

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

Université Mouloud Maameri de Tizi-Ouzou
Faculté des Sciences
Département de Mathématiques



THÈSE DE DOCTORAT ES SCIENCES

SPÉCIALISTE : Mathématiques

Option : Recherche Opérationnelle et Optimisation

Présentée par :

Mr CHEBBAH MOHAMMED

Sujet

RÉSOLUTIONS ET IMPLÉMENTATIONS DE PROBLÈMES EN OPTIMISATION GLOBALE.

Soutenue le : 07/07/2020.

Devant la commission d'examen composée de :

HAMADOUCHE DJAMAL	Professeur	U.M.M.T.O	Président
OUANES MOHAND	Professeur	U.M.M.T.O	Rapporteur
MOULAI MUSTAPHA	Professeur	U.S.T.H.B	Examineur
BOUROUBI SADEK	Professeur	U.S.T.H.B	Examineur
AIDER MEZIANE	Professeur	U.S.T.H.B	Examineur.

** Remerciements **

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

*Mes vifs et sincères remerciements vont à M^r OUANES Mohand (professeur à
UMMTO) qui, en acceptant de diriger ce travail, m'a permis de profiter de ses
conseils, son aide et ses encouragements, je lui témoigne ma respectueuse
gratitude.*

*Mes sincères remerciements s'adressent également aux membres du jury
pour avoir accepté d'examiner ce travail.*

En l'occurrence le professeur Hamadouche Djamel de l'UMMTO, les professeurs Moulai
Mustapha, Bouroubi Sadek et Aider Meziane de l'USTHB. *Je tiens également à remercier
tous ceux qui ont contribué de près ou
de loin à la réalisation et la réussite de ce travail.*

Merci beaucoup.

※ *Dédicaces* ※

C'est avec un énorme plaisir que je dédie ce travail :

*Aux deux personnes les plus nobles et les plus chères au monde :
Mon défunt Père et ma Mère qui ont sacrifié les plus belles années de
leurs vies pour me voir un jour réussir, et qui m'ont soutenu jusqu'à
la fin.*

*À mes frères et soeurs qui n'ont jamais
cessé de m'encourager.*

À mes chers granfs parents.

À toute ma famille.

À ma femme et mes enfants Rayan, Khadidja et Marya.

À mon directeur de thèse qui m'a vraiment été d'un grand soutien.

À mes très chers amis et camarades.

À tous ceux qui m'ont soutenu de près ou de loin.

MOHAMMED.

Table des matières

Table des matières	i
Table des figures	iv
Introduction générale	1
1 Convexité et Analyses Convexes	4
1.1 Introduction	4
1.2 Propriétés sur la convexité	5
1.2.1 Ensembles convexes	5
1.2.2 Propriétés des ensembles convexes	6
1.2.3 Polytopes et polyèdres convexes	6
1.3 Enveloppe convexe	7
1.4 Enveloppe affine	7
1.5 Enveloppe conique	7
1.6 Les fonctions convexes	8
1.6.1 Critères de Sylvester	11
1.6.2 Exemple	12
2 Optimisation et Théories	14
2.1 Classification des programmes mathématiques	15
2.1.1 Les programmes linéaires (la programmation linéaire) [16, 17]	16
2.1.2 Méthodes de résolution en programmation linéaire [100, 81, 16, 34, 17]	16
2.1.3 Exemple Illustratif de la programmation linéaire non différentiable	19
2.1.4 Exemple Illustratif (Implémentation) de la programmation linéaire	19
2.1.5 La programmation quadratique	24
2.1.6 Les programmes non linéaires (la programmation non linéaire) [11]	32
2.1.7 La programmation convexe [17, 34, 22, 81, 12, 9, 83]	41
2.1.8 La dualité en programmation convexe [17, 34, 22, 81]	45
2.1.9 La dualité en programmation mathématique non convexe [17, 34, 81]	48

3	Optimisation Globale et Contribution à l'Optimisation Globale	50
3.0.10	Introduction : L'optimisation globale [78, 86]	50
3.0.11	Domaines d'application de l'optimisation globale [34, 16]	50
3.0.12	Les techniques d'optimisation dans \mathbb{R} [34, 16, 17]	51
3.1	Problèmes pratiques en optimisation globale	56
3.2	Exemple	60
3.3	Méthodes de résolution en optimisation globale [68, 101, 95, 57]	63
3.4	Notre première (1°) contribution pour l'optimisation globale	71
3.5	Background	72
3.5.1	Programmation factorable, amélioration (relaxation)	72
3.5.2	La fonction borne inférieure dans la méthode α BB [11]	74
3.5.3	La fonction quadratique borne inférieure [51]	74
3.6	La nouvelle fonction borne inférieure	75
3.6.1	Test convexité/concavité	76
3.6.2	Pruning method	76
3.7	L'algorithme de Branch and Bound et sa convergence	78
3.7.1	Convergence	80
3.8	Résultats des calculs numériques	81
3.9	Conclusion	85
3.10	Perspectives	86
3.11	Notre deuxième (2°) contribution pour l'optimisation globale.	98
3.11.1	Motivations quant à l'utilisation de l'optimisation globale unidimensionnelle	99
3.11.2	La borne inférieure dans la méthode α BB [11]	100
3.11.3	La borne inférieure quadratique [51]	100
3.12	Nouvelle fonction borne inférieure	101
3.12.1	Test convexité/concavité	103
3.13	Algorithme et sa convergence	104
3.13.1	Algorithme	104
3.13.2	Convergence	105
3.14	Résultats numériques	105
3.15	Conclusion	106
4	Optimisation de systèmes dynamiques en contrôle optimal.	115
4.1	Introduction	115
4.2	Le cas discret	116
4.2.1	Position du problème	116
4.2.2	Notion de commandabilité	118
4.2.3	Accroissement de la fonctionnelle, critère d'optimalité et de suboptimalité.	119
4.2.4	Critère de suboptimalité	120
4.2.5	Méthode de résolution	120

4.2.6	Exemple	120
4.3	Le cas continu [8, 98, 48, 72, 23, 87, 76]	123
4.3.1	Position du problème	123
4.3.2	Définitions	123
4.3.3	Support et support-contrôle	124
4.3.4	Accroissement de la fonctionnelle et critères d'optimalité et suboptimalité.	125
4.3.5	Théorème (critère d'optimalité)	125
4.3.6	Théorème (critère de sub-optimalité)	125
4.3.7	Méthode de résolution	126
4.3.8	Changement du contrôle	126
4.3.9	Changement du support	127
4.3.10	Procédure finale	129
4.3.11	Conclusions et perspectives	130
4.3.12	Exemple	130
4.3.13	Exemple	130
4.3.14	Exemple	131
4.4	Conclusion	135

Conclusion générale **136**

A Généralités sur les fonctions à plusieurs variables **138**

A.1	Introduction	138
A.2	Généralités sur les fonctions à plusieurs variables et topologie. [16, 34, 38, 39, 70]	138
A.2.1	Éléments de la topologie	138
A.2.2	Les espaces normés	139
A.2.3	Les espaces métriques	140
A.2.4	Éléments de topologies sur \mathbb{R}^n	140
A.2.5	Les fonctions sur \mathbb{R}^n et théories	141
A.2.6	Limites et continuité	141
A.2.7	Dérivées partielles et directionnelles	141
A.2.8	Différentiabilité des fonctions	142
A.2.9	Matrice jacobienne	142
A.2.10	Formule de Taylor	143
A.2.11	Points critiques et extremas	144

Bibliographie **146**

Table des figures

1.1	Ensembles convexes et ensembles non convexes.	5
1.2	Représentation fonction f convexe.	9
1.3	Représentation épigraphe f convexe.	9
3.1	Trajectoires et commandes optimales avec la méthode indirecte.	61
3.2	Trajectoires et commandes optimales avec la méthode directe.	62
3.3	Trajectoires optimales.	62

Introduction générale

L'optimisation globale est de nos jours un sujet d'actualité par excellence, elle est une branche des mathématiques appliquées et de l'analyse numérique qui exhibe les minimas ou maximas globaux d'une fonction ou d'un ensemble de fonctions sur un ensemble donné avec multiplicité. Cette discipline partie intégrante de la recherche opérationnelle.

La recherche opérationnelle (R.O) [100, 81, 17] peut être définie comme l'ensemble des méthodes et techniques rationnelles orientées vers la recherche du meilleur choix dans la façon d'opérer en vue d'aboutir au résultat visé ou au meilleur résultat possible. Elle fait partie des « aides à la décision ». Dans la mesure où elle propose des modèles conceptuels en vue d'analyser et de maîtriser des situations complexes pour permettre aux décideurs de comprendre, d'évaluer les enjeux et d'arbitrer ou de faire les choix les plus efficaces. Ce domaine fait largement appel au raisonnement mathématique (logique, probabilités, analyse des données,...etc.) et à la modélisation des processus. Il est fortement lié à l'ingénierie des systèmes, ainsi qu'au management du système d'information entre autres.

L'optimisation [100, 81, 17, 16, 34] est une branche des mathématiques (branche de la recherche opérationnelle) cherchant à modéliser, à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à minimiser ou maximiser par exemple une fonction sur un ensemble [98, 72, 87, 23, 8, 62]. L'optimisation joue un rôle important en recherche opérationnelle (domaine à la frontière entre l'informatique, les mathématiques et l'économie). L'optimisation est 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.

L'optimisation globale a connu de nos jours, une avancée fulgurante en ce sens que beaucoup de papiers ont vu le jour dans ce contexte. Il n'y a pas si longtemps que ce domaine (l'optimisation globale) était à peine connu, d'ailleurs on peinait à trouver la meilleure solution dans le cadre de l'optimisation (programmation mathématique) en général. Donc pour chercher des solutions, les méthodes étaient le lagrangien augmenté, la méthode SQP, méthodes type Newton avec ses variantes, gradient réduit,...etc. Alors que la recherche de la multiplicité des solutions optimales paraissaient utopiques en général, cela occasionnant en outre des apports considérables en ressources (temps d'exécution, quantité de mémoires utilisées).

L'optimisation globale dans ce sens a ramené un nouveau souffle, de nouvelles orientations et surtout de nouvelles méthodes et a permis en même temps de résoudre les problèmes liés à l'optimisation citée plus haut.

Les travaux de l'équipe de PHAM DINH Tao et LE THI Hoai An (Metz France) [101], celle de Floudas (USA) [51], celle de Frédéric Messine et Jordan Ninin (Toulouse France) [78, 86] et d'autres [55, 99, 54] ont poussé la barre plus haut.

Dans le domaine de l'optimisation globale, on peut recenser deux types de méthodes.

Les méthodes de l'optimisation globale déterministes

Ces méthodes lorsque les problèmes d'optimisation sont bien définis, nous offrent la meilleure solution avec sa multiplicité. Les algorithmes utilisés sont très performants.

Les méthodes de l'optimisation globale non déterministes ou stochastiques

On peut citer par exemple : Branch and Bound stochastiques combinant : Colonies de fourmis, méthode Tabou, Essaim, méthode génétique,...etc.

Ces méthodes sont certes simples mais véhiculent des inconvénients à savoir, qu'on manipule des concepts qui ne cernent pas toute l'information du problème à résoudre.

Dans le cadre de l'élaboration de notre travail, on a érigé le plan des chapitres comme suit : L'annexe est consacré aux généralités sur les fonctions à plusieurs variables, des espaces normés en survolant la topologie vers la continuité et différentiabilité, deux éléments caractérisant les classes de fonctions. Les formules de Taylor proposant des approximations pour les fonctions. Enfin l'étude des extremas étroitement liée avec le critère de Sylvester.

Le premier chapitre est consacré aux propriétés des ensembles convexes vu leur importance dans la théorie de l'optimisation.

Ainsi sont abordées les enveloppes convexes, affines et coniques.

Les propriétés des fonctions convexes/concaves vu leur apport aussi dans la théorie de l'optimisation, ainsi nous évoquons les lemmes de Farkas et lemme de Motzkin.

Le deuxième chapitre évoque l'optimisation et sa théorie. Les programmes mathématiques sont classés.

Les programmes linéaires ont un impact direct et très significatif en matière de l'optimisation globale. L'algorithme de Karmarkar est présenté avec des extensions.

Les programmes quadratiques parmi les outils les plus privilégiés en matière de l'optimisation en général. Dualité et algorithmes de résolution évoqués cas contraint ou pas.

Le troisième chapitre présente l'optimisation globale et nos apports dans ce contexte.

Des exemples pratiques de la vie représentés ou modélisés par des problèmes en optimisation globale. Nous décelons les techniques de résolution en optimisation globale dans \mathbb{R} ou \mathbb{R}^n avec des algorithmes de résolution importants.

Notre première contribution publiée dans (A.I.P) [24], puis notre deuxième contribution dans le journal RAIRO [90] figurent aussi.

Dans le quatrième chapitre nous présentons une étude de cas sous forme d'un logiciel sur

le contrôle optimal et la technique de discrétisation (très utile en semi-infini), présentée et sélectionnée en deux reprises aux conférences internationales de Marakech 2013 / Algerian-Turkish day Annaba 2012. Ce logiciel trouve sa place après les techniques de linéarisation et convexification afin d'élaborer des méthodes simples possédant des atouts d'un bon comportement numérique, dans le cadre de l'optimisation globale.

CHAPITRE 1

Convexité et Analyses Convexes

1.1 Introduction

Le nombre de publications qui apparaissent chaque année dans le cadre de cette thématique est énorme. La convexité est très utilisée et en développement constant, vu qu'elle occupe des places de choix, comme par exemple en algèbre, analyse mathématique ou l'optimisation globale. Le mot convexe désignant dans la réalité plusieurs concepts selon son utilisation. En optique géométrique, spécialement pour désigner des miroirs ou des lentilles. Dans la langue courante en société le mot « convexité » est rattaché au concept mathématique d'ensemble convexe, pour ainsi la convexité d'un objet désignant la partie de ce dernier qui a une forme bombée. En économie, la convexité est un indicateur économique de risque de taux directement lié au concept mathématique de fonction convexe.

En mathématiques, le mot « convexe » est utilisé dans le but de désigner deux notions bien distinctes en général :

* / Formes géométriques, autrement-dit ensembles convexes.

* / Fonctions, au quel cas cela nous renvoie au concept de fonctions convexes.

Nous introduisons dans ce chapitre par exemple les principaux outils mathématiques nécessaires pour mener à bien le développement de notre étude. L'étude de la programmation convexe pour notre cas, est plus que primordiale. Nous nous intéressons dans le cadre de notre travail à l'optimisation globale ; et dans ce contexte bien précis beaucoup de méthodes tentent de reformuler les problèmes en optimisation globale en des problèmes d'optimisation convexe. Se dire que la programmation convexe est une plaque tournante dans le domaine de l'optimisation globale entre autres.

Notre domaine d'intérêt est donc entre autres :

- Les ensembles convexes et fonctions convexes [17].
- Les notions sur l'invéxité et fonctions invéxes [95].
- Les notions sur l'anti-convexité et ensembles anti-convexes [101].
- Les outils de la programmation convexe [101].

- Les conditions d’optimalité en programmation non linéaire [34].
- Les conditions d’optimalité type K.K.T et conséquences [17, 104].
- Les résultats de la dualité lagrangienne [34].
- Les utilisations de l’optimisation locale dans l’optimisation globale [101, 95, 57].
- Passages optimisation globale vers optimisation convexe, notions et techniques de convexification, reformulation et applications [16, 101, 95, 57].

1.2 Propriétés sur la convexité

La convexité est le noyau de l’optimisation, outil pour la recherche des conditions nécessaires et suffisantes ainsi que c’est un élément phare dans l’optimisation globale.

1.2.1 Ensembles convexes

Définition 1.2.1. Un ensemble E de \mathbb{R}^n est dit convexe, si $\forall \lambda \in [0, 1], \lambda x + (1 - \lambda)y \in E$

$$\forall x, y \in E.$$

Cela est illustré comme suit :

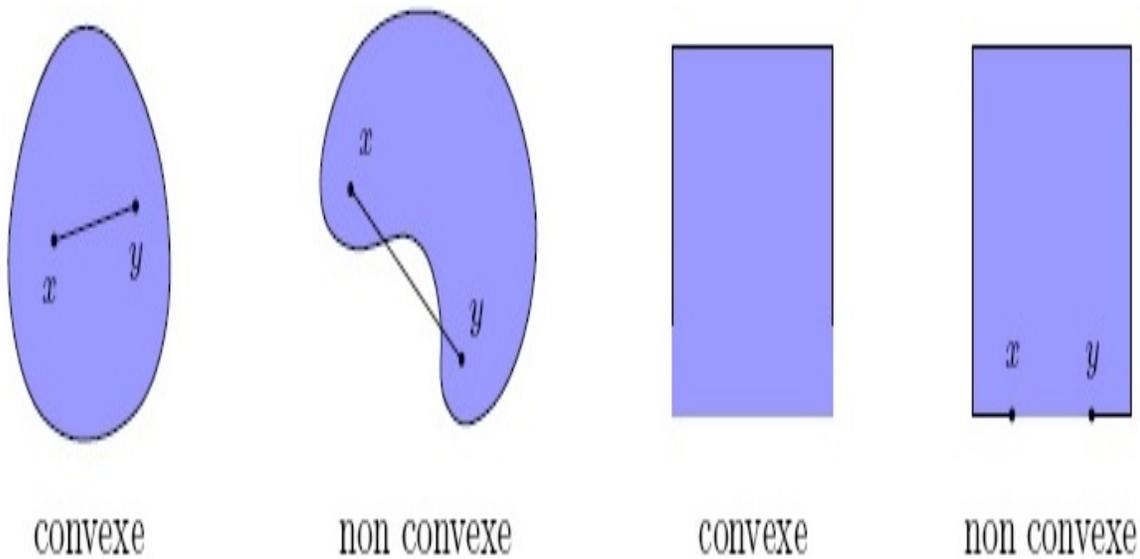


FIGURE 1.1 – Ensembles convexes et ensembles non convexes.

Remarque 1.2.1. En général, étant donné p points x_1, x_2, \dots, x_p de \mathbb{R}^n , on dit que $x \in \mathbb{R}^n$ est une combinaison convexe de ces p points, s’il existe des $\lambda_1, \lambda_2, \dots, \lambda_p$ tel que $x = \sum_{i=1}^p \lambda_i x_i$ avec

$$\sum_{i=1}^p \lambda_i = 1 \text{ et } \lambda_i \geq 0, i = 1, 2, \dots, p.$$

Et donc E est convexe, si pour toute combinaison convexe d'éléments de E , celle ci (combinaison convexe) est toujours dans E .

1.2.2 Propriétés des ensembles convexes

Soient E_1 et E_2 deux ensembles convexes de \mathbb{R}^n alors $C = E_1 \cap E_2$ est convexe et $K = \{x | x = x_1 + x_2, x_1 \in E_1 \text{ et } x_2 \in E_2\}$ est aussi un ensemble convexe.

Si E ensemble convexe de \mathbb{R}^n et $\lambda \in \mathbb{R}$ alors $C = \{x | x = \lambda x_*, x_* \in E\}$ est convexe.

1.2.3 Polytopes et polyèdres convexes

Définition 1.2.2. Soit $\lambda \in \mathbb{R}^n$ non nul. Un hyperplan de \mathbb{R}^n de dimensions $n - 1$, est l'ensemble $H = \{x \in \mathbb{R}^n | \lambda'x = \alpha\}$, $\alpha \in \mathbb{R}$.

L'hyperplan H divise l'espace \mathbb{R}^n en deux demi-espaces fermés comme suit : $H^+ = \{x \in \mathbb{R}^n | \lambda'x \geq \alpha\}$ et $H^- = \{x \in \mathbb{R}^n | \lambda'x \leq \alpha\}$. Avec H^+ et H^- convexes.

Définition 1.2.3. L'intersection d'un nombre fini de demi-espaces fermés est un polyèdre convexe. Un polyèdre borné est un polytope.

Définition 1.2.4. Soit E un ensemble convexe de \mathbb{R}^n non vide, x de E est un point extrême (sommet), s'il ne peut pas s'écrire comme combinaison convexe de deux points quelconques de E . Autrement-dit

$$\forall x_1, x_2 \in E, \forall \lambda \in [0, 1] : x = \lambda x_1 + (1 - \lambda)x_2 \implies x = x_1 = x_2.$$

Définition 1.2.5. Soit E un ensemble de \mathbb{R}^n non vide, est appelé cône de sommet zéro si :

$$\forall x \in E \implies \lambda x \in E, \forall \lambda \geq 0.$$

Une classe des cônes convexes est les cônes polaires.

Définition 1.2.6. Soit E un ensemble de \mathbb{R}^n non vide, le cône polaire de E est noté E^* défini comme suit :

$$E^* = \{\alpha | \alpha'x \leq 0, x \in E\}.$$

Définition 1.2.7. Soit E un ensemble fermé et convexe, le cône normal de E en a est défini comme suit :

$$\mathbb{N}(a) = \{x \in \mathbb{R}^n | x'(y - a) \leq 0, \forall y \in E\}.$$

Définition 1.2.8. Le cône tangent à E (fermé et convexe) en a , est le polaire du cône normal en a et qui est :

$$\mathbb{T}(a) = \mathbb{N}^*(a) = \{\alpha | \alpha = \lim_{y \rightarrow a} \lambda(y - a), \lambda \geq 0, y \in E\}.$$

Définition 1.2.9. On appelle simplexe unité de \mathbb{R}^n l'ensemble défini par S :

$$S = \{x \in \mathbb{R}^n | e'x = 1, x \geq 0\}.$$

$e' = (1, \dots, 1)$ de \mathbb{R}^n . C'est un convexe (S) (intersection de deux convexes : un sous espace affine et l'orthant positif). La dimension de S est $n - 1$.

1.3 Enveloppe convexe

Soit P une partie de E (espace vectoriel). L'intersection de convexes étant convexe, on peut parler du plus petit convexe contenant P , qui est donc l'intersection de tous les convexes contenant P . C'est ce que l'on appelle l'enveloppe convexe de P . On la note

$$\text{conv } P = \cap \{C \mid C \text{ est un convexe contenant } P\}.$$

Proposition 1.3.1. 1/ Un ensemble est convexe, si et seulement si il contient toutes les combinaisons convexes de ses éléments.

2) Si $P \subset E$, alors $\text{conv } P$ est l'ensemble des combinaisons convexes des éléments de P .

$$\text{conv } P = \left\{ \sum_{i=1}^m t_i x_i, t_i \geq 0, \sum_{i=1}^m t_i = 1, m \in \mathbb{N}^*, x_i \in P \right\}.$$

1.4 Enveloppe affine

Soit P une partie d'un espace vectoriel E . L'intersection de sous-espaces affines étant un sous-espace affine, on peut parler du plus petit sous-espace affine contenant P , qui est donc l'intersection de tous les sous-espaces affines contenant P . C'est ce que l'on appelle l'enveloppe affine de P . On la note

$$\text{aff } P = \cap \{A \mid A \text{ est un sous-espace affine contenant } P\}.$$

Proposition 1.4.1. 1) Un ensemble est un sous-espace affine si et seulement si il contient toutes les combinaisons affines de ses éléments.

2) Si $P \subset E$, alors $\text{aff } P$ est l'ensemble des combinaisons affines des éléments de P :

$$\text{aff } P = \left\{ \sum_{i=1}^m t_i x_i, t_i \in \mathbb{R}, \sum_{i=1}^m t_i = 1, m \in \mathbb{N}^*, x_i \in P \right\}.$$

1.5 Enveloppe conique

On dit qu'une partie K d'un espace vectoriel E est un cône si $tK \subset K$, pour tout $t \geq 0$, autrement dit si $tx \in K$ dès que $x \in K$ et $t \geq 0$. Soit P une partie de E . L'intersection de cônes convexes étant un cône convexe, on peut parler du plus petit cône convexe contenant P , qui est donc l'intersection de tous les cônes convexes contenant P . C'est ce que l'on appelle l'enveloppe conique de P (on devrait dire son enveloppe conique convexe). On la note

$$\text{cône } P = \cap \{K \mid K \text{ est un cône convexe contenant } P\}.$$

Proposition 1.5.1. 1) Un ensemble est un cône convexe, si et seulement si il contient toutes les combinaisons coniques de ses éléments.

2) Si $P \subset E$, alors $\text{cône } P$ est l'ensemble des combinaisons coniques des éléments de P :

$$\text{c\^one } P = \left\{ \sum_{i=1}^m t_i x_i, t_i \in \mathbb{R}, t_i \geq 0, m \in \mathbb{N}^*, x_i \in P \right\}.$$

1.6 Les fonctions convexes

Définition 1.6.1. Une fonction réelle f sur un convexe C de \mathbb{R}^n , est dite convexe, si pour tout point x_1 et x_2 de C et pour tout $\lambda \in [0, 1]$, on a :

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

f est dite strictement convexe si l'inégalité est stricte pour $x_1 \neq x_2$ et $\lambda \in]0, 1[$.
 f est dite affine si :

$$f(\lambda x_1 + (1 - \lambda)x_2) = \lambda f(x_1) + (1 - \lambda)f(x_2), \forall \lambda \in \mathbb{R}.$$

Propriété 1.6.1. Soit f une fonction réelle convexe sur un convexe $C \subset \mathbb{R}^n$ alors

- ♣ La fonction $-f$ est concave.
- ♣ La somme d'un nombre fini de fonctions convexes (comme f) est convexe.
- ♣ La fonction f est dite concave si $-f$ est convexe.

