] L'enveloppe convexe $\text{conv}(E)$ d'un ensemble E dans \mathbb{R}^n est l'ensemble de toutes les combinaisons convexes des points de E .

1.3.5 Propriétés algébriques

On a les propriétés élémentaires suivantes :

1. Si C_1, C_2 sont deux convexes de \mathbb{R}^n , λ_1 et λ_2 deux réels. Alors il existe $x_1 \in C_1$ et $x_2 \in C_2$ tel que : $\lambda_1 C_1 + \lambda_2 C_2$ est un convexe de \mathbb{R}^n .
2. Si $(C_j)_{j \in J}$ est une famille quelconque de convexes de \mathbb{R}^n . Alors $\bigcap_{j \in J} C_j$ est une convexe de \mathbb{R}^n .
3. Si C est un convexe de \mathbb{R}^n et $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ est une application affine, i.e $f(x) = Ax + b$ pour certains A et $b \in \mathbb{R}^m$, alors $f(C)$ est un convexe de \mathbb{R}^m .

1.3.6 Propriétés topologiques

On a les propriétés élémentaires suivantes :

1. Si C est un convexe de \mathbb{R}^n , son adhérence \bar{C} est un convexe de \mathbb{R}^n .
2. Si C est un convexe de \mathbb{R}^n , son intérieur \dot{C} est un convexe de \mathbb{R}^n .

Pour l'étude des ensembles convexes, il est commode d'adopter la notion de topologie relative : celle-ci qui n'est autre que la topologie au sens usuel, mais considérée dans le plus petit sous-espace affine comprenant l'en semble en question, autrement dit dans son enveloppe affine.

Définition 1.9 (intérieur relatif). Soit S un sous-ensemble quelconque de \mathbb{R}^n , l'intérieur relatif de S est l'ensemble :

$$\text{int}_r(S) = \{x \in S \mid \exists \epsilon > 0, B(x, \epsilon) \cap \text{aff}(S) \subset S\}$$

Bien sûr, on a toujours $S \subseteq \text{int}_r(S)$. Pour un ensemble convexe, on a de plus.

Théorème 1.8. [8] Si C est un convexe de \mathbb{R}^n , son intérieur relatif $\text{int}_r(C)$ est un convexe de \mathbb{R}^n . De plus, si C n'est pas vide, alors $\text{int}_r(C)$ non plus.

1.4 Fonctions convexes

Définition 1.10 (Fonction convexe). Soit $E \subset \mathbb{R}^n$ un ensemble convexe. Soit $f : E \rightarrow \mathbb{R}$ une fonction à valeurs réelles.

— f est convexe si :

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

— f est strictement convexe si :

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

Définition 1.11 (Fonctions concave). Soit $E \subset \mathbb{R}^n$ un ensemble convexe. Soit $f : E \rightarrow \mathbb{R}$ une fonction à valeurs réelles.

— f est concave si :

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

— f est strictement concave si :

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

1.4.1 Interprétation géométrique

L'inéquation (1.1) à l'interprétation géométrique suivante la valeur de la fonction f en tout point x du segment $[x_1, x_2]$ est situé sous la valeur de la "corde" (interprétation linéaire de f entre x_1 et x_2) au même x .

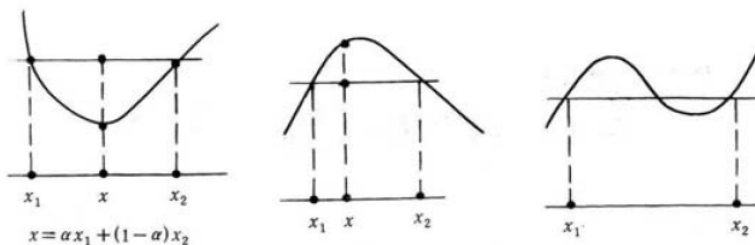


FIGURE 1.3 – Fonction convexe et fonction concave, fonction non convexe et non concave.

Propositions 1.1. [8]

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

- $f_1 + f_2$ est convexe.
- $\text{Max}\{f_1, f_2\}$ est convexe.

Remarque 1.2. On peut également restreindre λ à $]0; 1[$ pour la définition de la convexité.

Voici quelques exemples de fonctions convexes :

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

Proposition 1.1. [8] Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction. Les propriétés suivantes sont équivalentes :

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

Définition 1.12 (Épigraphe). Soit E un sous ensemble de \mathbb{R}^n . et soit $f : E \rightarrow \mathbb{R}$. on appelle épigraphe de f , notée $\text{epi}(f)$, le sous ensemble de \mathbb{R}^{n+1} donnée par :

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

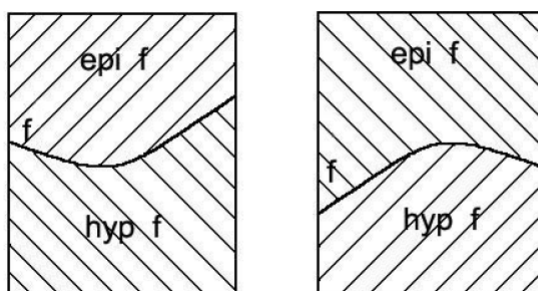


FIGURE 1.4 – épigraphe(f).

Théorème 1.9. [8]

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

Définition 1.13 (Ensemble de niveau). Soit E un ensemble convexe de \mathbb{R}^n , et soit f une fonction convexe sur E . On appelle ensemble de niveau α pour f l'ensemble :

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

Lemme 1.1. Soit $E \subset \mathbb{R}^n$ un ensemble convexe, et Soit $f : E \rightarrow \mathbb{R}$ une fonction convexe. Alors l'ensemble de niveau $E_\alpha = \{x \in E \mid f(x) \leq \alpha\}$, avec $\alpha \in \mathbb{R}$, est un ensemble convexe.

1.5 Fonctions convexes différentiables

Théorème 1.10. [2]

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

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

Définition 1.14. Soit E un ensemble non vide de \mathbb{R}^n , et soit $f : E \rightarrow \mathbb{R}$. Alors f est dite différentiable au point $\bar{x} \in \text{int}_r(E)$ s'il existe un vecteur $\nabla f(\bar{x})$, appelé vecteur gradient, et une fonction $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ tel que :

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

pour chaque $x \in E$, où

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

1.6 Fonctions convexes deux fois différentiables

Théorème 1.11. [2]

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

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

Définition 1.15. Soit E un ensemble non vide de \mathbb{R}^n , et soit $f : E \rightarrow \mathbb{R}$. Alors f est dite deux fois différentiable au point $\bar{x} \in \text{int}_r(E)$ s'il existe un vecteur $\nabla f(\bar{x})$, et une $n \times n$ -matrice symétrique $H(\bar{x})$, appelée la matrice Hessienne, et une fonction $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$ tel que :

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

pour chaque $x \in E$, où

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

1.7 Problèmes d'optimisation

1.7.1 Cadre et vocabulaire

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

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

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

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

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

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

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

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

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

Les contraintes peuvent prendre plusieurs formes distinctes :

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

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

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

On dit que le problème est *convexe* si U est convexe.

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

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

1.7.2 Définitions

Maximum, minimum :

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

Optimum local :

- La variable x^* est un *maximum* local d'une fonction f définie sur l'ensemble convexe S
 $\iff \exists \epsilon > 0$ tel que $f(x) \leq f(x^*) \quad \forall x \in S$ et $|x - x^*| \leq \epsilon.$
- La variable x^* est un *minimum* local d'une fonction f définie sur l'ensemble convexe S
 $\iff \exists \epsilon > 0$ tel que $f(x) \geq f(x^*) \quad \forall x \in S$ et $|x - x^*| \leq \epsilon.$

Optimum global :

- La variable x^* est un *maximum* global d'une fonction f définie sur l'ensemble convexe S
 $\iff \exists \epsilon > 0$ tel que $f(x) \leq f(x^*) \quad \forall x \in S$.
- La variable x^* est un *minimum* global d'une fonction f définie sur l'ensemble convexe S
 $\iff \exists \epsilon > 0$ tel que $f(x) \geq f(x^*) \quad \forall x \in S$.

Clairement **tout extremum global est aussi un extremum local**. La réciproque est évidemment fausse, comme le montre l'exemple de la figure 1.5.

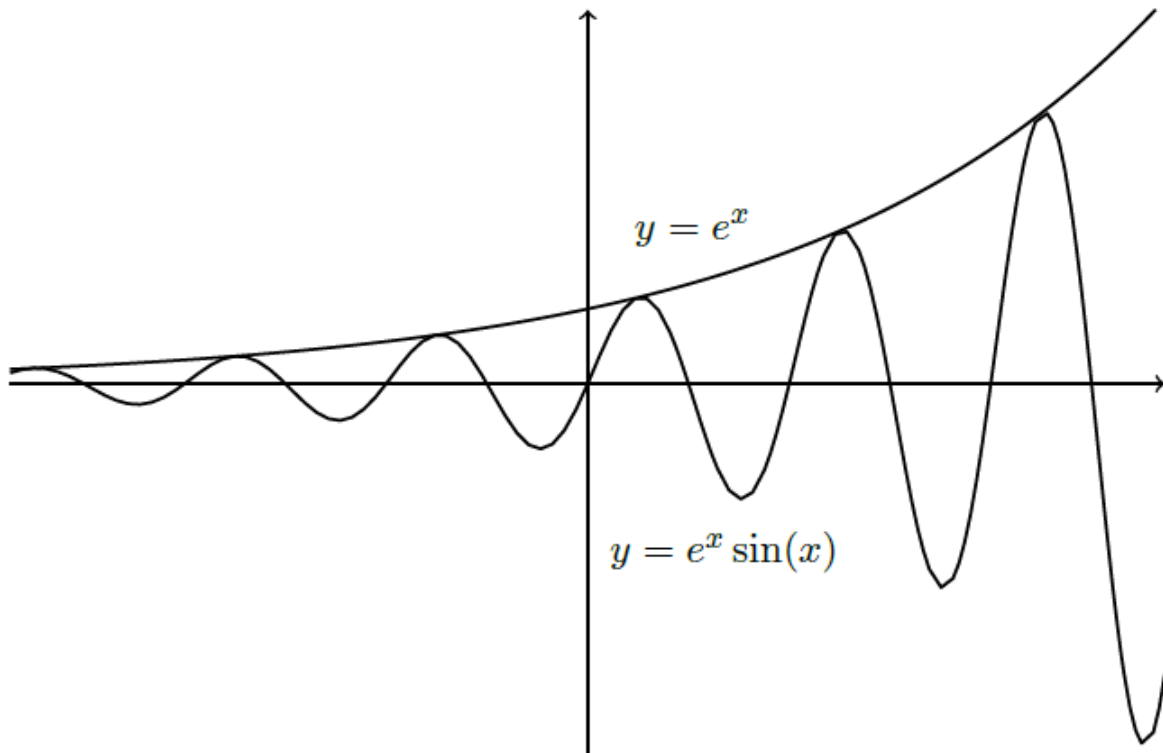


FIGURE 1.5 – L'application $x \rightarrow \exp(x) \sin(x)$ a une infinité de minima locaux (en $-\frac{\pi}{4}[2\pi]$) et de maxima locaux (en $-\frac{3\pi}{4}[2\pi]$) mais aucun extremum global.

Chapitre 2

Problèmes d'optimisation convexes

Définition générale

On rappelle qu'un problème d'optimisation

$$\left\{ \begin{array}{l} \text{Trouver } x^* \text{ tel que} \\ x^* \in U \text{ et } f(x^*) = \min_{x \in U} f(x) \end{array} \right. \quad (2.1)$$

est dit *convexe* si U et f sont convexes. Vis-à-vis de l'optimisation, la convexité joue un rôle crucial puisqu'elle permet d'assurer qu'un minimum local est en fait un minimum global, comme précisé par le résultat suivant.

Propositions 2.1. [2] *Soit $U \subset \mathbb{R}^n$ un ensemble convexe.*

- (a) *Si une fonction convexe $f : U \rightarrow \mathbb{R}$ admet un minimum local en un point x , elle y admet en fait un minimum global sur U .*
- (b) *Une fonction $f : U \rightarrow \mathbb{R}$ strictement convexe admet au plus un minimum local qui est en fait un minimum global strict.*

2.1 Problèmes d'optimisation sans contraintes

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

$$(\mathcal{P}) \left\{ \begin{array}{l} \min f(x) \\ x \in \mathbb{R}^n \end{array} \right.$$

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

2.1.1 Conditions d'existence et d'unicité de la solution

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

Définition 2.1 (Fonctions coercives). Une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$, est dite coercive si

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

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

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

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

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

Rappelons que parmi les normes ci-dessus, seule la norme ℓ_2 munit \mathbb{R}^n d'une structure d'espace de Hilbert. Nous noterons (\cdot, \cdot) le produit scalaire associé.

Exemple 2.1.

- $f(x) = \|x\|$ est coercive.
- Pour $n = 2$ et $x = (x_1, x_2)$, f définie par $J(x) = x_1^2 + x_2^2 - ax_1 - bx_2 - c$, est coercive pour tous réels a et b .
- Soit A une matrice carrée d'ordre n symétrique, définie positive et b un vecteur de \mathbb{R}^n . Alors f définie par

$$f(x) = \frac{1}{2}(Ax, x) - (b, x)$$

est coercive.

Théorème 2.1 (Existence). [2]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ propre, continue et coercive. Alors (\mathcal{P}) admet au moins une solution.

Le théorème suivant garantit l'existence d'une solution au problème de recherche de minimum.

Théorème 2.2 (Unicité). [2]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ strictement convexe. Alors le problème (\mathcal{P}) admet au plus une solution.

Donnons pour terminer un critère pour qu'une fonction soit strictement convexe et coercive :

Théorème 2.3. [2]

Soit f une fonction \mathcal{C}^1 de \mathbb{R}^n dans \mathbb{R} . On suppose qu'il existe $\alpha > 0$ tel que

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \quad \langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \alpha \|x - y\|^2 \quad (2.2)$$

Alors f est strictement convexe et coercive ; en particulier le problème (\mathcal{P}) admet une solution unique.

Il faut maintenant donner des conditions pour pouvoir calculer la (ou les) solutions. On va chercher à montrer que cette solution est solution de certaines équations, de sorte qu'il sera plus facile de la calculer.

2.1.2 Conditions d'optimalité

- **Condition nécessaire du premier ordre**

Dans cette sous section, on dérive les conditions pour un point x^* d'être un minimum local. étant donné un point $x^* \in \mathbb{R}$, on souhaite si c'est possible de déterminer un minimum local ou global pour la fonction f .

Pour cela on a besoin de caractériser la solution minimisante.

Théorème 2.4 (Direction de descente). [3]

supposons que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est différentiable en x^ . S'il existe un vecteur d tel que $\nabla f(x^*)^t d < 0$ et il existe un $\delta > 0$ tel que $f(x^* + \lambda d) < f(x^*)$ pour tout $\lambda \in (0, \delta)$, alors d est une direction de descente de la fonction f au point x^* .*

Théorème 2.5 (Condition nécessaire du premier ordre). [3]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction différentiable au point x^ . Si x^* est un minimum local, alors :*

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

- **Conditions nécessaires d'optimalité du deuxième ordre**

Théorème 2.6. [3]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois continûment différentiable dans au point x^ . Si x^* est un minimum local, alors :*

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

autrement dit la matrice hessienne $\nabla^2 f(x)$ est positive.

Comme le montre le résultat suivant, la dérivée seconde permet souvent de déterminer la nature d'un point critique c'est-à-dire de déterminer si un point critique est un bien minimum local, un maximum local, ou ni l'un ni l'autre.

- **Conditions suffisantes**

Dans Le théorème suivant nous donnons les conditions suffisantes pour l'optimalité local de point $x^* \in \mathbb{R}$

Théorème 2.7. [3]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction deux fois continûment différentiable en un point x^ . Si*

- $\nabla f(x^*) = 0$
- et*
- $\nabla^2 f(x^*) > 0$

alors x est un minimum local strict.

2.1.3 Conditions d'optimalité globale

les conditions décrites ci-dessus peuvent seulement garantir seulement une optimalité locale de points stationnaires puisque elles exploitent des informations locales : les valeurs du gradient et la matrice Hessienne au un point donné. Par contre, les conditions qui assurent l'optimalité globale doivent utiliser des informations globales. Par exemple, quand la matrice Hessienne d'une fonction est toujours semi définie positive, alors les points stationnaires sont aussi des minima globaux.

Théorème 2.8. [3]

Si f est deux fois continûment différentiable définie sur \mathbb{R}^n .

Supposons que $H_f(x^) \geq 0$ pour tout $x \in \mathbb{R}^n$, Soit $x^* \in \mathbb{R}^n$ un point stationnaire de f . Alors x^* est un minimum global de f*

Montrons maintenant que si la fonction objectif est continûment différentiable et convexe, alors la condition nécessaire du première ordre pour un minimum est aussi suffisante. Considérons le lemme suivant :

Lemme 2.1. *Soit $f : \Omega \rightarrow \mathbb{R}$ une fonction convexe définie sur un ensemble convexe $\Omega \subset \mathbb{R}^n$, et $f \in C^1$ sur un ensemble convexe ouvert contenant Ω . Supposons le point $x^* \in \Omega$ tel que pour tout $x \in \Omega$, $x \neq x^*$, on a :*

$$\nabla f(x^*)(x - x^*) \geq 0.$$

Alors x^ est un minimum globale de f sur Ω .*

En utilisant ce lemme, on a le théorème suivant :

Théorème 2.9. [3]

Soit $f : \Omega \rightarrow \mathbb{R}$ une fonction convexe définie sur un ensemble convexe $\Omega \subset \mathbb{R}^n$, et $f \in C^1$ sur un ensemble convexe ouvert contenant Ω . Supposons le point $x^ \in \Omega$ tel que pour toute direction admissible d au point x^* , on a :*

$$d^t \nabla f(x^*) \geq 0.$$

Alors x^ est un minimum global de f sur Ω .*

De ce théorème, on déduit le corollaire suivant :

Corollaire 2.1. *Soit $f : \Omega \rightarrow \mathbb{R}$, $f \in C^1$ une fonction convexe définie sur un ensemble convexe $\Omega \subset \mathbb{R}^n$. Supposons que le point $x^* \in \Omega$ est tel que :*

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

Alors x^ est un minimum globale de f sur Ω .*

2.2 Problèmes d'optimisation sous contraintes

Les problèmes de satisfaction de contraintes ou CSP (Constraint Satisfaction Problem) sont des problèmes mathématiques où l'on cherche des états ou des objets satisfaisant un certain nombre de contraintes ou de critères.