Définition 1.6.2. Une fonction réelle f sur un convexe C de \mathbb{R}^n , est dite quasi convexe, si et seulement si l'ensemble

$$\{x \in C, f(x) \leq \alpha, \forall \alpha \in \mathbb{R}\}$$

est convexe.

Propriété 1.6.2. Soit f une fonction réelle convexe sur un convexe $C \subset \mathbb{R}^n$, alors la fonction f est aussi quasi convexe.

Propriété 1.6.3. Soit f une fonction réelle sur $C \subset \mathbb{R}^n \rightarrow \mathbb{R}$, de classe C^1 .

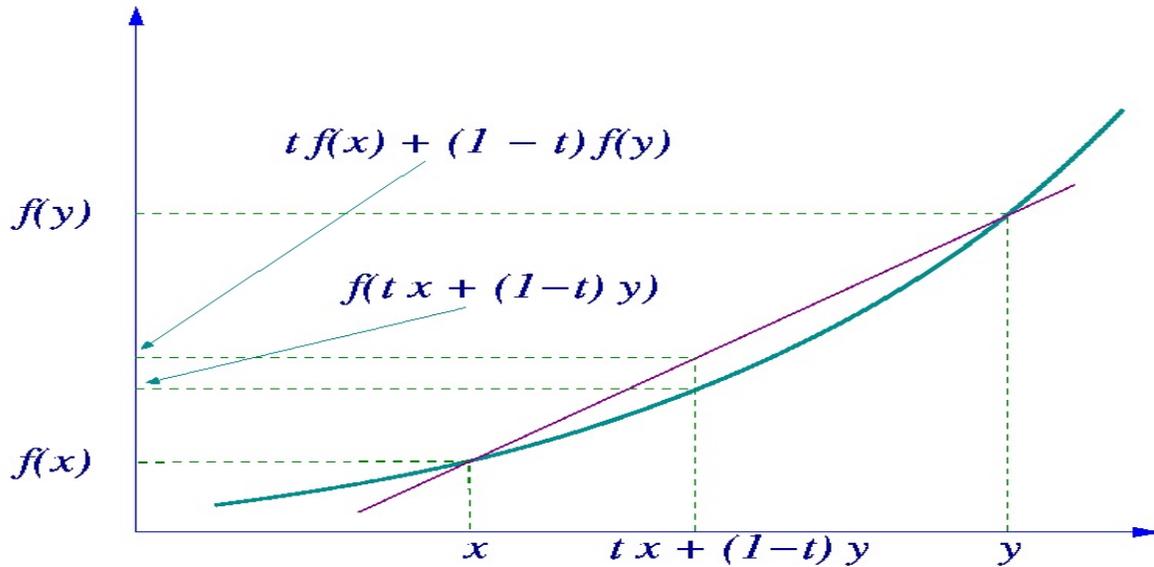
Si f est fortement convexe de rapport $\lambda > 0$ [34] alors

$$\forall x, y \in C, f(y) \geq f(x) + (y - x)' \nabla f(x) + \frac{\lambda}{2} \|x - y\|^2.$$

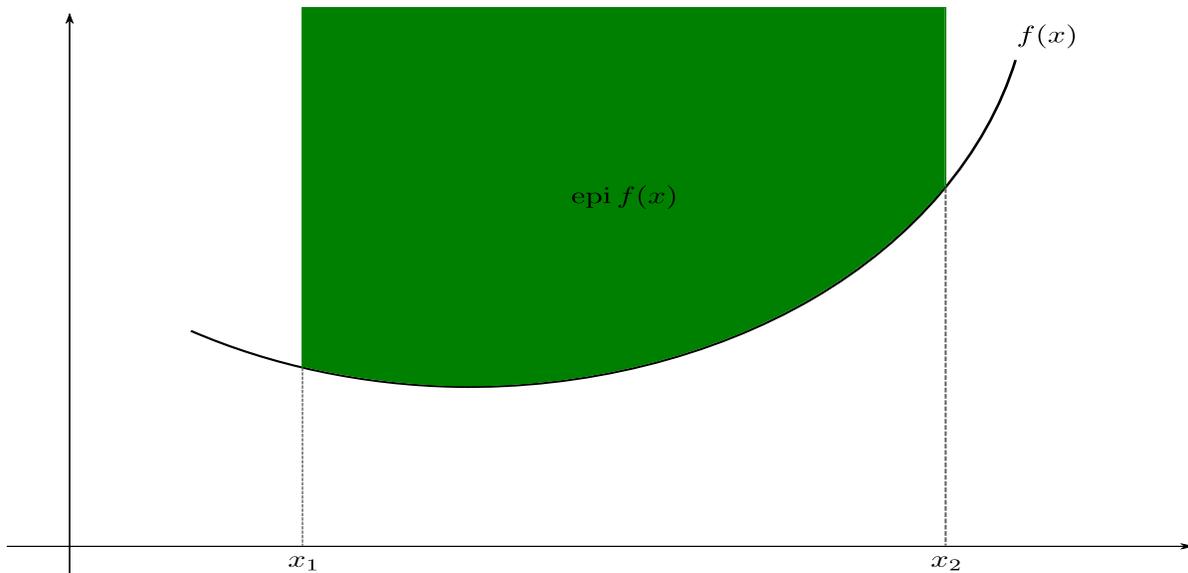
Si f est à gradient lipschitzien de constante $L > 0$ [34] alors

$$\forall x, y \in C, f(y) - f(x) \leq (y - x)' \nabla f(x) + \frac{L}{2} \|y - x\|^2.$$

Remarque 1.6.1. À Noter aussi l'existence des définitions de fonctions pseudo convexes, quasi convexes à l'instar de pseudo concaves et quasi concaves et entre autres quasi linéaires.

FIGURE 1.2 – Représentation fonction f convexe.

Propriété 1.6.4. Soit f une fonction réelle sur un convexe $C \subset \mathbb{R}^n$, alors la fonction f est convexe si et seulement si son épigraphe $\text{epi}(f) = \{(x, r) : x \in C, r \geq f(x)\} \subset \mathbb{R}^n \times \mathbb{R}$ est aussi convexe.

FIGURE 1.3 – Représentation épigraphe f convexe.

Propriété 1.6.5. Soit f une fonction réelle sur un convexe $C \subset \mathbb{R}^n$, de classe C^1 , alors la fonction f est convexe si et seulement si

$$f(x_2) - f(x_1) \geq (x_2 - x_1)' \nabla f(x_1), \forall x_1, x_2 \in C.$$

Propriété 1.6.6. Soit f une fonction réelle sur un convexe $C \subset \mathbb{R}^n$, de classe C^2 , alors la fonction f est convexe si et seulement si $(x_2 - x_1)'H(x_1)(x_2 - x_1) \geq 0, \forall x_1, x_2 \in C$, avec $H(x_1)$ le Hessien au point x_1 .

Propriété 1.6.7. Soit f une fonction réelle sur un convexe $C \subset \mathbb{R}^n$ ouvert, de classe C^2 , alors la fonction f est convexe si et seulement si

$$H(x) \geq 0, \forall x \in C.$$

Propriété 1.6.8. Soit f une fonction réelle sur un convexe $C \subset \mathbb{R}^n$, de classe C^1 , alors tout $x^* \in C$ tel que :

$$\forall x \in C, (x - x^*)'\nabla f(x^*) \geq 0,$$

est minimiseur global de f et réciproquement.

Exemple 1.6.1. La fonction f suivante est convexe de $\mathbb{R}_+^{*n} \rightarrow \mathbb{R}$.

$$f(x) = \sum_{i=1}^n x_i \log x_i - \left(\sum_{i=1}^n x_i \right) \log \left(\sum_{i=1}^n x_i \right).$$

Définition 1.6.3. Une forme généralisée de la convexité est la notion de l'invéxité, en ce sens que beaucoup de papiers ont vu le jour autour de cette notion.

(Hanson 1981) [95].

Soit D un ouvert non vide de \mathbb{R}^n , η une fonction vectorielle et f une fonction réelle.

$$\eta : D^2 \rightarrow \mathbb{R}^n; f : D \rightarrow \mathbb{R}.$$

La fonction f est dite invéxe au point x_0 de D par rapport à η si f est différentiable au point x_0 et de plus, pour tout $x \in D$

$$f(x) - f(x_0) \geq \eta(x, x_0)'\nabla f(x_0).$$

Si l'inégalité ci-dessus est stricte ($x \neq x_0$), alors on parlera de l'invéxité stricte.

Si on se place dans les problèmes invéxes, il y a l'impératif à considérer une fonction η ou plusieurs, pour les fonctions objectifs et contraintes. Pour cela des difficultés apparaissent dans les applications des conditions d'optimalité, à la dualité des problèmes de programmation non linéaire et multi-objectifs. En considérant l'invéxité généralisée ou de ses extensions, à ce moment de nouvelles conditions nécessaires et suffisantes de type Kuhn-Tucker (Fritz-John), des caractérisations des solutions et plusieurs résultats de dualité sont déduits pour des problèmes non-linéaires et multi-objectifs différentiables ou non différentiables avec contraintes d'inégalité

par exemples. L'étude de l'invéxité est importante dans les domaines de la R.O, l'optimisation, les mathématiques appliquées, l'ingénierie, etc.

Le théorème de Farkas joue un rôle important dans le cadre de l'obtention des conditions d'optimalité du premier ordre en optimisation linéaire et non linéaire, c'est aussi le cas pour le lemme de Motzkin utilisé en optimisation non linéaire par exemple.

Théorème 1.6.1. (Lemme de Farkas)

Soit A une matrice de type (m, n) et $c \in \mathbb{R}^n$. Alors exactement un des systèmes suivants admet une solution.

$$S_1 : Ax \leq 0, c'x > 0.$$

$$S_2 : A'y = c, y \geq 0.$$

Un lemme analogue, dans le même style.

Théorème 1.6.2. (Lemme de Motzkin)

Soit A_1 une matrice de type (m_1, n) , A_2 une matrice de type (m_2, n) et $c \in \mathbb{R}^n$. Alors exactement un des systèmes suivants admet une solution.

$$S_3 : A_1x = 0, A_2x \leq 0, c'x > 0.$$

$$S_4 : A'_1y_1 + A'_2y_2 = c, y_2 \geq 0.$$

1.6.1 Critères de Sylvester

Le critère de Sylvester présente un intérêt important, il permet de caractériser une forme quadratique.

Soit la matrice symétrique suivante :

$$D = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{pmatrix}.$$

Le mineur de la matrice D , formé des lignes i_1, i_2, \dots, i_p et des colonnes j_1, j_2, \dots, j_p est donné comme suit :

$$D = \begin{pmatrix} i_1, i_2, \dots, i_p \\ j_1, j_2, \dots, j_p \end{pmatrix} = \begin{vmatrix} d_{i_1 j_1} & \cdots & d_{i_1 j_p} \\ d_{i_2 j_1} & \cdots & d_{i_2 j_p} \\ \vdots & \cdots & \vdots \\ d_{i_p j_1} & \cdots & d_{i_p j_p} \end{vmatrix}.$$

Ce mineur est dit principal si on satisfait que $i_1 = j_1, i_2 = j_2, \dots, i_p = j_p$, c'est à dire formé de lignes et de colonnes portant les mêmes numéros.

Les mineurs suivants :

$$D_1 = d_{11}, D_2 = \begin{vmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{vmatrix}, \dots, D_n = \begin{vmatrix} d_{11} & \dots & d_{1n} \\ d_{21} & \dots & d_{2n} \\ \vdots & \dots & \vdots \\ d_{n1} & \dots & d_{nn} \end{vmatrix}.$$

Sont appelés des mineurs principaux successifs. Et donc le critère de Sylvester se présente comme suit :

Théorème 1.6.3. (Critère de Sylvester)

I / Pour qu'une matrice D soit définie positive ($D > 0$), il est nécessaire et suffisant que les mineurs principaux successifs de D soient positifs.

$$D_1 > 0, D_2 > 0, \dots, D_n > 0.$$

II / Pour qu'une matrice D soit semi-définie positive ($D \geq 0$), il est nécessaire et suffisant que les mineurs principaux de D soient non négatifs.

$$D = \begin{pmatrix} i_1, i_2, \dots, i_p \\ i_1, i_2, \dots, i_p \end{pmatrix} \geq 0, 1 \leq i_1 < i_2 < \dots < i_p, p = 1, 2, \dots, n.$$

Remarque 1.6.2. La condition

$$D_1 \geq 0, D_2 \geq 0, \dots, D_n \geq 0.$$

N'est pas suffisante pour que la matrice D soit semi-définie positive.

1.6.2 Exemple

Soit la matrice D tel que :

$$D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & -3 \end{pmatrix}.$$

Et donc

$$D_1 = 0, D_2 = 0.$$

Cela ne suffit pas car la forme quadratique F associée à la matrice a la forme suivante (par exemple) :

$$F(x) = d_{11}x_1^2 + 2d_{12}x_1x_2 + d_{22}x_2^2 = -3x_2^2.$$

Remarque 1.6.3. A noter que :

L'existence d'un théorème équivalent (un générique) à celui du ("critère de Sylvester"), en usant des valeurs propres. Consulter [70, 34, 17].

Exemple 1.6.2. Soit le problème suivant :

$$\begin{cases} xy \rightarrow \min \\ 4x^2 + y^2 = 4 \\ (x, y)' \in \mathbb{R}^2. \end{cases}$$

(1.1)

Le problème est assez intéressant à résoudre car celui-ci n'est pas convexe. Par application du théorème (condition nécessaire) d'analyse mathématique donné plus bas (annexe). C'est à dire les facteurs de lagrange, vont aboutir aux résultats :

La première ligne représentant les points critiques par les facteurs de Lagrange.

(x, y)	$(0, 2)$	$(0, -2)$	$(\frac{\sqrt{2}}{2}, \sqrt{2})$	$(\frac{\sqrt{2}}{2}, -\sqrt{2})$	$(-\frac{\sqrt{2}}{2}, \sqrt{2})$	$(-\frac{\sqrt{2}}{2}, -\sqrt{2})$
$f(x, y)$	0	0	1	-1	-1	1

Nous avons par exemples deux minimums globaux, on peut facilement le vérifier. Cela géométriquement ou par utilisation de logiciels comme : Lingo, Matlab,...etc.

CHAPITRE 2

Optimisation et Théories

Remarque 2.0.4. Pour le problème ci-dessous :

$$\left\{ \begin{array}{l} \frac{-x_1^2+5x_1-2x_1x_2+6x_2-4x_1x_3+7x_3-x_2^2-4x_2x_3-x_3^2-6}{x_1^2+x_2^2+1+x_3^2+x_1x_2} + \frac{-x_1^2+5x_1-2x_1x_2+6x_2-4x_1x_3+7x_3-x_2^2-4x_2x_3-x_3^2-6}{6x_1^2+7x_2^2+1+8x_3^2} \rightarrow \min \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \leq 15 \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \geq 1 \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \\ 0 \leq x_3 \leq 2 \\ (x_1, x_2, x_3)' \in \mathbb{R}_+^3. \end{array} \right. \quad (2.1)$$

Pour cet exemple :

- Le problème est non convexe, car au moins une fonction composant ce problème est non convexe, de ce fait ce problème fait partie de l'optimisation non convexe.
- Difficile à résoudre car problème difficile.
- Peut être pris en charge par l'optimisation globale pour résolution.

Dans ce chapitre, nous évoquerons les éléments mathématiques nécessaires afin de développer succinctement notre travail. En rappel, nous présentons une classification des programmes (problèmes d'optimisation) passant des programmes linéaires jusqu'aux programmes non linéaires sans pour autant négliger les programmes quadratiques (les formes quadratiques et en particulier les formes quadratiques définies et semi-définies positives) qui ont un rôle crucial au niveau de l'optimisation non linéaire. On exposera en outre entre autres.

- Les principaux résultats de la programmation non linéaire.
- Les principaux résultats concernant les conditions d'optimalité pour les problèmes de programmation non linéaire avec contraintes.
- Les éléments de la programmation convexe et de la dualité lagrangienne.

Nous pouvons affirmer l'existence entre autres des classifications suivantes pour les problèmes

d'optimisation et puis nous développerons une certaine classe.

1. Classification par type de décision d'optimalité.
 - (a) Optimisation continue.
 - (b) Optimisation discrète.
 - (c) Optimisation mixte.
2. Classification respectant le type de la fonction objectif et contraintes.
 - (a) Optimisation sans contraintes.
 - (b) Optimisation avec contraintes.
 - (c) Optimisation linéaire.
 - (d) Optimisation non linéaire.
 - (e) Optimisation convexe.
 - (f) Optimisation concave.
 - (g) Optimisation D.C (Difference convex).
 - (h) Optimisation non convexe.
3. Classification qui respecte le type de solution à obtenir.
 - (a) Optimisation locale.
 - (b) Optimisation globale.

2.1 Classification des programmes mathématiques

La classification d'un problème mathématique (d'optimisation) (PM) se fait en général à travers les propriétés qui sont : la convexité, la continuité, le lipschitzien (Holderien), la différentiabilité des fonctions du problème, (...etc). Dans ce sens, un (PM) est convexe si toutes les fonctions et domaines sont convexes. Si toutes les fonctions induites différentiables, on dit alors que le (PM) est différentiable. La classe de programmes mathématiques convexes différentiables est le modèle le mieux riche en théories et algorithmes, les programmes non convexes ou non différentiables sont difficiles mais ont fait l'objet de grandes attentions auprès des chercheurs en recherche opérationnelle, si bien que de nouveaux modèles et schémas de résolutions ont vu le jour, essentiellement en optimisation globale [100, 17, 16]. La programmation linéaire où toutes les fonctions induites sont linéaires sont simples à appréhender. Afin de bien clarifier la vue, nous présentons quelques programmes mathématiques.

2.1.1 Les programmes linéaires (la programmation linéaire) [16, 17]

Considérons le problème linéaire suivant :

$$(P) \begin{cases} f(x) = z(x) = c'x \rightarrow \min \\ Ax = b \\ x \geq 0 \end{cases} \quad (2.2)$$

où A est une matrice réelle de type $(m \times n)$ supposée de plein rang ($\text{rang}(A) = m \leq n$), $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et $x \in \mathbb{R}^n$.

L'es solutions réalisables définissent un polyèdre convexe fermé.

Le dual de (P) est :

$$(D) \begin{cases} w(y) = w = b'y \rightarrow \max \\ A'y \leq c \\ y \in \mathbb{R}^m. \end{cases} \quad (2.3)$$

Remarque 2.1.1. ♣ En général, on appelle le problème (P) le primal et le problème (D) son dual et vice versa.

♣ Trois cas à signaler pour les problèmes (P) et (D) : soient (P) et (D) possédant chacun une solution optimale à valeurs égales ou (P) non borné et (D) non réalisable et vice versa ou (P) et (D) non réalisables.

Les principaux théorèmes de la programmation linéaire sont :

Théorème 2.1.1. Soient x et y des solutions réalisables des problèmes (P) et (D) respectivement alors

$$c'x \geq b'y.$$

Théorème 2.1.2. Soient x et y des solutions réalisables des problèmes (P) et (D) respectivement et si $c'x = b'y$ alors x et y sont les solutions optimales des problèmes (P) et (D) respectivement.

Théorème 2.1.3. Soient x et y des solutions réalisables des problèmes (P) et (D) respectivement, x et y sont optimales ssi x et y satisfont les conditions de complémentarité [100].

2.1.2 Méthodes de résolution en programmation linéaire [100, 81, 16, 34, 17]

Pour la résolution des problèmes de programmation linéaire (PL), on peut citer la méthode géométrique, méthode d'élimination de Fourier ou l'utilisation des conditions de KHUN TUCKER

ou encore les méthodes combinatoires [100, 16, 17].

A / Méthode du Simplexe [100]

Depuis le modèle russe de Kantorovitch en 1939, les choses ont ainsi évoluées, si bien que George Dantzig à partir de 1947 a inventé un algorithme exact dit " Simplex" ou Simplexe". De ce fait, il a beaucoup contribué au démarrage de l'optimisation numérique. L'algorithme du simplexe a longtemps été la méthode la plus utilisée pour résoudre les problèmes d'optimisation linéaire. Depuis les années 1984-90, il est concurrencé par les méthodes de points intérieurs, mais garde une place de choix dans certaines circonstances. Cet algorithme évolue sur la frontière du domaine admissible de sommet en sommet adjacent, optimisant ainsi la valeur de l'objectif selon un critère d'optimalité régit par un théorème bien établi. A partir du fait que le nombre de sommets est fini, l'optimum fini et la non dégénérescence, l'algorithme converge en un nombre d'itérations ne dépassant pas le nombre C_n^m itérations. L'algorithme risque de cycler, mais cela est bien pris en charge, il existe des méthodes adéquates pour endiguer cela par exemple les techniques de Bland 1974. Le Simplex supporte une très large Gamme de problèmes (linéaires, ou non linéaires), de plus c'est un algorithme exact mais dont il souffre plus au moins, c'est dans l'ordre de sa complexité qui peut être dans les pires des cas exponentiel. Le Simplexe possède une version révisée très intéressante.

B/ Méthodes de programmation linéaire par approche non linéaire [34]

* Méthodes projectives qui sont une variante de l'algorithme de Karmarkar. On les appelle ainsi car ils font intervenir des transformations projectives d'un Simplexe dans lui même. L'algorithme de Karmarkar 1984 dont l'approche non linéaire pour la programmation linéaire a été probablement le mieux vendu (3 millions de dollars en 1985), le plus célèbre, le moins général et celui qui demande le plus de préparation du problème de départ. Sa publication s'est suivie par un impact dans le domaine des méthodes des points intérieurs, des fonctions potentielles, des transformations projectives et des problèmes de la recherches opérationnelle en général. À souligner sa complexité de l'ordre de $O(n^{3.5}L)$, L : la longueur de codage des données.

* Méthodes de chemins (points) intérieurs (méthodes primales - duales) qui consiste à suivre de plus ou moins près un chemin paramétré par un nombre $\nu > 0$, situé strictement à l'intérieur du domaine admissible. Initialement, ces méthodes ont été développées pour résoudre les problèmes linéaires et non linéaires dans les années 60. Quant à leur utilisation pour la programmation linéaire, ces méthodes n'ont pas reçu l'engouement escomptée vis à vis de leurs performances du moment qu'en ce temps, il y avait la suprématie et l'attractivité envers la méthode du Simplex. La publication des résultats de l'algorithme de Karmarkar en 1984 a été le tournant favorable vers ces méthodes de points intérieurs. On peut citer des variantes pour ces méthodes :

- Méthodes de réduction de potentiel : à l'instar de la méthode, citons par exemple : méthode de Karmarkar.
- Méthodes de chemin central : une complexité polynomiale et une convergence superlinéaire, ont une expansion dans les années 90.

* Méthodes affines qui fonctionnent en changeant de métrique à chaque itération. Citons pour cela l'exemple de l'algorithme du Russe Dikin en 1967, qui est resté inconnu jusqu'à 1985 - 1986. D'après les chercheurs, c'est l'algorithme qui pourrait remplacer celui de Karmarkar. Chèrement vendu pour l'U.S Navy pour sa flotte et sa logistique. Cet algorithme n'a pas besoin de transformation projective, ni préparation au préalable et ni fonction potentiel. Il est de type théorique, c'est à dire qu'il n'existe pas de démonstration prouvant que sa complexité est polynomiale sauf pour un cas primal dual. Son principal atout, est qu'il est souvent plus rapide que toutes les autres versions.

C/ Méthodes de programmation linéaire dites Adaptées ou de Supports [8, 48, 87, 23]

En utilisant la métrique du Simplexe, R.Gabassov et F.M .Kirrilova ont inventé durant les années 80 la méthode adaptée (Adaptée) et la méthode du support (Support). [48]

L'avantage de celle-ci, est une méthode de points intérieurs, elle permet aussi l'obtention d'une solution approchée et résout des problèmes de contrôle optimal. Au début de son invention, elle a été appliquée à différents types de problèmes de programmation mathématiques [48, 87, 23, 8], par la suite à des problèmes de contrôle optimal.

Étude comparative entre la méthode Adaptée et le Simplexe

- 1/ Les deux méthodes possèdent des algorithmes de résolution finis.
- 2/ Au vue de la résolution d'un problème par la méthode du simplexe, on constate que cela occasionne l'ajout de variables supplémentaires et d'équations, ce qui n'est pas le cas quant à l'utilisation de la méthode adaptée.
- 3/ Sachant que les solutions réalisables de bases sont des sommets du polyèdre décrit par les contraintes de plus la recherche de la solution optimale dans le simplexe s'effectue par saut sur les sommets du polyèdre alors que la méthode adaptée prend les points à l'intérieur du polyèdre ce qui fait que cette méthode convergera vraisemblablement plus rapidement vers la solution optimale. Donc on peut s'attendre à une amélioration de la complexité.
- 4/ Avec la méthode adaptée on peut trouver une solution epsilon optimale.
- 5/ Avec la méthode adaptée on peut commencer l'algorithme avec une solution initiale non réalisable.

2.1.3 Exemple Illustratif de la programmation linéaire non différentiable

Soit le problème non différentiable suivant pris en charge par la méthode adaptée sans reformulation :

$$f(x) = \min_k (c'_k x + \alpha_k) \rightarrow \max_x. k \in K.$$

$$Ax = b, d_1 \leq x \leq d_2.$$

Voici les définitions des paramètres du problème [8, 87, 23] :

$A = A[I, J]$: matrice (m, n) , $\text{Rang}(A) = m \leq n$, b est un m - vecteur. x, d_1, d_2 des n - vecteurs, α_k des scalaires. c'_k des n -vecteurs.