2.2.1 Problèmes d'optimisation sous contraintes égalités

1- Formulations du problème

Définition 2.2. *On définit un Problème d'optimisation sous contraintes égalité par*

$$(\mathcal{PE}) \quad \begin{cases} \min f(x) \\ \text{s.c} \\ h(x) = 0 \\ x \in \mathbb{R}^n \end{cases}$$

Où $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ et $h = [h_1, h_2, \dots, h_m]^t$ avec $m \leq n$. On suppose que la fonction h est continûment différentiable, i.e $g \in C^1$.

Définition 2.3. (Point régulier) Un point x^* satisfaisant les contraintes $h_1(x^*) = 0, h_2(x^*) = 0, \dots, h_m(x^*) = 0$ est dit un point régulier des contraintes si les vecteurs gradients $\nabla h_1(x^*) = 0, \nabla h_2(x^*) = 0, \dots, \nabla h_m(x^*) = 0$ sont linéairement indépendants.

Soit $Dh(x^*)$ la matrice Jacobienne de $h = [h_1, h_2, \dots, h_m]^t$ au point x^* , donnée par :

$$Dh(x^*) = \begin{pmatrix} Dh_1(x^*) \\ Dh_2(x^*) \\ \vdots \\ Dh_m(x^*) \end{pmatrix} = \begin{pmatrix} \nabla h_1(x^*)^t \\ \nabla h_2(x^*)^t \\ \vdots \\ \nabla h_m(x^*)^t \end{pmatrix}$$

Alors x^* est régulier si seulement si $\text{rang } Dh(x^*) = m$, c-à-d, la matrice Jacobienne est de rang plein.

L'ensemble des contraintes égalités $h_1(x^*) = 0, h_2(x^*) = 0, \dots, h_m(x^*) = 0$, $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, décrit une surface

$$S = \{x \in \mathbb{R}^n : h_1(x^*) = 0, h_2(x^*) = 0, \dots, h_m(x^*) = 0\}.$$

Supposons que les points dans S sont réguliers, alors la dimension de S est $n - m$.

2.3 Espace tangent et espace normal

Définition 2.4. Une courbe C sur une surface S est l'ensemble de points $\{x(t) \in S : t \in (a, b)\}$ continûment paramétrisée par $t \in (a, b)$; i.e : $x : (a, b) \rightarrow S$ est une fonction continue.

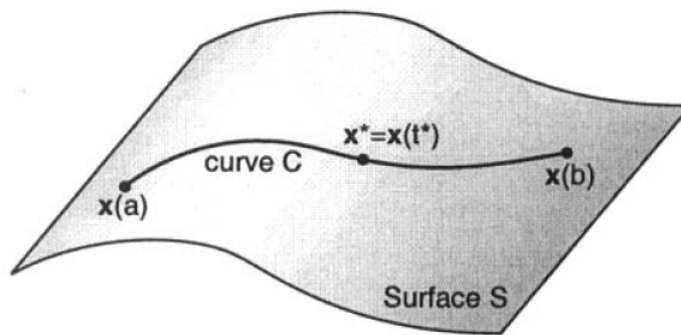


FIGURE 2.1 – Courbe sur une surface.

Intuitivement, une courbe $C = \{x(t) : t \in (a, b)\}$ est un chemin traversé par un point x se déplaçant sur une surface S . La position d'un point au temps t est donné par $x(t)$.

Définition 2.5. L'espace tangent au point x^* sur la surface $S = \{x \in \mathbb{R}^n : h(x) = 0\}$ est l'ensemble

$$T(x^*) = \{y : Dh(x^*)y = 0\}.$$

Notons que l'espace tangent $T(x^*)$ est le noyau de la matrice $Dh(x^*)$, i.e ;

$$T(x^*) = N(Dh(x^*)).$$

Supposons que x^* est régulier, la dimension de l'espace tangent est $n - m$, où c 'est le nombre de contraintes égalité $h_i(x) = 0$. Notons que l'espace tangent passe par l'origine. Cependant, il souvent convenable de représenter l'espace tangent comme un plan qui passe par le point x^* .

Pour cela, on définit le plan tangent au point x^* comme l'ensemble.

$$TP(x^*) = T(x^*) + x^* = \{x + x^*, x \in T(x^*)\}.$$

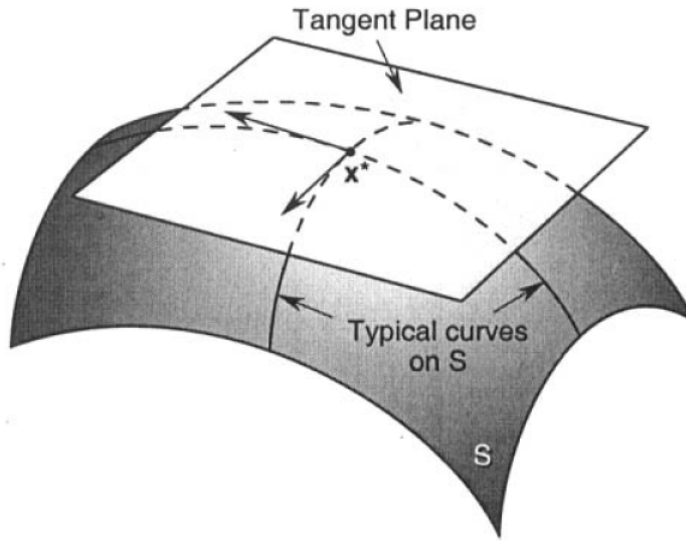


FIGURE 2.2 – Le plan tangent à la surface S au point x^* .

Définition 2.6. L'espace normal $N(x^*)$ au point x^* sur la surface $S = \{x \in \mathbb{R}^n : h(x) = 0\}$ est l'ensemble .

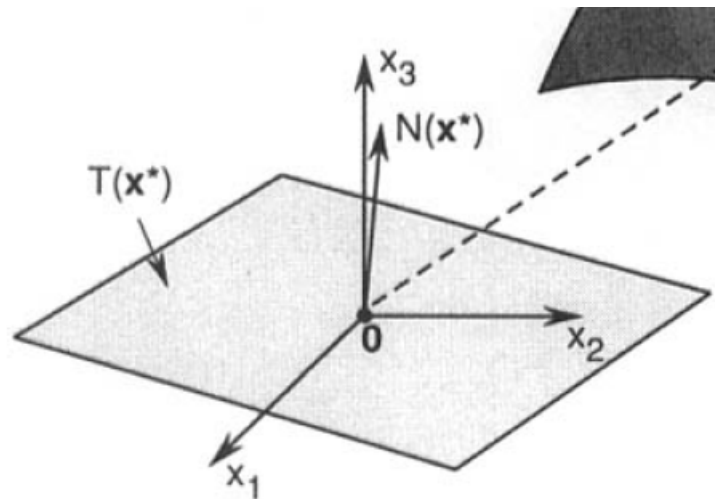
$$N(x^*) = \{x \in \mathbb{R}^n : x \in Dh(x^*)^t z, z \in \mathbb{R}^m\}.$$

$$= \{x \in \mathbb{R}^n : x = z_1 \nabla h_1(x^*) + \dots + z_m \nabla h_m(x^*), z_1, \dots, z_m \in \mathbb{R}\}$$

Notons que l'espace normal contient le vecteur zéro. Supposons que x^* est régulier, la dimension de l'espace normal $N(x^*)$ est m , comme dans le cas de l'espace tangent, il est souvent convenable de représenter l'espace normal $N(x^*)$ passant par le point x^* . Pour cela, on définit le plan normal au point x^* comme l'ensemble

$$NP(x^*) = N(x^*) + x^* = \{x + x^* \in \mathbb{R}^n : x \in N(x^*)\}$$

Lemme 2.2. On a $T(x^*) = N(x^*)^\perp$ et $T(x^*)^\perp = N(x^*)$


 FIGURE 2.3 – Espace normale dans \mathbb{R}^3 .

2- Le lagrangien (fonction de Lagrange)

Définition 2.7. On appelle la fonction Lagrange (le Lagrangien) du problème (P) la fonction L définie par :

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i h_i(x)$$

- Le vecteur $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \in \mathbb{R}^m$ est appelé le multiplicateur de Lagrange.
- Les λ_i indique un forme de l'influence de la contrainte dans le coût de la solution.

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i \nabla h_i(x^*) = 0$$

il y a deux façons d'interpréter cette équation :

- le gradient de coût $\nabla f(x^*)$ étant le sous-espace recouvert par les contraintes gradients à x^* .
- le gradient de coût $\nabla f(x^*)$ est orthogonal au sous-espace de variations réalisables de premier ordre.

$$v(x^*) = \{\Delta x : \nabla h_i(x)^t \Delta x = 0, i = 1, \dots, m\}.$$

3- Conditions de Lagrange

• Condition nécessaire du première ordre

Théorème 2.10. (Théorème de Lagrange). [3]

Soit x^* un minimum locale (ou maximum locale) de $f : \mathbb{R}^n \rightarrow \mathbb{R}$, sujet à $h(x) = 0$, tel que $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, et $m \leq n$. Suppose que x^* est un point régulier. Alors, il existe $\lambda^* \in \mathbb{R}^m$ tel que :

$$\nabla f(x^*) + \lambda^{*t} \nabla h(x^*) = 0$$

• Condition second ordre

Théorème 2.11. [3]

On suppose que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ sont deux fois continument différentiable, i.e ; $f, h \in C^2$. soit

$$l(x, \lambda) = f(x) + \lambda^t h(x) = f(x) + \lambda_1^t h_1(x) + \lambda_2^t h_2(x) + \dots + \lambda_m^t h_m(x)$$

la fonction Lagrange.

Soit $L(x, \lambda)$ la matrice Hessienne de $l(x, \lambda)$ par rapport à x , i.e ;

$$L(x, \lambda) = F(x) + \lambda^t H(x) = F(x) + \lambda_1^t H_1(x) + \lambda_2^t H_2(x) + \dots + \lambda_m^t H_m(x)$$

Où $F(x)$ la matrice Hessienne de f en point x .

$H_k(x)$ est la matrice Hessienne de h_k en point x .

• **Condition nécessaire du deuxième ordre**

Théorème 2.12. [3]

Supposons que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, avec $i = 1, \dots, m$ sont deux fois continûment différentiables. Soit

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \nabla h_i(x)$$

la fonction de Lagrange, et soit

$$\nabla_{xx}^2 L(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x)$$

la matrice Hessienne de $L(x, \lambda)$.

Théorème 2.13 (Condition nécessaire du deuxième ordre). [3]

Soit x^* un minimum local de la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ dans les contraintes $h_i(x) = 0$ avec $i=1, \dots, m$ et $m \leq n$ et $f, h_i \in C^2$. Supposons que x^* est régulier. Alors, il existe $\lambda^* \in \mathbb{R}^m$ tel que :

- $\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) = 0$.
- Pour tout $y \in T(x^*)$, on a $y^t \nabla_{xx}^2 L(x^*, \lambda^*) y \geq 0$.

• **Condition suffisante de deuxième ordre**

La condition nécessaire du première ordre peut être satisfaite pour un minimum local et pour un maximum local (et possible d'autres points critiques). Pour garantir qu'un point stationnaire donné est un minimum local, on a besoin d'une condition suffisante d'optimalité qui est donnée par le théorème suivant.

Théorème 2.14. [3]

Supposons que f et g sont deux fois continûment différentiables, et soit $x^* \in \mathbb{R}^n$ et $\lambda \in \mathbb{R}^m$ satisfait :

- $\nabla_x L(x^*, \lambda^*) = 0, \nabla_\lambda L(x^*, \lambda^*) = 0$.
 - $y^t \nabla_{xx}^2 L(x^*, \lambda^*) y > 0$, pour tout $y \neq 0$ avec $\nabla h(x^*)^t y = 0$
- Alors x^* est un minimum local strict de $f(x)$ sujet à $h(x) = 0$.

Supposons que l'ensemble admissible est convexe. le théorème suivant indique si l'ensemble admissible est convexe, la condition nécessaire de Lagrange devient suffisante pour un minimum.

Théorème 2.15. [3]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^1$, est une fonction convexe sur un ensemble admissible de points

$$\Omega = \{x \in \mathbb{R}^n | h(x) = 0\},$$

Où $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $h \in C^1$, et Ω est convexe. Supposons qu'il existe $x^* \in \Omega$ et $\lambda^* \in \mathbb{R}^m$ tel que :

$$\nabla f(x^*) + \lambda^{*t} \nabla h(x^*) = 0.$$

Alors x^* est un minimum globale de f sur Ω .

2.3.1 Problème d'optimisation avec contraintes inégalités

Définition 2.8. : On définit un problème d'optimisation avec contraintes inégalités par :

$$(\mathcal{PI}) \quad \begin{cases} \min f(x) \\ \text{s.c} \\ g_i(x) \leq 0, \quad i = 1, \dots, m \\ x \in \mathbb{R}^n \end{cases}$$

Où $f, g_i (i = 1, \dots, m)$ sont des fonctions continument différentiables sur \mathbb{R}^n .

Définition 2.9 (Contraintes actives). La contrainte $g_i(x) \leq 0$ est dite active ou saturée en x^0 si :

$$g_i(x^0) = 0.$$

On note l'ensemble des contraintes actives en x^0 par : $E = \{i : g_i(x^0) = 0\}$.

Définition 2.10 (Point régulier). Un point x^0 admissible c-à-d $g_i(x) \leq 0, \forall i = 1, \dots, m$ est dit régulier si $\text{rang}(JG_E(x^0)) = |E|$, (JG_E jacobienne associée à l'ensemble E).

Avec $G = (g_1, \dots, g_m)$

Autrement dit si les vecteurs gradients des contraintes active $\{\nabla g_i(x)\}_{i \in E}$ sont linéairement indépendants.

• Condition nécessaire du premier ordre

Le théorème de Karush-Kuhn-Tucker (KKT) suivant nous donne une condition nécessaire du première ordre pour qu'un point x^* soit un minimum local pour le problème \mathcal{PI} .

Théorème 2.16. [3]

Soit x^* un minimum local pour le problème

$$(\mathcal{PI}) \quad \begin{cases} \min f(x) \\ \text{s.c} \\ g_i(x) \leq 0, \quad i = 1, \dots, m \\ x \in \mathbb{R}^n \end{cases}$$

Où $f, g_i (i = 1, \dots, m)$ sont des fonctions continument différentiables sur \mathbb{R}^n .

Soit $E = \{i : g_i(x^*) = 0\}$ l'ensemble des contraintes actives. Supposons que les gradients des contraintes actives $\{g_i(x^*)\}_{i \in E}$ sont linéairement indépendants. Alors ils existent des multiplicateurs de Lagrange $\lambda_1, \lambda_2, \dots, \lambda_m \geq 0$ tel que :

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

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

2.3.2 Problème d'optimisation avec contraintes égalités et inégalités

Définition 2.11. : On définit un problème d'optimisation avec contraintes égalités et inégalités par :

$$(\mathcal{PIE}) \quad \begin{cases} \min f(x) \\ s.c \\ h_j(x) = 0, \quad j = 1, \dots, p \\ g_i(x) \leq 0, \quad i = 1, \dots, m \\ x \in \mathbb{R}^n \end{cases}$$

Où $f, h_j (j = 1, \dots, p), g_i (i = 1, \dots, m)$ sont des fonctions continument différentiables sur \mathbb{R}^n .

• **Condition d'optimalité du première ordre**

Théorème 2.17. [3]

Soit $f, h_j (j = 1, \dots, p), g_i (i = 1, \dots, m)$ sont des fonctions continument différentiables sur \mathbb{R}^n . Soit x^* un minimum local régulier pour le problème (\mathcal{PIE}) .

Alors il existe $\lambda_1^*, \lambda_2^*, \dots, \lambda_m^* \geq 0$ et $\mu_1^*, \mu_2^*, \dots, \mu_p^* \in \mathbb{R}$ tel que

$$\begin{aligned} \nabla f(x^*) + \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* \nabla g_i(x^*) &= 0, \quad i = 1, \dots, m \end{aligned}$$

dans le théorème ci-dessus, les multiplicateurs $\lambda_i^* (i=1, \dots, m)$ sont appelés multiplicateurs de KKT et $\mu_j^* (j=1, \dots, p)$ sont appelés multiplicateurs de Lagrange.

Définition 2.12. (Point de Karush-Kuhn-Tucker (KKT)) Considérons le problème de minimisation (PIE) , où $f, h_j (j = 1, \dots, p), g_i (i = 1, \dots, m)$ sont des fonctions continument différentiables sur \mathbb{R}^n .

Un point admissible x^* est appelé point de KKT s'ils existent $\lambda_1, \dots, \lambda_m \geq 0$ et $\mu_1, \dots, \mu_p \in \mathbb{R}$ tel que :

$$\begin{aligned} \nabla f(x^*) + \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* \nabla g_i(x^*) &= 0, \quad i = 1, \dots, m \end{aligned}$$

• **Conditions d'optimalité du deuxième ordre**

Dans cette sous section, on donne les conditions nécessaires et suffisantes pour les extremas des problèmes d'optimisation avec contraintes égalités et inégalités. Pour cela, on a besoin de définir la matrice suivante :

$$H_l(x, \lambda, \mu) = F(x) + \lambda G(x) + \mu H(x).$$

Où $F(x)$ est la matrice Hessienne associée à f au point x .

$$\lambda G(x) = \lambda_1 G_1(x) + \lambda_2 G_2(x) + \dots + \lambda_m G_m(x).$$

$$\mu H(x) = \mu_1 H_1(x) + \mu_2 H_2(x) + \dots + \mu_p H_p(x).$$

Où $G_k(x)$ est la matrice Hessienne de $g_k(x)$ donnée par

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

Et $H_k(x)$ est la matrice Hessienne de $h_k(x)$ donnée par

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

Dans le théorème qui suit, on utilise

$$T(x^*) = \{y \in \mathbb{R}^n \mid J_h(x^*)y = 0, J_{g_j}(x^*)y = 0, j \in J(x^*)\}$$

c-à-d l'espace tangent à la surface défini par contraintes actives.