$C[K, J]$: matrice (p, n) formée des lignes des vecteurs $c'_k, \forall k \in K$.

$K = 1 \dots p$: l'ensemble des indices des composantes de la fonctionnelle.

$I = 1 \dots m$: l'ensemble des indices de lignes de A .

$J = 1 \dots n$: l'ensemble des indices de colonnes de A .

$$\omega_k = (c'_k x + \alpha_k) - f(x), \forall k \in K.$$

$$\omega_k(x) \geq 0, \forall k \in K.$$

$$\min_{k \in K} \omega_k(x) = 0, \forall k \in K.$$

$$\omega_k(\bar{x}) - \omega_k(x) = \Delta \omega_k(x) \geq -\omega_k(x), \forall k \in K.$$

Pour la résolution de ce problème, cela est établi et compris dans une de nos contributions pour notre travail et cela au chapitre 4 suivant où il est question d'un déploiement de logiciel [23]. Ce dernier peut prendre en charge la résolution des problèmes linéaires (Simplex / Adaptée / Support/méthode primale Adaptée/méthode duale Adaptée) [23] et des extensions vers les problèmes non linéaires, sans pour autant oublier la résolution des problèmes en contrôle optimal.

2.1.4 Exemple Illustratif (Implémentation) de la programmation linéaire

Avec la méthode Karmarkar [34, 62, 84, 67, 66, 64]

L'algorithme de Karmarkar permet de résoudre le problème de Type :

$$(PLK) \begin{cases} f(x) = z(x) = c'x \rightarrow \min = z^* \dots(1) \\ Ax = 0 \\ x \in S_n. \end{cases} \quad (2.4)$$

On connaît la valeur optimale $z^* = 0$ et une solution réalisable par exemple le point $a = \frac{1}{n}e_n$ (centre du Simplexe S_n), e_n désigne le vecteur dont toutes les composantes sont égales à 1.

A est une matrice de dimension $(m \times n)$ et de plein rang ($\text{rang}(A) = m \leq n$).

$$S_n = \{x \in \mathbb{R}, x \geq 0, e'x = 1\},$$

est le simplexe de dimension $(n - 1)$.

Description des Itérations de l'algorithme [34, 62, 63]

À partir d'une solution admissible $x^0 = a$ l'algorithme construit une suite de points intérieurs qui converge vers une solution optimale avec un temps polynomial. Afin de ramener l'objectif à zéro, on le minimise localement autour d'une sphère inscrite dans la région réalisable.

À itération k , l'itéré $x^k > 0$ est ramené au centre de S_n avec la transformation projective T_k donnée par :

$$T_k : x \in S_n \rightarrow T_k(x) = y \in S_n.$$

$$T_k(x) = \frac{D_k^{-1}x}{e'_n D_k^{-1}x} = y, T_k^{-1}(y) = \frac{D_k y}{e'_n D_k y}.$$

Avec

$$D_k = \text{diag}\{x^k\}.$$

La transformation T_k appliquée avec l'itéré $x^k > 0$ projeté au centre de S_n , le transformé du programme linéaire (PLK) est le suivant :

$$\left\{ \begin{array}{l} c' \frac{D_k y}{e'_n D_k y} \rightarrow \min \\ A \frac{D_k y}{e'_n D_k y} = 0 \quad \dots(2) \\ e'_n y = 1 \\ y \geq 0. \end{array} \right. \quad (2.5)$$

Par la suite, on trouve le problème équivalent suivant :

$$\left\{ \begin{array}{l} c' D_k y \rightarrow \min \\ A D_k y = 0 \quad \dots(3) \\ e'_n y = 1 \\ y \geq 0. \end{array} \right. \quad (2.6)$$

On utilise le lemme suivant pour faciliter la résolution du problème ci-dessus :

Lemme 2.1.1. [34, 62, 63] Si pour un programme linéaire donné on connaît une solution réalisable y^0 tel que $(y_i^0 > 0, i = 1 : n + 1)$, alors l'ellipsoïde :

$$E = \{y \in \mathbb{R}^{n+1} : \sum_{i=1}^{n+1} \frac{(y_i - y_i^0)^2}{(y_i^0)^2} < \beta^2, 0 < \beta < 1\},$$

est dans l'intérieur de l'orthant positif de \mathbb{R}^{n+1} .

Démonstration. Supposons qu'il existe $j \in \{1, 2, \dots, n + 1\}$ tel que $y_j \leq 0$ et donc :

$$\sum_{i=1}^{n+1} \frac{(y_i - y_i^0)^2}{(y_i^0)^2} \geq \frac{(y_j - y_j^0)^2}{(y_j^0)^2} \geq 1 > \beta^2.$$

D'après le lemme ci-dessus, si on ajoute au dernier problème la contrainte :

$$y \in \mathbb{R}^n / \|y - a\| \leq \alpha r, 0 < \alpha < 1, r = \frac{1}{\sqrt{n(n-1)}}.$$

Alors la contrainte de positivité $y \geq 0$ devient redondante. On aura le problème :

$$\begin{cases} c' D_k y \rightarrow \min \\ A D_k y = 0 & \dots(4) \\ e'_n y = 1 \\ (\|y - a\|)^2 \leq (\alpha r)^2. \end{cases} \quad (2.7)$$

Minimisation d'une fonction linéaire sur une sphère, la solution optimale est établie à travers le théorème suivant :

□

Théorème 2.1.4. La solution optimale du problème (2.6) est donnée par : $y^k = a - \alpha r d^k$, avec $d^k = \frac{p^k}{\|p^k\|}$ et $p^k = p_{B_k}(D_k c)$, $B_k = \begin{pmatrix} A_k \\ e'_n \end{pmatrix}$.

Démonstration. [34, 62, 63]

Indications

- * Poser les conditions nécessaires et suffisantes d'optimalité de KKT.
- * Utilisation de la transformée projective.

□

Algorithme de résolution

- * $\epsilon > 0$ une précision donnée, $x^0 = a = \frac{1}{n} e_n, k = 0$.

Tant que $c'x^k > \epsilon$ do

$$D_k = \text{diag}\{x^k\}, A_k = AD_k, B_k = \begin{pmatrix} A_k \\ e'_n \end{pmatrix}$$

$$p^k = (I - B'_k(B_k B'_k)^{-1} B_k) D_k c, d^k = \frac{p^k}{\|p^k\|}$$

$$* y^k = a - \alpha r d^k, r = \frac{1}{\sqrt{n(n-1)}}, 0 < \alpha < 1.$$

$$* x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k}{e'_n D_k y^k}, k = k + 1.$$

Fin tant que.

Théorème 2.1.5. Pour $0 < \alpha < \frac{1}{4}$ et en partant de $x^0 = \frac{1}{n}e_n$, l'algorithme atteint un point x après $0(nq + n \log(n))$ itérations, avec

* $c'x = 0$ ou

* $\frac{c'x}{c'x^0} \leq \epsilon = 2^{-q}$, q est une précision fixée.

Généralisation de la méthode (cas linéaire) [34, 62, 63]

Considérons le programme linéaire suivant :

$$(P) \begin{cases} z(x) = c'x \rightarrow \min = z^* \dots (1) \\ Ax = b \\ x \geq 0. \end{cases} \quad (2.8)$$

A est une matrice $m \times n$, $c \in \mathbb{R}^n, b \in \mathbb{R}^m$.

On définit le simplexe S_{n+1} de dimension n contenu dans \mathbb{R}^{n+1} par :

$$S_{n+1} = \{x \in \mathbb{R}^{n+1} : x \geq 0, e'_{n+1}x = 1\}.$$

Avec les hypothèses suivantes :

- ♣ La matrice A type $(m, n) : \text{Rang}(A) = m \leq n$.
- ♣ On dispose d'un (point initial) x^0 strictement réalisable ($Ax^0 = b; x^0 > 0$).
- ♣ La valeur optimale z^* de l'objectif est connue au départ.

À partir de la transformée projective

$T_k : \mathbb{R}_+^n \rightarrow S_{n+1}$ définie par : $T_k(x) = y$

comme suit :

$$\begin{cases} y_i = \frac{x_i/x_i^k}{1 + \sum_{i=1}^n x_i/x_i^k}, i = 1 : n \\ y_{n+1} = 1 - \sum_{i=1}^n y_i \end{cases} \quad (2.9)$$

avec

$$y_i = \frac{x_i}{x_i^k} y_{n+1}, i = 1, \dots, n$$

$y[n] = (D_k^{-1}x)y_{n+1}$ avec $y[n]$ désignant les n premières composantes de y .

$$x = T_k^{-1}(y) = \frac{D_k y[n]}{y_{n+1}}, D_k = \text{diag}(x^k).$$

Et donc par la transformée T_k , on aura :

$$\begin{cases} c' \frac{D_k y[n]}{y_{n+1}} \rightarrow \min \\ A \frac{D_k y[n]}{y_{n+1}} = b \\ e'_{n+1} y = 1 \\ y[n] \geq 0, y_{n+1} > 0 \end{cases} \quad (2.10)$$

ou

$$\begin{cases} \bar{c}' y \rightarrow \min \\ \bar{A} y = 0 \\ e'_{n+1} y = 1 \\ y \geq 0 \end{cases} \quad (2.11)$$

avec

$$\begin{cases} \bar{c}_i = c_i x_i^k, i = 1, \dots, n \\ \bar{c}_{n+1} = -z^* \end{cases} \quad (2.12)$$

$$y = \begin{pmatrix} y[n] \\ y_{n+1} \end{pmatrix} \text{ et } \bar{A} = [AD_k \ -b].$$

Remarque 2.1.2. [34, 62, 63] Une extension de la méthode de Karmarkar est effectuée et assurée pour le problème non linéaire suivant :

$$\begin{cases} f(x) \rightarrow \min = z^* \\ Ax = b \\ x \geq 0 \end{cases} \quad (2.13)$$

f une fonction non linéaire convexe et différentiable

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}.$$

- ♣ La matrice A type $(m, n) : \text{Rang}(A) = m \leq n$.
- ♣ On dispose d'un (point initial) x^0 strictement réalisable ($Ax^0 = b; x^0 > 0$).
- ♣ La valeur optimale z^* de l'objectif est connue au départ.

Voici l'algorithme correspondant :

Algorithme de résolution

* $\epsilon > 0$ une précision donnée, x^0 strictement réalisable.

Tant que $f(x^k) - z^* > \epsilon$ do

$$D_k = \text{diag}\{x^k\}, A_k = [AD_k \quad -b], B_k = \begin{pmatrix} A_k \\ e'_{n+1} \end{pmatrix}$$

$$p^k = (I - B'_k(B_k B'_k)^{-1} B_k) \nabla g(a), d^k = \frac{p^k}{\|p^k\|}$$

$$* y^k = a - \alpha d^k, 0 < \alpha < 1.$$

$$* x^{k+1} = T_k^{-1}(y^k) = \frac{D_k y^k [n]}{y_{n+1}^k}, k = k + 1.$$

Fin tant que.

2.1.5 La programmation quadratique

Consulter [17, 34, 81]

On peut considérer la programmation quadratique comme étant une discipline à part. Par ses théories et ses applications, le programme quadratique apparait souvent comme sous problème dans l'optimisation SQP (sequentiel quadratics problems). Ces théories aussi de type complémentarité, entre programme quadratique et programme linéaire et les formes quadratiques qui ont un grand apport dans le cadre de l'optimisation,...etc.

Les cas pratiques en programmation quadratique sont par exemples :

- ♣ Optimisation de portefeuilles en finance.
- ♣ Optimisation en contrôle optimal.
- ♣ Optimisation en économie, le problème de transport.
- ♣ La régression et optimisation.

La programmation quadratique, branche de l'optimisation non linéaire où la fonction objectif à minimiser (à optimiser) est une fonction quadratique et les contraintes sont linéaires et/ou

quadratiques.

Beaucoup de chercheurs se sont reliés dans le cadre de la programmation quadratique, on peut citer : Franck Wolf, Billionnet, Pardalos, Elloumi, Adams-Sherali, Le Thi-Pham Dinh et la liste reste très longue.

Exemple 2.1.1. Soit le problème en programmation quadratique :

$$\left\{ \begin{array}{l} -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 - 6 \rightarrow \min \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \leq 5 \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \geq 2 \\ 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \\ 0 \leq x_3 \leq 1 \\ (x_1, x_2, x_3)' \in \mathbb{R}_+^3. \end{array} \right. \quad (2.14)$$

Les formes quadratiques et propriétés

Définition 2.1.1. Une fonction f , tel que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, définie par :

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}x_i x_j + \sum_{j=1}^n c_j x_j = x'Ax + c'x.$$

Et dont les paramètres sont :

$x' = (x_1, x_2, \dots, x_n)$, c_j des n-vecteurs de \mathbb{R}^n . $A = (a_{ij}, 1 \leq i \leq n, 1 \leq j \leq n)$ matrice carrée d'ordre n . ($'$) désignant l'opérateur de transposition matriciel (vectoriel).

La redéfinition de la forme quadratique peut s'écrire comme suit, sans perte de généralité :

$$f(x) = x'Ax + c'x = x'Dx + c'x, \forall x \in \mathbb{R}^n.$$

Avec D matrice symétrique d'ordre n .

Et donc, on peut considérer que la matrice d'une forme quadratique est toujours symétrique.

Gradient d'une forme quadratique

Définition 2.1.2. Une fonction f , tel que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, f étant forme quadratique, f continûment différentiable et son gradient au point x est :

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix} = 2Dx + c.$$

Avec $\frac{\partial f(x)}{\partial x_i}$ est la dérivée partielle de f par rapport à x_i .

Définition 2.1.3. Une fonction f , tel que $f : \mathbb{R}^n \rightarrow \mathbb{R}$, f étant forme quadratique, f 2 fois continûment différentiable et son Hessien au point x est :

$$\begin{aligned} \nabla^2 f(x) &= \left(\nabla \frac{\partial f(x)}{\partial x_1}, \nabla \frac{\partial f(x)}{\partial x_2}, \dots, \nabla \frac{\partial f(x)}{\partial x_n} \right) = \\ \mathbb{H}(x) = \nabla^2 f(x) &= \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{pmatrix} = \left(\left(\frac{\partial^2 f}{\partial x_j \partial x_i}(x) \right) \right)_{1 \leq i, j \leq n} \\ &= 2D. \end{aligned}$$

A / La programmation quadratique sans contraintes [17, 5, 34, 81, 12, 31]

Conditions d'optimalité pour les problèmes quadratiques

Théorème 2.1.6. *Considérons le problème suivant :*

$$\min_{x \in \mathbb{R}^n} = \frac{1}{2} x' Q x + g' x + c.$$

Où Q est une matrice symétrique de type (n, n) , $g \in \mathbb{R}^n$ et $c \in \mathbb{R}$.

1/ Si Q n'est pas semi définie positive, alors le problème ne possède pas de solution, c'est à dire qu'il n'existe aucun $x \in \mathbb{R}^n$ qui soit un minimum local.

2/ Si Q est définie positive, alors le problème possède une unique solution

$$x^* = -Q^{-1}g.$$

Qui est le minimum global.

Démonstration. 1 / Nous avons $\nabla f(x) = Qx + g$ et $\nabla^2 f(x) = Q$.

Raisonnons par l'absurde, on suppose qu'on a un minimum x^* , ce qui implique que $\nabla^2 f(x) = Q$ est semi définie positive, ce qui est en soit une contradiction.

2/ Comme Q est définie positive, le point $x^* = -Q^{-1}g$ est bien défini, de plus

$$\nabla f(x^*) = -QQ^{-1}g + g = 0.$$

f strictement convexe, x^* est l'unique minimum global.

□

Les méthodes utilisées dans le cadre de la programmation quadratique sans contraintes [17, 34, 81]

- ♣ Méthode des directions conjuguées.
- ♣ Méthode des gradients conjugués.
- ♣ Méthode de Newton locale.
- ♣ Méthode de descente avec recherche linéaire (Wolf/Armijo).
- ♣ Méthode de région de confiance.
- ♣ Méthode de la quasi -Newton.
- ♣ Méthode de Nelder Mead.

Remarque 2.1.3. S'agissant des conditions d'optimalité ou autres, cela sera pris en charge dans la partie d'étude de la programmation non linéaire (cas général).

B / La programmation quadratique avec contraintes [17, 34, 81, 12, 31]

La programmation quadratique avec contraintes a connu et connaît un vaste champs d'applications dans plusieurs domaines. On l'utilise souvent comme procédures intervenant (procédures intermédiaires) pour des programmes non linéaires par exemples.

Un cas de programme quadratique sous contraintes peut se présenter de la forme :

Considérons le problème quadratique suivant :

$$(QP) \begin{cases} f(x) = \frac{1}{2}x'Qx + c'x \rightarrow \min \\ Ax = b \\ x \geq 0. \end{cases} \quad (2.15)$$

Q matrice symétrique d'ordre n , où A est une matrice réelle de type $(m \times n)$ supposée de plein rang ($\text{rang}(A) = m \leq n$), $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et $x \in \mathbb{R}^n$.

L'ensemble des solutions réalisables définit un polyèdre convexe fermé.

Remarquons de suite que le problème (QP) est différentiable et la fonction objectif indéfiniment différentiable. La convexité du problème (QP) dépend seulement de la matrice Q .

Remarque 2.1.4. ♣ Le problème (QP) , est convexe si et seulement si Q semi-définie positive.

♣ Le problème (QP) devient **très intéressant** à résoudre dès que la matrice Q non semi-définie positive, dans ce cas le problème (QP) sera classé comme étant un problème de **l'optimisation globale**, beaucoup de méthodes pouvant prendre en charge ce problème, objet des chapitres suivants.

Les méthodes utilisées dans le cadre de la programmation quadratique convexe avec contraintes [17, 34, 81]

- ♣ Méthode de Wolf.

- ♣ Méthode adaptée (support) avec ses 02 versions.
- ♣ Méthode duale adaptée (support) avec ses 02 versions.
- ♣ Méthode des points intérieurs.
- ♣ Méthode de Karmarkar.

La liste est très longue pour toute la citer (d'autres méthodes seront citées plus tard).

Quelques propriétés importantes pour les problèmes quadratiques convexes :

Parmi lesquelles, on peut citer les propriétés de la complémentarité et celle de la dualité.

- ♣ La dualité en programmation convexe.

Pour le programme convexe (QP) , on lui associe son programme dual de la forme suivante :

$$(DQP) \begin{cases} L(z, y, v) = \frac{1}{2}z'Qz + c'z + y'(Az - b) - v'z \rightarrow \max \\ \frac{\partial L}{\partial z}(z, y, v) = Qz + c + A'y - v = 0 \\ z \in \mathbb{R}^n, y \in \mathbb{R}^m, v \in \mathbb{R}^n, v \geq 0. \end{cases} \quad (2.16)$$

Le problème dual (DQP) deviendra le suivant par un simple artifice de calcul.

$$(DQP) \begin{cases} L(z, y, v) = -\frac{1}{2}z'Qz - y'b \rightarrow \max \\ \frac{\partial L}{\partial z}(z, y, v) = Qz + c + A'y - v = 0 \\ z \in \mathbb{R}^n, y \in \mathbb{R}^m, v \in \mathbb{R}^n, v \geq 0. \end{cases}$$

Une autre version du problème (QP) est :

$$(QP) \begin{cases} f(x) = \frac{1}{2}x'Qx + c'x \rightarrow \min \\ Ax = b \\ d^- \leq x \leq d^+. \end{cases} \quad (2.17)$$

Pour le programme convexe (QP) , on lui associe son programme dual de la forme suivante :

$$(DQP) \begin{cases} L(z, y, v, w) = \frac{1}{2}z'Qz + c'z + y'(Az - b) + v'(-z + d^-) + w'(z - d^+) \rightarrow \max \\ \frac{\partial L}{\partial z}(z, y, v, w) = Qz + c + A'y - v + w = 0 \\ z \in \mathbb{R}^n, y \in \mathbb{R}^m, v \in \mathbb{R}^n, v \geq 0, w \in \mathbb{R}^n, w \geq 0. \end{cases} \quad (2.18)$$

Le problème dual (DQP) deviendra le suivant par un simple artifice de calcul.

$$(DQP) \begin{cases} L(z, y, v, w) = -\frac{1}{2}z'Qz - y'b + v'd^- - w'd^+ \rightarrow \max \\ \frac{\partial L}{\partial z}(z, y, v, w) = Qz + c + A'y - v + w = 0 \\ z \in \mathbb{R}^n, y \in \mathbb{R}^m, v \in \mathbb{R}^n, v \geq 0, w \in \mathbb{R}^n, w \geq 0. \end{cases}$$

♣ La complémentarité en programmation convexe [66, 65, 2, 93, 63, 62].

La complémentarité en programmation convexe pour le cas de notre problème a suscité l'attention de plusieurs chercheurs, cette notion se retrouve dans plusieurs domaines : programmation linéaire, optimisation quadratique, optimisation en général, mécanique,...etc.

Le programme complémentaire a la forme suivante :

$$(PCL) \left\{ \begin{array}{l} x, y \in \mathbb{R}^n, \text{ avec} \\ y = Mx + q \\ x'y = 0 \\ (x, y) \geq 0. \end{array} \right.$$

M : matrice carré d'ordre n .

Transformons le programme quadratique suivant en un programme complémentaire :

$$(QPC) \left\{ \begin{array}{l} f(x) = \frac{1}{2}x'Qx + c'x \rightarrow \min \\ Ax \leq b \\ x \geq 0. \end{array} \right. \quad (2.19)$$

Q matrice symétrique d'ordre n semi-définie positive, où A est une matrice réelle de type $(m \times n)$ supposée de plein rang ($\text{rang}(A) = m \leq n$), $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ et $x \in \mathbb{R}^n$.

Par simple application des conditions K.K.T (Karush KHUN TUCKER), cela donne, en tenant compte que :

$$\exists y \in \mathbb{R}_+^m, \lambda \in \mathbb{R}_+^n$$

$$\left\{ \begin{array}{l} c + Qx + A'y - \lambda = 0 \\ y'(b - Ax) = 0 \\ -\lambda'x = 0 \\ y \geq 0, \lambda \geq 0 \end{array} \right.$$

$$\left\{ \begin{array}{l} \lambda = c + Qx + A'y \\ v = (b - Ax) \\ \lambda'x = 0, y'v = 0 \\ x \geq 0, y \geq 0, \lambda \geq 0, v \geq 0 \end{array} \right.$$

$$\left\{ \begin{array}{l} \begin{pmatrix} \lambda \\ v \end{pmatrix} = \begin{pmatrix} Q & A' \\ -A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c \\ b \end{pmatrix} \\ \langle \begin{pmatrix} \lambda \\ v \end{pmatrix}; \begin{pmatrix} x \\ y \end{pmatrix} \rangle = 0 \\ \begin{pmatrix} \lambda \\ v \end{pmatrix} \geq 0; \begin{pmatrix} x \\ y \end{pmatrix} \geq 0. \end{array} \right.$$

En posant les paramètres comme suit :

$$w = \begin{pmatrix} \lambda \\ v \end{pmatrix}; z = \begin{pmatrix} x \\ y \end{pmatrix}; q = \begin{pmatrix} c \\ b \end{pmatrix}; M = \begin{pmatrix} Q & A' \\ -A & 0 \end{pmatrix}.$$

Le problème complémentaire se déduit :

$$(PCL) \left\{ \begin{array}{l} z \in \mathbb{R}^{m+n}, \text{ avec} \\ w = Mz + q \geq 0, z \geq 0 \\ w'z = 0. \end{array} \right.$$

Remarque 2.1.5. Concernant la notion de complémentarité :

♣ soit le programme linéaire (P)

$$(P) \left\{ \begin{array}{l} c'x \rightarrow \min \\ Ax \leq b \\ x \geq 0. \end{array} \right.$$

Ce qui implique son programme linéaire complémentaire (PCL)

$$(PCL) \begin{cases} z \in \mathbb{R}^{m+n}, \text{ avec} \\ w = Mz + q \geq 0, z \geq 0 \\ w'z = 0. \end{cases}$$

Avec les paramètres comme suit :

$$w = \begin{pmatrix} \lambda \\ v \end{pmatrix}; z = \begin{pmatrix} x \\ y \end{pmatrix}; q = \begin{pmatrix} c \\ b \end{pmatrix}; M = \begin{pmatrix} 0 & A' \\ -A & 0 \end{pmatrix}.$$

La transformation inverse d'un programme linéaire complémentaire vers un programme quadratique convexe (QP) ne peut se faire à moins que la matrice M soit semi-définie positive (programme linéaire complémentaire monotone) [34, 62, 63].



A partir d'un (PCL) monotone, on peut construire un programme quadratique convexe et puis moyennant les notions de fonctions barrières, on résout le programme quadratique avec par exemple la méthode des points intérieurs (sous multiples variantes) ce qui permet ensuite de résoudre des problèmes du type suivant par la méthode des points intérieurs :

$$\begin{cases} f(x) \rightarrow \min \\ Ax = b \\ x \geq 0. \end{cases} \quad (2.20)$$

f une fonction non linéaire convexe et différentiable.

$$f : \mathbb{R}^n \longrightarrow \mathbb{R},$$

la matrice A type $(m, n) : \text{Rang}(A) = m \leq n$.