• **Conditions nécessaires du deuxième ordre**

Théorème 2.18. [3]

Soit x^* est un minimum local de la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ sous les contraintes $h(x) = 0$ et $g(x) \leq 0$, où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$ avec $p \leq n$ et $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ avec $m \leq n$ et $f, g, h \in C^2$.

supposons que x^* est régulier. Alors il existe $\lambda^* \in \mathbb{R}^m, \mu^* \in \mathbb{R}^p$ tel que :

1. $\lambda^* \geq 0, \nabla f(x^*) + \mu^{*t} \nabla h(x^*) + \lambda^{*t} \nabla g(x^*) = 0$
 $\lambda^{*t} g(x^*) = 0.$
2. pour tout $y \in \mathbb{R}^n$, on a : $y^t l(x^*, \lambda^*, \mu^*) y \geq 0.$

• **Conditions suffisantes du deuxième ordre**

pour formuler les conditions suffisantes du deuxième ordre pour un extremum du problème avec contraintes inégalités, on utilise l'ensemble suivant dans la formulation du résultat :

$$\bar{T}(x^*, \lambda^*) = \{y : J_h(x^*)y = 0, \nabla g_i(x^*)y = 0, i \in \bar{J}(x^*, \lambda^*)\}$$

Où $\bar{J}(x^*, \lambda^*) = \{i : g_i(x^*) = 0, \lambda_i^* > 0\}.$

Théorème 2.19. [3]

Supposons que $f, g, h \in C^2$ et il existe un point admissible $x^* \in \mathbb{R}^n$ et deux vecteurs $\lambda^* \in \mathbb{R}^m$ et $\mu^* \in \mathbb{R}^p$, tel que :

1. $\lambda^* \geq 0, \nabla f(x^*) + \sum_{j=1}^p \mu_j \nabla h_j(x^*) + \sum_{i=1}^m \lambda_i \nabla g_i(x^*) = 0$
 $\lambda_i^* g_i(x^*) = 0, i = 1, \dots, m.$
2. Pour tout $y \in \bar{T}(x^*, \lambda^*), y \neq 0$, on a : $y^t H_l(x^*, \lambda^*, \mu^*) y > 0.$

Alors, x^* est un minimum local (strict) pour la fonction f sous les contraintes $h(x) = 0$ et $g(x) \leq 0$.

Considérons maintenant le problème suivant, dans le cas convexe

$$(PIE) \begin{cases} \min f(x) \\ s.c \\ h(x) = 0 \\ g(x) \leq 0 \\ x \in \mathbb{R}^n \end{cases}$$

On suppose que l'ensemble admissible est convexe. c'est le cas où, les deux ensembles $\{x : h(x) = 0\}$ et $\{x : g(x) \leq 0\}$ sont convexes.

Théorème 2.20. [3]

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f \in C^1$, une fonction convexe sur l'ensemble admissible

$$\Omega = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \leq 0\}$$

Où $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $h, g \in C^1$ et Ω est convexe.

Supposons qu'il existent $x^* \in \Omega$, $\lambda^* \in \mathbb{R}^m$, $\Omega^* \in \mathbb{R}^p$, tel que :

1. $\lambda^* \geq 0$.
2. $\nabla f(x^*) + \mu^{*t} \nabla h(x^*) + \lambda^{*t} \nabla g(x^*) = 0$.
3. $\lambda^{*t} g(x^*) = 0$.

Alors, x^* est un minimum globale de f sur Ω .

2.3.3 Problèmes d'optimisation linéaires

Un problème d'optimisation linéaire demande de minimiser une fonction linéaire sur un polyèdre convexe. La fonction que l'on minimise ainsi que les contraintes sont décrites par des fonctions linéaires, d'où le nom donné à ces problèmes.

Elle est également désignée par le nom de *programmation linéaire* (Dantzig 1947), mais cette appellation tend à être abandonnée à cause de la confusion possible avec la notion de *programmation informatique*.

Formulations du problème

On peut représenter un polyèdre convexe de différentes manières. Lorsqu'on le voit comme ci-dessus, à savoir comme une intersection d'un nombre fini de demi-espaces :

$$\{x \in \mathbb{R}^n : Ax \leq b\}$$

On dit que le polyèdre (ou le problème d'optimisation linéaire avec un tel polyèdre) est écrit sous **forme canonique**. Lorsque le polyèdre convexe est vu comme l'intersection de l'orthant positif et d'un sous-espace affine :

$$\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$$

On dit que le polyèdre (ou le problème d'optimisation linéaire avec un tel polyèdre) est écrit sous **forme standard**.

Les analyses du problème d'optimisation linéaire se font le plus souvent sur le problème dont l'ensemble admissible est représenté sous la forme standard, problème que nous écrirons comme suit :

$$(P_L) \begin{cases} \max c^t x \\ Ax = b \\ x \geq 0 \end{cases}, \quad (P_L) \begin{cases} \min c^t x \\ Ax = b \\ x \geq 0 \end{cases}$$

L'ensemble des solutions peut être vide, et peut être infini

Sommet d'un polyèdre convexe sous forme standard

Soient $P := \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ un polyèdre convexe non vide et $x \in P$. Alors les propriétés suivantes sont équivalentes :

1. x est un sommet de P .
2. les colonnes $\{A_j \in \mathbb{R}^m : x_j > 0\}$ de A sont linéairement indépendantes.

De plus P a au moins un sommet.

Exemple 2.2. Une firme a le projet de construire deux produits nécessitant l'utilisation de 3 machines. la machine A ne peut travailler que 150 heures par mois, la machine B, 210 heures, et la machine C, 180 heures. Le premier produit P1 nécessite 1 heure de la machine A et 1 heure de la machine B et il est vendu 300 euros à l'unité. Le second produit P2 nécessite une heure de la machine A, trois heures de la machine B et 3 heures de la machine C et il est vendu 500 euros à l'unité. La firme cherche à définir le programme de fabrication lui permettant de rendre maximal son prix de revient.

Une modélisation mathématique conduit à appeler x le nombre de produits P1 à fabriquer et y le nombre de produits P2.

Les contraintes

Les contraintes de fabrication s'expriment à l'aide des inégalités suivantes :

$$\begin{array}{ll} x + y \leq 150 & \text{(contraintes de fabrication de la machine A)} \\ x + 3y \leq 210 & \text{(contraintes de fabrication de la machine B)} \\ 3y \leq 180 & \text{(contraintes de fabrication de la machine C)} \\ x \geq 0, y \geq 0 & \text{(contraintes logiques)} \end{array}$$

La fonction objectif

Sur cet ensemble admissible des contraintes, on définit la fonction revenu net f que l'on cherche à rendre maximale, où :

$$f(x, y) = 300x + 500y$$

Le modèle mathématique

$$\begin{array}{l} (P_L) \quad \{ \text{Max } f(x, y) = 300x + 500y \\ \quad \quad \quad x + y \leq 150 \\ \text{s.c} \quad \quad x + 3y \leq 210 \\ \quad \quad \quad 3y \leq 180 \\ \quad \quad \quad x \geq 0, y \geq 0 \end{array}$$

Chacune de ces contraintes définit un demi-plan auquel le point M de coordonnées (x, y) doit appartenir. L'intersection de ces demi-plans dessine un polygone convexe (OABCD) appelé ensemble admissible.

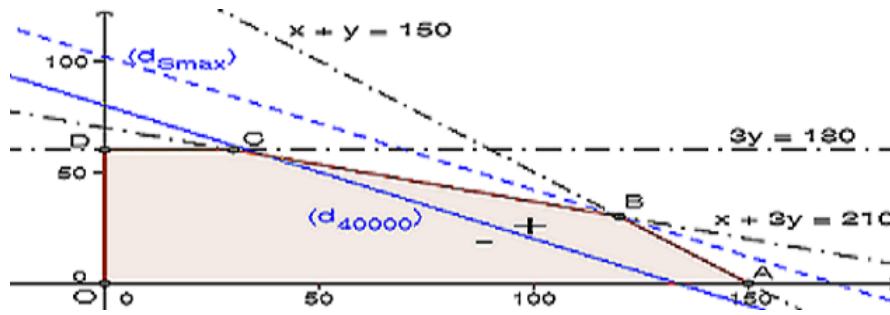


FIGURE 2.4 – Exemple pratique d'un problème d'optimisation linéaire

2.3.4 Conditions d'existence et d'unicité de solution

Conditions d'existence :

Il y a exactement deux cas (exclusifs) dans lesquels le problème d'optimisation linéaire n'a pas de solution.

- Le premier est celui où les contraintes ne sont pas compatibles (par exemple $x \geq 2$ et $x \leq 1$). La valeur optimale du problème de minimisation vaut alors $+\infty$, par convention. Dans ce cas, on dit que le problème n'est pas **réalisable**.
- Le second se produit lorsque le problème de minimisation est réalisable mais que sa valeur optimale vaut $-\infty$ (par exemple lorsqu'on cherche à minimiser x sous la contrainte $x \leq 0$). Dans ce cas, on dit que le problème n'est pas borné ou est **n'est pas borné**.

Dans tous les autres cas, la valeur optimale du problème d'optimisation linéaire est finie et le problème a une solution.

Théorème 2.21. *Si le problème (P_L) a une solution, il a une solution en un sommet de son ensemble admissible.*

Ce résultat est important pour *l'algorithme du simplexe*, car celui-ci, générant ses itérés sur des sommets, cherche une solution-sommet (qui doit donc exister pour qu'il en trouve une !).

Unicité de la solution optimale :

- ★ Si dans le tableau optimal de (P) (Algorithme du Simplexe), on a pour toute variable x_j hors base $\Delta_j < 0$, alors la solution optimale est unique.
- ★ Sinon la solution optimale n'est pas unique.

2.3.5 Conditions d'optimalité

Les conditions d'optimalité du problème d'optimisation linéaire (P_L) peuvent être obtenues comme cas particulier de la théorie générale des problèmes d'optimisation différentiables en dimension finie (conditions de *Karush, Kuhn et Tucker*), avec la simplification supplémentaire de ne pas avoir à s'occuper de la qualification des contraintes du problème.

Grâce à la convexité du problème OL , les conditions énoncées ci-dessous sont nécessaires et suffisantes à l'optimalité :

Conditions d'optimalité

Le point $x \in \mathbb{R}^n$ est solution de (P_L) si et seulement si, il existe deux vecteurs $y \in \mathbb{R}^m$ et $s \in \mathbb{R}^n$ tels que

$$\begin{cases} (a) & A^t y + s = c, \quad s \geq 0 \\ (b) & Ax = b, \quad x \geq 0 \\ (c) & x^t s = 0 \end{cases}$$

Les vecteurs y et s introduites par ces conditions d'optimalité portent le nom de **multiplicateurs de Lagrange** ou de **variables duales**.

Les relations (a) expriment l'admissibilité duale et la première de ces relations est le gradient en x du **Lagrangien** du problème, qui est la fonction

$$\ell : (x, y, s) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \longmapsto \ell(x, y, s) = c^t x - y^t (Ax - b) - s^t x.$$

Les relations (b) expriment l'admissibilité primale et la relation (c) exprime la complémentarité existant entre les variables primales x et leurs multiplicateurs s : x_i ou s_i est nul (ou les deux).

2.4 Problèmes d'optimisation quadratiques

Un problème d'optimisation quadratique est un problème d'optimisation dans lequel on minimise (ou maximise) une fonction quadratique sur un polyèdre convexe. Les contraintes peuvent donc être décrites par des fonctions linéaires. L'optimisation linéaire peut être vue comme un cas particulier de l'optimisation quadratique.

2.4.1 Formulation du problème

Un problème d'optimisation quadratique consiste à minimiser ou maximiser une fonction quadratique $q : \mathbb{R}^n \rightarrow \mathbb{R}$, non nécessairement convexe, sur un polyèdre convexe.

Le critère à minimiser

La fonction q est définie en $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ par

$$q(x) = c^t x + \frac{1}{2} x^t H x = \sum_{i=1}^n c_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n H_{ij} x_i x_j$$

Dans la première expression de $q(x)$, l'expression sous forme matricielle, $c \in \mathbb{R}^n$ est un vecteur et $H \in \mathbb{R}^{n \times n}$ est une matrice réelle symétrique (celle-ci est généralement supposée symétrique, car q ne voit pas la partie antisymétrique éventuelle de H). Il ne sert à rien de garder le terme constant dans q , car celui-ci n'affecte pas la solution du problème de minimisation.

Rappelons qu'une telle fonction quadratique q est

- Convexe si, et seulement si, la matrice H est semi-définie positive.
- Strictement convexe si, et seulement si, la matrice H est définie positive.

Les contraintes

On impose également au vecteur recherché x d'appartenir à un polyèdre convexe, ce qui revient à dire que x doit vérifier un nombre fini de contraintes affines. Celles-ci peuvent prendre des formes variées comme

$$\begin{aligned} l_B &\leq x \leq u_B && \text{(contraintes de borne)} \\ l_I &\leq A_I x \leq u_I && \text{(contraintes d'inégalité affines)} \\ A_E x &= b_E && \text{(contraintes d'égalité affines),} \end{aligned}$$

expressions dans lesquelles on a noté

- l_B et u_B des vecteurs de \mathbb{R}^n pouvant donc prendre des valeurs infinies et vérifiant $l_B < u_B$.
- $A_I \in \mathbb{R}^{m_I \times n}$ une matrice réelle de type $m_I \times n$ ($A_I x$ désigne le produit de la matrice A_I par le vecteur x).
- l_I et u_I des vecteurs de \mathbb{R}^{m_I} pouvant donc prendre des valeurs infinies et vérifiant $l_I < u_I$.
- $A_E \in \mathbb{R}^{m_E \times n}$ une matrice réelle de type $m_E \times n$.
- $b_E \in \mathbb{R}^{m_E}$ un vecteur réel.

Il sera intéressant d'utiliser la notation compacte

$$[l_B, u_B] := \{x \in \mathbb{R}^n : l_B \leq x \leq u_B\}$$

et une définition similaire pour $[l_I, u_I]$. On note X_Q l'ensemble admissible défini par toutes les contraintes ci-dessus, à savoir

$$X_Q := \{x \in \mathbb{R}^n : x \in [l_B, u_B], A_I x \in [l_I, u_I], A_E x = b_E\}$$

Formulation compacte

De manière compacte, on peut donc écrire le problème d'optimisation quadratique de la manière suivante

$$(P_Q) \quad \min_{x \in X_Q} q(x)$$

On dit que ce problème est convexe si le critère q est convexe, ce qui est le cas si et seulement si H est semi-définie positive.

2.4.2 Conditions d'existence et d'unicité de solution

Conditions d'existence :

Le résultat fondamental est dû à *Frank* et *Wolfe*. Il est du même type que celui connu en optimisation linéaire. Rappelons que

- Un problème d'optimisation est dit réalisable si son ensemble admissible est non vide (ce qui revient à dire que sa valeur optimale ne vaut pas $+\infty$)
- Un problème d'optimisation réalisable est dit borné si sa valeur optimale ne vaut pas $-\infty$ (on ne peut pas trouver une suite de points admissibles faisant tendre le critère vers $-\infty$)

L'étude des problèmes quadratiques convexes constitue un domaine propre de la théorie de la programmation quadratique ; le résultat le plus remarquable est le suivant :

Théorème 2.22. [3] *Tout problème quadratique convexe dont la valeur optimal est finie admet au moins une solution.*

Pour un problème d'optimisation quadratique (non nécessairement convexe), les propriétés suivantes sont équivalentes :

- le problème a une solution.
- le problème est réalisable et borné.
- la valeur optimale du problème est finie.

Unicité de solution :

L'unicité de la solution aura certainement lieu si q est strictement convexe, mais pourra se produire sans cela. C'est le cas par exemple pour le problème à une unique variable $\inf\{x : x \geq 0\}$, dont le critère est linéaire (donc pas strictement convexe).

2.4.3 Conditions d'optimalité

La méthode du simplexe peut être adaptée pour résoudre un problème quadratique convexe.

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^t H x + c^t x \quad \text{sous} \quad \begin{cases} Ax - b \geq 0 \\ x \geq 0 \end{cases} \quad \text{avec} \quad \begin{matrix} H \in \mathbb{R}^{n \times n} , c \in \mathbb{R}^n \\ A \in \mathbb{R}^{m \times n} , b \in \mathbb{R}^m \end{matrix}$$

La méthode de Lagrangien est aussi peut être adaptée pour résoudre ce problème.

$$L(x, \mu, \lambda) = \frac{1}{2} x^t H x + c^t x - \mu^t (Ax - b) - \lambda^t x$$

Avec des multiplicateurs positifs μ et λ pour les contraintes inégalité.

Conditions suffisantes d'optimalité

$$\begin{aligned} \text{— 1}^{\text{er}} \text{ordre :} & \quad \begin{cases} Hx + c - A^t \mu - \lambda = 0 \\ Ax - b \geq 0 \\ x \geq 0 \\ \mu, \lambda \geq 0 \\ \lambda_i x_i = 0 \quad 1 \leq i \leq n \\ \mu_j (Ax - b)_j = 0 \quad 1 \leq j \leq m \end{cases} \quad \implies \quad \left(\begin{array}{c} \text{conditions de} \\ \text{complémentarité} \end{array} \right) \\ \text{— 2}^{\text{ème}} \text{ordre : } H \text{ définie positive (problème convexe)} & \quad \implies \quad \left(\begin{array}{c} \text{condition} \\ \text{suffisante} \end{array} \right) \end{aligned}$$

Chapitre 3

Méthodes de résolution

Dans ce chapitre, nous allons présenter quelques algorithmes permettant de calculer (de manière approchée) la ou les solutions du problème (\mathcal{P}) de départ. Bien entendu, nous ne pouvons pas être exhaustifs ; nous présentons les méthodes "de base" les plus classiques. Toutefois, la plupart de ces algorithmes exploitent les conditions d'optimalité dont on a vu qu'elles permettaient (au mieux) de déterminer des minima locaux. La question de la détermination de minima globaux est difficile et dépasse le cadre que nous nous sommes fixés. Néanmoins, nous décrirons dans la chapitre suivante, un algorithme probabiliste permettant de "déterminer" un minimum global.

Remarquons aussi que nous avons fait l'hypothèse de différentiabilité de la fonction J . Il existe des méthodes permettant de traiter le cas non différentiable (ou non régulier). Nous n'en parlerons pas ici. Nous commencerons par quelques définitions :

Définition 3.1 (Algorithme). *Un algorithme est défini par une application \mathcal{A} de \mathbb{R}^n permettant la génération d'une suite d'éléments de \mathbb{R}^n par la formule :*

$$\begin{cases} x_0 \in \mathbb{R}^n \text{ donné, } k = 0 & \text{Etape d'initialisation} \\ x_{k+1} = \mathcal{A}(x_k), \quad k = k + 1 & \text{Itération } k. \end{cases} \quad (3.1)$$

Ecrire un algorithme n'est ni plus ni moins que se donner une suite $(x_k)_{k \in \mathbb{N}}$ de \mathbb{R}^n ; étudier la convergence de l'algorithme, c'est étudier la convergence de la suite $(x_k)_{k \in \mathbb{N}}$.

Définition 3.2 (Convergence d'un algorithme). *On dit que l'algorithme \mathcal{A} converge si la suite $(x_k)_{k \in \mathbb{N}}$ engendrée par l'algorithme converge vers une limite x^* .*

Il est bien entendu très important d'assurer la convergence d'un algorithme, mais la vitesse de convergence et la complexité sont aussi des facteurs à prendre en compte lors de l'utilisation (ou de la génération) d'un algorithme ; on a en effet "intérêt" à ce que la méthode soit la plus rapide possible tout en restant précise et stable. Un critère de mesure de la vitesse (ou taux) de convergence est l'évolution de l'erreur commise ($e_k = \|x_k - x^*\|$)

Définition 3.3 (Taux de convergence d'un algorithme). *Soit $(x_k)_{k \in \mathbb{N}}$ une suite de limite x^* définie par la donnée d'un algorithme convergent \mathcal{A} .*

On dit que la convergence de \mathcal{A} est :

- **linéaire** si l'erreur ($e_k = \|x_k - x^*\|$) décroît linéairement :

$$\exists C \in]0, 1[, \exists k_0, \forall k \geq k_0 \quad e_{k+1} \leq C e_k \quad (3.2)$$

- **super-linéaire** si l'erreur e_k décroît de la manière suivante :

$$e_{k+1} \leq \alpha_k e_k \quad (3.3)$$

où α_k est une suite positive convergente vers 0. Si α_k est une suite géométrique, la convergence de l'algorithme est dite géométrique.

– **d'ordre** p si l'erreur e_k décroît de la manière suivante :

$$\exists C \geq 0, \exists k_0, \forall k \geq k_0 \quad e_{k+1} \leq C[e_k]^p \quad (3.4)$$

si $p = 2$, la convergence de l'algorithme est dite *quadratique*.

Enfin, la convergence est dite locale si elle n'a lieu que pour des points de départ x_0 dans un voisinage de x^* . Dans le cas contraire la convergence est globale.

Remarque 3.1.

La "classification" précédente des vitesses de convergence renvoie à la notion de comparaison des fonctions au voisinage de $+\infty$. En effet, si on suppose que l'erreur e_k ne s'annule pas, une convergence linéaire revient à dire que $\frac{e_{k+1}}{e_k} = O(1)$, alors qu'une convergence super-linéaire est équivalente à $\frac{e_{k+1}}{e_k} = O(1)$. De manière analogue, un algorithme d'ordre $p \geq 2$ est tel que $\frac{e_{k+1}}{e_k} = O(e_k^{p-2})$. On a bien entendu intérêt à ce que la vitesse de convergence d'un algorithme soit la plus élevée possible (afin d'obtenir la solution avec un minimum d'itérations pour une précision donnée).

3.1 Problèmes d'optimisation sans contraintes

3.1.1 Méthode du Gradient

La méthode (ou algorithme) du Gradient fait partie d'une classe la plus grande de méthodes numériques appelées méthodes de descente. Expliquons rapidement l'idée directrice de ces méthodes.

On veut minimiser une fonction J . Pour cela on se donne un point de départ arbitraire x_0 . Pour construire l'itéré suivant x_1 il faut penser qu'on veut se rapprocher du minimum de J ; on veut donc que $J(x_1) < J(x_0)$. On cherche alors x_1 sous la forme $x_1 = x_0 + \rho_1 d_1$ où d_1 est un vecteur non nul de \mathbb{R}^n et ρ_1 un réel strictement positif.

En pratique donc, on cherche d_1 et ρ_1 pour que $J(x_0 + \rho_1 d_1) < J(x_0)$. On ne peut pas toujours trouver d_1 . Quand d_1 existe on dit que c'est une direction de descente et ρ_1 est le pas de descente. La direction et le pas de descente peuvent être fixes ou changer à chaque itération. Le schéma général d'une méthode de descente est le suivant :

$$\begin{cases} x_0 \in \mathbb{R}^n & \text{donné} \\ x_{k+1} = x_k + \rho_k d_k, & d_k \in \mathbb{R}^n - \{0\}, \rho_k \in \mathbb{R}_+^* \end{cases} \quad (3.5)$$

où ρ_k et d_k sont choisis de telle sorte que $J(x_k + \rho_k d_k) \leq J(x_k)$.

Une idée naturelle pour trouver une direction de descente est de faire un développement de Taylor (formel) à l'ordre 2 de la fonction J entre deux itérés x_k et $x_{k+1} = x_k + \rho_k d_k$:

$$J(x_k + \rho_k d_k) = J(x_k) + \rho_k (\nabla J(x_k), d_k) + o(\rho_k d_k) \quad (3.6)$$

Comme on veut $J(x_k + \rho_k d_k) < J(x_k)$ on peut choisir en première approximation $d_k = -\nabla J(x_k)$. La méthode ainsi obtenue s'appelle l'algorithme du **Gradient**. Le pas ρ_k est choisi constant ou variable.

Algorithme du Gradient

1. Initialisation

$k = 0$: choix de x_0 et de $\rho_0 > 0$.

2. Itération k

$x_{k+1} = x_k - \rho_k \nabla J(x_k)$.

3. Critère d'arrêt

si $\|x_{k+1} - x_k\| < \epsilon$, **STOP**

sinon, on pose $k = k + 1$ et on retourne à 2.

Dans tout ce qui suit, ϵ est un réel positif (petit) donné qui représente la précision désirée.

Cette méthode a pour avantage d'être très facile à mettre en oeuvre. Malheureusement, les conditions de convergence sont assez lourdes (c'est essentiellement de la stricte convexité) et la méthode est en général assez lente. Nous donnons ci-dessous un critère de convergence :

Théorème 3.1. :[5]

Soit J une fonction \mathcal{C}^1 de \mathbb{R}^n dans \mathbb{R} , coercive et strictement convexe. On suppose qu'il existe une constante M strictement positive telle que

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \quad \|\nabla J(x) - \nabla J(y)\| \leq M\|x - y\| \quad (3.7)$$

Alors, si on choisit le pas ρ_k dans un intervalle $[\beta_1, \beta_2]$ tel que $0 < \beta_1 < \beta_2 < \frac{2}{M}$, la méthode du gradient converge vers le minimum de J .

Remarque 3.2. Lorsque J vérifie (1, 5), on peut aussi interpréter l'algorithme du gradient à pas constant comme la méthode des approximations successives appliquée à la recherche du point fixe de la fonction

$$S_\rho(x) = x - \rho \nabla J(x).$$

où $\rho \neq 0$. On peut en effet montrer que S_ρ est lipschitzienne de rapport $(1 - 2\rho\alpha + \rho^2 M^2)$. C'est donc une contraction stricte si $\rho \in]0, 2\alpha/M^2[$, elle possède alors un unique point fixe. Pour ρ/M^2 , le taux de contraction est $(1 - \alpha/M^2)$: c'est le meilleur possible. La convergence est alors celle d'une série géométrique de raison $(1 - \alpha/M^2)$.

On utilise le plus souvent la méthode du gradient à **pas constant** ($\rho_k \equiv \rho$ constant). Toutefois, on peut faire varier le pas à chaque itération : on obtient alors la méthode du gradient à **pas variable**.

La méthode du gradient à **pas optimal** propose un choix du pas qui rend la fonction coût minimale le long de la direction de descente choisie. Plus précisément, l'étape 2, devient :

2. Itération k :

$$x_{k+1} = x_k - \rho_k \nabla J(x_k)$$

où ρ_k réalise le minimum sur \mathbb{R}^+ de la fonction Φ_k définie par

$$\Phi_k(\rho) = J(x_k - \rho \nabla J(x_k))$$

En pratique, on ne cherche pas le minimum de Φ_k et on détermine ρ_k en effectuant une **recherche linéaire** de pas optimal suivant une règle de la forme suivante par exemple :

Règle de recherche linéaire de Wolfe

1. **Initialisation** $\rho = 1$ (par exemple), $\rho_- = \rho_+ = 0$. On se donne $0 < \beta_1 < \beta_2 < 1$.
2. Si $\Phi_k(\rho) \leq \Phi_k(0) + \beta_1 \rho \Phi'_k(0)$ et $\Phi_k(\rho) \geq \Phi_k(0) + \beta_2 \rho \Phi'_k(0)$, **STOP** : $\rho_k = \rho$.
3. Sinon
 - $\Phi_k(\rho) > \Phi_k(0) + \beta_1 \rho \Phi'_k(0)$, on pose $\rho_+ = \rho$.
 - $\Phi_k(\rho) \leq \Phi_k(0) + \beta_1 \rho \Phi'_k(0)$ et $\Phi_k(\rho) \leq \Phi_k(0) + \beta_2 \rho \Phi'_k(0)$, on pose $\rho_- = \rho$.
 et on va à 4.
4. Choix d'un nouveau ρ :
 - Si $\rho_+ = 0$, on cherche $\rho > \rho_-$ (par exemple $\rho = 2\rho_-$).
 - Si $\rho_+ > 0$, on cherche $\rho \in]\rho_-, \rho_+[$ (par exemple $\rho = \frac{\rho_- + \rho_+}{2}$).
 Retour à 2.

La règle apparaissant à l'étape 2. est connue sous le nom générique de règle d'**Armijo**. Il existe beaucoup d'autres règles de recherche linéaire.

Exemple 3.1. Les conditions du théorème peuvent paraître compliquées, aussi nous donnons un exemple. Soit J la fonction de \mathbb{R}^n vers \mathbb{R} déjà évoquée plusieurs fois (car elle joue un rôle important) définie par

$$J(x) = \frac{1}{2}(Ax, x) - (b, x)$$

où A est une matrice carrée, symétrique et définie positive et $b \in \mathbb{R}^n$. Cette fonction J vérifie les hypothèses du théorème ci-dessus avec pour constantes α et M la plus petite et la plus grande valeur propre de A (respectivement).

Remarque 3.3. La notion d'ellipticité est très importante, car elle conditionne la convergence de la plupart des algorithmes qui vont être décrits par la suite. Toutefois, les conditions de convergence que nous donnons sont toujours des conditions **suffisantes**. L'algorithme converge si elles sont vérifiées mais **il peut éventuellement converger, même si elles ne le sont pas...**

En pratique, on ne calcule pas α et M . Pour trouver l'intervalle de convergence de ρ , on fait plusieurs tests pour différentes valeurs. La non convergence se traduit en général, soit par une explosion de la solution (elle va clairement vers $+\infty$) soit par des oscillations (périodiques ou non) qui empêchent la suite des itérés de converger vers une valeur.

3.1.2 Méthode de Newton

L'algorithme de Newton en optimisation est une application directe de l'algorithme de Newton pour la résolution d'équations du type $J(x) = 0$. En optimisation sans contrainte, l'algorithme de Newton cherche les solutions de l'équation $\nabla J(x^*) = 0$. Ceci est une équation non linéaire (ou plutôt un système d'équations non linéaires) dans \mathbb{R}^n et nous allons utiliser la méthode de Newton pour la résoudre. *Toutefois nous n'obtiendrons que les points critiques de J : il faudra ensuite vérifier que ce sont bien des minima.*

Ici $f = \nabla J$ est bien une fonction de \mathbb{R}^n dans \mathbb{R}^n . La dérivée de f n'est autre que la matrice hessienne de J : $H(x) = D^2 J(x)$. La méthode de Newton s'écrit alors :

Algorithme de Newton dans \mathbb{R}

1. Initialisation

$k = 0$: choix de $x_0 \in \mathbb{R}$ dans un voisinage de x^* .

2. Itération k

$$x_{k+1} = x_k - [H(x_k)]^{-1} \nabla J(x_k).$$

3. Critère d'arrêt

si $|x_{x+1} - x_k| < \epsilon$, **STOP**

sinon, on pose $k = k + 1$ et on retourne à 2.

L'étape 2 de la méthode revient à résoudre le système linéaire suivant :

$$H_k \delta_k = \nabla J(x_k)$$

où $H_k = H(x_k)$ puis à poser $x_{k+1} = x_k - \delta_k$.

3.1.3 Méthode du Gradient conjugué

Méthode du Gradient conjugué : cas linéaire

La présentation des deux algorithmes précédents montrent qu'en réalité on ne calcule pas les extrema d'une fonction mais les points stationnaires (ou points critiques) où J est quadratique nous

avons vu que cela revient à résoudre un système linéaire : $Ax = b$. Nous allons donc présenter ici une méthode de résolution d'un système linéaire issue de la théorie de l'optimisation et convergente dans le cas des matrices symétriques définies positives. Dans ce cas l'application qui au couple (x, y) associe le produit (Ax, y) est un produit scalaire sur \mathbb{R}^n qu'on note $(x, y)_A$. La méthode qui suit est une méthode de descente inspirée de la méthode du gradient. La direction de descente w_k n'est plus égale au gradient $g_k = Ax_k - b$: le gradient g_k est "corrigé" de façon que toutes les directions w_k obtenues soient orthogonales (ou conjuguées) pour le produit scalaire $(\cdot, \cdot)_A$. Plus précisément on pose :

$$w_k = g_k + \alpha_k w_{k-1}$$

Tel que : $(w_k, w_{k-1})_A = 0$.

Algorithme du Gradient conjugué

1. Initialisation

$k = 0$: choix de $x_0 \in \mathbb{R}^n$ et calcul de $g_0 = Ax_0 - b$.

2. Itération k

(a) Si $g_k = 0$ **STOP** ;

(b) Sinon :

$$\bullet w_k = \begin{cases} g_0 & \text{si } k = 0 \\ g_k + \alpha_k w_{k-1} & \text{si } k \geq 1 \end{cases} \quad \text{avec} \quad \alpha_k = -\frac{(g_k, Aw_{k-1})}{(Aw_{k-1}, w_{k-1})}.$$

$$\bullet \rho_k = \frac{(g_k, w_k)}{(Aw_k, w_k)}.$$

$$\bullet x_{k+1} = x_k - \rho_k w_k.$$

$$\bullet g_{k+1} = Ax_{k+1} - b.$$

(c) $k = k + 1$.

Une fois de plus, outre la convergence de la suite des itérés, nous devons assurer son existence c'est-à-dire $w_k \neq 0$ à l'étape 2b. de l'algorithme, ce que nous allons faire en démontrant le résultat suivant.

Théorème 3.2. :[5]

La méthode du gradient conjugué trouve le minimum d'une fonction quadratique J , où A est symétrique, définie positive, en au plus n itérations où n est l'ordre de A .

Remarque 3.4. *Cette méthode est très stable même pour des matrices mal conditionnées. Elle demande $2n^3$ opérations dans le cas d'une matrice pleine et de n itérations. Pour une matrice creuse, le nombre d'opérations diminue beaucoup.*

En pratique, la convergence n'est pas finie à cause des erreurs de précision dues à la machine. On ajoute donc en général une étape 3. "Critère d'arrêt" analogue à celle des algorithmes précédents.

3.1.4 Méthode de Relaxation

La dernière méthode que nous présentons permet de ramener un problème de minimisation dans \mathbb{R}^n à la résolution successive de n problèmes de minimisation dans \mathbb{R} (à chaque itération).

On cherche à minimiser $J : \mathbb{R}^n \rightarrow \mathbb{R}$; posons $X = (x_1, x_2, \dots, x_n)$. Le principe de la méthode est le suivant : étant donné un itéré X^k de coordonnées (x_1^k, \dots, x_n^k) , on fixe toutes les composantes sauf la première et on minimise sur la première :

$$\min J(x, x_2^k, x_3^k, \dots, x_n^k), \quad x \in \mathbb{R}$$

On obtient ainsi la première coordonnée de l'itéré suivant X^{k+1} que l'on note x_1^{k+1} ; on peut, pour effectuer cette minimisation dans \mathbb{R} , utiliser par exemple la méthode de Newton dans \mathbb{R} . On recommence ensuite en fixant la première coordonnée à x_1^{k+1} et les $n - 2$ dernières comme précédemment. On minimise sur la deuxième coordonnée et ainsi de suite. L'algorithme obtenu est le suivant :

Méthode de relaxation successive

1. Initialisation

$k = 0$: choix de $X^0 \in \mathbb{R}^n$

2. Itération k

pour i variant de 1 à n , on calcule la solution x_i^{k+1} de

$$\min J(x_1^{k+1}, x_2^{k+1}, \dots, x_{i-1}^{k+1}, x, x_{i+1}^k, \dots, x_n^k), \quad x \in \mathbb{R}$$

3. Critère d'arrêt

si $\|x_{k+1} - x_k\| < \epsilon$, **STOP**

sinon, on pose $k = k + 1$ et on retourne à 2.

Nous ne détaillerons pas les conditions de convergence de cette méthode. Elles sont analogues aux conditions de convergence de la méthode du gradient et de Newton.

3.2 Problèmes d'optimisation avec contraintes

Problèmes linéaires

3.2.1 Rappel

Définition 3.4 (fonction linéaire). Si c_1, c_2, \dots, c_n sont des constantes réelles, la fonction $f(x)$ définie par :

$$f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n = \sum_{i=1}^n c_i x_i \quad (3.8)$$

est une *fonction linéaire*.

Définition 3.5. Si b, c_1, c_2, \dots, c_n sont des constantes réelles, alors l'équation

$$\sum_{j=1}^n c_j x_j = b \quad (3.9)$$

est appelée une *équation linéaire*.

Définition 3.6. Un PL est dit sous forme standard s'il est de la forme :

$$(P) \quad \begin{cases} \max_x \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = 1, \dots, m \\ x_j \geq 0 \quad j = 1, \dots, n \end{cases}$$

Remarque 3.5. Il est toujours possible de ramener un programme linéaire quelconque sous forme-standard.

Définition 3.7. — La fonction linéaire maximisée (ou minimisée) dans un programme linéaire est appelée **fonction objectif**.

- Une solution (x_1, \dots, x_n) est dite **réalisable** si elle satisfait à toutes les contraintes (incluant les bornes).
- Une solution (x_1^*, \dots, x_n^*) est dite **optimale** si elle est réalisable et si elle maximise la fonction objectif parmi toutes les solutions réalisables. La valeur correspondante de l'objectif est appelée la valeur optimale.

Remarque 3.6. La solution optimale (si elle existe) n'est pas nécessairement unique.

On peut se demander s'il existe toujours au moins une solution optimale. La réponse est non ; deux situations peuvent se présenter.

1. Un programme linéaire qui n'a pas de solution réalisable est appelé **non réalisable**.
2. Un programme linéaire réalisable qui n'a pas de valeur optimale finie est appelé **non borné**.