Une variante assez générale de programme quadratique (programme quadratique mixte) :

$$(QPC) \begin{cases} f(x) = \frac{1}{2}x'Qx + c'x \rightarrow \min \\ Ax \leq b \\ x'L_jx + c'_jx \leq \alpha_j : j = 1, \dots, p \\ x \in \mathbb{R}^n. \end{cases} \quad (2.21)$$

* Q et $L_j : j = 1, \dots, p$ matrices symétriques semi-définies positives entre autres.

* Résolutions : méthode de points intérieurs - variante de Newton - SDP (méthode semi-définie positive). [9]

* Q et/ou $L_j : j = 1, \dots, p$ matrices symétriques. Résolution par l'optimisation globale.

2.1.6 Les programmes non linéaires (la programmation non linéaire) [11]

Dans le domaine de la programmation non linéaire (la généralisation), on va traiter des programmes sans contraintes et des programmes avec contraintes. Cela englobera l'étude pour la programmation quadratique.

A / La programmation non linéaire sans contraintes [17, 5, 34, 81, 12, 31]

Nous rappelons les principaux résultats vu qu'on se ramène en général sous certaines conditions de la programmation non linéaire avec contraintes vers la programmation non linéaire sans contraintes.

Exemple 2.1.2. Soit le problème en guise d'illustration :

$$\begin{cases} 7x_1^2 - 7x_1x_2 - 2x_2^2 - x_1^4 - x_2^4 - x_1^6 - x_2^6 - \exp(x_1 + x_2) \rightarrow \max \\ (x_1, x_2)' \in \mathbb{R}^2. \end{cases} \quad (2.22)$$

Exemple 2.1.3. Soit le problème :

$$\begin{cases} 9x_1^{-0.9}x_2^{-1.6}x_3^{-4} + 6x_4^{-0.3}x_5^{-3.6} + 3x_6^{-1.9}x_7^{-0.6}x_8^{-1} + x_1 \log(x_1) + x_2 \log(x_2) \rightarrow \min \\ (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)' \in \mathbb{R}_+^{*8}. \end{cases} \quad (2.23)$$

Définition 2.1.4. f étant une fonction réelle définie, sur un ouvert $D \subset \mathbb{R}^n$. f admet un minimum local en $x^* \in D$, si $\exists B(x^*, \epsilon) = \{x / \|x - x^*\| < \epsilon\} \subset D$ et telle que :

$$f(x) \geq f(x^*), \forall x \in B(x^*, \epsilon).$$

Définition 2.1.5. f étant une fonction réelle définie, sur un ouvert $D \subset \mathbb{R}^n$. f admet un minimum local en $x^* \in D$ au sens strict, si $\exists B(x^*, \epsilon) = \{x / \|x - x^*\| < \epsilon\} \subset D$. et telle que :

$$f(x) > f(x^*), \forall x \in B(x^*, \epsilon), x \neq x^*.$$

Définition 2.1.6. f Étant une fonction réelle définie, sur $D \subset \mathbb{R}^n$. f admet un minimum global en $x^* \in D$, si :

$$f(x) \geq f(x^*), \forall x \in D.$$

Définition 2.1.7. Le vecteur $d \in \mathbb{R}^n$, $d \neq 0$, est direction admissible au point $x \in D$, s'il existe un réel $\alpha > 0$, tel que $x + \theta d \in D$, $\forall \theta \in [0, \alpha]$.

Si x est un point intérieur, alors toutes les directions sont admissibles.

Définition 2.1.8. f étant une fonction réelle définie, sur $D \subset \mathbb{R}^n$ de classe C^1 . La direction d admissible en $x \in D \subset \mathbb{R}^n$ est dite de descente ssi

$$d' \nabla f(x) < 0.$$

Lemme 2.1.2. f étant une fonction réelle définie, sur $D \subset \mathbb{R}^n$ de classe C^1 . Si f admet un minimum local en $x^* \in D$, alors pour toute direction d admissible en x^* :

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

De plus, en particulier si $x^* \in D$ est un point intérieur alors $\nabla f(x^*) = 0$.

Démonstration. Soit d

une direction admissible en $x^* \in D$, avec $\alpha > 0$ et θ très petit tel que :

$$\begin{aligned} \forall \theta \in [0, \alpha], x = x^* + \theta d \in D \\ \implies \forall \theta \in [0, \alpha], f(x) = f(x^* + \theta d) \geq f(x^*) \end{aligned}$$

du fait que x^* soit minimum local et compte tenu que f soit de classe C^1

$$\begin{aligned} f(x^* + \theta d) - f(x^*) &= \theta d' \nabla f(x^*) + o(\theta) \\ \theta &\rightarrow 0 (o(\theta) \rightarrow 0) \\ \implies 0 &\leq f(x^* + \theta d) - f(x^*) = \theta d' \nabla f(x^*) \\ \implies d' \nabla f(x^*) &\geq 0. \end{aligned}$$

Cas particulier : Si x^* est intérieur de D : toutes les directions d sont admissibles de plus $d' \nabla f(x^*) \geq 0, \forall d$

$$\implies d' \nabla f(x^*) = 0, \forall d \implies \nabla f(x^*) = 0.$$

□

Théorème 2.1.7. Si x^* est minimum local de f dans \mathbb{R}^n (ou $x^* \in D$) et si de plus f différentiable en x^* alors :

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

(x^* est dit point stationnaire).

Démonstration. x^* minimum local de f , considérons $d \in \mathbb{R}^n$ et soit une fonction réelle ψ définie comme suit :

$\psi(\theta) = f(x^* + \theta d)$, la fonction ψ étant sur un intervalle ouvert de \mathbb{R} contenant l'origine, puisque

x^* minimum local de f alors il existe $\beta > 0$, tel que :

$$\psi(0) \leq \psi(\theta), \forall \theta \in]-\beta, \beta[.$$

0 est minimum local de ψ .

f est différentiable en x^* , donc ψ est dérivable en 0 et $\psi'(0) = \langle \nabla f(x^*), d \rangle$.

$\psi'(0) = 0$ et donc $\langle \nabla f(x^*), d \rangle = 0$ pour tout $d \in \mathbb{R}^n$ et finalement $\nabla f(x^*) = 0$.

□

Théorème 2.1.8. Si x^* est minimum local (global) de f dans \mathbb{R}^n (ou $x^* \in D$) et si de plus f deux fois différentiable en x^* alors :

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

♣ $\nabla^2 f(x^*) = H(x^*)$ (Hessienne) est semi-définie positive.

Démonstration. x^* minimum local de f , considérons le développement de Taylor de la fonction f au voisinage de x^* comme suit :

$$f(x) - f(x^*) = \nabla f(x^*)(x - x^*) + \frac{1}{2}(x - x^*)'H(x^*)(x - x^*) + \|(x - x^*)\|^2 \epsilon(x - x^*).$$

$\epsilon(x - x^*) \rightarrow 0$ avec $x \rightarrow x^*$.

Procédant par l'absurde ($\nabla f(x^*) \neq 0$), et posant $x = x^* - \theta \nabla f(x^*)$, cela aboutira à ce que $f(x) < f(x^*)$ pour des θ suffisamment petits strictement positifs, ce qui contredit que : x^* minimum local de f , donc la première condition est nécessaire.

De façon identique pour démontrer la première condition nécessaire.

Supposant la matrice $H(x^*)$ n'est pas semi-définie positive pour ($d \in \mathbb{R}^n$), de plus posant $x = x^* + \theta d$, ($\theta > 0$), ($d \in \mathbb{R}^n, d \neq 0$) cela aboutira à ce que $f(x) < f(x^*)$ pour des θ suffisamment petits strictement positifs, ce qui contredit que : x^* minimum local de f , donc la deuxième condition est aussi nécessaire.

□

Théorème 2.1.9. Si $x^* \in \mathbb{R}^n$ (ou $x^* \in D$) et f est deux fois différentiable en x^* de plus si

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

♣ $\nabla^2 f(x^*) = H(x^*)$ (Hessienne) est définie positive.

Alors x^* est un minimum local strict de f sur \mathbb{R}^n (sur D).

Démonstration. Soit $x^* \in \mathbb{R}^n$, qui satisfait les deux conditions, considérons le développement de Taylor de la fonction f au voisinage de x^* comme suit :

$$f(x) - f(x^*) = \nabla f(x^*)(x - x^*) + \frac{1}{2}(x - x^*)'H(x^*)(x - x^*) + \|(x - x^*)\|^2 \epsilon(x - x^*).$$

$\epsilon(x - x^*) \rightarrow 0$ avec $x \rightarrow x^*$.

de plus posant $x = x^* + \theta d$, ($\theta > 0$), ($d \in \mathbb{R}^n, d \neq 0$), d direction admissible.

$$f(x) = f(x^* + \theta d) = f(x^*) + \frac{\theta^2}{2} d' \nabla^2 f(x^*) d + \theta^2 \epsilon(\theta).$$

$\epsilon(\theta) \rightarrow 0$ avec $\theta \rightarrow 0$. Du fait que $d' \nabla^2 f(x^*) d > 0$ pour des θ suffisamment petits strictement positifs, ce qui aboutit au résultat escompté.

□

Les méthodes utilisées dans le cadre de la programmation non linéaire sans contraintes [17, 34, 81, 12, 31, 100, 9]

- ♣ Méthode des directions conjuguées.
- ♣ Méthode des gradients conjugués.
- ♣ Méthode de Newton locale.
- ♣ Méthode de descente avec recherche linéaire (Wolf/Armijio).
- ♣ Méthode de région de confiance.
- ♣ Méthode de la quasi -Newton.
- ♣ Méthode de Nelder Mead.

Exemple 2.1.4. Soit le problème suivant :

$$\begin{cases} x^2 - 4xy + y^3 + 4y \rightarrow \min \\ (x, y) \in \mathbb{R}^2. \end{cases}$$

(2.24)

Le problème est assez intéressant à résoudre car celui-ci n'est pas convexe. Par application du théorème ci-dessous (annexe) d'analyse mathématique. La représentation des points critiques est :

<i>(Point – critique test – convexité</i>	<i>Conclusion</i>
(4, 2) Positif	(4,2) minimum Local
($\frac{4}{3}, \frac{2}{3}$) Négatif	($\frac{4}{3}, \frac{2}{3}$) Point selle

Nous avons par exemple un minimum local, on ne peut pas affirmer donc que c'est la meilleure solution.

B / La programmation non linéaire avec contraintes [17, 34, 22, 81, 12, 31, 16]

La programmation non linéaire a connu beaucoup de développement, nous dressons dans ce sens, les résultats les plus importants car liés étroitement à l'optimisation globale.

La formulation dans \mathbb{R}^n , d'un problème non linéaire est :

$$(PNL) \left\{ \begin{array}{l} f(x) \rightarrow \min \\ g_i(x) = 0 \quad , i = 1 \dots l_E \\ g_i(x) \leq 0 \quad , i = l_{E+1} \dots l \\ x \in \mathbb{R}^n. \end{array} \right. \quad (2.25)$$

Les fonctions f et g_i ($i=1 \dots l$) sont des fonctions réelles définies et dérivables dans leurs domaines de définitions (une des fonctions est non linéaire) (en général différentiables selon le contexte : l'ordre des conditions 1° ou 2° ordre comme cela présenté ci-dessous).

Posons $E = \{1, \dots, l_E\}$ et $I = \{l_{E+1}, \dots, l\}$, ensembles des indices des contraintes d'égalités et d'inégalités.

Considérons le domaine de définition (domaine des solutions admissibles) S_AD pour le problème (PNL) comme suit :

$$S_AD = \{x \in \mathbb{R}^n : g_i(x) = 0, i \in E, g_i(x) \leq 0, i \in I\}.$$

Définition 2.1.9. La contrainte d'inégalité $g_i(x) \leq 0$ au point x^* est dite : active si $g_i(x^*) = 0$, inactive si $g_i(x^*) < 0$ et violée si $g_i(x^*) > 0$.

Pour $x \in \mathbb{R}^n$ de S_AD , l'ensemble des indices des contraintes actives est donné par :

$$A(x) = E(x) \cup I_a(x).$$

$I_a(x) = \{i : g_i(x) = 0, i \in I\}$ ensemble des indices de contraintes d'inégalités actives en x .
($E = E(x)$)

Définition 2.1.10. Soit,

$$D_AD(x, S_AD) = \{d \in \mathbb{R}^n : x + \theta d \in S_AD, \forall \theta \in [0, \alpha], \alpha > 0, d \neq 0\}.$$

Est l'ensemble des directions admissibles au point $x \in S_AD$ (pour les contraintes).

Définition 2.1.11. Soit $x \in S_AD$ et $d \in \mathbb{R}^n$, l'ensemble suivant :

$$FL(x) = \left\{ \begin{array}{l} d \in \mathbb{R}^n : \\ d' \nabla g_i(x) = 0 \quad , i = 1 \dots l_E \\ d' \nabla g_i(x) \leq 0 \quad , i = l_{E+1} \dots l \end{array} \right\}. \quad (2.26)$$

Est l'ensemble des directions admissibles linéaires au point $x \in S_AD$.

Définition 2.1.12. Soit $x \in S_AD$ et $d \in \mathbb{R}^n$, l'ensemble suivant :

$$TAS(x) = \left\{ \begin{array}{l} d \in \mathbb{R}^n : \\ x + \theta_k d_k \in S_AD \quad , \forall k = 1, 2, \dots \\ d_k \rightarrow d \quad \quad \quad , \theta_k \rightarrow 0; \theta_k > 0 \end{array} \right\}. \quad (2.27)$$

Est l'ensemble des directions admissibles séquentielles au point $x \in S_AD$.

Lemme 2.1.3. Soit $x \in S_AD$.

Si toutes les fonctions contraintes sont différentiables au point x et donc

$$D_AD(x, S_AD) \subseteq TAS(x) \subseteq FL(x).$$

Démonstration. Voir,

les références suivantes [17, 34, 22, 81].

□

Concernant et s'agissant de $FL(x) \subseteq TAS(x)$, n'est pas toujours vrai, mais importante pour énoncer les conditions du premier ordre Kuhn et Tucker (1951), qui ont fait la supposition en posant $FL(x) = TAS(x)$ ce qui se traduit par les conditions de qualifications des contraintes (QC) (régularité des contraintes).

Lemme 2.1.4. Soit $x \in S_AD$.

Pour que (QC) soit vérifiée en tout point $x \in S_AD$, il suffit que l'une des conditions (1) ou (2) soit réalisée :

(1) Toutes les contraintes g_i sont linéaires (Karlin).

(2) Toutes les contraintes g_i sont convexes et S_AD a un intérieur non vide (Slater).

Pour que (QC) soit vérifiée en un point $x^* \in S_AD$, il suffit :

(3) Les gradients $\nabla g_i(x^*)$ des contraintes actives en x^* soient linéairement indépendants (Fiacco et McCormick).

Définition 2.1.13. Soit $x \in S_AD$.

La direction $d \in D_AD(x, S_AD)$ est dite de descente si :

$$d' \nabla f(x) < 0.$$

$$\mathbb{D} = \{ d \in \mathbb{R}^n : d' \nabla f(x) < 0 \} \quad (2.28)$$

est l'ensemble des directions admissibles de descente au point $x \in S_AD$.

Théorème 2.1.10. *Soit $x^* \in S_{AD}$.*

Un minimum local du problème (PNL), avec toutes les fonctions f, g_i $i = (1, \dots, l)$, différentiables en x^ alors :*

$$d' \nabla f(x^*) \geq 0, \forall d \in T_{AS}(x^*).$$

Démonstration. Indication :

Pour la preuve, pour ce théorème, pouvant s'effectuer par analogie et en s'inspirant de la preuve déjà faite sur le même cas pour l'optimisation non linéaire sans contraintes.

□

Les conditions (QC) et l'existence des facteurs de lagrange grâce à des lemmes intermédiaires nous permettent de poser le théorème de Karush Kuhn et Tucker (KKT).

La fonction de lagrange étant :

$$L(x, \lambda) = f(x) + \sum_{i=1}^l \lambda_i g_i(x).$$

Théorème 2.1.11. *(Condition nécessaire d'optimalité du premier ordre (K.K.T))*

Soit $x^ \in S_{AD}$,*

un minimum local du problème (PNL), si les conditions de qualifications de contraintes (QC) sont satisfaites (vérifiées), alors il existe des multiplicateurs de lagrange λ_i^ , ($i = 1, \dots, l$) et tel que pour (x^*, λ^*) , on a :*

$$\left\{ \begin{array}{l} \nabla f(x^*) + \sum_{i=1}^l \lambda_i^* \nabla g_i(x^*) = 0 \\ g_i(x^*) = 0, \forall i \in E \\ g_i(x^*) \leq 0, \forall i \in I \\ \lambda_i^* \geq 0, \forall i \in I \\ \lambda_i^* g_i(x^*) = 0, \forall i \in I. \end{array} \right. \quad (2.29)$$

Pour les conditions nécessaires du 2° ordre, peuvent s'énoncer de la manière suivante moyennant des restrictions comme c'est le cas pour l'établissement des conditions du 1° ordre.

Théorème 2.1.12. *(Condition nécessaire d'optimalité du premier ordre et deuxième ordre)*

Soit $x^ \in S_{AD}$.*

Un minimum local du problème (PNL), si les conditions de qualifications de contraintes (QC) sont satisfaites (vérifiées), alors il existe des multiplicateurs de lagrange λ_i^ , ($i = 1, \dots, l$) et tel que pour (x^*, λ^*) , on a :*

$$\left. \begin{array}{l} \nabla f(x^*) + \sum_{i=1}^l \lambda_i^* \nabla g_i(x^*) = 0 \\ g_i(x^*) = 0, \forall i \in E \\ g_i(x^*) \leq 0, \forall i \in I \\ \lambda_i^* \geq 0, \forall i \in I \\ \lambda_i^* g_i(x^*) = 0, \forall i \in I \\ d' \nabla_{xx}^2 L(x^*, \lambda^*) d \geq 0, \forall d \neq 0 \\ d' \nabla g_i(x^*) = 0, i \in E \\ d' \nabla g_i(x^*) = 0, i \in I, g_i(x^*) = 0. \end{array} \right\}. \quad (2.30)$$

Exemple 2.1.5. Soit le problème suivant :

$$\left\{ \begin{array}{l} 2x^2 + xy + 12y^2 - x + 4y \rightarrow \min \\ x + y \leq 3 \\ 2x + y \leq 2 \\ (x, y)' \in \mathbb{R}_+^2. \end{array} \right.$$

(2.31)

Ce problème quadratique convexe est assez intéressant à résoudre, on peut utiliser la méthode du support avec ses variantes (primale,duale,primale duale), la méthode de lemke ou tout simplement une variante de l'algorithme classique du Simplex linéaire autrement dit la méthode de Wolf (avec les facteurs de Lagrange) ou toutes les méthodes locales sont valables. Une variante de la méthode de Karmarkar est fortement recommandée, conseillée et indiquée si on est fortement soucieux du paramètre temps (la complexité).

Remarque 2.1.6. Comme on le verra un peu plus loin à travers un exemple, il n'est pas toujours aisé de travailler avec les facteurs de Lagrange dans le cas général.

Théorème 2.1.13. (Condition suffisantes d'optimalité du deuxième ordre)

Soit $x^* \in S_{AD}$,

un point régulier satisfaisant les conditions de KKT et de plus (entre autres les conditions KKT vérifiées) :

$$\left\{ \begin{array}{l} \nabla f(x^*) + \sum_{i=1}^l \lambda_i^* \nabla g_i(x^*) = 0 \\ g_i(x^*) = 0, \forall i \in E \\ g_i(x^*) \leq 0, \forall i \in I \\ \lambda_i^* \geq 0, \forall i \in I \\ \lambda_i^* g_i(x^*) = 0, \forall i \in I \\ \lambda_i^* > 0, \forall i \in A(x^*) \\ d' \nabla_{xx}^2 L(x^*, \lambda^*) d > 0, \forall d \neq 0 \\ d' \nabla g_i(x^*) = 0, i \in E \\ d' \nabla g_i(x^*) = 0, i \in I, g_i(x^*) = 0. \end{array} \right. \quad (2.32)$$

Alors x^* est minimum local strict.

Exemple 2.1.6. Soit le problème suivant :

$$(2.33) \quad \left\{ \begin{array}{l} \frac{1}{2}(x^2 + y^2) \rightarrow \min \\ y^2 = (x - 1)^3 \\ (x, y) \in \mathbb{R}^2. \end{array} \right.$$

Il est aisé de vérifier graphiquement que le point $(1, 0)$ est la solution optimale. Par application des multiplicateurs de lagrange, le Lagrangien est :

$$L(x, y, \lambda) = \frac{1}{2}(x^2 + y^2) + \lambda(-y^2 + (x - 1)^3).$$

Impossible de trouver un triplet $(1, 0, \lambda)$ qui puisse annuler à la fois :

$$\frac{\partial L}{\partial x} = x + 3\lambda(x - 1)^2 : \frac{\partial L}{\partial y} = y - 2y\lambda : \frac{\partial L}{\partial \lambda} = (-y^2 + (x - 1)^3).$$

Le point $(1, 0)$ n'est pas qualifié (régulier).

Exemple 2.1.7. Reprenons l'exemple précédent

Nous allons exprimer le problème avec des multiplicateurs de lagrange généralisés connus sous la forme de multiplicateurs de lagrange généralisés (Fritz John) [34] (Forme généralisée du Théorème de Karush Khun et Tucker (Fritz John)).

Théorème 2.1.14. (Fritz John), Soit le (PNL) suivant :
Une formulation dans \mathbb{R}^n , d'un problème non linéaire est :

$$(PNL) \begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0 & , i = 1 \dots l \\ h_j(x) = 0 & , j = 1 \dots m \\ x \in \mathbb{R}^n. \end{cases} \quad (2.34)$$

Les fonctions f , g_i et h_j sont des fonctions réelles définies et dérivables dans leurs domaines de définitions (une des fonctions est non linéaire) (en général différentiables).

Si x^* est solution de ce (PNL) alors il existe $\lambda_0 \in \mathbb{R}_+$, $\lambda \in \mathbb{R}_+^l$ et $\mu \in \mathbb{R}^m$ tel que :

$$(PNL) \begin{cases} \lambda_0 g(x^*) = 0 \\ (\lambda_0, \lambda, \mu) \neq 0 \\ \lambda_0 \nabla f(x^*) + \lambda' \nabla g(x^*) + \mu' \nabla h(x^*) = 0. \end{cases} \quad (2.35)$$

Finalement l'application de ce dernier théorème à l'exemple précédent, confirme que la solution optimale $x^* = (1, 0)$.

Les méthodes utilisées dans le cadre de la programmation non linéaire avec contraintes [17, 34, 81, 16, 78, 104, 79, 85]

Entre autres :

- ♣ Méthode de Newton sous plusieurs variantes.
- ♣ Méthode du Lagrangien augmenté.
- ♣ Méthode SQP (Sequentiel Quadratics Problems).
- ♣ Méthode des points intérieurs.
- ♣ Méthode des coupes.

2.1.7 La programmation convexe [17, 34, 22, 81, 12, 9, 83]

La programmation convexe concerne une classe de problèmes (PNL) dont la fonction objectif et domaine des solutions admissibles sont convexes, c'est un domaine très intéressant comme c'est le cas pour la programmation mathématique locale pour des perspectives d'optimisation globale. Une formulation du problème (PNL) convexe est comme suit dans \mathbb{R}^n :

$$(PNLC) \begin{cases} f(x) \rightarrow \min \\ g_i(x) = a'_i x + b_i = 0 & , i = 1 \dots l_E \\ g_i(x) \leq 0 & , i = l_{E+1} \dots l \\ x \in \mathbb{R}^n \end{cases} \quad (2.36)$$

$a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$.

Théorème 2.1.15. *Pour un terme positif signomial ([73]) :*

$s(z) = c \prod_{i=1}^I z_i^{p_i}$ est convexe ssi

(i) $p_i \leq 0, \forall, i = 1 \dots I$.

(ii) $\exists k, p_k + \sum_{i \neq k} p_i \geq 1$ quand $p_i \leq 0; \forall, i = 1 \dots I, i \neq k$.

Pour un terme négatif signomial :

$s(z) = c \prod_{i=1}^I z_i^{p_i}$ est convexe ssi $p_i \geq 0; \forall, i = 1 \dots I$ et $\sum_{i=1}^I p_i \leq 1$.

Théorème 2.1.16. *Tout minimum local x^* du problème (PNLC) est minimum global et l'ensemble des minimums globaux est un ensemble convexe.*

Démonstration. Soit x^* un minimum local pour le problème (PNLC), mais pas global.

Soit \bar{x} , un vecteur du domaine de définition tel que $f(\bar{x}) \leq f(x^*)$.

Soit $0 < \xi < 1$ et suffisamment petit.

$$x_\xi = \xi \bar{x} + (1 - \xi)x^* = x^* + \xi(\bar{x} - x^*).$$

$0 < \xi < 1, \implies$ calculons donc une combinaison de $f(\bar{x})$ et $f(x^*)$

$$f(x_\xi) \leq \xi f(\bar{x}) + (1 - \xi)f(x^*).$$

$$\leq f(x^*) + \xi(f(\bar{x}) - f(x^*)).$$

$$< f(x^*).$$

Ce qui donne une contradiction, donc x^* minimum Global.

Soient x_1 et x_2 deux minimums globaux.