Théorème 3.3. [3] Dans tout programme linéaire, une seule des éventualités suivantes peut se présenter :

1. il existe au moins une solution optimale.
2. le programme linéaire est non-réalisable.
3. le programme linéaire est non borné.

3.2.2 Méthode du Simplexe

Nous voyons dans cette section la méthode du simplexe pour la résolution d'un programme linéaire. Cette méthode, due à Dantzig, est la première qui a permis de résoudre un programme linéaire. Une description informelle est faite dans cette section. La description formelle de l'algorithme est présentée à la section suivante.

Dictionnaire

Soit un programme linéaire sous forme standard :

$$(\mathcal{PL}) \quad \begin{cases} \max \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = 1, \dots, m \\ x_j \geq 0 \quad j = 1, \dots, n \end{cases}$$

On introduit les variables d'écart $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ et on note l'objectif par z . On définit ces variables par

$$\frac{x_{n+i}}{z} = \frac{\sum_{j=1}^n a_{ij} x_j}{\sum_{j=1}^n c_j x_j} \quad i = 1, \dots, m$$

Dans le cadre de la méthode du simplexe, chaque solution réalisable (x_1, \dots, x_n) de (\mathcal{PL}) est représentée par $n + m$ variables non-négatives. À chaque itération, la méthode du simplexe fournit une solution réalisable qui améliore la fonction objectif. À chaque solution réalisable correspond un système d'équations linéaires qu'on appelle dictionnaire.

Dans un dictionnaire :

1. Toute solution non négative des équations linéaires comprises dans le dictionnaire est une solution de (\mathcal{PL}) , et vice-versa.

2. Les équations d'un dictionnaire doivent exprimer m des variables (x_1, \dots, x_{n+m}) et la variable z en fonction des n autres variables.

On appelle un dictionnaire réalisable, si en posant à 0 les variables du côté droit, on obtient une solution réalisable.

Définition 3.8.

— Les variables x_j qui se trouvent du côté gauche du dictionnaire sont appelées **variables de base**.

— Les variables x_j qui se trouvent du côté droit du dictionnaire sont appelées variables **hors-base**.

L'ensemble des variables de base constituent une base. Ainsi, à chaque itération la base change : une variable entre dans la base et une variable sort de la base. Le choix de la variable d'entrée est dicté par le désir d'améliorer l'objectif z . Le choix de la variable de sortie est dicté par le désir de conserver toutes les variables non-négatives.

La variable de sortie apparaît dans la ligne du pivot. On appelle pivotage le processus permettant de passer d'un dictionnaire à un autre (ou encore, passer d'une base à une autre).

L'algorithme du simplexe

On peut résumer l'algorithme du simplexe comme suit :

Étape 1) **Initialisation** : former le dictionnaire réalisable initial.

Étape 2) **Itérations** :

- choix de la variable d'entrée.
- choix de la variable de sortie.
- pivotage.

Étape 3) **Critère d'optimalité** :

- Si oui, arrêt.
- Sinon, aller à l'étape 2.

Programme auxilliaire :

le programme linéaire auxilliaire suivant :

$$(\mathcal{P}\mathcal{L}_{\text{AUX}}) \quad \begin{cases} \min x_0 \\ \sum_{j=1}^n a_{ij}x_j - x_0 \leq b_i \quad i = 1, \dots, m \\ x_j \geq 0 \quad j = 1, \dots, n \end{cases}$$

Ce problème a :

1. une solution réalisable facile à obtenir (après un pivot).
2. le programme initial a une solution réalisable si et seulement si le problème auxilliaire a comme valeur optimale 0.

Donc, pour trouver un dictionnaire réalisable initial pour un problème qui n'est pas réalisable à l'origine, on résoud dans un premier temps le programme auxilliaire.

INITIALISATION

1. Si le programme linéaire est réalisable à l'origine (tous les $b_i \geq 0$), alors aller à l'étape 3.
2. PHASE 1 : Trouver une solution réalisable.
 - (a) Construire le programme auxilliaire.
 - (b) Construire le 1^{er} dictionnaire non-réalisable.

- (c) Entrer x_0 et sortir la variable la plus négative \implies dictionnaire réalisable pour le programme auxiliaire.
- (d) Itérations du simplexe (en cas d'égalité, sortir x_0 de la base) jusqu'à l'optimalité.
- (e) Si la solution optimale est 0, aller à l'étape 3 ; sinon, le programme initial est non réalisable.

3. PHASE 2 :

- (a) Construire le 1^{er} dictionnaire réalisable (du problème initial).
- (b) Itérations du simplexe.

Itérations

Étant donné un dictionnaire réalisable, on doit choisir une variable d'entrée qui détermine une variable de sortie afin d'obtenir le prochain dictionnaire.

1. **Choix de la variable d'entrée**

La variable d'entrée est une variable hors-base x_j dont le coefficient \bar{c}_j dans la dernière ligne du dictionnaire courant est positif. Plus précisément, considérons la dernière ligne du dictionnaire courant

$$z = z^* + \sum_{j \in N} \bar{c}_j x_j \tag{3.10}$$

où N est l'ensemble des indices j des variables hors-base x_j . La solution courante, $x_j = 0$ pour tout $j \in N$ donne à l'objectif la valeur z^* .

– Si

$$\bar{c}_j \leq 0, \quad \forall j \in N$$

alors la solution courante est optimale puisque tout autre solution avec $x_j \geq 0, j \in N$ donne une valeur numérique à l'objectif inférieure ou égale à z^* .

– S'il existe des $j \in N$ tels que $\bar{c}_j > 0$, alors les x_j correspondants sont toutes des variables candidates à entrer dans la base.

2. **Variable de sortie**

La variable de sortie est une variable de base dont la non négativité impose la contrainte la plus forte sur la borne supérieure de la variable d'entrée. On peut obtenir l'une des situations suivantes :

- (a) On trouve une seule variable de sortie et on pivote.
- (b) On trouve plusieurs variables de sortie : on choisit alors une variable de sortie parmi ces dernières et on pivote.
- (c) On ne trouve aucune variable de sortie : alors le problème est non borné, c'est-à-dire que l'objectif peut prendre une valeur aussi grande que désirée ($z = z^* + t\bar{c}_j$ où \bar{c}_j est le coefficient > 0 de la variable d'entrée dans la dernière ligne du dictionnaire et t peut prendre une valeur aussi grande que voulue).

Critère d'arrêt

On sait que si dans la dernière ligne du dictionnaire courant

$$z = z^* + \sum_{j \in N} \bar{c}_j x_j$$

on a $\bar{c}_j \leq 0, \forall j \in N$ la solution de base courante est optimale. Peut-on ne jamais obtenir une telle condition, c'est-à-dire générer un nombre infini de dictionnaires ? La réponse est oui.

Théorème 3.4. [3] *Si la méthode du simplexe ne se termine pas, alors elle doit cycliser.*

Exemple 3.2.

$$\begin{cases} 2x_1 + 3x_2 + x_3 \leq 5 \\ 4x_1 + x_2 + 2x_3 \leq 11 \\ 3x_1 + 4x_2 + 2x_3 \leq 8 \\ x_i \geq 0, \quad i = 1, \dots, 3 \\ 5x_1 + 4x_2 + 3x_3 = z(\max) \end{cases} \quad (3.11)$$

On commence par introduire des variables d'écart, lesquelles mesurent en fait l'écart entre le second membre et le premier membre au niveau de chaque contrainte et qui de ce fait doivent être non négatives pour les solutions réalisables. On obtient ceci :

$$\begin{cases} x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ x_i \geq 0, \quad i = 1, \dots, 5 \\ 5x_1 + 4x_2 + 3x_3 = z(\max) \end{cases} \quad (3.12)$$

Les variables x_1, x_2 et x_3 sont les variables de décision et les variables x_4, x_5 et x_6 sont les variables d'écart. Ce format s'appelle format dictionnaire (les variables se trouvant à gauche s'expriment (s'expliquent) uniquement en fonction des variables se trouvant à droite. Le format tableau quant à lui se présente comme suit :

x_1	x_2	x_3	x_4	x_5	x_6	b
2	3	1	1	0	0	5
4	1	2	0	1	0	11
3	4	2	0	0	1	8
5	4	3	0	0	0	z

Remarque :

Lorsqu'on fixe les variables de décision à 0, on obtient une solution réalisable à savoir $x = (0, 0, 0, 5, 11, 8)$. Notons que $z(x) = 0$.

On supposera dans la suite, sans perte de généralité, que l'on a un problème de maximisation.

Principe de l'algorithme de simplexe

L'algorithme du simplexe consiste à partir d'une solution réalisable et à tenter de l'améliorer itérativement. Ceci veut dire qu'une itération typique de l'algorithme part d'une solution réalisable non optimale x et détermine une autre solution réalisable, disons x' telle que $z(x) \leq z(x')$. Le but est d'arriver après un certain nombre d'itérations à une solution optimale.

Comment trouver cet x' ? Ici, si on maintient $x_2 = x_3 = 0$ et si on fait croître x_1 , on a $z = 5x_1$. Ceci est donc prometteur. Ce qu'il faut alors déterminer c'est la plus grande valeur que peut prendre x_1 sans pour autant qu'une autre variable devienne strictement négative. On a :

$$\begin{aligned} x_4 \geq 0 &\implies 5 - 2x_1 \geq 0 \implies x_1 \leq \frac{5}{2} \\ x_5 \geq 0 &\implies 11 - 4x_1 \geq 0 \implies x_1 \leq \frac{11}{4} \\ x_6 \geq 0 &\implies 8 - 3x_1 \geq 0 \implies x_1 \leq \frac{8}{3} \end{aligned}$$

Lorsque x_1 prend la plus grande valeur admissible à savoir $5/2$, on obtient une solution réalisable $x' = (5/2, 0, 0, 0, 1, 1/2)$ laquelle donne la valeur $25/2$ à la fonction objective. L'objectif annoncé est atteint. Il s'agit maintenant d'associer à cette nouvelle solution x' un dictionnaire de manière à ce que la

direction d'amélioration soit aussi simple que lors de la première itération. Pour cela, il faut exprimer les composantes non nulles de x' et z uniquement en fonction des composantes nulles de x' . Il suffit alors d'utiliser la contrainte la plus astreignante à savoir $x_4 = 5 - 2x_1 - 3x_2 - x_3$ pour exprimer x_1 en fonction des composantes nulles de x' et ensuite substituer cette expression à x_1 partout ailleurs. On obtient :

$$\begin{cases} x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x_6 = \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ x \geq 0 \\ z = \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{cases} \quad (3.13)$$

Lorsqu'on observe la fonction objective, on constate que la seule composante de x' qui peut améliorer la valeur de la fonction objective est x_3 . C'est la seule dont le coefficient dans la fonction objective est strictement positif. Notons que si toute variable a un coefficient négatif ou nul dans la fonction objective, alors on peut conclure que la solution courante est optimale.

Revenons à notre exemple. Lorsque la variable x_3 prend la plus grande valeur admissible à savoir la valeur 1, x_6 s'annule et on obtient la solution réalisable (2,0,1,0,1,0). Ci-dessous est représenté le dictionnaire correspondant à cette solution :

$$\begin{cases} x_3 = 1 + x_2 + 3x_4 - 2x_6 \\ x_1 = 2 - 2x_2 - 2x_4 + x_6 \\ x_5 = 1 + 5x_2 + 2x_4 \\ x \geq 0 \\ z = 13 - 3x_2 - x_4 - x_6 \end{cases} \quad (3.14)$$

Les coefficients des variables dans la fonction objective étant tous non positifs, la solution courante $x^* = (2, 0, 1, 0, 1, 0)$ est optimale et l'optimum du programme linéaire est $z^* = 13$.

Terminologie :

Dans le format dictionnaire, les variables du 1^{er} membre sont appelées variables de base ; celles du second membre sont dites hors base.

Problèmes quadratiques

De façon similaire au cas sans contraintes, les méthodes de résolution des problèmes contraints sont très nombreuses. Nous allons en présenter quelques unes, qui conduisent à des algorithmes numériques simples et utilisables en pratique.

Le problème auquel nous nous intéressons, dans tout ce chapitre, est le suivant :

$$\text{Trouver } u \in K \text{ tel que } J(u) = \min_{v \in K} J(v)$$

Ici, J est la fonctionnelle qui à v associe $J(v) = \frac{1}{2}(Av, v) - (b, v)$, où la matrice A est supposée symétrique définie positive de $\mathbb{R}^{n \times n}$, b un vecteur quelconque de \mathbb{R}^n et K est un fermé convexe de \mathbb{R}^n .

3.2.3 Méthode du gradient projeté

D'abord insistons sur l'hypothèse que K est un convexe fermé. Nous savons que la solution optimale u du problème contraint, vérifie l'inéquation d'Euler suivante :

$$(\nabla J(u), u - v) \leq 0 \quad \forall v \in K \quad (3.15)$$

D'autre part, rappelons que pour tout $w \in \mathbb{R}^n$, il existe une unique projection de w sur K , notée $P_K(w) \in K$, solution de

$$\|P_K(w) - w\| = \min_{v \in K} \|v - w\| \quad (3.16)$$

De plus, pour tout $w \in \mathbb{R}^n$, la projection $P_K(w)$ est caractérisée par :

$$(P_K(w) - w, P_K(w) - v) \leq 0 \quad \forall v \in K \quad (3.17)$$

On arrive alors à la proposition suivante :

Proposition 3.1. [3] *Soit K un convexe fermé de \mathbb{R}^n et $\rho > 0$. $u \in K$ est solution du problème contraint $J(u) = \min_{v \in K} J(v)$ si et seulement si*

$$u = P_K(u - \rho \nabla J(u)) \quad (3.18)$$

L'algorithme du gradient projeté se base sur la propriété ci-dessus et propose de construire une suite $(u_k)_k$ qui converge vers u en tant que point fixe de l'application $v \mapsto P_K(v - \rho \nabla J(v))$. Plus précisément, l'algorithme s'écrit :

Algorithme du gradient projeté

1. **Initialisation** $u_0 \in \mathbb{R}^n$, un pas $\rho > 0$ et une précision $\eta > 0$.
On calcule $u_1 = P_K(u_0 - \rho \nabla J(u_0))$
2. **Itération** k tant que $\|u_k - u_{k-1}\| > \eta$, on définit u_{k+1} par

$$u_{k+1} = P_K(u_k - \rho \nabla J(u_k))$$

Remarque 3.7. *Noter que dans cet algorithme le test d'arrêt est différent de celui qu'on avait utilisé dans les cas des algorithmes d'optimisation sans contrainte.*

Théorème 3.5. [3] *Soit $K \neq \emptyset$ un convexe fermé de \mathbb{R}^n , et Soit*

$$J : v \mapsto \frac{1}{2}(Av, v) - (b, v) \quad (3.19)$$

où $A \in \mathbb{R}^{n \times n}$ est symétrique, définie positive, et $b \in \mathbb{R}^n$. Si $0 < \rho < \frac{2}{\lambda_{\max}(A)}$, alors quel que soit $u_0 \in \mathbb{R}^n$ la suite (u_k) définie par le gradient projeté converge vers le minimum u . ($\lambda_{\max}(A)$ désigne la plus grande valeur propre de A)

L'application $w \in \mathbb{R}^n \mapsto P_K(w)$ est 1-Lipshitz :

$$\|P_K(w) - P_K(v)\|_2 \leq \|w - v\|_2 \quad \forall w, v \in \mathbb{R}^n \quad (3.20)$$

Problèmes non linéaires

Les algorithmes newtoniens sont basés sur la linéarisation d'équations caractérisant les solutions que l'on cherche, fournies par les conditions d'optimalité d'ordre 1. Ces algorithmes sont *primaux-duaux* dans le sens où ils génèrent à la fois une suite *primale* (x_k) convergeant vers une solution x^* du problème considéré, et une suite *duale* (λ^k) de multiplicateurs convergeant vers un multiplicateur optimal λ^* associé à x^* .

3.2.4 Méthode SQP (Sequential Quadratic Programming)

La méthode séquentielle de programmation quadratique abrégée par SQP (ce nom en clature est due à son nom en anglais sequential quadratic programming) est une méthode d'optimisation qui s'est avéré beaucoup plus performante que celles basées sur le gradient. Dans cette section, on étudie deux algorithmes SQP pour les problèmes non linéaires, l'un pour les contraintes de type égalité et l'autre pour les contraintes de type inégalité.

Principe de la méthode

A chaque itération, une approximation du hessien du lagrangien du problème est faite en utilisant une méthode quasi-newtonienne, ce qui va permettre de générer un sous-problème QP dont la solution va être utilisée pour former la direction de recherche.

Néanmoins, la méthode SQP ne donne pas l'optimum global dans le cas où la fonction objectif n'est pas convexe.

3.2.5 La méthode SQP pour les problèmes sous contraintes de type égalité

les méthodes SQP sont généralement introduites, dans un premier temps, les problèmes avec contraintes égalités. Considérons le programme non linéaire suivant :

$$\left\{ \begin{array}{l} \min_x f(x) \\ s.c \\ g_i(x) = 0 \quad , i = 1.., k, ..m \\ x \in \mathbb{R}^n \end{array} \right. \quad (3.21)$$

où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction convexe et deux fois continûment différentiable, est une fonction convexe.

À chaque itération k , on construit un sous problème quadratique en remplaçant la fonction objectif f par une approximation quadratique locale de la forme :

$$f(x) = f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)H_f(x_k)(x - x_k). \quad (3.22)$$

et la contrainte g sera remplacée par l'approximation affine :

$$g(x) = g(x_k) + \nabla g(x_k)(x - x_k). \quad (3.23)$$

Posons $p_k = x - x_k$.