Posons $\xi \in]0, 1[$

$$x_\xi = \xi x_1 + (1 - \xi)x_2.$$

$$f(x_\xi) \geq f(x_1) = f(x_2).$$

Mais,

$$f(x_\xi) \leq \xi f(x_1) + (1 - \xi)f(x_2) = f(x_2), \text{ (convexité).}$$

Donc

$$f(x_\xi) = f(x_2) = f(x_1).$$

Ce qui prouve que l'ensemble des minimums Globaux est convexe.

□

Exemple 2.1.8. Soit le problème suivant :

$$(2.37) \quad \left\{ \begin{array}{l} 3x_1^{-0.9}x_2^{-1.6}x_3^{-4} + 2x_4^{-0.3}x_5^{3.6} + 5x_6^{-1.9}x_7^{-0.6}x_8^{-1} + x_1^4 + x_2^4 + x_1^6 + x_2^6 \rightarrow \min \\ 4.5x_1^{-3.8}x_2^{-2.2}x_3^8 + 3.5x_4^{-0.7}x_5^{-1.6} + 2.5x_6^{-4.5}x_7^{-1.9}x_8^{-1.2} \leq 2 \\ 5x_1^{-1.5}x_2^{-1.9}x_3^{5.5} \leq 2 \\ 6x_4^{-3.2}x_5^{-1.2} \leq 2 \\ 6.2x_6^{4.5}x_7^{-2.5}x_8^{-0.6} \leq 2 \\ x_1^{5.5}x_2^{-0.5}x_3^{-3.5} \leq 2 \\ 2.5x_4^{5.5}x_5^{-1.4} \leq 2 \\ 7x_6^{-1.2}x_7^{-7.3}x_8^{-5.4} \leq 2 \\ (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)' \in \mathbb{R}_+^{*8}. \end{array} \right.$$

Ce problème est convexe (très simple à vérifier et à résoudre).

Lemme 2.1.5. Si f .

Avec f strictement convexe sur l'ensemble des solutions admissibles convexe alors le minimum global pour le (PNLC) est unique.

Démonstration. Soient x_1, x_2 deux solutions globales et $x_1 \neq x_2$.

$$f(x_1) = f(x_2) = \min_{x, \text{admissible}} f(x)$$

⇒ par convexité

$$\delta = \frac{x_1 + x_2}{2} = \frac{1}{2}x_1 + \frac{1}{2}x_2.$$

⇒ par stricte convexité

$$f(\delta) = f\left(\frac{1}{2}x_1 + \frac{1}{2}x_2\right) < \frac{1}{2}f(x_1) + \frac{1}{2}f(x_2).$$

⇒ contradiction.

Ou autrement :

Soit $\xi \in]0, 1[$.

$$x_\xi = \xi x_1 + (1 - \xi)x_2; x_\xi \neq x_1.$$

$$f(x_\xi) < \xi f(x_1) + (1 - \xi)f(x_2).(\text{convexité/strict})$$

$$< f(x_1).$$

Ce qui donne une contradiction, sachant x_1 minimum global. [(contradiction)].

Ce qui prouve le lemme.

□

Exemple 2.1.9. Soit le problème suivant :

$$(2.38) \quad \left\{ \begin{array}{l} 5x_1^{0.9}x_2^{-1.5}x_3^{-3} + 7x_4^{-0.3}x_5^{2.6} + 3x_6^{-1.8}x_7^{-0.5}x_8^1 + x_1^4 + x_2^4 + \exp(x_1 + x_2) \rightarrow \min \\ 8.2x_1^{-3.8}x_2^{2.2}x_3^{4.3} + 0.2x_4^{-0.7}x_5^{-1.6} + 2.2x_6^{4.3}x_7^{-1.9}x_8^{8.5} \leq 1 \\ 2x_1^{2.3}x_2^{1.7}x_3^{4.5} \leq 1 \\ 1.6x_4^{-2.1}x_5^{0.4} \leq 1 \\ 7.2x_6^{4.5}x_7^{-2.7}x_8^{-0.6} \leq 1 \\ 1.1x_1^{1.6}x_2^{0.4}x_3^{-3.8} \leq 1 \\ 1.7x_4^{5.4}x_5^{1.3} \leq 1 \\ 1.3x_6^{-1.1}x_7^{7.3}x_8^{-5.6} \leq 1 \\ (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)' \in \mathbb{R}_+^{*8}. \end{array} \right.$$

Ce problème est non convexe (pas simple à résoudre mais pris en charge par l'optimisation globale).

Théorème 2.1.17. *Pour le problème (PNLC), si les conditions de K.K.T sont satisfaites en x^* alors x^* est minimum global.*

Démonstration. Soient (x^*, λ^*) un point K.K.T.

Il est normal d'avoir,

$$L(x, \lambda^*) \leq f(x).$$

De plus

$$\begin{aligned} L(x, \lambda^*) &\geq L(x^*, \lambda^*) + (x - x^*)' \nabla_x L(x^*, \lambda^*) \\ &\geq L(x^*, \lambda^*) \\ &\geq f(x^*). \end{aligned}$$

Et donc

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

□

Exemple 2.1.10. Soit le problème suivant :

$$(2.39) \quad \left\{ \begin{array}{l} -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 - 6 \rightarrow \min \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \leq 5 \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \geq 2 \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \\ 0 \leq x_3 \leq 2 \\ (x_1, x_2, x_3)' \in \mathbb{R}_+^3. \end{array} \right.$$

Ce problème quadratique est assez intéressant à résoudre car celui-ci n'est pas convexe (très simple à vérifier).

Par application des théorèmes ci-dessus d'analyse mathématique, le problème est assez complexe à résoudre. Donc nous pouvons le situer dans le cadre des problèmes en optimisation globale et nous allons présenter son prototype de résolution (la solution) en fin de chapitre suivant consacré à "l'optimisation globale."

2.1.8 La dualité en programmation convexe [17, 34, 22, 81]

Le but de la dualité est de trouver une transformation pour le problème initial, le problème original est le primal celui transformé est le dual. La théorie de la dualité est souvent attachée à la notion des multiplicateurs de Lagrange, si bien que en optimisation, on trouve des méthodes primales, des méthodes duales et des méthodes primales duales par exemples.

Définition 2.1.14. Le problème dual associé au problème convexe (PNLC) est le suivant :

$$(PNLCD) = \left\{ \begin{array}{l} L(x, \lambda) \rightarrow \max_{x, \lambda} \\ \nabla_x L(x, \lambda) = 0 \quad , \lambda_i \geq 0, \forall i \in I. \end{array} \right. \quad (2.40)$$

Théorème 2.1.18. Soit $(x^*, \lambda^*)'$ maximum du problème dual (PNLCD) et si le Hessian du Lagrangien est régulier en x^* et si les contraintes du (PNLCD) sont qualifiées en $(x^*, \lambda^*)'$, alors x^* est un minimum du (PNLC).

Démonstration. De manière équivalente

Soient les problèmes P et D primal et dual respectivement :

$$P \begin{cases} f(x) \rightarrow \min \\ g_i(x) \leq 0 \end{cases}, x \in \mathbb{R}^n. \quad (2.41)$$

$$g(x) = (g_1(x), g_2(x), \dots, g_I(x))'$$

$$D \begin{cases} L(x, \lambda) = f(x) + \lambda'g(x) \rightarrow \max_{x, \lambda} \\ \nabla_x L(x, \lambda) = 0 \end{cases}, \lambda_i \geq 0, \forall i \in I, x \in \mathbb{R}^n \quad (2.42)$$

$\beta^* \in \mathbb{R}^n$ les multiplicateurs de lagrange existants associés au problème D et son maximum $(x^*, \lambda^*)'$. Dans cet ordre d'idées,

$$\begin{aligned} g(x^*) + \beta^* g'(x^*) &\leq 0. \\ \nabla_x L(x^*, \lambda^*) + \beta^* \nabla_x^2 L(x^*, \lambda^*) &= 0. \\ \beta^* \nabla_x^2 L(x^*, \lambda^*) &= 0. \\ \beta^* &= 0. \end{aligned}$$

Donc à partir de cela :

$$g(x^*) \leq 0.$$

Et donc

$$x^*$$

minimum de P . □

Théorème 2.1.19. *(Dualité faible)*

Soit x^* minimum du problème convexe primal (PNLC). Si f et $g_i, \forall i \in I$ sont continument différentiables (la condition de régularité vérifiée), alors (x^*, λ^*) résout (PNLCD) et de plus :

$$f(x^*) = L(x^*, \lambda^*).$$

Démonstration. De part le théorème (K.K.T), il existe des facteurs de lagrange λ_i^* avec :

$$\nabla_x L(x^*, \lambda^*) = 0, \lambda_i^* g_i(x^*) = 0.$$

Cela pour tous les indices.

$$\implies f(x^*) = L(x^*, \lambda^*).$$

Considérons une solution du dual (x, λ) .

$$L(x^*, \lambda^*) = f(x^*).$$

$$\begin{aligned}
&\geq f(x^*) + \sum_{i=1}^l \lambda_i g_i(x^*). \\
&\geq L(x^*, \lambda). \\
&\geq L(x, \lambda) + (x^* - x)' \nabla_x L(x, \lambda). \\
&\geq L(x, \lambda).
\end{aligned}$$

Ce qui prouve le résultat.

□

Théorème 2.1.20. (*Dualité forte*)

Soit \bar{x} une solution réalisable du problème convexe primal (PNLC) et (x, λ) une solution réalisable du problème dual (PNLCD) alors :

$$f(\bar{x}) \geq L(x, \lambda).$$

Démonstration. Soient \bar{x} , solution du primal et (x, λ) , solution du dual.

Par convexité :

$$\begin{aligned}
f(\bar{x}) - f(x) &\geq \nabla' f(x)(\bar{x} - x). \\
&\geq - \sum_{i=1}^l \lambda_i \nabla' g_i(x)(\bar{x} - x). \\
&\geq - \sum_{i=1}^l \lambda_i (g_i(\bar{x}) - g_i(x)). \\
&\geq \sum_{i=1}^l \lambda_i g_i(x).
\end{aligned}$$

Et donc

$$f(\bar{x}) \geq f(x) + \sum_{i=1}^l \lambda_i g_i(x) = L(x, \lambda).$$

Ce qui prouve le Résultat.

□

Définition 2.1.15. Point "selle"

On distingue un point important dans la théorie de la dualité, autour duquel sont bâtis beaucoup d'algorithmes de résolution de programmes mathématiques, en occurrence le point "selle".

C'est un point $(x^*, \lambda^*)'$ tel que :

$$L(x, \lambda^*) \leq L(x^*, \lambda^*) \leq L(x^*, \lambda).$$

Pour tout $(x, \lambda)'$ satisfaisants les contraintes.

Théorème 2.1.21. *Soit,*

$(x^*, \lambda^*)'$ est un point selle du Lagrangien pour (PNLC) $\iff (x^*, \lambda^*)'$ satisfait les conditions KKT pour (PNLC).

2.1.9 La dualité en programmation mathématique non convexe [17, 34, 81]

Soit la formulation générale dans $X(\text{convexe}) \subseteq \mathbb{R}^n$, d'un problème non linéaire est :

$$(PNLG) \begin{cases} f(x) \rightarrow \min \\ h(x) = 0 \\ g(x) \leq 0 \\ x \in X; X = \mathbb{R}^n. \end{cases} \quad (2.43)$$

Les fonctions f de \mathbb{R}^n vers \mathbb{R} , h de \mathbb{R}^n vers \mathbb{R}^m et g de \mathbb{R}^n vers \mathbb{R}^p .

Son problème dual est le suivant :

$$(PNLGD) = \begin{cases} \psi(\lambda, \nu) \rightarrow \max_{\lambda, \nu} \\ \nu \geq 0 \\ (\lambda, \nu) \in X_q \end{cases} \quad (2.44)$$

$$\psi(\lambda, \nu) = L(x, \lambda, \nu) \rightarrow \min_{x \in \mathbb{R}^n} \quad ; \quad X_q = \{\lambda, \nu : \psi(\lambda, \nu) > -\infty\}.$$

Théorème 2.1.22. *Étant donné :*

x^* solution du problème primal (PNLG) et considérons la fonction dual ψ . Soient $\lambda \in \mathbb{R}^m$ et $\nu \in \mathbb{R}^p$, $\nu \geq 0$ alors :

$$\psi(\lambda, \nu) \leq f(x^*).$$

Démonstration. Étant donné :

$$\psi(\lambda, \nu) = \min_{x \in \mathbb{R}^n} L(x, \lambda, \nu).$$

$$\min_{x \in \mathbb{R}^n} L(x, \lambda, \nu) \leq L(x^*, \lambda, \nu) = f(x^*) + \lambda' h(x^*) + \nu' g(x^*) = f(x^*) + \nu' g(x^*).$$

$$\psi(\lambda, \nu) \leq f(x^*).$$

□

Théorème 2.1.23. (Concavité - convexité du problème dual) :

Étant donné le problème dual (PNLGD) du primal (PNLG) alors la fonction objectif du dual est concave et les contraintes du dual sont convexes.

Démonstration. Étant donné :

x variable primal, θ et $\bar{\theta}$ variables duales.

$$\theta \neq \bar{\theta}, \theta = (\lambda, \nu), \bar{\theta} = (\bar{\lambda}, \bar{\nu}).$$

$$\theta, \bar{\theta} \in X_q, \nu \geq 0, \bar{\nu} \geq 0$$

$\beta \in \mathbb{R}$ tel que $0 \leq \beta \leq 1$ et considérons la fonction L :

$$L(x, \beta\theta + (1 - \beta)\bar{\theta}) = \beta L(x, \theta) + (1 - \beta)L(x, \bar{\theta})$$

$$\min_{x \in \mathbb{R}^n} L(x, \beta\theta + (1 - \beta)\bar{\theta}) \geq \min_{x \in \mathbb{R}^n} \beta L(x, \theta) + \min_{x \in \mathbb{R}^n} (1 - \beta)L(x, \bar{\theta})$$

$$\psi(\beta\theta + (1 - \beta)\bar{\theta}) \geq \beta\psi(\theta) + (1 - \beta)\psi(\bar{\theta}).$$

Donc ψ est concave. De plus, on remarque pour θ et $\bar{\theta}$ pris à partir de X_q , la combinaison convexe $\beta\theta + (1 - \beta)\bar{\theta} \in X_q$. □

Théorème 2.1.24. Étant donné :

x^* solution du problème primal (PNLG) et considérons la fonction dual ψ . Soit (λ^*, ν^*) la solution associée du problème dual (PNLGD) alors

$$\psi(\lambda^*, \nu^*) \leq f(x^*).$$

Démonstration. Étant donné :

Le théorème précédent, on prend tout simplement $\lambda = \lambda^*$ et $\nu = \nu^*$. □

Corollaire 2.1.1. Étant donné :

Le problème primal (PNLG) et considérons son problème dual (PNLGD) alors :

- Si le problème primal est non borné, alors le problème dual est non admissible.
- Si le problème dual est non borné, alors le problème primal est non admissible.

Démonstration. Soit,

le coût du primal optimal tendant vers - l'infini, il ne peut y avoir de variable duale $(\lambda, \nu)'$ pour cela. On procède de façon identique pour le coût du dual optimal tendant vers + l'infini. □

CHAPITRE 3

Optimisation Globale et Contribution à l'Optimisation Globale

3.0.10 Introduction : L'optimisation globale [78, 86]

L'optimisation globale est une branche au carrefour des mathématiques appliquées et de l'analyse numérique consistant en l'optimisation d'une fonction ou d'un ensemble de fonctions suivant certains critères. Typiquement, un ensemble de bornes et plus général les contraintes sont également présentées et les variables de décision sont optimisées en tenant compte des contraintes. L'optimisation globale se distingue de l'optimisation locale par son objectif de trouver le maximum ou le minimum absolu sur l'ensemble des valeurs du domaine de définition, par opposition à la recherche de minimas ou maximas locaux.

3.0.11 Domaines d'application de l'optimisation globale [34, 16]

- **L'ajustement de courbes** : Par l'analyse des moindres carrés linéaires ou non linéaires et d'autres généralisations, pour ajuster des paramètres de modèles aux données expérimentales en chimie, physique, biologie, économie, finance, médecine, astronomie, ingénierie,...etc.
- **Prédiction de structures** : Comme dans les cas de problèmes économiques, météorologiques, psychologiques, médicaux,...etc.
- **La logique floue** : Avec des applications potentielles : scindées sous trois catégories : le raisonnement dans l'incertain (systèmes experts, configurations systèmes, la maintenance de parcs matériel,...etc), la commande automatique floue et la robotique. Sans oublier que les problèmes de type flou en optimisation globale peuvent être imprégnés du caractère stochastique.
- **Les problèmes de Ratios** : En économie essentiellement. Problème du voyageur de commerce vers les problèmes de tournées de véhicules en grandes dimensions avec options

par exemples de fenêtres temps.

- Conception de circuits électriques.
- Chimie et contrôle optimal : Les résolutions des équations de Schrödinger.
- Vérification de la sécurité : Ingénierie de la sécurité (par exemple, structures mécaniques, bâtiments).
- Analyse des cas les plus défavorables.
- Verres à lunettes.
- Problèmes mathématiques spéciaux (par exemples, les conjectures).
- Problèmes d’emballage d’objets (conception de configuration).

Dans le cadre de l’optimisation globale, on s’intéressera à l’optimisation à une dimension et à l’optimisation à plusieurs dimensions.

3.0.12 Les techniques d’optimisation dans \mathbb{R} [34, 16, 17]

En général l’optimisation unidimensionnelle n’est pas toujours aisée, en effet on retrouve les mêmes difficultés que dans l’optimisation en dimension supérieure (différentiabilité, convexité, continuité, lipschitzien, minimas locaux et globaux,...etc.).

Définition 3.0.16. Soit f une fonction définie sur l’intervalle $[a, b]$, et soit x^* son argument minimum sur cet intervalle. On dit que f est unimodale sur $[a, b]$, si elle est monotone strictement décroissante sur $[a, x^*[$ et monotone strictement croissante sur $]x^*, b]$.

♣ Méthode dichotomique

Pour calcul de x^* . Méthode simple mais peu efficace, à chaque itération nécessite 02 évaluations et permet de réduire de moitié l’intervalle, avec une vitesse de convergence linéaire de rapport $\frac{1}{2}$. Pour f unimodale sur $[a, b]$: $a = x_1 < x_2 < x_3 < x_4 < x_5 = b$, seuls les cinq cas suivants peuvent se présenter, avec x_3 milieu de $[a, b]$ et x_2, x_4 milieux respectivement de $[x_1, x_3]$, $[x_3, x_5]$.

$$1^\circ / f(x_1) < f(x_2) < f(x_3) < f(x_4) < f(x_5).$$

$$2^\circ / f(x_1) > f(x_2) < f(x_3) < f(x_4) < f(x_5).$$

$$3^\circ / f(x_1) > f(x_2) > f(x_3) < f(x_4) < f(x_5).$$

$$4^\circ / f(x_1) > f(x_2) > f(x_3) > f(x_4) < f(x_5).$$

$$5^\circ / f(x_1) > f(x_2) > f(x_3) > f(x_4) > f(x_5).$$

♣ Méthode du nombre d'or

La méthode de la section dorée est plus économique et conduit à une vitesse de convergence linéaire en fonction de $\frac{\sqrt{5}-1}{2}$ (inverse du nombre d'or) pour une économie d'évaluations de la fonction f à chaque itération. L'intervalle est découpé en 3 au lieu de 4 (cas de la méthode dichotomique).

Pour calcul de x^* . Pour f unimodale sur $[a, b] : a = x_1 < x_2 < x_3 < x_4 = b$ seuls les cas suivants peuvent se présenter.

$$1^\circ / f(x_1) < f(x_2) < f(x_3) < f(x_4).$$

$$2^\circ / f(x_1) > f(x_2) < f(x_3) < f(x_4).$$

$$3^\circ / f(x_1) > f(x_2) > f(x_3) < f(x_4).$$

$$4^\circ / f(x_1) > f(x_2) > f(x_3) > f(x_4).$$

Deux méthodes de détermination des zéros d'une fonction : la méthode de Newton-Raphson et la méthode de la sécante.

Les motivations d'utilisation de ces deux méthodes sont par exemples dans les méthodes dites de descentes (recherches linéaires) ou dans le calcul matriciel (calcul des valeurs propres).

♣ Méthode de Newton

Soit f une fonction de classe C^1 dont on sait qu'elle admet au moins un zéro. Supposons que l'on connaisse la valeur de f au point x^0 ainsi que la valeur de $f'(x^0)$. Une approximation g de f au voisinage de x^0 à travers le développement de Taylor à l'ordre 1, donne :

$$g(x) = f(x^0) + f'(x^0)(x - x^0).$$

Algorithme

La fonction g est linéaire ($f'(x^0) \neq 0$), on calcule son zéro x^1 donné par :

$$x^1 = x^0 - \frac{f(x^0)}{f'(x^0)}.$$

En réitérant le procédé, on peut ainsi engendrer une suite $(x^k)_{k \in \mathbb{N}}$, qui sous certaines conditions - converge vers un zéro de f .

Théorème 3.0.25. *Soit f de classe C^2 et x^* son zéro. Si $f'(x^*) \neq 0$ et si x^0 est suffisamment proche de x^* , la suite engendrée par la méthode de Newton converge vers x^* et la vitesse de convergence est quadratique.*

♣ Méthode de la sécante

Méthode similaire à la méthode de Newton précédente. La fonction f approximée par g passant par les points $(x^0, f(x^0))$ et $(x^1, f(x^1))$, on définit x^2 le zéro de g , autrement-dit

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

Algorithme

On définit une suite récurrente de type :

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

Que l'on peut arrêter en fixant un nombre maximal d'itérations ou test de proximité entre x^{k+1} et x^k .

Propriété 3.0.1. *La vitesse de convergence asymptotique de cette méthode est superlinéaire dans le voisinage x^* , racine de f .*

Ces méthodes sont restrictives, n'exploitent pas assez les éléments des fonctions à optimiser ou à traiter.

C'est dans ce sens qu'on utilise des méthodes plus rigoureuses afin d'obtenir des résultats plus raffinés à savoir entre autres, optimums globaux avec multiplicité.

♣ **Méthode Branch and Bound** . [68, 101, 95, 57, 47, 18]

La méthode de Branch and Bound (*B.B*) est une méthode pour l'optimisation globale. Méthode basée sur la subdivision du domaine réalisable en plusieurs sous ensembles. Pour chaque sous ensemble on construit par exemple une borne supérieure et une borne inférieure pour la fonction objectif afin d'inhiber les parties non réalisables ainsi que les parties non susceptibles de contenir l'optimum global et de travailler sur le reste des parties. À noter que cette méthode exploite les outils de l'analyse convexe.

Principe de la méthode

Soit (*PG*) un problème d'optimisation globale.

$$(PG) \begin{cases} f(x) \rightarrow \min \\ x \in D. \end{cases} \quad (3.1)$$

f continue non convexe. D convexe ou pas. D (subdivisé en D_k).

L'algorithme (*B.B*) consiste à engendrer des suites $\{UB_k\}$ et $\{LB_k\}$ des bornes supérieures et inférieures respectivement de la valeur minimale de la fonction objectif de (*PG*).

Les bornes supérieures et inférieures à l'itération k sur l'ensemble D_k (subdivisé en D_{ki}) :

$$\begin{cases} UB_k = \min(UB_{ki}). \\ LB_k = \min(LB_{ki}). \end{cases} \quad (3.2)$$

Le principe est le meilleur d'abord. Tout sous ensemble sur lequel la borne inférieure dépasse UB_k sera évidemment éliminé.

♣ **Méthode Hölder mixée (combinée)** (B.B) [68, 101, 57, 47, 18].

Soit f une fonction continue pour le problème d'optimisation globale (P) suivant :

$$(P) \begin{cases} f(x) \rightarrow \min \\ x \in I. \end{cases} \quad (3.3)$$

$I = [a, b] = I_0$: intervalle de \mathbb{R} . f continue vérifiant la propriété de Hölder.

(i.e.)

$$\forall x \in I, \forall y \in I, |f(x) - f(y)| \leq L|x - y|^{\frac{1}{\alpha}}.$$

L : constante strictement positive (constante Lipschitzienne), $\alpha \geq 1$.

Remarque 3.0.7. L'algorithme pour la résolution de (P) se base sur la construction de fonctions bornes inférieures combiné à la méthode Branch and Bound.

Algorithme HOL^1 dans \mathbb{R}

1. $x \in I_j = [a^j, b^j]$, $j = 0, 1, 2, \dots, k$, $k \in \mathbb{N}$.
2. $j = k$, $I_k = [a^k, b^k]$, $\epsilon > 0$, $L > 0$.
3. Évaluations : $f_a^k(x)$, $f_b^k(x)$, $x = x^k$, LB_k , UB_k .

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

(3.4)

4. Si $UB_k - LB_k \leq \epsilon$, Stop on a la solution pour (P) .

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

(3.5)

5. Sinon subdiviser I_k en sous intervalles, soient : I_{k1} et I_{k2} .

$$\bigcup_{i=1}^2 I_{ki} = I_k; I_{k1} \cap I_{k2} = \emptyset.$$

6. Construire les problèmes des bornes inférieures et supérieures sur I_{k1} et I_{k2} , $i = 1..2$.
7. Soit

$$\begin{cases} UB_{k+1} = \min(UB_{k1}, UB_{k2}, UB_k) \\ LB_{k+1} = \min(LB_{k1}, LB_{k2}) = LB_{k*} \end{cases}$$

(3.6)

8. $I_k = \{I_{k1}, I_{k2}\}$
9. Éliminer tous les sous ensembles I_{kj} , $j = 1, 2$ tel que

$$LB_{kj} > UB_{k+1}.$$

$$I_{k+1} = I_{k*}$$

10 . $k = k + 1$ aller à 4.

Théorème 3.0.26. (*Gourdin et al pour la minimisation*) [47, 18]

L'algorithme HOL^1 est fini ou bien converge asymptotiquement vers l'optimum.

♣ **Méthode α BB** [11].

Soit le problème suivant en optimisation globale :

$$(P) \begin{cases} \min b(s) \\ s \in [s^0, s^1] \subset \mathbb{R}. \end{cases}$$

Avec b non convexe et de classe C^2 , fonction réelle sur l'intervalle $[s^0, s^1]$. Beaucoup de méthodes ont été élaborées pour étudier ce type de problème en littérature, (voir [42]).

Cette méthode se basant essentiellement sur la borne de floudas pour la résolution.

♣ **Méthode KBBm.** [3]

Cette méthode traite le même problème que celui ci-dessus.

♣ **Méthode Efficient BB.** [24]

Soit le problème suivant en optimisation globale :

$$(P) \begin{cases} \min b(s) \\ s \in [s^0, s^1] \subset \mathbb{R}. \end{cases}$$

Avec b non convexe et de classe C^2 , fonction réelle sur l'intervalle $[s^0, s^1]$.

Cette méthode a été présentée en conférence internationale de Constantine, était sélectionnée et puis publiée dans le journal (A.I.P 2017).

Sera présentée en détails et fait partie du chapitre contributions à l'optimisation globale.

♣ Méthode Best Efficient BB. [90]

Soit le problème suivant en optimisation globale :

$$(P) \begin{cases} \min b(s) \\ s \in [s^0, s^1] \subset \mathbb{R}. \end{cases}$$

Avec b non convexe et de classe C^2 , fonction réelle sur l'intervalle $[s^0, s^1]$.

Cette méthode a été soumise au journal RAIRO, acceptée et puis publiée aussi dans le journal RAIRO (RAIRO 2018).

Sera présentée en détails et fait partie du chapitre contributions à l'optimisation globale.

3.1 Problèmes pratiques en optimisation globale

Pour avoir une idée sur les cas réels en optimisation globale, choisissons d'expliciter cet échantillon qui suit en forme et type de résolution :

⊗ a) En optimisation ou l'ajustement de courbes [78, 86, 16]

Par exemple en optimisation, s'agissant du problème suivant :

$$\begin{cases} (\frac{1}{7} - \frac{x_2 x_3}{x_1 x_4})^2 \rightarrow \min \\ 5 \leq x_i \leq 80 \\ x_i \in \mathbb{N}, i = 1 \dots 4. \end{cases} \quad (3.7)$$

* Les problèmes de transport, production et affectation classiques dont il s'agit de minimiser les couts alors que les fonctions objectifs respectives ne sont pas convexes.

* Soit par exemple ce problème :

$$\begin{cases} 3x_1^2 + x_2^2 - 2x_1 x_2 \rightarrow \max \\ -3x_1^2 - 2x_2^2 + 5x_1 x_2 \leq -3 \\ 7x_1^2 + 8x_2^2 - 3x_1 x_2 \geq -200 \\ -3x_1^2 - 2x_2^2 - 8x_1 x_2 \leq 350 \\ 10x_1 - 4x_2 \leq 50 \\ -10 \leq x_1 \leq 10 \\ -10 \leq x_2 \leq 10. \end{cases}$$

Donc la difficulté de résolution de ces problèmes se sent.

⊗ b) Prédiction de structures [7]

Considérons le programme linéaire stochastique sous la forme suivante :

$$(P_s) \begin{cases} c'(\omega)x \rightarrow \min \\ x \in D = \{x \geq 0, Ax \leq b\} \end{cases} \quad (3.9)$$

c est un vecteur aléatoire, $A = (a_{ij})$; $b = (b_i)$ sont déterministes.

Parmi les méthodes offertes pour la résolution de ce type de problème, on recense la méthode de Katota et donc :

La minimisation de α - fractile de la fonction de distribution de l'objectif avec α choisi par le décideur.

$$(P'_s) \begin{cases} u \rightarrow \min \\ P(\omega/c'(\omega)x \leq u) = \alpha \\ x \in D = \{x \geq 0, Ax \leq b\}. \end{cases} \quad (3.10)$$

Dans le cas gaussien et donc :

$$P(\omega/c'(\omega)x \leq u) = \alpha \iff$$

$$P(\omega/c'(\omega)x \leq u) = \alpha = P(\omega/\frac{c'(\omega)x - \bar{c}'(\omega)x}{\sqrt{x'Vx}} \leq \frac{u - \bar{c}'(\omega)x}{\sqrt{x'Vx}}) = \phi(\frac{u - \bar{c}'(\omega)x}{\sqrt{x'Vx}})$$

ϕ est la fonction de répartition de la variable aléatoire normale centrée réduite et V matrice variance covariance.

Cela revient donc à résoudre :

$$\begin{cases} \bar{c}'(\omega)x + \phi^{-1}(\alpha)\sqrt{x'Vx} \rightarrow \min \\ x \in D = \{x \geq 0, Ax \leq b\}. \end{cases} \quad (3.11)$$

$\bar{c}'(\omega)x + \phi^{-1}(\alpha)\sqrt{x'Vx}$ est convexe si $\phi^{-1}(\alpha) \geq 0 \iff \alpha \geq \frac{1}{2}$.

On sent déjà la difficulté de résolution.

Si de plus A et b sont des variables aléatoires, alors on complique la résolution en imposant par exemple un seuil de probabilité individuelle α_i pour chaque contrainte avec $0 \leq \alpha_i \leq 1$. Donc La convexité de chaque contrainte non assurée.

⊗ c) La logique floue [59, 7, 17, 68, 101, 57]

Le domaine flou est très visible en médecine, industrie ou autres :

Soit le problème non linéaire suivant (PNL) :

$$(PNL) \begin{cases} f(x) \rightarrow \max \\ g_i(x) \leq 0, i = 1, \dots, m. \end{cases} \quad (3.12)$$

Les fonctions f, g_i , sont de $\mathbb{R}^n \rightarrow \mathbb{R}$.

À noter que dans le domaine du flou, on recense entre autres, le modèle Robuste et celui du flexible.

À ce dernier on peut lui associer le problème non linéaire flou flexible suivant :

$$(PNLFF) \begin{cases} f(x) \rightarrow m\tilde{a}x \\ g_i(x) \tilde{\leq} 0, i = 1, \dots, m. \end{cases} \quad (3.13)$$

Les symboles $m\tilde{a}x$ et $\tilde{\leq}$ représentent les modèles flexibles ou flous des symboles classiques de maximiser et \leq respectivement.

Toute exigence par rapport à la fonction objectif f et aux contraintes $g_i, i = 1..m$ peut se représenter par des fonctions d'appartenances $\mu_i(g_i(x)), i = 0..m$, on a par convention $\mu_0(f(x))$ qui est $\mu_0(g_0(x))$.

Afin de déterminer le degré de satisfaction, on doit déterminer les fonctions d'appartenances $\mu_i, i = 0..m$, de la forme suivante :

$$\mu_i(g_i(x)) = \begin{cases} 1, & g_i(x) \leq g_i^1 \\ d_i(x), & g_i^1 \leq g_i(x) \leq g_i^0, i = 0..m \\ 0, & g_i^0 \leq g_i(x). \end{cases} \quad (3.14)$$

En se basant sur les principes de Bellman, Zadeh et sakawa définissant le degré total de satisfaction eu égard aux fonctions d'appartenances, cela s'exprime par :

$$\mu_D(x) = \min_{i=0..m}(\mu_i(g_i(x))).$$

La solution optimale x^* réalisant le meilleur degré satisfaction est donnée :

$$\max(\min_{i=0..m}(\mu_i(g_i(x)))).$$

Ce dernier problème peut se reformuler comme suit :

$$\begin{cases} \lambda \rightarrow \max \\ \lambda \leq \mu_i(g_i(x)), i = 0..m \\ 0 \leq \lambda \leq 1. \end{cases} \quad (3.15)$$

Ce type de problèmes fera appel aux principes de l'optimisation globale.

⊗ d) Les problèmes de Ratios [78, 86, 16, 68, 101, 57]

Ce sont des problèmes assez fréquents en économie, il s'agit de maximiser une somme de plusieurs rations.

$$\begin{cases} \sum_{i=1}^m \left(\frac{a_i + \sum_{j=1}^n a_{ij}x_j}{b_i + \sum_{j=1}^n b_{ij}x_j} \right) \rightarrow \max \\ x \in \{0, 1\}^n, \text{ ou, } (x \in \mathbb{N}^n), \text{ ou, } (x \in \mathbb{R}_+^{*n}). \end{cases} \quad (3.16)$$

Dans ce cadre, on voit la difficulté quant à la résolution de problèmes de la sorte.

⊗ e) Le contrôle optimale [98, 86, 16, 68]

Par exemple, soit à résoudre ce type de problème en commande optimale :

$$\begin{cases} J(u) = (1/2)x(T_1)'Qx(T_1) + c'x(T_1) \rightarrow \min \\ \frac{dx(t)}{dt} = Ax(t) + bu(t), x(0) = x_0 \\ H(x(T_1)) = g \\ d_* \leq u(t) \leq d^*; t \in [0, T_1]. \end{cases}$$

Donc, c'est à la fois un problème non convexe et dynamique, impliquant beaucoup de difficultés dans la résolution.

►►► Méthodes de résolution numérique

Les problèmes de contrôle optimal que ce soit en économie ou autres ne sont pas toujours solvables analytiquement avec aisance. Donc, il est opportun d'user de méthodes adéquates numériques ou autres. On peut trouver existantes différentes méthodes pour les problèmes de commande optimale, chacune avec ses avantages et ses inconvénients. Le choix de la méthode dépend du problème. On distingue deux types de méthodes en contrôle optimal, les méthodes directes et les méthodes indirectes (après transformation du problème éventuellement).

►►►►► Méthodes indirectes

Pour des informations et commentaires sur ces méthodes, on peut consulter l'article d'Emmanuel Trélat, ([98] contrôle optimal : théorie et applications).

Basées sur le principe du maximum de Pontryagin ([98, 91], [92]) au quel cas on dispose d'une condition nécessaire d'optimalité, il faut vérifier à posteriori l'optimalité de la trajectoire calculée. Ces méthodes sont précises mais très sensibles aux conditions initiales, nécessitant une étude théorique et par exemple la structure de commutation doit être connue à l'avance. Elles sont efficaces dans toute dimension. Il existe des versions probabilistes pour ces méthodes. Pour les résolutions ces méthodes utilisent, entre autres des outils géométriques et algébriques.

►►►►► Méthodes directes

L'optimisation globale prend une part importante dans ce sens (Linéarisation par exemple).

Pour les méthodes directes, on exhibe (par exemple) la méthode de résolution par la programmation linéaire, qui est la méthode adaptée appelée aussi méthode du support /ou simplex ([8, 72, 87, 23, 48]). Ces méthodes permettant d'avoir des solutions approchées très précises ou une solution exacte. Une autre méthode directe est la méthode de discrétisation du problème initial. Pour un problème de départ, on fait une discrétisation de la commande sur l'espace du temps par exemple. On obtient un problème de programmation linéaire par exemple ou autres qui peut être pris en charge par les méthodes de résolution à grandes dimensions (méthode Adaptée) ou [34]. L'inconvénient de cette version est l'obtention d'une solution approchée. La mise en oeuvre des méthodes directes est simple, car ne nécessitant pas une étude théorique préalable, on n'étudie pas les variables adjointes ou on ne connaît pas à l'avance la structure des commutations.

3.2 Exemple

Soit le problème suivant ([98]) :

$$\begin{cases} \dot{x}(t) = y(t), & x(0) = 0, \\ \dot{y}(t) = u(t), & y(0) = 0. \end{cases} \quad (3.17)$$

Soit le contrôle $u(t)$, $t \in I = [0, T]$ vérifiant $|u(t)| \leq 1$. Soit à résoudre le problème en temps minimal pour atteindre le point final $(0, -1)$, en partant de l'origine.

Exhibons la solution : Avec les deux méthodes directe et indirecte.

Soit l'Hamiltonien du système (3.17) est :

$$H(x(t), y(t), p_x(t), p_y(t), u) = p_x(t)y(t) + p_y(t)u + p^0.$$

Où p_x et p_y sont les composantes du vecteur adjoint. Elles sont solutions du système

$$\begin{cases} \dot{p}_x(t) = \frac{-\partial H}{\partial x} = 0, \\ \dot{p}_y(t) = \frac{-\partial H}{\partial y} = -p_x(t). \end{cases}$$

Et donc ce système sera équivalent à :

$$\begin{cases} p_x(t) = Cste = \gamma, \\ p_y(t) = -\gamma t + \mu. \end{cases}$$

Soit l'expression de l'hamiltonien :

$$H = p_x(t)y(t) + p_y(t)u + p^0.$$

Alors quelque soit la valeur de p^0 ,

$$\frac{\partial H}{\partial u} = \frac{\partial}{\partial u}(p_y(t)u + p_x(t)y(t)),$$

$$u \text{ extrémal} \Leftrightarrow u = \text{sign}(p_y(t)).$$

Selon Emmanuel Trélat, ([98]).

De là, le contrôle extrémal sera le suivant :

$$u(t) = \begin{cases} -1 & \text{si } p_y(t) < 0, \\ +1 & \text{si } p_y(t) > 0, \quad t \in I. \end{cases}$$

On a exécuté une méthode indirecte, les résultats sont tracés dans la figure (3.1).

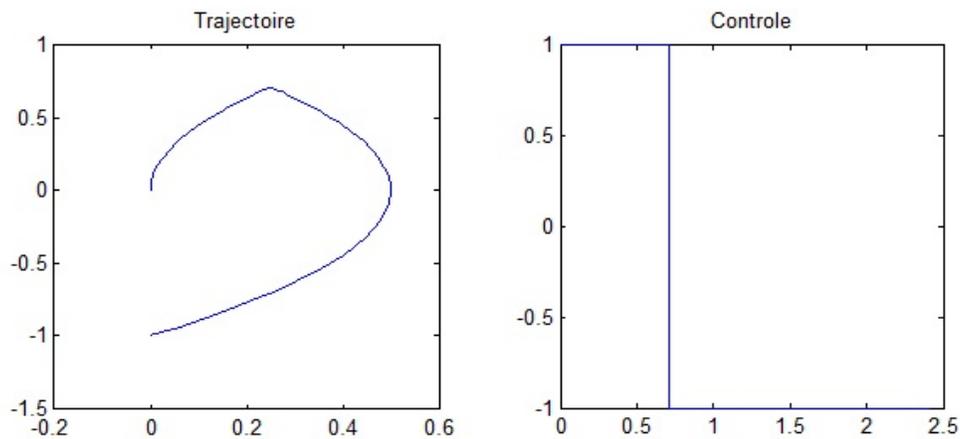


FIGURE 3.1 – Trajectoires et commandes optimales avec la méthode indirecte.

Dans le cas d'une méthode directe, on discrétise la commande $u(\cdot)$ et l'état $x(\cdot)$ par utilisation de la subdivision :

$$0 = t_0 < t_1 < t_2 < \dots < t_N = T.$$

Le problème (3.17) deviendra le suivant :

$$T \rightarrow \min$$

Sous les contraintes :

$$x[i + 1] = x[i] + \frac{T}{N}y[i],$$

$$y[i + 1] = y[i] + \frac{T}{N}u[i]$$

Les résultats de ce problème sont tracés dans la figure (3.2).

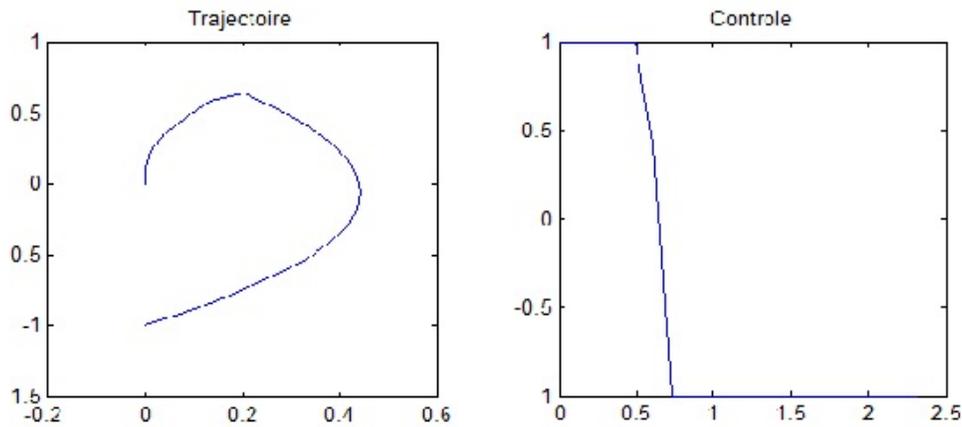


FIGURE 3.2 – Trajectoires et commandes optimales avec la méthode directe.

Les contrôles extrémaux auront en plus une commutation. Soit t_1 , ce temps de commutation. La trajectoire obtenue pour $u(t) = 1$ sur $[0, t_1]$ et pour $u(t) = -1$ sur $[t_1, T]$ est :

- ✦ Si $t \in [0, t_1]$, on aura $y(t) = t$ et $x(t) = \frac{1}{2}t^2$.
- ✦ Si $t \in [t_1, T]$, on obtient $y(t) = -t + 2t_1$ et $x(t) = -\frac{1}{2}t^2 + 2t_1t - t_1^2$.

Les trajectoires obtenues avec $u = +1$ puis $u = -1$ sont montrées dans la figure (3.3).

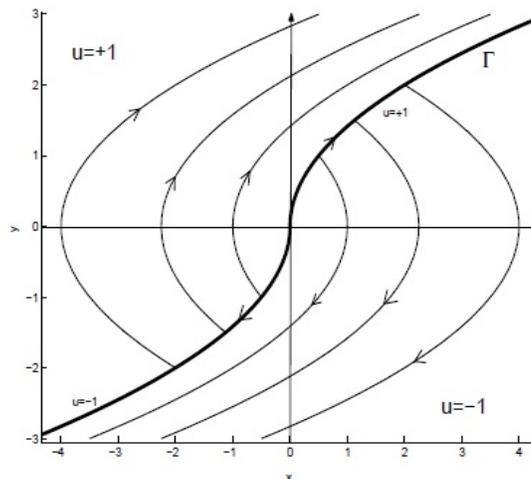


FIGURE 3.3 – Trajectoires optimales.

3.3 Méthodes de résolution en optimisation globale [68, 101, 95, 57]

Remarque 3.3.1. Il existe par exemple des méthodes en optimisation globale pour des cas particuliers.

C'est ainsi, pour des cas de généralisation de la notion de la convexité (pseudo convexe/quasi convexe /etc.) voir [108, 69]

Pour les techniques d'optimisation globale, nous recensons au moins les méthodes suivantes.

A . Les méthodes déterministes (exactes)

♣ a1) La programmation anti-convexe en général [68, 101, 95, 57, 53, 56]

On définit un problème d'optimisation globale anti-convexe par :

$$\begin{cases} f(x) \rightarrow \min \\ x \in D \\ g(x) \geq 0. \end{cases} \quad (3.18)$$

f, g des fonctions convexes finies de \mathbb{R}^n et D un ensemble convexe fermé dans \mathbb{R}^n .

$$G = \{x \in \mathbb{R}^n, g(x) \geq 0\}.$$

Ou bien encore

$$\begin{cases} f(x) \rightarrow \min \\ x \in D \cap G. \end{cases} \quad (3.19)$$

On a bâti toute une théorie autour du type de problème ci-dessus, au point qu'il existe une interaction entre ce type de problème et la programmation (D.C) (différences convexes).

Théorème 3.3.1. (Pour le problème d'optimisation globale anti-convexe)

Toute solution optimale se trouve dans $D \cap \partial G$. Avec ∂G désignant la frontière de G .

On définit un problème d'optimisation globale D.C par :

$$\begin{cases} f(x) \rightarrow \min \\ x \in D \\ g_i(x) \leq 0. \end{cases} \quad (3.20)$$

$f, g_i, i = 1..m$ des fonctions (D.C) (différences convexes) de \mathbb{R}^n et D un ensemble convexe dans \mathbb{R}^n .

On définit un problème d'optimisation globale D.C canonique par :

$$\begin{cases} c'x \rightarrow \min \\ h(x) \leq 0 \\ g(x) \geq 0. \end{cases} \quad (3.21)$$

h, g des fonctions convexes de \mathbb{R}^n . $c \in \mathbb{R}^n$.

En particulier, on définit un problème linéaire d'optimisation globale anti-convexe par :

$$(P) \begin{cases} c'x \rightarrow \min \\ Ax \leq b, x \geq 0 \\ g(x) \geq 0. \end{cases} \quad (3.22)$$

A matrice $m \times n$; $c, x \in \mathbb{R}^n$; $b \in \mathbb{R}^m$ et g convexe.

Dans ce cadre plusieurs variantes de méthodes sont proposées, entre autres :

- * /Annexation Polyédrale. [68, 101, 95, 57, 53, 56]
- * /Approximation Externe. [102, 68, 101, 95, 57, 53, 56]
- * /Techniques des plans coupants. [82, 102, 68, 101, 95, 57, 53, 56]

♣ a2) La méthode de Branch and Bound/Branch and cut/Branch and Prune.

[82, 102, 68, 101, 95, 57, 53, 56]

Pour ce type de méthodes, le principe est similaire comme pour le cas à une seule dimension.

Par exemple, la résolution pour les fonctions de Hölder [47, 18] comme suit :

Soit f une fonction réelle continue pour le problème d'optimisation globale (P) suivant :

$$(P) \begin{cases} f(x) \rightarrow \min \\ x \in D. \end{cases} \quad (3.23)$$

D est un hyperrectangle non vide de \mathbb{R}^n .

$D = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3] \times \dots \times [a_n, b_n] \subseteq \mathbb{R}^n$.

f continue vérifiant la propriété de Hölder

(i.e.)

$$\forall x \in D, y \in D, |f(x) - f(y)| \leq L \|x - y\|^\alpha.$$

L : constante strictement positive (constante Lipschitzienne), $\alpha \geq 1$.

$\|\cdot\|$ La norme euclidienne.

Remarque 3.3.2. L'algorithme pour la résolution de (P) se base sur la construction non pas de fonctions bornes inférieures mais de points bornes puis combiné à la méthode Branch and Bound.

Algorithme HOL^n dans \mathbb{R}^n [47, 18]

1. $x \in D$ tel que $x = (x_1, x_2, x_3, \dots, x_n)'$.

$$D_j = [a_1^j, b_1^j] \times [a_2^j, b_2^j] \times \dots \times [a_n^j, b_n^j] \subseteq \mathbb{R}^n$$

$$j = 0, 1, 2 \dots k, k \in \mathbb{N}$$

2. $j = k$, pour $D_k, \epsilon > 0, L > 0$.

3. Évaluations : $x^k = (c_1^k, c_2^k, c_3^k, \dots, c_n^k)', LB_k, UB_k$.

$$\left\{ \begin{array}{l} x_1^k \in [a_1^k, b_1^k] \\ x_2^k \in [a_2^k, b_2^k] \\ \dots \\ \dots \\ \dots \\ x_n^k \in [a_n^k, b_n^k] \end{array} \right. \implies \left\{ \begin{array}{l} c_1^k = \frac{a_1^k + b_1^k}{2} \\ c_2^k = \frac{a_2^k + b_2^k}{2} \\ \dots \\ \dots \\ \dots \\ c_n^k = \frac{a_n^k + b_n^k}{2} \end{array} \right. \quad (3.24)$$

$$x^k = (c_1^k, c_2^k, c_3^k, \dots, c_n^k)', UB_k = f(x^k)$$

$$LB_k = f(x^k) - L \left(\frac{\|B-A\|}{2} \right)^{\frac{1}{\alpha}}, B = (b_1^k, b_2^k, b_3^k, \dots, b_n^k)', A = (a_1^k, a_2^k, a_3^k, \dots, a_n^k)'$$

4. Si $UB_k - LB_k \leq \epsilon$, Stop on a la solution pour (P).

$$\left\{ \begin{array}{l} x^k = \bar{x} \\ UB_k = f(\bar{x}) = \min(f(x)) \end{array} \right. \quad (3.25)$$

5. Sinon subdiviser D_k en sous intervalles, soient : D_{k1} et D_{k2} .

$$\bigcup_{i=1}^2 D_{ki} = D_k, D_{k1} \cap D_{k2} = \emptyset$$

6. Construire les problèmes des bornes inférieures et supérieures sur D_{k1} et $D_{k2}, i = 1..2$.

7. Soit

$$\left\{ \begin{array}{l} UB_{k+1} = \min(UB_{k1}, UB_{k2}, UB_k) \\ LB_{k+1} = \min(LB_{k1}, LB_{k2}) = LB_{k*} \end{array} \right.$$

(3.26)

8. $D_k = \{D_{k1}, D_{k2}\}$.

9. Éliminer tous les sous ensembles $D_{ki}, i = 1, 2$ tel que

$$LB_{ki} > UB_{k+1}.$$

$$D_{k+1} = D_{k*}$$

10 . $k = k + 1$ aller à 4.