Et construisons le sous problème quadratique suivant :

$$\left\{ \begin{array}{l} \min_p \frac{1}{2}p_k^t H_f(x_k)p_k + \nabla f(x_k)^t p_k + f(x_k) \\ s.c \\ \nabla g(x_k)^t p_k + g(x_k) = 0. \\ x \in \mathbb{R}^n \end{array} \right. \quad (3.24)$$

Le problème quadratique est lié à un modèle quadratique local du Lagrangien L

$$L(x, \lambda) = f(x) + \lambda^t g(x) \quad (3.25)$$

Comme la fonction objectif

$$\left\{ \begin{array}{l} \min_p \frac{1}{2} p_k^t H_L(x_k, \lambda_k) p_k + \nabla f(x_k)^t p_k + f(x_k) \\ s.c \\ \nabla g(x_k)^t p_k + g(x_k) = 0. \\ p_k \in \mathbb{R}^n \end{array} \right. \quad (3.26)$$

3.2.6 Algorithme de la méthode SQP pour les problèmes sous contraintes de type égalités

Choisir une paire initiale (x_0, λ_0) , une tolérance $\varepsilon = 10^{(-6)}$;
Poser $k = 0$;

Répéter

Evaluer $f(x_k), \nabla f(x_k), \nabla^2 L(x_k, \lambda_k), g(x_k), \nabla g(x_k)$

Résoudre le problème (3.26) pour obtenir p_k et l_k

Poser $x_{k+1} = x_k + p_k$ et λ_{k+1}

$k \leftarrow k + 1$

Jusqu'à $\|x_{k+1} - x_k\|$

Fin

Exemple 3.3. Considérons le problème non linéaire suivant :

$$\left\{ \begin{array}{l} \text{minimiser} \quad x_1^2 + 2(x_2 - 2)^2 \\ s.c \\ x_1^2 + x_2^2 - 1 = 0 \\ x_1, x_2 \in \mathbb{R} \end{array} \right. \quad (3.27)$$

Soit le Lagrangien : $L(x, \lambda) = f(x) + \lambda g(x)$

$$L(x, \lambda) = (x_1^2 + 2(x_2 - 2)^2) + \lambda(x_1^2 + x_2^2 - 1)$$

$$\nabla^2 L(x_k, \lambda_k) = \begin{pmatrix} 2 + 2\lambda_k & 0 \\ 0 & 4 + 2\lambda_k \end{pmatrix}$$

$$\nabla f(x_k) = (2x_1^k, 4(x_2^k - 2))$$

$$\nabla g(x_k) = (2x_1^k, 2x_2^k)$$

Itération 0

$$k = 0$$

$$x^0 =, \left(\frac{-1}{2}, \frac{1}{2}\right), \quad \lambda_0 = 1$$

$$\nabla^2 L_0 = \begin{pmatrix} 4 & 0 \\ 0 & 6 \end{pmatrix}$$

$$\nabla f(x^0) = (-1, -6)^t$$

$$\nabla g(x^0) = (-1, 1)^t$$

$$g(x^0) = \frac{-1}{2}$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & 2p_1^2 + 3p_2^2 - p_1 - 6p_2 + 9.25 \\ & s.c \\ & -p_1 + p_2 - 1/2 = 0 \end{cases} \quad (3.28)$$

$$\begin{pmatrix} \nabla^2 L_0 & \nabla g(x_0)^t \\ \nabla g(x_0) & 0 \end{pmatrix} \begin{pmatrix} p_0 \\ l_0 \end{pmatrix} = - \begin{pmatrix} \nabla f(x_0) \\ g(x_0) \end{pmatrix}$$

$$\Leftrightarrow \nabla g(x_0)^t \nabla^2 L_0^{-1} \nabla g(x_0) l_0 = -(g(x_0) - \nabla g(x_0)^t \nabla^2 L_0^{-1} \nabla f(x_0))$$

$$\Leftrightarrow (-1 \ 1) \begin{pmatrix} 1/4 & 0 \\ 0 & 1/6 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} l_0 = - \left(-1/2 - (-1 \ 1) \begin{pmatrix} 1/4 & 0 \\ 0 & 1/6 \end{pmatrix} \begin{pmatrix} -1 \\ -6 \end{pmatrix} \right)$$

On obtient :

$$l_0 = 3.15$$

$$p_0 = \begin{pmatrix} 1.03 \\ 0.475 \end{pmatrix}$$

Donc :

$$x^1 = (0.53, 0.975) \quad \text{et} \quad \lambda_1 = 3.15$$

$$\|x^1 - x^0\| > \varepsilon, \quad k = k + 1$$

Itération 1

Comme dans l'itération précédente on calcule les quantités suivantes :

$$\nabla^2 L_1 = \begin{pmatrix} 8.3 & 0 \\ 0 & 10.3 \end{pmatrix}$$

$$\nabla f(x^1) = (1.06, -4.1)^t$$

$$\nabla g(x^1) = (1.06, 1.95)^t$$

$$g(x^1) = 0.23$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & \frac{8.3}{2} p_1^2 + \frac{10.3}{2} p_2^2 + 1.06 p_1 - 4.1 p_2 \\ & s.c \\ & 1.06 p_1 + 1.95 p_2 + 0.23 = 0 \end{cases} \quad (3.29)$$

On répétant le même procédé que l'itération précédente on obtient :

$$l_1 = 1.72$$

$$p_1 = \begin{pmatrix} 0.35 \\ -0.073 \end{pmatrix}$$

Donc :

$$x^2 = (0.88, 0.902) \quad \text{et} \quad \lambda_2 = 1.72$$

$$\|x^2 - x^1\| > \varepsilon, \quad k = k + 1$$

Itération 2

$$\nabla^2 L_2 = \begin{pmatrix} 5.44 & 0 \\ 0 & 7.44 \end{pmatrix}$$

$$\nabla f(x^2) = (1.76, -4.392)^t$$

$$\nabla g(x^2) = (1.76, 1.804)^t$$

$$g(x^2) = 0.588$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & \frac{5.44}{2}p_1^2 + \frac{7.44}{2}p_2^2 + 1.76p_1 - 4.392p_2 \\ & s.c \\ & 1.76p_1 + 1.804p_2 + 0.588 = 0 \end{cases} \quad (3.30)$$

On répétant le même procédé que l'itération précédente on obtient :

$$l_2 = 1.07$$

$$p_2 = \begin{pmatrix} -0.68 \\ 0.328 \end{pmatrix}$$

Donc :

$$x^3 = (0.2, 1.23) \quad \text{et} \quad \lambda_3 = 1.07$$

$$\|x^3 - x^2\| > \varepsilon, \quad k = k + 1$$

Itération 3

$$\nabla^2 L_3 = \begin{pmatrix} 4.14 & 0 \\ 0 & 6.14 \end{pmatrix}$$

$$\nabla f(x^3) = (0.4, -3.08)^t$$

$$\nabla g(x^3) = (0.4, 2.46)^t$$

$$g(x^3) = 0.55$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & \frac{4.14}{2}p_1^2 + \frac{6.14}{2}p_2^2 + 0.4p_1 - 3.08p_2 \\ & s.c \\ & 0.4p_1 + 2.46p_2 + 0.55 = 0 \end{cases} \quad (3.31)$$

On répétant le même procédé que l'itération précédente on obtient :

$$l_3 = 1.16$$

$$p_3 = \begin{pmatrix} -0.2 \\ 0.02 \end{pmatrix}$$

Donc :

$$x^4 = (0, 1.25) \quad \text{et} \quad \lambda_4 = 1.16$$

$$\|x^4 - x^3\| > \varepsilon,$$

$$k = k+1$$

Itération 4

$$\nabla^2 L_4 = \begin{pmatrix} 4.32 & 0 \\ 0 & 6.32 \end{pmatrix}$$

$$\nabla f(x^4) = (0, -3)^t$$

$$\nabla g(x^4) = (0, 2.5)^t$$

$$g(x^4) = 0.562$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & \frac{4.32}{2}p_1^2 + \frac{6.32}{2}p_2^2 - 3p_2 \\ & s.c \\ & 2.5p_2 + 0.562 = 0 \end{cases} \quad (3.32)$$

On répétant le même procédé que l'itération précédente on obtient :

$$l_4 = 1.76$$

$$p_4 = \begin{pmatrix} 0 \\ -0.224 \end{pmatrix}$$

Donc :

$$x^5 = (0, 1.025) \quad \text{et} \quad \lambda_5 = 1.76$$

$$\|x^5 - x^4\| > \varepsilon, \quad k = k + 1$$

Itération 5

$$\nabla^2 L_5 = \begin{pmatrix} 5.52 & 0 \\ 0 & 7.52 \end{pmatrix}$$

$$\nabla f(x^5) = (0, -3.896)^t$$

$$\nabla g(x^5) = (0, 2.052)^t$$

$$g(x^5) = 0.052$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & \frac{5.52}{2}p_1^2 + \frac{7.52}{2}p_2^2 - 3.896p_2 \\ & s.c \\ & 2.052p_2 + 0.052 = 0 \end{cases} \quad (3.33)$$

On répétant le même procédé que l'itération précédente on obtient :

$$l_5 = 2$$

$$p_5 = \begin{pmatrix} 0 \\ -0.025 \end{pmatrix}$$

Donc :

$$x^6 = (0, 1) \quad \text{et} \quad \lambda_6 = 2$$

$$\|x^6 - x^5\| > \varepsilon, \quad k = k + 1$$

Itération 6

$$\nabla^2 L_6 = \begin{pmatrix} 6 & 0 \\ 0 & 8 \end{pmatrix}$$

$$\nabla f(x^6) = (0, -4)^t$$

$$\nabla g(x^6) = (0, 2)^t$$

$$g(x^6) = 0$$

Le sous problème quadratique s'écrit comme suit :

$$\begin{cases} \min & 3p_1^2 + 4p_2^2 - 4p_2 \\ & s.c \\ & 2p_2 = 0 \end{cases} \quad (3.34)$$

On répétant le même procédé que l'itération précédente on obtient :

$$l_6 = 2$$

$$p_6 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Donc :

$$x^7 = (0, 1) \quad \text{et} \quad \lambda_7 = 2$$

$$\|x^7 - x^6\| = 0 \quad \text{implique} \quad (0, 1) \text{ est la solution optimale.}$$

3.2.7 La méthode SQP pour les contraintes inégalités

Le fonctionnement de la méthode SQP peut être facilement élargi pour les problèmes de programmation non linéaire généraux de la forme :

$$\left\{ \begin{array}{l} \text{minimiser } f(x) \\ \text{s.c} \\ g_j(x) \leq 0 \quad , j = 1.., k, ..m \\ h_j(x) = 0 \quad , j = 1.., k, ..p \\ x \in \mathbb{R}^n \end{array} \right. \quad (3.35)$$

À chaque itération k , on construit un sous problème quadratique en remplaçant la fonction objectif f par une approximation quadratique locale de la forme :

$$f(x) = f(x_k) + \nabla f(x_k)(x - x_k) + \frac{1}{2}(x - x_k)H_f(x_k)(x - x_k). \quad (3.36)$$

et la contrainte g sera remplacée par l'approximation affine :

$$g(x) = g(x_k) + \nabla g(x_k)(x - x_k). \quad (3.37)$$

Posons $p_k = x - x_k$.

Et construisons le sous problème quadratique suivant :

$$\left\{ \begin{array}{l} \text{minimiser } \frac{1}{2}p_k^t H_f(x_k)p_k + \nabla f(x_k)^t p_k + f(x_k) \\ \text{s.c} \\ \nabla g(x_k)^t p_k + g(x_k) \leq 0. \\ x \in \mathbb{R}^n \end{array} \right. \quad (3.38)$$

Le problème quadratique est lié à un modèle quadratique local du Lagrangien L

$$L(x, \lambda) = f(x) + \lambda^t g(x) \quad (3.39)$$

Comme la fonction objectif

$$\left\{ \begin{array}{l} \text{min}_p \frac{1}{2} p_k^t H_L(x_k, \lambda_k)p_k + \nabla f(x_k)^t p_k + f(x_k) \\ \text{s.c} \\ \nabla g(x_k)^t p_k + g(x_k) \leq 0. \\ p_k \in \mathbb{R}^n \end{array} \right. \quad (3.40)$$

3.2.8 Algorithme de la méthode des contraintes actives

Calculer un point de départ admissible

Poser d'être le sous ensemble de contraintes active en

Pour $k = 0, 1, \dots$

Résoudre le problème (3.40) pour trouver p_k .

Si $p_k = 0$

Calculer les multiplicateurs de Lagrange $\bar{\lambda}_j$, Avec $\bar{w} = w_k$

si $\bar{\lambda}_j \geq 0$ pour tout $j \in w_k \cap I$
 Stop avec la solution $x^* = x_k$.

sinon $j \leftarrow \operatorname{argmin}_j \in W_{k \cap I} \bar{\lambda}_j$
 $x_{k+1} \leftarrow x_k$; $w_{k+1} \leftarrow w_k \setminus \{j\}$

Sinon ($p_k \neq 0$)

$x_{k+1} = x_k + p_k$

si

il y a de contraintes de blocage, obtenir w_{k+1} en ajoutant l'une des contraintes de blocage à w_k

sinon

$w_{k+1} = w_k$

Fin pour **Fin**.

Exemple 3.4. Considérons le problème non linéaire suivant :

$$\left\{ \begin{array}{l} \text{minimiser} \quad (x_1 - 1)^2 + (x_2 - 2.5)^2 \\ \text{s.c} \\ -x_1 + 2x_2 - 4 \leq 0 \\ x_1 + 2x_2 - 6 \leq 0 \\ x_1 - 2x_2 - 2 \leq 0 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{array} \right. \quad (3.41)$$

le problème équivalent est :

$$\left\{ \begin{array}{l} \text{minimiser} \quad (x_1 - 1)^2 + (x_2 - 2.5)^2 \\ \text{s.c} \\ -x_1 + 2x_2 - 4 \leq 0 \\ x_1 + 2x_2 - 6 \leq 0 \\ x_1 - 2x_2 - 2 \leq 0 \\ -x_1 \leq 0, \quad -x_2 \leq 0 \end{array} \right. \quad (3.42)$$

Soit le Lagrangien : $L(x, \lambda) = f(x) + \sum_{i=1}^5 \lambda_i g_i(x)$

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

Itération 0

On choisit un point de départ admissible $x^0 = (2, 0)$

$$\nabla f(x^0) = (2(x_1^0 - 1), 2(x_2^0 - 2.5)) = (2, -5)$$

$$\nabla g_1(x^0) = (-1, 2)$$

$$\nabla g_2(x^0) = (1, 2)$$

$$\nabla g_3(x^0) = (1, -2)$$

$$\nabla g_4(x^0) = (-1, 0)$$

$$\nabla g_5(x^0) = (0, -1)$$

$$g_1(x^0) = -6$$

$$g_2(x^0) = -4$$

$$g_3(x^0) = 0$$

$$g_4(x^0) = -2$$

$$g_5(x^0) = 0$$

Le modèle quadratique est :

$$\left\{ \begin{array}{l} \min \quad p_1^2 + p_2^2 - 2p_1 - 5p_2 + 7.25 \\ \quad \quad \quad s.c \\ \quad \quad \quad -p_1 + 2p_2 - 6 \leq 0 \\ \quad \quad \quad p_1 + 2p_2 - 4 \leq 0 \\ \quad \quad \quad p_1 - 2p_2 = 0 \\ \quad \quad \quad -p_1 \leq 0 \\ \quad \quad \quad -p_2 = 0 \end{array} \right. \quad (3.43)$$

L'ensemble de fonctionnement est $w_0 = \{3, 5\}$

Le sous problème est :

$$\left\{ \begin{array}{l} \min \quad p_1^2 + p_2^2 - 2p_1 - 5p_2 + 7.25 \\ \quad \quad \quad s.c \\ \quad \quad \quad p_1 - 2p_2 = 0 \\ \quad \quad \quad -p_2 = 0 \end{array} \right. \quad (3.44)$$

$$\text{donc : } p = (0, 0)$$

On évalue les multiplicateurs de Lagrange avec la formule :

$$A^t \lambda = -\nabla f(x^k)$$

$$\begin{pmatrix} 1 \\ -2 \end{pmatrix} \lambda_3 + \begin{pmatrix} 0 \\ -1 \end{pmatrix} \lambda_5 = - \begin{pmatrix} 2 \\ -5 \end{pmatrix}$$

On trouve $\lambda_3 = -2$, $\lambda_5 = -1$

$$\min \{\lambda_3, \lambda_5\} = \min \{-2, -1\} = -2 = \lambda_3$$

La contrainte (3) quitte l'ensemble de fonctionnement w_0

Itération 1

$$\begin{aligned} w_1 &= w_0 \setminus \{3\} = \{5\} \\ x^1 &= (2, 0) \end{aligned}$$

Le sous problème quadratique est :

$$\begin{cases} \min & p_1^2 + p_2^2 2p_1 - 5p_2 + 7.25 \\ & s.c \\ & -p_2 = 0 \end{cases} \quad (3.45)$$

donc : $p = (-1, 0)$

$$x^2 = x^1 + p = (1, 0)$$

Itération 2

l'ensemble de fonctionnement $\bar{w} = w_2 = w_1 = \{5\}$

Car il n'y a pas de contraintes de blocage au point x^2

$$\nabla f(x^2) = (0, -5)$$

$$\begin{aligned} g_1(x^2) &= -5 \\ g_2(x^2) &= -5 \\ g_3(x^2) &= -1 \\ g_4(x^2) &= 1 \\ g_5(x^2) &= 0 \end{aligned}$$

Le modèle quadratique est :

$$\begin{cases} \min & p_1^2 + p_2^2 - 5p_2 + 6.25 \\ & s.c \\ & -p_1 + 2p_2 - 5 \leq 0 \\ & p_1 + 2p_2 - 5 \leq 0 \\ & p_1 - 2p_2 - 1 = 0 \\ & -p_1 \leq 0 \\ & -p_2 = 0 \end{cases} \quad (3.46)$$

Le sous problème quadratique est :

$$\begin{cases} \min & p_1^2 + p_2^2 - 5p_2 + 6.25 \\ & s.c \\ & -p_2 = 0 \end{cases} \quad (3.47)$$

on trouve $p = (-1, 0)$

On évalue le multiplicateur de Lagrange, on trouve $\lambda_5 = -5$

La contrainte (5) quitte l'ensemble de fonctionnement

Itération 3

$$w_3 = \emptyset$$

On résout le problème sans contraintes pour obtenir p

$$\min p_1^2 + p_2^2 - 5p_2 + 6.25$$

On obtient $p = (0, 2.5)$

$$x^3 = (1, 2.5)$$

Au point x^3

il y a une contrainte de blocage qui est la contrainte (1),
donc $w_4 = w_3 \cup \{1\} = \{1\}$

Itération 4

$$\nabla f(x^4) = (0, 0)$$

Le sous problème quadratique est

$$\left\{ \begin{array}{l} \min p_1^2 + p_2^2 \\ \text{s.c} \\ -p_1 + 2p_2 = 0 \end{array} \right. \quad (3.48)$$

On obtient $p = (0, 0)$

On évalue le multiplicateur de Lagrange, on trouve $\lambda_1 = 0$.

On arrête, la solution est $x^* = (1, 2.5)$.

Chapitre 4

Introduction à l'optimisation globale

Introduction

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. Les algorithmes d'optimisation ont été développés dans pratiquement plusieurs problèmes économiques et industriels dans le but de maximiser ou 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 objets ou produits. Ainsi, l'optimisation entre en jeu dans beaucoup de domaines scientifiques : conception de moteurs électriques, changement d'orbite d'un satellite, météo, biologie-mathématique, génie des procédés chimiques ...