Théorème 3.3.2. (Gourdin et al pour la minimisation) [47, 18]

L'algorithme HOL^n est fini ou bien converge asymptotiquement vers l'optimum.

♣ a3) La méthode d'approximation externe [102, 68, 53, 56]

Soit le problème

$$\begin{cases} f(x) \rightarrow \min \\ h_i(x) \leq 0, i = 1..m \\ g(x) \geq 0. \end{cases} \quad (3.27)$$

$x \in \mathbb{R}^n, H = \{x \in \mathbb{R}^n, h(x) \leq 0\}, f, g, h_i$ des fonctions convexes de $\mathbb{R}^n \rightarrow \mathbb{R}$.

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

$$G = \{x \in \mathbb{R}^n, g(x) \geq 0\}.$$

Remarque 3.3.3. Pour cette méthode :

- Il existe au moins 02 variantes d'algorithmes.
- La convergence a été établie.
- Soit w solution optimale du problème sans la contrainte anti-convexe.

Algorithme de la méthode

Soit $\bar{x}^1 \in H \cap \partial G$, posons $\alpha_1 = f(\bar{x}^1)$, ensuite on génère un polytope S_1 contenant le compact convexe $\{x \in H, f(x) \leq \alpha_1\}, k = 1$.

Itérations $k = 1, 2, 3, \dots$

Résoudre le sous problème (Q_k) .

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

Soit z^k , la solution optimale pour (Q_k)

Si $g(z^k) = 0$ alors stop (solution optimale).

Sinon on cherche x^k avec le segment $[w, z^k]$ qui intersecte la surface $\max\{f(x) - \alpha_k, g(x)\} = 0$.

- (a) Si $x^k \in H, (h(x^k) \leq 0)$ alors $p^k \in \partial f(x^k)$

$$l_k(x) = \langle p^k, (x - x^k) \rangle .$$

- (b) Si $x^k \notin H, (h(x^k) > 0)$ alors $p^k \in \partial h(x^k)$

$$l_k(x) = \langle p^k, (x - x^k) \rangle + h(x^k)$$

$$S_{k+1} = S_k \cap \{x \in \mathbb{R}^n, l_k(x) \leq 0\}$$

$$\bar{x}^{k+1} = x^k, \text{ si } x^k \in H, g(x^k) = 0$$

$$\bar{x}^{k+1} = \bar{x}^k, \text{ sinon}$$

$\alpha_{k+1} = f(\bar{x}^{k+1})$ aller à l'itération $k + 1$.

♣ a4) La méthode des Plans coupants [82, 102, 57, 53, 56].

Soit le problème

$$\begin{cases} c'x \rightarrow \min \\ Ax \leq b \\ g(x) \geq 0, \\ x \geq 0. \end{cases} \quad (3.28)$$

$c, x \in \mathbb{R}^n, D = \{x \in \mathbb{R}^n, Ax \leq b, x \geq 0\}, g$ fonction convexe.

A matrice $m \times n, G = \{x \in \mathbb{R}^n, g(x) \geq 0\}, D \neq \phi.$

$G \neq \phi, G^c \neq \phi, D \cap G$ ensemble des solutions réalisables.

Le schéma de la méthode est :

Soit x^0 la solution optimale non dégénérée pour le problème

$$\begin{cases} c'x \rightarrow \min \\ x \in D. \end{cases} \quad (3.29)$$

Soit $N(x^0)$, l'ensemble des voisins de x^0 .

$$N(x^0) = \{x^1, x^2, \dots, x^n\}.$$

En choisissant $x^j \in N(x^0)$ et calculons la direction $z^j = x^j - x^0$.

Algorithme de la méthode

Supposons que les sommets voisins de x^0 sont admissibles pour le problème initial.

Dans ce contexte les points à rechercher y^j sous hypothèses que : $g(x^0) < 0$ et $g(x^j) \geq 0, x^j \in N(x^0)$.

Cela se traduit par le problème mathématique unidimensionnel :

$$\begin{aligned} & t \rightarrow \min \\ & g(x^0 + t(x^j - x^0)) = 0 \\ & t \geq 0 \\ & y^j = x^0 + t(x^j - x^0) \\ & Ay^j \leq b \\ & g(y^j) = g(x^0 + t(x^j - x^0)) = 0 \\ & c'y^j = c'(x^0 + t(x^j - x^0)) \\ & \text{Conduisant} :: \grave{a} \\ & \implies \\ & c'y^j \leq c'(x^j). \end{aligned}$$

On note x^* la solution optimale actuelle, soit $x^k \in N(x^0)$ avec $c'x^k < c'x^*$ alors on applique de nouveau le problème mathématique unidimensionnel avec x^k afin de mettre à jour x^* , on continue le processus jusqu'à épuisement de $N(x^0)$.

Remarque 3.3.4. Pour cette méthode :

Il existe au moins 03 variantes d'algorithmes.

♣ a5) La méthode DC (Difference Convex) et DCA (Difference Convex Algorithm) [104, 79, 85].

En résumé, la méthode D.C est une méthode inventée par Pham Dinh Tao et Le Thi Hoai an. Cette méthode est souvent combinée à la méthode de Branch and Bound et Variantes, dans le cadre de la résolution en optimisation globale.

Objet de la programmation D.C

La programmation D.C est l'étude et la résolution de programmes mathématiques, avec comme hypothèse celle que la fonction objectif étant la différence de 02 fonctions convexes, et donc sans perdre de généralité, un programme D.C est présenté comme suit :

$$\begin{cases} g(x) - h(x) \rightarrow \min \\ x \in \mathbb{R}^n. \end{cases}$$

(3.30)

g, h sont 02 fonctions convexes.

Domaines d'applications de la programmation D.C

- */ Résolution de problèmes non linéaires, quadratiques, linéaires et autres.
- */ La programmation linéaire en variables mixtes (télécommunication, logistique et autres).
- */ Le problème du Bin Packing.
- */ Combinée avec les algorithmes génétiques pour les problèmes de fouilles de données.
- */ Traitements d'images par ordinateurs.
- */ Combinée avec les méthodes Méta heuristiques pour problèmes divers.

Philosophie de la méthode D.C

Soit une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$

* f est dite semi continue inférieurement en x_0 si pour tout $\epsilon > 0$, il existe un voisinage V tel que :

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

* f est dite propre si elle ne prend pas la valeur $-\infty$ et elle n'est pas partout égale à $+\infty$.

Conjugué de Fenchel-Legendre

On définit la conjuguée f^* de f par :

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

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

* f^* peut nécessiter la résolution d'un problème d'optimisation.

Le sous Gradient

Soit f une fonction convexe et propre sur \mathbb{R}^n , y^0 est sous Gradient de f au point x^0 tel que : $f(x^0) \neq +\infty$, si :

$$\langle y^0, x - x^0 \rangle + f(x^0) \leq f(x), \forall x \in \mathbb{R}^n.$$

On note par $\partial f(x^0)$ l'ensemble des sous Gradients de f au point x^0 , appelé sous - différentiel.

Conséquences

Soit Γ_0 l'ensemble des fonctions convexes, semi-continues inférieurement et propres sur \mathbb{R}^n .

$$f \in \Gamma_0 \iff f^* \in \Gamma_0, f = f^{**}.$$

$$y \in \partial f(x) \iff f(x) + f^*(y) = \langle x, y \rangle \text{ et } y \in \partial f(x) \iff x \in \partial f^*(y).$$

Dualité en optimisation D.C

Soit le problème primal (P)

$$\alpha = \inf\{g(x) - h(x), x \in \mathbb{R}^n\}.$$

$$= \inf\{g(x) - \sup\{\langle x, y \rangle - h^*(y) : y \in \mathbb{R}^n\} : x \in \mathbb{R}^n\}.$$

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

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

$\alpha = \inf\{h^*(y) - g^*(y), y \in \mathbb{R}^n\}$ ce dernier problème exprime le problème (D) dual du primal (P).

Condition nécessaire d'optimalité local

Si x^* est minimum local de $g - h$ alors

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

Condition suffisante d'optimalité local

Si x^* admet un voisinage V tel que

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

Alors x^* est un minimum local de $g - h$.

La théorie de la dualité en optimisation D.C permet le transport des minimas locaux prouvant ainsi que l'algorithme pour l'optimisation D.C (D.C.A) est primal dual.

Algorithme D.C.A

1. Soit x^0 une solution initiale, $k = 0$ et $\epsilon > 0$.
2. x^k étant connue, calculer $y^k \in \partial h(x^k)$.
3. Calculer $x^{k+1} \in \partial g^*(y^k)$.
4. $\|x^{k+1} - x^k\| < \epsilon$ Stop, $k = k + 1$ aller à 2.

Remarque 3.3.5. L'algorithme D.C.A :

Construit 02 suites décroissantes $(g - h)(x^k)$, $(h^* - g^*)(y^k)$ avec $\{x^k\}$, $\{y^k\}$.

D.C.A sans calcul de pas.

D.C.A est primal dual.

le point initial réalisable ou non réalisable.

Théorème 3.3.3. Pour l'algorithme D.C.A (Convergence finie).

Pour les suites $(g - h)(x^k)$, $(h^* - g^*)(y^k)$ sont décroissantes.

Lorsque $(g - h)(x^{k+1}) = (g - h)(x^k)$ l'algorithme s'arrête à la $(k + 1)^{ième}$ itération et le point x^k (resp y^k) est un point critique de $(g - h)$ (resp $(h^* - g^*)$).

Exemple 3.3.1. D.C et D.C.A

Soit f la fonction tel que $f(x) = \frac{1}{2}x'Qx + b'x \rightarrow \min$

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

D'après la D.C :

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

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

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

Algorithme D.C.A

1. Soit x^0 une solution initiale, $k = 0$ et $\epsilon > 0$.
2. x^k étant connue, calculer $y^k \in \partial h(x^k) = (\rho I - Q)x^k$.
3. Calculer $x^{k+1} \in \operatorname{argmin}\{\frac{\rho}{2}\|x\|^2 - x'(y^k - b)\}$.
4. $\|x^{k+1} - x^k\| < \epsilon$ Stop, $k = k + 1$ aller à 2.

♣ a6) La méthode d'analyse des Intervalles [78].

♣ a7) La méthode d'analyse Affine [86].

♣ a8) La méthode d'annexion Polyédrale [102].

♣ a9) La méthode Homotopique.

♣ a10) La méthode Dynamic Programming.

♣ a11) La méthode Branch and Bound (Mixed Integer Programming) ([16],[71]).

Parmi les méthodes les plus récentes et à la fois efficaces, on peut recenser dans ce sens et entre autres :

*/ Les Techniques de linéarisation pour les problèmes d'optimisation globale.

*/ Les Techniques de convexification pour les problèmes d'optimisation globale.

*/ Les Techniques de reformulation de problèmes en Semi Défini Positif [9].

Beaucoup de méthodes liées à ces techniques s'apparentent de la méthode de Branch and Bound et variantes.

Les chercheurs qui sont à la pointe, on peut citer Alain Billionnet, son équipe et Elloumi coté français [16] et feu Floudas [45] coté américain, mais cela ne prouve pas que les autres sont à la traîne [101, 95, 57]. Les Logiciels ont aussi suivi l'évolution : Cplex, Mapple, Mathematica, Lingo, Matlab,...etc.

- ♣ a12) Branch and Bound (Constraint Satisfaction Techniques)/C.S.P.
- ♣ a13) Methods based on real algebraic geometry.
- ♣ a14) Méthodes dites Barrières. [34].
- ♣ a15) Méthode dite Aliénor [25, 26].
- ♣ a16) New Quadratic lower Bound in Global Optimization [89].

La liste des méthodes déterministes (exactes) ci-dessus est tronquée, si bien qu'il existe encore d'autres méthodes efficaces de résolution, donc la liste est ouverte.

B . Les méthodes non déterministes

- ♣ b1) A Base recuit simulé.
- ♣ b2) A Base Monté Carlo.
- ♣ b3) A Base stochastique.

La liste des méthodes non déterministes ci-dessus est tronquée, si bien qu'il existe encore d'autres méthodes efficaces de résolution, donc la liste est ouverte.

3.4 Notre première (1°) contribution pour l'optimisation globale

Consulter [24]

Remarque 3.4.1. Pour notre Première (1°) contribution.

On a préféré présenter presque le contenu de ce qu'il a été soumis et publié.

Simulation of Efficient combination of two lower bound functions in univariate global optimization.

AIP Conference Proceedings 1863, 250004 (2017) ;

https://doi.org/10.1063/1.4992412 : : Published Online : 21 July 2017

CHEBBAH¹Mohammed Ouanes¹mohand Zidna²ahmed

chbbhea@yahoo.fr¹ ouanes_mohand@yahoo.fr¹ ahmed.zidna@univ – lorraine.fr²

LITA UFR Metz, Univ Paul Verlaine - Metz, Ile du Saulcy, 57045 Metz, France²

Laboratoire LAROMAD University Tizi ouzou Algeria DZ¹.

Considérons le problème suivant

$$(P) \begin{cases} \min f(x) \\ x \in [x^0, x^1] \subset \mathbb{R}. \end{cases}$$

Avec $f(x)$ (continue) non-convexe C^2 sur $[x^0, x^1]$ de \mathbb{R} .

L'optimisation globale de problèmes unidimensionnels attire l'attention de plusieurs chercheurs non seulement car cela survient dans plus d'une application en réalité mais encore les méthodes pour ces problèmes sont utiles pour des extensions aux cas multidimensionnels ou à la réduction du cas multidimensionnel vers le cas unidimensionnel. Une classe d'approches déterministes, utilisant des méthodes avec bornes inférieures, s'avère une stratégie efficace pour trouver le minimum global. Ces méthodes permettent de se concentrer sur les régions prometteuses susceptibles de contenir le minimum global.

Afin de résoudre un problème d'optimisation globale, plusieurs enveloppes de méthodes y sont proposées (voir [42] et références). Plusieurs méthodes ont été étudiées en littérature pour des problèmes d'optimisation globale en dimension 1. Parmi ces dernières, nous pouvons citer la méthode classique α BB développée dans [11], d'autres méthodes utilisant une borne inférieure quadratique sont développées dans [51] pour le cas en dimension 1. Ces dernières sont généralisées dans [88]. Dans [45], une bonne borne inférieure convexe pour cas dimension 1 continue de classe C^2 est proposée en utilisant des bornes inférieures quadratiques continues par morceaux obtenues à travers la méthode α BB.

Dans [94], une variante à base branch and prune est proposée.

Les principales contributions de ce travail sont :

1. Amélioration de la borne inférieure quadratique donnée en [51].
2. La combinaison de cette borne inférieure quadratique améliorée avec la fonction borne inférieure de la méthode α BB (voir [11]). Cette combinaison résultante est non différentiable mais donnant une meilleure fonction borne inférieure. Pour calculer ces minimums, nous résolvons un programme convexe.
3. Un test Convexe/concave et d'étapes pruning pour accélérer la convergence de l'algorithme proposé.

La structure du travail est comme suit. Les deux fonctions bornes dans [11] et dans [51], leurs propriétés et la fonction borne inférieure quadratique améliorée sont présentées. Puis la nouvelle fonction borne, propriétés, l'algorithme de résolution, sa convergence et enfin la présentation de résultats numériques.

3.5 Background

3.5.1 Programmation factorable, amélioration (relaxation)

Définition 3.5.1. Les fonctions factorables sont des fonctions qui sont des sommes et produits récursifs de fonctions univariées. Cette classe de fonctions suffit pour décrire la plupart des domaines d'intérêt d'application surtout en optimisation non linéaire non convexe.

Exemple 3.5.1. Soient des fonctions factorables

$$f(x) = \sum_{i=1}^n \log(x_i), f(x, y) = xy, f(x, y) = \frac{x}{y}, f(x, y, z, w) = \sqrt{\exp(xy + z \log w) z^3}.$$

Remarque 3.5.1. McCormick [75], [74] a mis au point des techniques de délimitation pour programmes non linéaires. Ces techniques jouent actuellement un rôle central dans de nombreuses implémentations liées aux problèmes non linéaires (PNL). Le principe est qu'un facteur (PNL) peut être converti en une (PNL) séparable équivalente récursive. Puis par exemple, introduction de nouvelles variables et contraintes. La (PNL) séparable peut être traitée par des sous-estimateurs et des sur-estimateurs appropriés des fonctions univariées impliquées.

Exemple 3.5.2. Soit la fonction

$$\begin{aligned} f(x, y, z, w) &= \sqrt{\exp(xy + z \log w) z^3} \\ x_1 &= xy, x_5 = \exp(x_4) \\ x_2 &= \log(w), x_6 = z^3 \\ x_3 &= zx_2, x_7 = x_5 x_6 \\ x_4 &= x_1 + x_3, f = \sqrt{x_7}. \end{aligned}$$

Une manière de convexifier le produit $t = x_j x_k$ selon (McCormick's envelopes, [75],[74]) sur une région $[x^L, x^U]$ est :

$$\begin{aligned} t &\geq x_j^L x_k + x_k^L x_j - x_j^L x_k^L \\ t &\geq x_j^U x_k + x_k^U x_j - x_j^U x_k^U \\ t &\leq x_j^L x_k + x_k^U x_j - x_j^L x_k^U \\ t &\leq x_j^U x_k + x_k^L x_j - x_j^U x_k^L. \end{aligned}$$

Remarque 3.5.2. Il existe plusieurs variantes de l'approche factorable. Ces techniques introduites dans le but de conversions à l'instar du dual pour le primal.

La factorisation et la séparation (en guise de nouvelles contributions) pour prendre le maximum de domaines d'intérêt d'application en optimisation non linéaire non convexe par exemple pour notre cas.

Exemple 3.5.3. Soit le problème d'optimisation dans \mathbb{R}^n suivant en forme séparable (factorable)

$$\begin{aligned} f(x) &= \sum_{i=1}^n x_i \log(x_i) \rightarrow \min \\ a_i &\leq x_i \leq b_i, a_i > 0. \end{aligned}$$

Exemple 3.5.4. Soit la classe dans \mathbb{R}^n de problèmes d'optimisation suivante en forme séparable (factorable).

$$f(x) = \sum_{i=1}^n f_i(x_i) \rightarrow \min$$

$$a_i \leq x_i \leq b_i, a_i > 0.$$

Prise en charge facilement par notre algorithme dans \mathbb{R} ci-dessous.
(Dans d'autres cas cet algorithme, étant étape intermédiaire).

3.5.2 La fonction borne inférieure dans la méthode αBB [11]

La fonction borne inférieure dans la méthode αBB sur l'intervalle $[x^0, x^1]$ est donnée par :

$$LB_{\alpha}(x) = f(x) - \frac{K_{\alpha}}{2}(x - x^0)(x^1 - x)$$

avec $K_{\alpha} \geq \max\{0, -f''(x)\}, \forall x \in [x^0, x^1]$. Les principales propriétés de cette fonction borne inférieure sont :

1. Elle est convexe (i.e. $LB_{\alpha}''(x) = f''(x) + K_{\alpha} \geq 0, \forall x \in [x^0, x^1]$).
2. Elle coïncide avec la fonction $f(x)$ aux extrémités de l'intervalle $[x^0, x^1]$ (i.e. par construction de $(LB_{\alpha}(x))$).
3. Elle est une fonction borne inférieure (i.e. $f(x) - LB_{\alpha}(x) = \frac{K_{\alpha}}{2}(x - x^0)(x^1 - x) \geq 0, \forall x \in [x^0, x^1]$).

Pour plus de détails sur cela voir [11].

3.5.3 La fonction quadratique borne inférieure [51]

La fonction Quadratique borne inférieure développée dans [51] sur l'intervalle $[x^0, x^1]$ est donnée par :

$$LB_{LO}(x) = f(x^0) \frac{x^1 - x}{x^1 - x^0} + f(x^1) \frac{x - x^0}{x^1 - x^0} - \frac{K}{2}(x - x^0)(x^1 - x)$$

avec $K \geq |f''(x)|, \forall x \in [x_0, x_1]$. Les principales propriétés de cette fonction borne inférieure sont :

1. Elle est convexe (i.e. $LB_{LO}''(x) = K \geq 0$).
2. Elle coïncide avec la fonction $f(x)$ aux extrémités de l'intervalle $[x^0, x^1]$ (i.e. par construction de $LB_{LO}(x)$).
3. Elle est une fonction borne inférieure (i.e. $(f(x) - LB_{LO}(x))'' = f''(x) - K \leq 0, \forall x \in [x^0, x^1]$.) qui implique que $(f(x) - LB_{LO}(x))$ est concave, est minimum aux extrémités de $[x^0, x^1]$ alors $f(x) \geq LB_{LO}(x), \forall x \in [x^0, x^1]$.

3.6 La nouvelle fonction borne inférieure

Nous commençons l'amélioration de la fonction borne inférieure quadratique LB_{LO} en prenant la constante $K_q \geq \max\{0, f''(x)\}, \forall x \in [x_0, x_1]$ au lieu de $K \geq |f''(x)|$, comme dans [51]. Soit la fonction borne inférieure quadratique améliorée

$$LB_q(x) = f(x^0) \frac{x^1 - x}{x^1 - x^0} + f(x^1) \frac{x - x^0}{x^1 - x^0} - \frac{K_q}{2} (x - x^0)(x^1 - x).$$

Proposition 1

$LB_q(x) \geq LB_{LO}(x), \forall x \in [x_0, x_1]$.

Proof Par considération de $LB_q(x) - LB_{LO}(x) = \frac{K - K_q}{2} (x - x^0)(x^1 - x) \geq 0, \forall x \in [x^0, x^1]$ (i.e. $K \geq K_q$) et la proposition est prouvée.

Proposition 2

1. $LB_q(x)$ est convexe sur $[x^0, x^1]$.
2. $LB_q(x)$ coïncide avec la fonction $f(x)$ aux extrémités de l'intervalle $[x^0, x^1]$.
3. $LB_q(x)$ est fonction borne inférieure de $f(x)$ sur l'intervalle $[x^0, x^1]$.

Démonstration. 1. $LB_q''(x) = K_q \geq 0, \forall x \in [x^0, x^1]$ alors $LB_q(x)$ est convexe.

2. Par construction de $LB_q(x)$.

3. Par considération $(f(x) - LB_q(x))'' = f''(x) - K_q = f''(x) - \max\{0, f''(x)\} \leq 0, \forall x \in [x^0, x^1]$ qui implique que $(f(x) - LB_q(x))$ est concave sur $[x^0, x^1]$, est minimum aux extrémités de $[x^0, x^1]$ alors $f(x) \geq LB_q(x), \forall x \in [x^0, x^1]$.

□

Nous présentons maintenant la nouvelle fonction borne inférieure sur l'intervalle $[x^0, x^1]$,

$$LB(x) = \max\{LB_\alpha(x), LB_q(x)\}.$$

Proposition 3

1. $LB(x)$ est convexe sur l'intervalle $[x^0, x^1]$.
2. Elle coïncide avec la fonction objectif aux extrémités de l'intervalle $[x^0, x^1]$.
3. Elle est fonction borne inférieure de $f(x)$ sur $[x^0, x^1]$ et meilleure que les deux bornes inférieures $LB_\alpha(x)$ et $LB_q(x)$ sur $[x^0, x^1]$.

- Démonstration.*
1. Elle est maximum de fonctions convexes sur l'intervalle $[x^0, x^1]$ alors elle est convexe.
 2. Par construction de $LB(x)$.
 3. Par considération de $LB_\alpha(x) \leq f(x), \forall x \in [x^0, x^1]$ et $LB_q(x) \leq f(x), \forall x \in [x^0, x^1]$ alors $LB(x) = \max\{LB_\alpha(x), LB_q(x)\} \leq f(x), \forall x \in [x^0, x^1]$, de plus elle est meilleure que les fonctions bornes inférieures par construction.

□

Cette nouvelle fonction borne inférieure n'est pas une fonction lisse. Afin de calculer ses minimums sur l'intervalle $[x^0, x^1]$, nous résolvons le problème $\min_{x \in [x^0, x^1]} \max\{LB_\alpha(x), LB_q(x)\}$ qui est équivalent au problème convexe suivant :

$$\begin{cases} \min z \\ \text{s.t. } LB_\alpha(x) \leq z, \quad LB_q(x) \leq z \\ x \in [x^0, x^1], z \in \mathbb{R}. \end{cases}$$

3.6.1 Test convexité/concavité

À l'itération k nous calculons $K_q^k \geq \max\{0, f''(x)\}$ et $K_\alpha^k \geq \max\{0, -f''(x)\}$ sur l'intervalle $[a_k, b_k]$.

1. Si $K_\alpha^k = 0$ (i.e. $-f''(x) \leq 0, \forall x \in [a_k, b_k]$) alors $f(x)$ est convexe sur l'intervalle $[a_k, b_k]$, toute locale recherche donne le minimum global sur cet intervalle.
2. Si $K_q^k = 0$ (i.e. $f''(x) \leq 0, \forall x \in [a_k, b_k]$) alors $f(x)$ est concave sur l'intervalle $[a_k, b_k]$ et ses minimums sont indiqués aux extrémités de cet intervalle.

Remarque 1

L'algorithme peut s'arrêter à travers le test convexité/concavité si il est satisfait pour tous les sous-intervalles.