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

Les approches déterministes, comme leur nom l'indique, nous offrent la certitude d'obtenir l'optimum global et ne laissent aucune place au hasard et conduiront pour un contexte initial donné à une même solution finale. Pour ces méthodes, l'exploration de l'espace de solutions se fait grâce à des procédures de recherche qui sont élaborées à partir de la constante de Lipschitz ou celle de Hölder, des dérivées ou d'autres informations locales et globales concernant la fonction objectif. La théorie mathématique de l'optimisation globale est assez développée et possède de nombreuses applications importantes.

Dans le cas où la fonction objectif est donnée sous forme de "boite noire" le problème d'optimisation est particulièrement difficile. Les modèles déterministes ne traitent pas adéquatement les informations disponibles sur la fonction objectif.

Les approches stochastiques peuvent souvent faire face à ce genre de problèmes plus facilement et plus efficacement que les algorithmes déterministes. Les algorithmes stochastiques explorent l'espace de solutions grâce en partie à des procédures de transitions aléatoires. Ainsi, plusieurs exécutions successives de ces algorithmes, pourront conduire à des résultats différents (pour un même point initial).

4.1 Optimisation globale

Définition 4.1. On définit un problème d'optimisation globale par :

$$\begin{cases} \text{Min } f(x) \\ x \in D \end{cases} \quad (4.1)$$

Où f est une fonction de classe C^2 , D est un compact non vide.

4.1.1 Problème optimisation globale

Minimisation concave

Définition 4.2. définit un problème de minimisation concave par :

$$\begin{cases} \text{Min } f(x) \\ g_i(x) \leq 0, \quad i = 1, \dots, m. \\ x \in D \end{cases} \quad (4.2)$$

Où $f : A \rightarrow \mathbb{R}$ est concave sur l'ensemble A contenant D .

$D \subset \mathbb{R}^n$ est un ensemble non vide, fermé et convexe.

Une propriété importante de la minimisation concave est quand la fonction objectif est concave sur un domaine admissible convexe A , alors elle atteint toujours son minimum globale sur la frontière de A , comme l'indique le théorème suivant.

Théorème 4.1. [3]

Soit $f : D \rightarrow \mathbb{R}$ une fonction concave et $D \subset \mathbb{R}^n$ un ensemble compact et convexe, alors f atteint toujours son minimum globale sur la frontière de D .

4.1.2 Optimisation DC

Définition 4.3. Soit $C \subset \mathbb{R}^n$ un ensemble convexe, on dit qu'une fonction $h : C \rightarrow \mathbb{R}$ est DC (différence convexe) dans C s'il existe deux fonctions convexes : $p : C \rightarrow \mathbb{R}$ et $q : C \rightarrow \mathbb{R}$ tel que :

$$h(x) = p(x) - q(x), \forall x \in C. \quad (4.3)$$

Une inégalité $h(x) \leq 0$ est appelée une inégalité DC quand h est DC.

Définition 4.4. On définit un problème d'optimisation DC par :

$$\begin{cases} \text{Min } f(x) \\ g_i(x) \leq 0, \quad i = 1, \dots, m. \\ x \in C \end{cases} \quad (4.4)$$

Où $C \subset \mathbb{R}^n$ un ensemble convexe, et toutes les fonctions f, g_i , avec $i = 1, \dots, m$ sont DC.

4.1.3 Optimisation Lipshitzienne

Définition 4.5. Une fonction $h : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite Lipshitzienne sur un ensemble $M \subset \mathbb{R}^n$ s'il existe une constante réelle $L = L(h, M) > 0$. tel que :

$$\|h(x) - h(y)\| \leq L\|x - y\|, \quad \forall x, y \in M. \quad (4.5)$$

Supposons qu'une borne supérieure μ de L est convexe, alors on peut utiliser μ comme une constante de Lipshitz au lieu de L , car en général il est difficile de trouver une valeur exacte de L .

Propositions 4.1. [3] *Soit $X \subset \mathbb{R}^n$ un ensemble convexe, et f une fonction continument différentiable sur un ensemble ouvert contenant X avec un gradient borné sur X alors f est Lipschitzienne sur X avec la constante*

$$L = \sup\{\|\nabla f(x)\|, x \in X\}. \quad (4.6)$$

4.2 Les méthodes de résolution

4.2.1 La méthode de branch-and-bound

La méthode de Branch-and-Bound a été d'abord proposée par Land et Doig améliorée pour résoudre les problèmes de programmation linéaire mixtes. Elle a été élargie pour le cas des problèmes non linéaires mixtes par Dakin et améliorée pour résoudre des problèmes généraux, comme les problèmes d'optimisation globale.

La méthode branch-and-bound (appelée parfois dans la littérature francophone méthode de partition et évaluation), joue un rôle très important dans l'optimisation globale. L'idée générale est de partitionner le problème de départ en sous problèmes parallèles qui peuvent être graduellement plus faciles à manipuler, puis évaluer les bornes inférieures et supérieures des valeurs des solutions optimales de ces problèmes secondaires. Par conséquent, le procédé de branch-and-bound peut être représenté sous forme d'un arbre : le problème de départ est situé comme racine de cet arbre et les branches sont les sous problèmes hiérarchiquement construits. Cette construction est régie par la stratégie de la recherche qui déterminera la suite de solutions des problèmes secondaires ; la séquence de la décomposition et la recherche de la solution continue jusqu'à ce qu'il puisse vérifier que l'un ou l'autre sur la branche indiquée ne peut pas apporter une meilleure solution que celle déjà trouvée.

4.2.2 Principe de la méthode

L'algorithme de séparation et évaluation, plus connu sous son appellation anglaise Branch and Bound (B&B), repose sur une méthode arborescente de recherche d'une solution optimale par séparations et évaluations, en représentant les états solutions par un arbre d'états, avec des noeuds, et des feuilles.

Le branch-and-bound est basé sur trois axes principaux :

- L'évaluation,
- La séparation,
- La stratégie de parcours.

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. Le branch-and-bound 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 noeud parcouru à celui de la meilleure solution. Si le coût du noeud 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.

La séparation :

La séparation consiste à diviser le problème en sous-problèmes. 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 nœuds 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 nœuds actifs.

La stratégie de parcours :

- **La largeur d'abord :** Cette stratégie favorise les sommets les plus proches de la racine en faisant moins de séparations du problème initial. Elle est moins efficace que les deux autres stratégies présentées.
- **La profondeur d'abord :** Cette stratégie avantage les sommets les plus éloignés de la racine (de profondeur la plus élevée) en appliquant plus de séparations au problème initial. Cette voie mène rapidement à une solution optimale en économisant la mémoire.
- **Le meilleur d'abord :** Cette stratégie consiste à explorer des sous problèmes possédant la meilleure borne. Elle permet aussi d'éviter l'exploration de tous les sous-problèmes qui possèdent une mauvaise évaluation par rapport à la valeur optimale.

4.2.3 Algorithme de Branch-and-bound

Itération 0 : initialisation

Donner une tolérance $\epsilon = 10^{-6}$

Poser $S = S_0$ le plus petit polyèdre contenant D .

$S_{M_0} \subset D$ ensemble de points choisis dans l'ensemble admissible.

Calculer β_0, α_0 tel que

$$\alpha_0 = \min f(S_{M_0})$$

$$\beta_0 = \beta(S_0)$$

Si $\alpha_0 < +\infty$, alors

poser $x_0 = \operatorname{argmin} f(S_{M_0})$, i.e $f(x_0) = \alpha_0$

Si $\alpha_0 - \beta_0 \leq \epsilon$ stop.

la solution est ϵ -optimale.

Sinon, aller à l'itération k

Itération $k = 1, 2, \dots$

Etape 1 : partition

Diviser l'ensemble S_{k-1} en deux sous ensemble S_{k_1} et S_{k_2} .

Etape 2 : construction des bornes

Construire les bornes inférieures β_{k_1}, β_{k_2} et les bornes supérieures $\alpha_{k_1}, \alpha_{k_2}$.

Etape 3 Élimination

Poser $\beta_k = \min(\beta_{k_1}, \beta_{k_2}, \beta_{k-1}^{res})$

$$\alpha_k = \min(\alpha_{k_1}, \alpha_{k_2}, \alpha_{k-1}).$$

Éliminer tous les nœuds où

$$\beta_{ji} \geq \alpha_k; j = 1 \dots k, i = 1, 2$$

Etape 4 : test d'arrêt

Si $\alpha_k - \beta_k \leq \epsilon$ stop.

on a une solution ϵ -optimale globale.

Etape 5 : sélection

$\beta_k = \min(\beta_{k_1}, \beta_{k_2}, \beta_{k-1}^{res})$ on obtient le x_k^* correspondant à ce minimum,

Poser $k = k + 1$, aller à l'étape 1.

Remarque 1

La réalisation d'un algorithme Branch and bound dépend du choix des procédures suivantes :

- Division des ensembles S_{k_i} .
- Sélection des ensembles S_{k_i} .
- Estimation des bornes inférieures β_{k_i} .

Remarque 4.1. β_{k-1}^{res} Correspond aux bornes inférieures des sous ensembles qui ne sont ni subdivisés ni éliminés jusqu'à l'itération $(k-1)$.

Exemple 4.1. Considérons le problème de minimisation concave suivant :

$$\begin{cases} \text{minimiser} & -(x_1 - 20)^2 - (x_2 - 10)^2 \\ \text{s.c} & \\ & -\frac{1}{2}x_1 + x_2 \leq 10 \\ & x_1^2 + (x_2 - 10)^2 \leq 500 \\ & x_1 \geq 0, x_2 \geq 0 \end{cases}$$

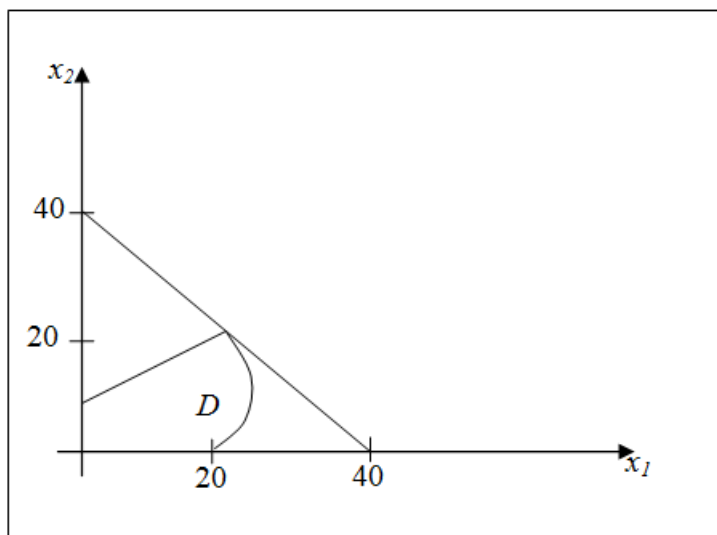


FIGURE 4.1 – Le domaine admissible D et le simplexe S_0

Itération 0

On choisit S_0 le simplexe conv $(0, 0), (0, 40), (40, 0)$ ayant les sommets $(0, 0), (0, 40), (40, 0)$

Pour obtenir les bornes inférieures et supérieures, nous considérons la fonction affine $\varphi(x) = a_1x_1 + a_2x_2 + a_3$

qui coincide avec les sommets de S_0 . Résolvant le système d'équations linéaires correspondant :

$$\begin{cases} a_3 = 500 \\ 40a_2 + a_3 = -1300 \\ 40a_1 + a_3 = -500 \end{cases}$$

On trouve $\varphi(x) = -20x_2 - 500$ en raison de concavité de f la fonction φ est un sous estimateur de f , alors on aura :

$$\varphi(x) \leq f(x), \forall x \in S_0$$

Une borne inférieure β_0 est obtenue en résolvant le problème d'optimisation convexe suivant (avec la fonction objective linéaire)

$$\begin{aligned} & \begin{cases} \text{minimiser } \varphi(x) \\ \text{s.c} \\ x \in S_0 \cap D = D \end{cases} \\ \Leftrightarrow & \begin{cases} \text{minimiser } -20x_2 - 500 \\ \text{s.c} \\ -\frac{1}{2}x_1 + x_2 \leq 10 \\ x_1^2 + (x_2 - 10)^2 \leq 500 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

La solution de ce sous problème est $x = (20; 20)$ On obtient $\beta_0 = -900$ (au point $(20, 20)$). Alors les points admissibles sont $(0,0)$ et $(20,20)$, et on pose $S_{M_0} = (0,0);(20,20)$, $\alpha_0 = \min f(S_{M_0}) = f(0,0) = -500 \leq \infty$
 $x^0 = (0, 0)$

$\alpha_0 - \beta_0 \geq \epsilon \Rightarrow$ la solution x^0 n'est pas optimale.

Itération 1

Etape 1 On divise S_0 en deux simplexes

$$S_{1,1} = \text{conv}(0, 0), (0, 40), (20, 20)$$

$$S_{1,2} = \text{conv}(0, 0), (20, 40), (40, 0)$$

Comme dans l'itération 0, nous construisons les bornes inférieures $\beta(S_{1,1})$ et $\beta(S_{1,2})$ en minimisant sur $S_{1,1} \cap D$ et $S_{1,2} \cap D$ les fonctions affines φ_{11} et φ_{12} qui coincident avec la fonction aux sommets de $S_{1,1}$ et $S_{1,2}$ respectivement.

On obtient

$$\varphi_{11}(x) = 40x_1 + 20x_2 - 500$$

$$\varphi_{12}(x) = 20x_1 - 500$$

En résolvant les deux sous problèmes suivant :

$$\begin{cases} \text{minimiser } \varphi_{11}(x) = 40x_1 + 20x_2 - 500 \\ \text{s.c} \\ -\frac{1}{2}x_1 + x_2 \leq 10 \\ x_1^2 + (x_2 - 10)^2 \leq 500 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

et

$$\left\{ \begin{array}{l} \text{minimiser } \varphi_{12}(x) = 20x_1 - 500 \\ \text{s.c} \\ -\frac{1}{2}x_1 + x_2 \leq 10 \\ x_1^2 + (x_2 - 10)^2 \leq 500 \\ x_1 \geq 0, x_2 \geq 0 \end{array} \right.$$

On obtient $\beta(S_{1,1}) = -700$ correspond à la solution (0,10) et $\beta(S_{1,2}) = -400$ correspond a la solution (0,0) ce qui implique que $\beta_1 = -700$

Les ensembles des points admissibles de $S_{1,1}$ et $S_{1,2}$ sont

$$S_{M_{1,1}} = (0; 0), (0; 10), (20; 20)$$

$$S_{M_{1,2}} = (0; 0), (20; 20)$$

$$\text{D'ou } \alpha(S_{M_{1,1}}) = \alpha(S_{M_{1,2}}) = f(0, 0) = -500 \text{ et } x^1 = (0, 0)$$

Etape 2

L'ensemble $S_{1,2}$ peut être supprimé puisque $\beta(S_{1,2}) = \alpha_1 = -500$

On devise $S_{1,1}$ en deux simplexes

$$S_{2,1} = \text{conv}(0, 0), (0, 20), (20, 20)$$

$$S_{2,2} = \text{conv}(0, 20), (0, 40), (20, 20)$$

De même façon, on calcule les sous fonctions $\varphi_{21}(x) = 20x_1 - 500$ la fonction $\varphi_{22}(x)$ n'a pas besoin d'être calculée, puisque $S_{2,2} \cap D = (20, 20)$

On obtient $\beta(S_{2,1}) = -500$ atteint au point (0, 0)

$$\beta(S_{2,2}) = -100 \text{ atteint au point } (20, 20)$$

d'où $\beta_2 = -500$

$$S_{M_{2,1}} = (0, 0), (0, 10), (0, 20), (20, 20), \alpha(S) = \min f(S_{M_{2,1}}) = -500$$

$$S_{M_{2,2}} = (20, 20), \alpha(S_{M_{2,2}}) = -100$$

d'où $\alpha_2 = -500$ et $x^2 = (0, 0)$ puisque $\alpha_2 = \beta_2$ on arrête, $x^2 = (0, 0)$ est la solution optimale.

4.3 Conclusion

Les codes d'optimisation globale sont de plus en plus demandés ; et cela dans beaucoup de domaines de conception et d'ingénierie. Plusieurs méthodes sont proposées pour répondre à cette demande : des méthodes spécifiques à une situation donnée, des méthodes à portée limitée à une classe de fonctions données, des méthodes utilisant les acquis en optimisation locale, des méthodes basées sur l'observation des comportements naturels (sur la physique, les statistiques, l'évolution), etc.

Malgré leur nombre important, on peut classer ces catégories principales, selon qu'elles incluent ou pas, des stratégies probabilistes : la classe des méthodes déterministes.

Pour toutes ces classes de méthodes, il subsiste encore des problèmes pour la validation de codes. En effet, il n'y a encore aucun test général reconnu décisif en dehors des fonctions tests de type académique ; et surtout il n'existe aucune convention établie sur des critères de performances à vérifier.

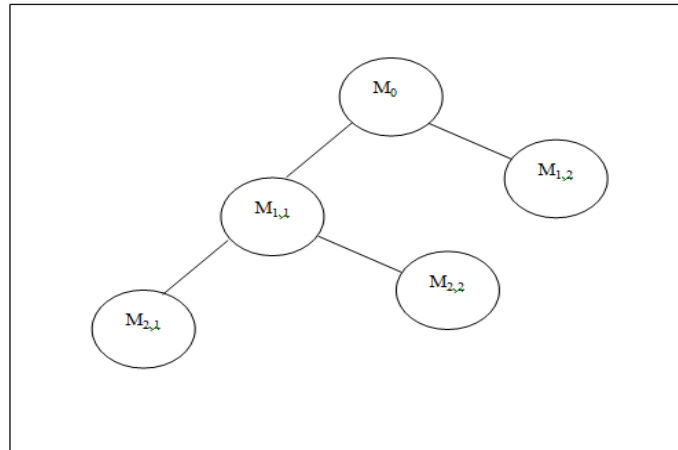


FIGURE 4.2 – L'arbre correspondant à l'exemple

Les problèmes de succès des méthodes déterministes, quant à eux, sont beaucoup plus liés aux stratégies employées pour échapper à un état de minimum local. En effet, ces stratégies, même si elles sont bien justifiées théoriquement mais leur implémentation est difficile et surtout leur complexité croît, au fur et à mesure, que se déroule l'exécution des codes.

Application

Raffinage des produits pétroliers

1- Problème

la raffinerie de Donges fabrique du butane, du carburant automobile, du gazole moteur et du fuel domestique à partir de deux pétrole bruts B_1 et B_2 . Pour fabriquer tous ces produits, le raffineur a quatre familles de traitement : séparation, conversion, amélioration et mélange. La phase de séparation consiste à distiller le produit brut afin d'obtenir, entre autres, du butane, de l'essence, du gazole et des distillats.

les distillats passent ensuite par une phase de conversion (craquage catalytique) qui permet d'obtenir des produits plus légers. Les différents produits issus de la distillation sont respectés pour respecter la réglementation portant sur les produits commercialisés. Le schéma simplifié de cette raffinerie est donné à la figure 1

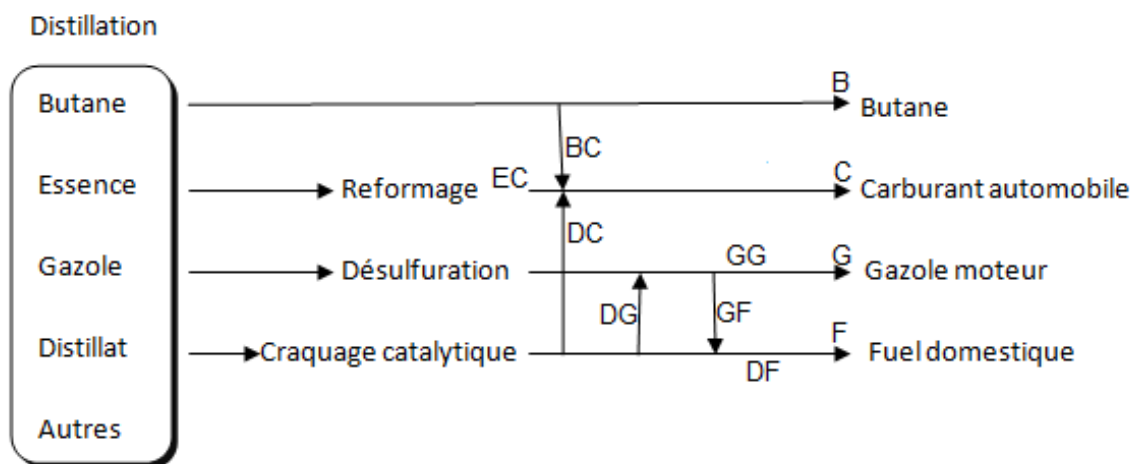


Figure 1- Schéma simplifié d'une raffinerie

Après distillation, le brut B_1 fournit 3 % de butane, 15 % d'essence, 40 % de gazole 15 % de distillat. Le brut B_2 fournit 5 % de butane, 20 % d'essence, 35 % de gazole, 10 % de distillat. Le carburant automobile est fabriqué à partir de trois ingrédients l'essence, le butane et le distillat. Le gazole moteur est obtenu par mélange de gazole et de distillat. Et le fuel domestique peut contenir en proportions libres du distillat et du gazole.

Les réglementations imposent, entre autres, certaines contraintes sur la qualité du carburant automobile et du gazole. Trois caractéristiques du carburant automobile sont importantes : l'indice d'octane, la tension de vapeur et la volatilité.

L'indice d'octane est une mesure de pouvoir antidétonant du carburant dans le moteur. La tension de vapeur est une mesure de risque d'explosion au stockage, notamment par temps chaud. La volatilité est une mesure de la facilité de démarrage du moteur par temps froid. Enfin, une quantité maximale soufre dans le gazole moteur est imposée par des normes antipollution. Les caractéristiques à respecter pour ces produits ainsi que celles des composants sont résumées au tableau 1. Les cases vides

signifient l'absence de norme particulière.

Tableau 1 - Caractéristiques des composants et produits finis

Caractéristiques	Butane	Essence reformé	Gazole désulfuré	Distillat	Carburant	Gazole moteur
Indice d'octane	120	100	–	74	≥ 94	–
Tension de vapeur	60	2.6	–	4.1	≤ 12.7	–
Volatilité	105	3	–	12	≥ 17	–

La raffinerie doit produire en un mois au moins 20000 tonnes de butane, 40000 tonnes de carburant automobile, 35000 tonnes de gazole moteur et 50000 tonnes de fuel domestique. Elle dispose pour cela de 250000 tonnes de brut B_1 et de 500000 tonnes de brut B_2 en stock. La capacité mensuelle de reformage est limitée à 30000 tonnes, celle de désulfuration à 40000 tonnes et celle de craquage à 50000 tonnes. Les coûts de reformage, de désulfuration et de craquage sont respectivement de 250 F, 450 F et 350 F par tonne.

Déterminer la composition de chaque produit fini de façon à minimiser les coûts de production de la raffinerie pour la période considérée, tout en respectant les contraintes de capacité et les réglementations portant sur ces produits.

2- Modélisation :

Soit B, C, G et F les quantités de butane, carburant automobile, gazole moteur et fuel à fabriquer respectivement. Nous devons déterminer la composition de chacun de ces produits finis. Pour cela, nous utilisons les variables suivantes : BC, EC et DC sont respectivement les quantités de butane, essence et distillat issus de la distillation et utilisées pour fabriquer du carburant automobile. De la même façon, nous noterons GG et DG respectivement les quantités de gazole et de distillat utilisées dans la fabrication gazole moteur, et GF et DF les quantités de gazole et distillat composant le fuel domestique.

Notre objectif est de minimiser les coûts de production comprenant les coûts de reformage, les coûts de désulfuration et les coûts de craquage catalytique. D'où la fonction objectif (1).

$$(1) \text{ Min } 250 \text{ EC} + 450 (\text{GG} + \text{GF}) + 350 (\text{DC} + \text{DG} + \text{DF})$$

Les contraintes (2), (3), (4), et (5) suivantes imposent des restrictions sur les quantités maximales de chaque composant :

$$(2) \text{ B} + \text{BC} \leq 0.03 B_1 + 0.05 B_2$$

$$(3) \text{ EC} \leq 0.15 B_1 + 0.20 B_2$$

$$(4) \text{ GC} + \text{GF} \leq 0.40 B_1 + 0.35 B_2$$

$$(5) \text{ DC} + \text{DG} + \text{DF} \leq 0.15 B_1 + 0.10 B_2$$

Ainsi, la contrainte (2) indique que la quantité totale de butane utilisée ($\text{B} + \text{BC}$) ne peut excéder la quantité maximale de butane issue de la distillation des deux bruts B_1 et B_2 ($0.03 B_1 + 0.05 B_2$).

Les contraintes (6) à (8) suivantes traduisent le fait que la quantité fabriquée de chaque produit fini est égale à la somme des quantités des ingrédients le composant. On notera que le butane n'apparaît pas dans ces contraintes, car il est fabriqué directement à partir du butane issu de la distillation des produits bruts.

$$(6) \text{ C} = \text{BC} + \text{EC} + \text{DC}$$

$$(7) \text{ G} = \text{GC} + \text{DG}$$

$$(8) \text{ F} = \text{GF} + \text{DF}$$

$$(9) \frac{120 \text{ BC} + 100 \text{ EC} + 74 \text{ DC}}{\text{C}} \geq 94.$$

Cependant, cette contrainte n'est pas linéaire puisqu'elle contient une variable en dénominateur. Pour la rendre linéaire, il suffit de la multiplier par C . On obtient alors la contrainte suivante (9').

$$(9') \quad 120 BC + 100 EC + 74 DC \geq 94C$$

Les contraintes (10) et (11) imposent respectivement que la tension de vapeur soit inférieure à 12.7 et la volatilité supérieure à 17 dans le carburant automobile, et que le taux de soufre soit inférieur à 0.5 dans le gazole moteur.

$$(10) \quad 60 BC + 2.6 EC + 4.1 DC \leq 12.7 C$$

$$(11) \quad 105 BC + 3 EC + 12 DC \geq 17 C$$

$$(12) \quad EC \leq 30000$$

$$(13) \quad GC + GF \leq 40000$$

$$(14) \quad DC + DG + DF \leq 50000$$

$$(15) \quad B \geq 20000$$

$$(16) \quad C \geq 40000$$

$$(17) \quad G \geq 35000$$

$$(18) \quad F \geq 50000$$

$$(19) \quad B, C, G, F, BC, EC, CG, GF, DC, DG, DF \geq 0$$

Enfin, les contraintes (12), (13) et (14) sont les contraintes de respect des capacités de production et les contraintes (15) à (18) les contraintes de satisfaction de la demande. A ce système de contraintes s'ajoutent les contraintes de positivité des variables (19). Le programme linéaire final est composé des lignes (1) à (8), (9'), (10) à (19).

Résolution d'un problème par "Solveur"

Introduction de "solveur"

Qu'est-ce que le Solveur ?

Le solveur est une fonction qui se présente dans EXCEL. C'est un outil très puissant qui permet à la fois d'optimiser et d'allouer des ressources. Cet outil est souvent utilisé pour résoudre des équations. En effet, il permet de trouver le minimum, le maximum ou la valeur la plus proche d'une donnée tout en respectant les contraintes que l'on a émises. Le solveur a donc le pouvoir de donner la meilleure solution, c'est-à-dire l'optimum.

En règle générale, le solveur est utilisé lorsque l'on recherche la valeur optimale d'une cellule donnée (la fonction économique) par l'ajustement des valeurs d'autres cellules (les variables) en respectant des conditions limitées par des valeurs numériques (les contraintes).

La programmation linéaire avec le Solveur Un programme linéaire est un problème d'optimisation qui maximiser ou minimiser une expression linéaire en les variables (x_1, x_2, \dots, x_n) sous des contraintes. Le Solveur d'EXCEL peut résoudre ces problèmes par l'algorithme simplexe. Le simplexe est une méthode de calcul basée sur la méthode de Gauss-Jordan pour la résolution de systèmes d'équations linéaires.

La programmation non linéaire avec le Solveur Lorsque le modèle n'est pas linéaire, le Solveur doit tenter dans ce cas, de trouver une solution par approximations successives appelées "itérations".

Qu'est-ce que la Valeur Cible ?

La Valeur Cible est une fonction d'EXCEL qui a le même objectif que le Solveur mais sans les contraintes. Cet outil est utilisé lorsqu'on veut qu'une cellule atteigne une valeur particulière. La

cellule doit obligatoirement contenir une formule. La Valeur Cible a besoin de 3 paramètres pour fonctionner :

la référence de la cellule pour laquelle on veut affecter la valeur particulière la valeur déterminée pour cette cellule la cellule variable qui doit être modifiée pour atteindre la valeur cible.

Application sur le problème

Dans l'illustration suivante, nous verrons que la première étape de toute décision utilisant Excel consiste à sélectionner la cellule contenant la forme générale de la fonction cible(fonction objectif).

SOMME									
=250*G4+450*H4+450*I4+350*J4+350*K4+350*L4									
	A	B	C	D	E	F	G	H	
1									
2									
3	X	B	C	G	F	BC	EC	GG	G
4			0	0	0	0	0	0	0
5									
6	z	=250*G4+450*H4+450*I4+350*J4+350*K4+350*L4							
7									
8	contrainte		0						

FIGURE 4.3 –

Ensuite, nous sélectionnons la fonction Solveur puis suivons les étapes ci-dessous.

Cellule cible à définir : on vous demande d'identifier la position de la fonction à laquelle vous souhaitez effectuer une opération. Dans notre exemple, la fonction avait été placée à la cellule B6.

Égale à : on vous demande d'identifier l'opération que vous souhaitez effectuer à la fonction située en B6 (max ? min ? valeur ?).

Cellules variables : on vous demande d'identifier les cellules qui contiennent les variables de votre fonction. Dans notre exemple, les cellules sont de B4 à L4.

Contraintes : nous définissons les contraintes en cliquant sur Ajouter puis en sélectionnant la cellule et en spécifiant sa forme(≤ 0 ou $= 0$ ou ≥ 0).

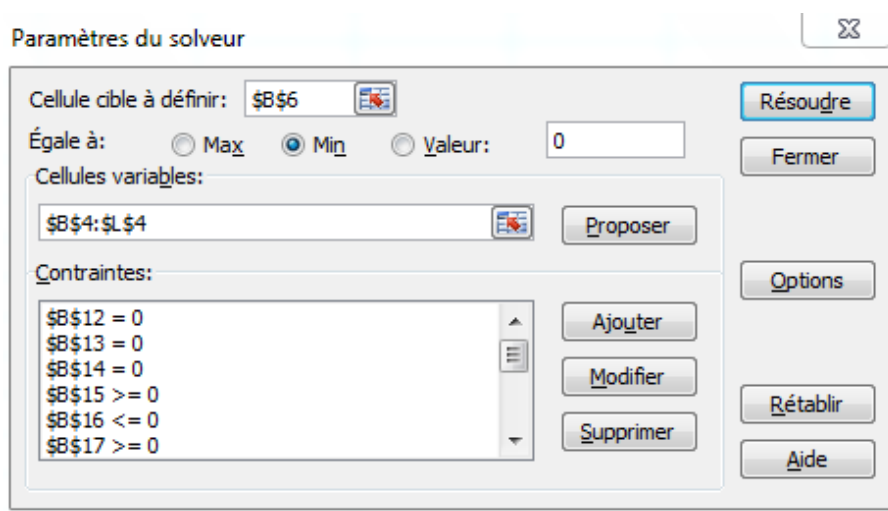


FIGURE 4.4 –

Après tout cela, nous appuyons sur "Résoudre", et trouvons les résultats finaux montrés dans la figure suivante :

Cellule cible (Min)

Cellule	Nom	Valeur initiale	Valeur finale
\$B\$6	z B	0	42118515,21

Cellules variables

Cellule	Nom	Valeur initiale	Valeur finale
\$B\$4	B	0	20000
\$C\$4	C	0	40000
\$D\$4	G	0	35000
\$E\$4	F	0	50000
\$F\$4	BC	0	6958,855098
\$G\$4	EC	0	30000
\$H\$4	GG	0	35000
\$I\$4	GF	0	3041,144902
\$J\$4	DC	0	3041,144902
\$K\$4	DG	0	0
\$L\$4	DF	0	46958,8551

FIGURE 4.5 – Solutions

Conclusion générale

Dans ce mémoire, nous avons exposé beaucoup de méthodes utilisées dans la résolution des programmes d'optimisation statique (par opposition programmes d'optimisation dynamique qui n'ont pas été traités ici).

Le cadre dit "convexe" de tels problèmes est celui où la fonction coût (fonction objectif) est convexe et l'ensemble admissible est convexe.

Dans le premier chapitre, on a choisi d'aborder l'étude des ensembles convexes après celle des fonctions convexes. On a vu qu'il est possible de passer de diverses façons des fonctions aux ensembles (en considérant les épigraphes ou les ensembles de niveau).

Dans le deuxième chapitre, on a abordé les problèmes d'optimisation convexe sans et avec contraintes et les problèmes d'optimisation quadratique et on a analysé les conditions d'existence et d'unicité, et les conditions d'optimalité et d'optimalité globale des problèmes abordés dans ce chapitre.

Dans le troisième chapitre, nous avons discuté des algorithmes permettant de résoudre des problèmes d'optimisation sans contrainte, la plupart de ces algorithmes exploitent les conditions d'optimalité dont on a vu qu'elles permettaient (au mieux) de déterminer des minima locaux, et pour le problème d'optimisation sous contraintes nous avons appliqué la méthode de simplexe si le problème est linéaire et la méthode de SQP si le problème est non linéaire et la méthode de gradient projeté si le problème est quadratique.

Dans le quatrième chapitre, nous avons fourni un aperçu de l'optimisation globale et l'optimisation DC et l'optimisation Lipshitzienne, avec présentation du méthode de Branch-and-bound et leur algorithme.

Bibliographie

- [1] **A.Rondepierre.** *Méthodes numériques pour l'optimisation non linéaire déterministe, Méthodes numériques pour l'optimisation non linéaire déterministe 4 ème année, INSA Toulouse 2017-2018.*
- [2] **B.GALERNE.** *Optimisation (MMLIE31) Notes de cours Master 1 Mathématiques et Modélisation (MM), Université Paris Descartes 2017-2018.*
- [3] **E.K.P. CHONG and S.H. ZAK.** *An Introduction to Optimization, Second Edition, WILEY-Interscience 2001.*
- [4] **J.GRENET.** *Vade mecum : Optimisation statique.Lagrangien et conditions de Kuhn et Tucker.TD d'Économie, École Normale Supérieure Année 2007-2008.*
- [5] **M.Bergounioux.** *Optimisation sans contraintes, Département de Mathématiques Orléans 2003-2004 .*
- [6] **M.MEROUANI** *Stabilité différentielle dans un problème d'optimisation paramétré ; mémoire de magister en mathématique Université d'Oran.*
- [7] **M.Z. ES-SADEK,** *Contribution à l'optimisation globale : approche déterministe et stochastique et application. thèse en cotutelle pour obtenir le grade de docteur en sciences appliquées et L'INSA de Rouen pour obtenir le grade de docteur de L'INSA DE Rouen, Université de Mohammed V-Agdal 21 Novembre 2009.*
- [8] **N.ABDESSEMED.** *Contribution à l' étude de la dualité quasi-convexe en optimisation, Option : Analyse Mathématiques Appliquée.Mémoire Présenté pour obtenir le diplôme de Magister En Mathématiques , Université El Hadj Lakhdar Batna 12-02-2007.*

Résumé

Dans ce travail; nous avons d'abord donné quelques rappels sur les concepts fondamentaux des matrices et des ensembles et des fonctions convexes. Ensuite, nous avons présenté les problèmes d'optimisation sans contraintes et sous contraintes et les conditions d'optimalités de ces problèmes. Après cela, nous nous sommes penchés sur la programmation linéaire et programmation non linéaire et les méthodes de les résoudre. Enfin, nous avons fait une étude générale sur l'optimisation globale et une petite application pour terminer.

Mots Clés : Ensembles et fonctions convexes, Problèmes d'optimisation convexes, Optimisation globale, Optimisation DC.

ملخص

في هذا العمل؛ قدمنا في البداية بعض التذكيرات حول المفاهيم الأساسية للمصفوفات والمجموعات والوظائف المحدبة. ثم قدمنا مشاكل التحسين دون قيود و تحت قيود والشروط الأفضل لهذه المشاكل. بعد ذلك انتقلنا إلى البرمجة الخطية والبرمجة غير الخطية وطرق حلها. وأخيرًا أجرينا دراسة عامة حول التحسين العالمي وتطبيقًا صغيرًا لنهاية العمل

كلمات مفتاحية : مجموعات و دوال محدبة، مشاكل تحسين محدبة، التحسين

العالمي، تحسين دي سي