3.6.2 Pruning method

Soit $LB_q^k(x)$ la borne quadratique inférieure sur l'intervalle $[a_k, b_k]$ et UB_k la borne supérieure courante dans l'algorithme du Branch and prune. Nous résolvons, l'équation quadratique $LB_q^k(x) = UB_k$ comme vu dans [50][49]. Nous avons trois cas :

1. On n'a pas de solution alors les entrées de l'intervalle $[a_k, b_k]$ sont inhibées (i.e. $LB_q^k(x) > UB_k$ sur $[a_k, b_k]$ alors cet intervalle ne peut pas contenir le minimum global).
2. C'est une solution double \tilde{x}^k , si $f(\tilde{x}^k) = UB_k = LB_q^k(\tilde{x}^k)$, l'intervalle est réduit en un point (cette solution) nous mettons à jour cet ensemble (points) qui donne la borne supérieure, et cet intervalle $[a_k, b_k]$ est inhibé (i.e. le minimum de $f(x)$ sur $[a_k, b_k]$ est rattaché à \tilde{x}^k) sinon les entrées de l'intervalle $[a_k, b_k]$ sont inhibées (i.e. $f(x) > UB_k$ sur $[a_k, b_k]$).

3. On a deux solutions distinctes $a_k^{r_1}$ et $b_k^{r_1}$ alors l'intervalle $[a_k, b_k]$ est réduit à $[a_k^{r_1}, b_k^{r_1}]$ (i.e. $f(x) \geq LB_q^k(x) > UB_k$ sur les deux intervalles $[a_k, a_k^{r_1}[$ et $]b_k^{r_1}, b_k]$ alors les deux intervalles ne peuvent pas contenir le minimum global). Nous répétons cette procédure jusqu'à $f(a_k^{r_{j_k}}) = f(b_k^{r_{j_k}}) = UB_k$.

Remarque 2

Si la solution optimale est trouvée à l'itération k et la condition d'arrêt $UB_k - LB_k < \varepsilon$ n'est pas satisfaite, la méthode pruning nous permet de confirmer cette solution et de stopper l'algorithme.

Nous présentons deux simples exemples.

Exemple 1

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

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

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

Remarque 3

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

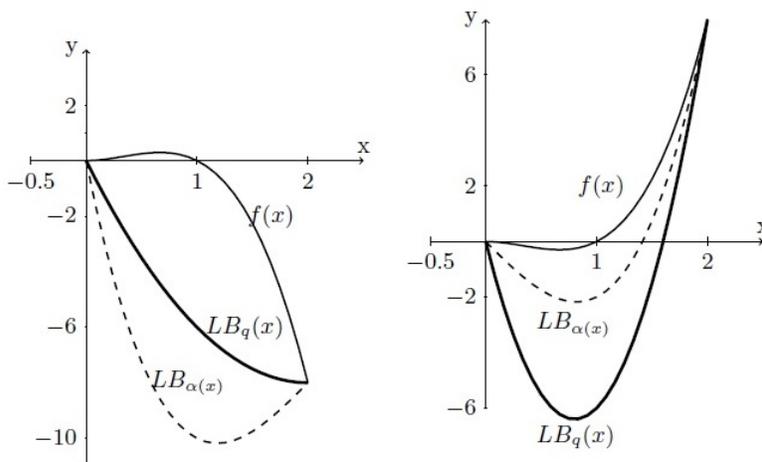


Fig. 1. (left) : $f(x) = -x^3 + x^2$ and its two lower bounds $LB_\alpha(x)$ (dashed) and $LB_q(x)$ (bold). (Right) : $f(x) = x^3 - x^2$ and its two lower bounds $LB_\alpha(x)$ (dashed) and $LB_q(x)$ (bold)

Exemple 2

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

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

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

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

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

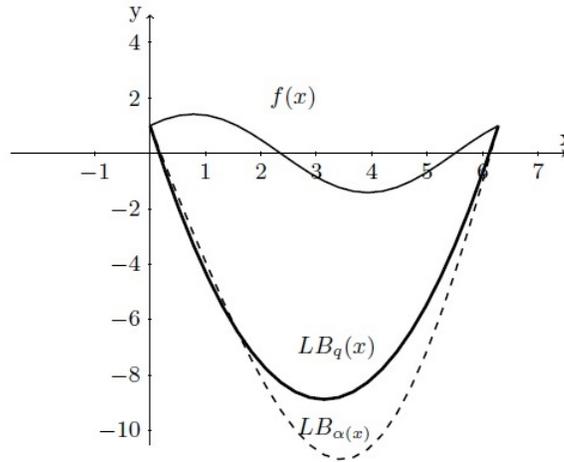


Fig. 2. The function $f(x) = \sin(x) + \cos(x)$ (bold) and its the lower bounds $LB_\alpha(x)$ (dashed) and $LB_q(x)$ (bold)

3.7 L'algorithme de Branch and Bound et sa convergence

Remarque 3.7.1. L'algorithme que nous proposons, est une variante d'algorithmes améliorables et améliorés soumis dans des conférences internationales ou dans des journaux scientifiques mathématiques.

Nous pouvons améliorer la technique Pruning (nouvelle contribution par exemple), ou tout simplement, ajout de procédures d'accélération de la convergence pour l'algorithme (pour performance).

Méthode basée (sur le principe de séparation et évaluation / Branch-and-bound) (BB), est une des méthodes la plus populaire dans le domaine des applications en optimisation globale déterministe. Elle consiste en la subdivision de l'espace des solutions en petites régions où la borne supérieure et inférieure de la fonction objectif sont calculées. Selon ces bornes, chaque région est explorée ou inhibée pour construire l'arbre de Branch and Bound. La solution globale est obtenue une fois que la courante meilleure borne supérieure (UB) est proche de la courante meilleure borne inférieure (LB) compte tenu d'une tolérance spécifiée ε . Nous introduisons l'algorithme pour recherche de la solution globale du problème (P) et nous montrerons sa convergence.

Algorithme Branch and Bound (BB)**Étape 1 : Initialisation**

- a) Soit ε donné un petit nombre et soit $[a_0, b_0]$ l'intervalle initial.
- b) Calculer $K_\alpha^0 = \max\{0, \sup_{x \in [a_0, b_0]}(-f''(x))\}$ et $K_q^0 = \max\{0, \sup_{x \in [a_0, b_0]}f''(x)\}$.
- c) Appliquer Test Convexité/concavité.
- d) Appliquer le pruning test afin de réduire et mise à jour de l'intervalle recherché.
- e) Soit $k := 0, T^0 = [a_0, b_0], M := T^0$.
- f) Calculer $LB_\alpha^0(x)$ et $LB_q^0(x)$ sur T^0 , et résoudre le programme convexe pour obtenir la solution optimale z^0 et $s_0^*(s_0^* = x_0^*)$.

$$\min \{z : LB_\alpha^0(x) \leq z, LB_q^0(x) \leq z, z \in \mathbb{R}, x \in T^0\} \quad (3.31)$$

- g) Soit $UB_0 := \min\{f(a_0), f(b_0), f(s_0^*)\} = f(\bar{s}^0)$, $LB_0 = LB(T^0) := z^0$.
- h) Si $UB_0 - LB_0 \leq \varepsilon$ alors imprimer \bar{s}^0 comme ε -solution optimale; **Sortie** algorithme.
sinon Set $M \leftarrow \{T^0\}$, $k \leftarrow 1$

Étape 2 : Itération**a) Étape Sélection**

- Selection $T^k = [a_k, b_k] \in M$, l'intervalle tel que $LB_k = \min LB(T^k)$

b) Étape Bisection

- Bisection T^k par deux sous-rectangles $T_1^k = [a_k^1, b_k^1], T_2^k = [a_k^2, b_k^2]$ par procédure subdivision via s_k^* .

c) Étape Calculs

- **Pour** $i = 1, 2$ **faire**

1. Calculer K_α^{ki} et K_q^{ki} sur l'intervalle T_i^k
2. test Convexité : si $K_\alpha^{ki} = 0$ alors mettre à jour $LB(T_i^k)$ et $UB(T_i^k)$ et aller à **étape d**
3. test Concavité : si $K_q^{ki} = 0$ alors mettre à jour $LB(T_i^k)$ et $UB(T_i^k)$ et aller à **étape d**
4. Test Pruning : Calculer LB_q^{ki} et résoudre $LB_q^{ki} = UB_k$ pour réduire l'intervalle recherché $[a_k^i, b_k^i]$
5. Calculer $LB_\alpha^{ki}(x)$. Soit z^{ki} et s_{ki}^* solution pour le problème convexe

$$\min \{z : LB_\alpha^{ki}(x) \leq z, LB_q^{ki}(x) \leq z, z \in \mathbb{R}, x \in T_i^k\} \quad (3.32)$$

$$\text{et } LB(T_i^k) = z^{ki}$$

6. Soit $M \leftarrow M \cup \{T_i^k : UB_k - LB(T_i^k) \geq \varepsilon, i = 1, 2\} \setminus \{T^k\}$.

d) Étape mise à jour

- Mettre à jour borne supérieure : $UB_{k+1} := \min\{UB_k, f(a_k^i), f(b_k^i), f(s_{ki}^*)\} := f(\bar{s}^k)$

- Mettre à jour borne inférieure : $LB_{k+1} = \min\{LB(T) : T \in M\}$.
- Suppression à partir de M tous les intervalles T tel que $LB(T) > UB_k - \varepsilon$.

e) Étape arrêt

- **Si** $M = \emptyset$ **alors Sortie** \bar{s}^k solution optimale et sortie algorithme
- **sinon** soit $k \leftarrow k + 1$, et aller à **Étape 2a**.
- test d'arrêt $UB_k - LB_k < \varepsilon$.

3.7.1 Convergence

Dans le théorème suivant, nous allons voir la convergence de notre algorithme.

Théorème La séquence $\{\bar{s}^k\}$ ($\{\bar{x}^k\}$) générée par l'algorithme converge vers la solution optimale pour le problème (P).

Démonstration. Si l'algorithme s'arrête à l'itération k qui peut être obtenu par la condition d'arrêt $UB_k - LB_k < \varepsilon$ ou le test convexité/concavité ou la méthode pruning alors on obtient une solution exacte ou une ε -solution optimale.

Supposons maintenant que l'algorithme est infini, alors il doit générer une infinité de séquences $\{T^k\}$ avec $T^k = [a_k, b_k]$ dont les longueurs décroissent vers zéro, alors ces séquences $\{T^k\}$ décroissent vers un singleton. De plus nous devons voir que $\lim_{k \rightarrow \infty} (UB_k - LB_k) = 0$. Soit UB_{LO}^k et LB_{LO}^k la borne supérieure et inférieure obtenues dans [51]. On peut voir aisément que $UB_k - LB_k \leq UB_{LO}^k - LB_{LO}^k$. En effet, nous avons

1. $UB_{LO}^k \geq UB_k$, car nous ajoutons dans notre algorithme le test pruning étape et le test convexité/concavité qui améliore la borne supérieure.
2. $LB_{LO}^k \leq LB_k$ par construction de $LB_k(x)$.

Par conséquent, nous avons

$$\begin{aligned} 0 &\leq UB_k - LB_k \leq UB_{LO}^k - LB_{LO}^k \\ &= f(\bar{s}^k) - f(a_k) \frac{b_k - s_k^*}{b_k - a_k} - f(b_k) \frac{s_k^* - a_k}{b_k - a_k} + \frac{K}{2} (s_k^* - a_k)(b_k - s_k^*). \end{aligned}$$

Puisque, nous avons [[36],[51]]

$$f(\bar{s}^k) - f(a_k) \frac{b_k - s_k^*}{b_k - a_k} - f(b_k) \frac{s_k^* - a_k}{b_k - a_k} \leq \frac{K}{8} (b_k - a_k)^2$$

et

$$\frac{K}{2} (s_k^* - a_k)(b_k - s_k^*) \leq \max_{x \in [a_k, b_k]} \left(\frac{K}{2} (x - a_k)(b_k - x) \right) = \frac{K}{8} (b_k - a_k)^2.$$

Donc $(UB_k - LB_k) \rightarrow 0$ quand $k \rightarrow \infty$ et la séquence $\{\bar{s}^k\}$ converge vers la solution optimale du problème (P). Le théorème est prouvé. \square

3.8 Résultats des calculs numériques

Nous commençons par des exemples illustratifs :

Exemple 3

On considère la fonction $f(x) = \sin x, x \in [0, 2\pi]$. Nous avons

$$f''(x) = -\sin x,$$

$-1 \leq f''(x) \leq 1, K_\alpha = K_q = 1$. Les bornes inférieures :

$$LB_q(x) = -\frac{1}{2}x(2\pi - x), LB_\alpha(x) = \sin x - \frac{1}{2}x(2\pi - x).$$

Puisque $f(0) = f(2\pi) = 0$, donc pas de pruning étape. Par résolution du problème convexe

$$\min \left\{ z : \sin x - \frac{1}{2}x(2\pi - x) \leq z, -\frac{1}{2}x(2\pi - x) \leq z, z \in \mathbb{R}, x \in [0, 2\pi] \right\}.$$

Nous obtenons, $z^0 = -\frac{1}{2}\pi^2, s_0^* = \pi(x_0^* = \pi)$ et $LB_0 = z^0, UB_0 = 0$. Nous scindons $[0, 2\pi]$ par subdivision via $s_0^* = \pi$.

Nous considérons en premier l'intervalle $[0, \pi]$, nous calculons $K_\alpha^1 = 1, K_q^1 = 0$. Pour le test Convexité/concavité : $K_q^1 = 0 \Rightarrow f$ est concave sur $[0, \pi]$ ses minimums sont atteints en 0 et π . De la même manière, pour l'intervalle $[\pi, 2\pi]$, nous calculons $K_\alpha^2 = 0$ et $K_q^2 = 1$. Par utilisation du test Convexité/concavité : puisque $K_\alpha^2 = 0$, alors f est convexe sur l'intervalle $[\pi, 2\pi]$. Minimum est atteint à $\frac{3\pi}{2}$. L'algorithme s'arrête au global minimum $\bar{s}^1 = \frac{3\pi}{2}$ avec $f(\frac{3\pi}{2}) = -1$.

Exemple 4

Pour l'exemple qui suit. Nous avons $f(x) = \sin(x) + \cos(x), x \in [0, 2\pi]$.

$$f''(x) = -\sin(x) - \cos(x).$$

$-2 \leq f''(x) \leq 2; K_\alpha = K_q = 2$. Les bornes inférieures sont données par

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

Puisque $f(0) = f(2\pi) = 1$ alors pas de pruning étape. Par résolution du problème convexe

$$\min \left\{ z : \sin(x) + \cos(x) - \frac{1}{1}x(2\pi - x) \leq z, 1 - \frac{1}{1}x(2\pi - x) \leq z, z \in \mathbb{R}, x \in [0, 2\pi] \right\}.$$

Nous obtenons $z^0 = 1 - \pi^2, x_0^* = \pi$ et $LB_0 = z^0, UB_0 = -1, \bar{s}^0 = \pi$. Ensuite nous scindons $[0, 2\pi]$ par subdivision via $s_0^* = \pi$. On commence par l'intervalle $[0, \pi]$, nous calculons $K_\alpha^{11} = \sqrt{2}$ et $K_q^{11} = 1$. Par utilisation du test Convexité/concavité, aucune modification. Par utilisation du Pruning étape, nous calculons $LB_q^{11}(x) = 1 - \frac{2x}{\pi} - \frac{1}{2}x(\pi - x) = UB_0 = -1$. Nous trouvons deux solutions, 1.2732 et π alors l'intervalle $[0, \pi]$ est réduit à l'intervalle $[1.2732, \pi]$ (i.e. la portion $[0, 1.2732]$ écartée). En plus, nous calculons $LB_q^{1r11}(x)$ sur l'intervalle $[1.2732, \pi]$ et nous résolvons

l'équation quadratique $LB_q^{1r1}(x) = -1$, cette procédure s'arrête sur l'intervalle avec le point π optimal et $UB_{11} = LB_{11} = -1$.

Considérons le second intervalle $[\pi, 2\pi]$, nous calculons $K_\alpha^{12} = 0$ alors par test convexité/concavité, f est convexe sur l'intervalle $[\pi, 2\pi]$, nous appliquons une locale recherche pour trouver le minimum $s_{12}^* = \frac{5\pi}{4}$; $f(s_{12}^*) = -\sqrt{2} = UB_{12} = LB_{12}$.

Nous avons $UB_1 = LB_1 = -\sqrt{2}$ et l'algorithme s'arrête après deux itérations avec la solution optimale $\bar{s}^1 = \frac{5\pi}{4}$ avec $f(\frac{5\pi}{4}) = -\sqrt{2}$.

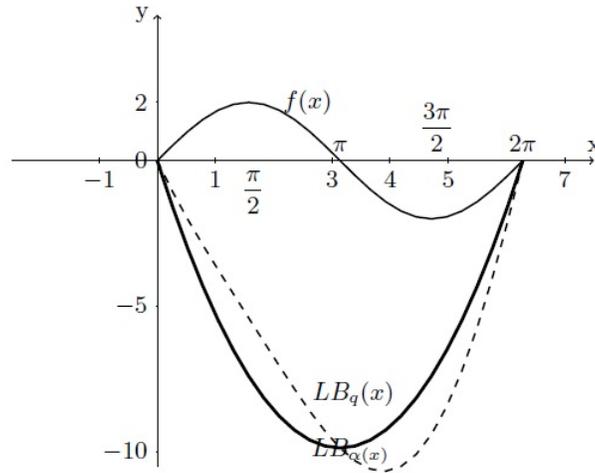


Fig. 3. The function $\sin(x)$ (bold), the lower bound $LB_\alpha(x)$ (dashed) and the lower bound $LB_q(x)$ (bold)

Maintenant nous donnons quelques problèmes test et comparons notre méthode avec les méthodes présentées dans [51] (par exemple Numeva entre autres) en termes de nombre d'itérations. L'environnement expérimental est implémenté autour du logiciel MATLAB et exécuté sur ordinateur DELL avec la configuration Intel Core I3 CPU M370 / 2.40 GHz and 4GB RAM.

Remarque 4

Nous constatons à la table ci-dessous que le nombre d'itérations dans notre algorithme de branch and prune est moins important que le nombre d'itérations dans les algorithmes de [51] pour toutes les fonctions test. La nouvelle fonction borne inférieure combinée avec le pruning étape et le test convexité/concavité nous ont permis des réductions significatives des nombres d'itérations.

Sans comparaison des deux bornes inférieures et par seulement la résolution du problème convexe dans notre algorithme BB proposé par exemple on aboutit à des résultats plus que satisfaisants.

TABLE 3.1 – Comparatif de résultats de calculs avec les algorithmes (par exemple Numeva) dans [51] avec $\varepsilon = 10^{-6}$.

Problem	Function	$[a, b]$	$Nb_{Iteration}$ our algorithm	$Nb_{Iteration}$ of algorithm in [51]	optimal solution
1	$x^6 - 15x^4 + 27x^2 + 250$	$[-4, 4]$	2	17	2.999934
2	$\frac{x^2-5x+6}{x^2+1}$	$[-5, 5]$	2	120	2.414197
3	$(3x - 1.4) \sin(18x)$	$[0, 1]$	1	3	0.966086
4	$2(x - 3)^2 + e^{\frac{x^2}{2}}$	$[-3, 3]$	1	13	1.590704
5	$(x + \sin(x))e^{-x^2}$	$[-10, 10]$	5	100	-0.679576
6	$-\sum_{i=1}^{i=5} \sin((i + 1)x + i)$	$[-10, 10]$	20	31	-6.720123 -0.436912 5.846301
7	$\sin(x) + \sin(\frac{3x}{10}) + \ln(x) - 0.84x$	$[2.5, 7.5]$	2	11	5.455267
8	$\sin(x) + \sin(\frac{2x}{3})$	$[3.1, 20.4]$	7	23	17.039213

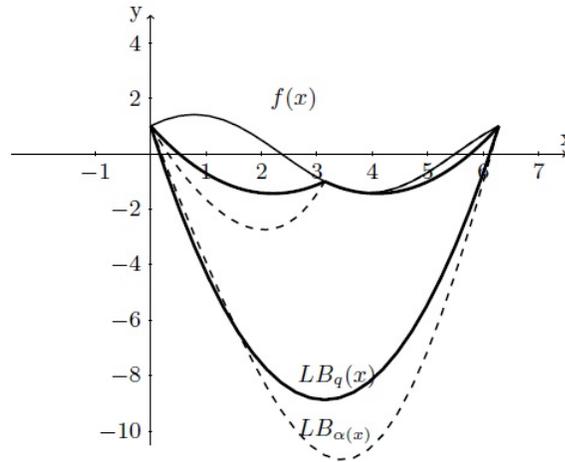


Fig. 4. The function $\sin(x) + \cos(x)$, the lower bound $LB_\alpha(x)$ (dashed) and the lower bound $LB_q(x)$ (bold) on the interval $[0, 2\pi]$. Also the lower bounds $LB_\alpha^1(x)$ and $LB_q^1(x)$, respectively $LB_\alpha^2(x)$ and $LB_q^2(x)$ are given on $[0, \pi]$ respectively $[\pi, 2\pi]$. Note that $LB_\alpha^2(x)$ coincides with f since $K_q^2 = 0$

On donne maintenant la comparaison entre notre méthode et la méthode dans [45] (Floudas).

Numéro du Problème	Nombre d'intervalles Floudas	Global Optimum Floudas	Nombre d'intervalles Nouvelle Méthode	Global Optimum Nouvelle Méthode	x* Nouvelle Méthode Notre Méthode	OBSERVATIONS
1	16	-1	4	-1	7.853981522644390 ; 14.137166777212814	
2	8	-1	2	-1	6.283185305508640	
3	1	0	1	0	0.99999157232	
4	64	-17.58287	19	-17.582871815180280	6.894525986331936	
5	1024	-0.020903	1	-0.020903	0.06781274	
6	16	-0.952897	7	-0.952896792547349	2.839347106361496	
7	8	-6.262872	1	-6.262872094177839	6.920063241795233	
8	16	0.077590	2	-0.077589685038973	0.902209774288853	0.07758968503897
9	16	0.211315	1	0.211314612658343	0.224882434878652	
10	16	-0.478362	3	-0.478361865691453	0.724896896650575	
11	4	-5.815675	2	-5.815674542978210	5.872866367472354	
12	8	-7.047444	3	-7.047444074140848	5.134338369777209	
13	8	-4.60138	3	-4.601307546066266	5.199786838870376	
14	4	-0.14110	2	-0.141100487712490	0.408237030060961	
15	16	-0.870885	5	-0.870885482641672	4.858047188917565	
16	16	-9.031249	10	-9.031249441398060	5.791796698749343	
17	1	0.475689	1	0.475688626380756	-0.787891419247242	
18	8	0	4	0	3.141584237231236 ; 6.283180830887630	
19	64	-1	2	-1	-0.000044732677344	
20	1	1	1	1	0.000000000000000	
21	16	1	2	1	0.000055473416505	
22	4	-0.918397	3	-0.918397340885203	3.251079686033197	
23	64	-0.824239	5	-0.824239398456395	-0.679575338135825	
24	4	-0.027864	3	-0.027864070194811	3.926986265091935	
25	8	3.5	3	3.5	2.094394271217541 ; 4.188791036091182	
26	8	0.367879	4	0.3678794411	5.759584452678632 ; 3.665190847312067 ; 1.570799741097562	

27	8	-0.451388	4	-0.45138781886	5.006390855245968 ; 1.864797813097123	
28	8	-1	4	-1	4.712397606983803 ; 10.995573997239797 17.278759613821912	
29	2	-0.410135	2	-0.410315206527357	3.862077972154709	
30	16	-0.718282	5	-0.718281789521809	2.617937458512833	
31					WE HAVE NOT PARAMETERS	
32	1024	-1	8	-1	0,212206487 0,090945688 0,057874517 0,042441322 0,033506303 0,027679122 0,023578509 0,020536123	
33	32	-12.03125	19	-12.031249441334111	-6.774573824849375	
34	4	-0.535534	3	-0.535533905932071	2.414211465942532	
35					WE HAVE NOT PARAMETERS	
36	1	-0.35	1	5.05000000093115	2.000012866129532	5.0500000009311
37	4	-32.78126	2	-32.781261290208491	0.713679797555900	
38	8	7	6	7	2.9999917166	
39	16	-1	4	-1	1.381976898751345 ; 3.618035944847370	
40	4	-89	4	-89	2.000058945943136	

3.9 Conclusion

Dans ce travail nous avons proposé l'algorithme de branch and prune pour calculer tous les minimiseurs globaux pour des fonctions unidimensionnelles sous contraintes de variables bornées. L'algorithme utilise la combinaison de deux bornes inférieures et use la technique du pruning aussi bien que le test convexité/concavité afin d'accélérer le processus de recherche. Les résultats numériques montrent que notre méthode proposée est efficace.

Le développement software est basé autour de l'algorithme présenté dans notre article. On peut considérer la procédure **Prunning**, par simple exemple comme étant une méthode d'optimisation type CSP et par conséquent l'accélération de la convergence.

Le software (développé sur matlab 2009), est testé sur 40 problèmes test [45]. L'article [45] est intéressant même seulement avec le nombre d'itérations (paramètre très important). Dans nos prochains travaux nous envisagerons le paramètre temps d'exécution CPU.

3.10 Perspectives

*/Nous prévoyons d'améliorer le travail dans \mathbb{R}^n .

**/ Obtenir un algorithme pour le multi-objectifs, dans \mathbb{R}^n .

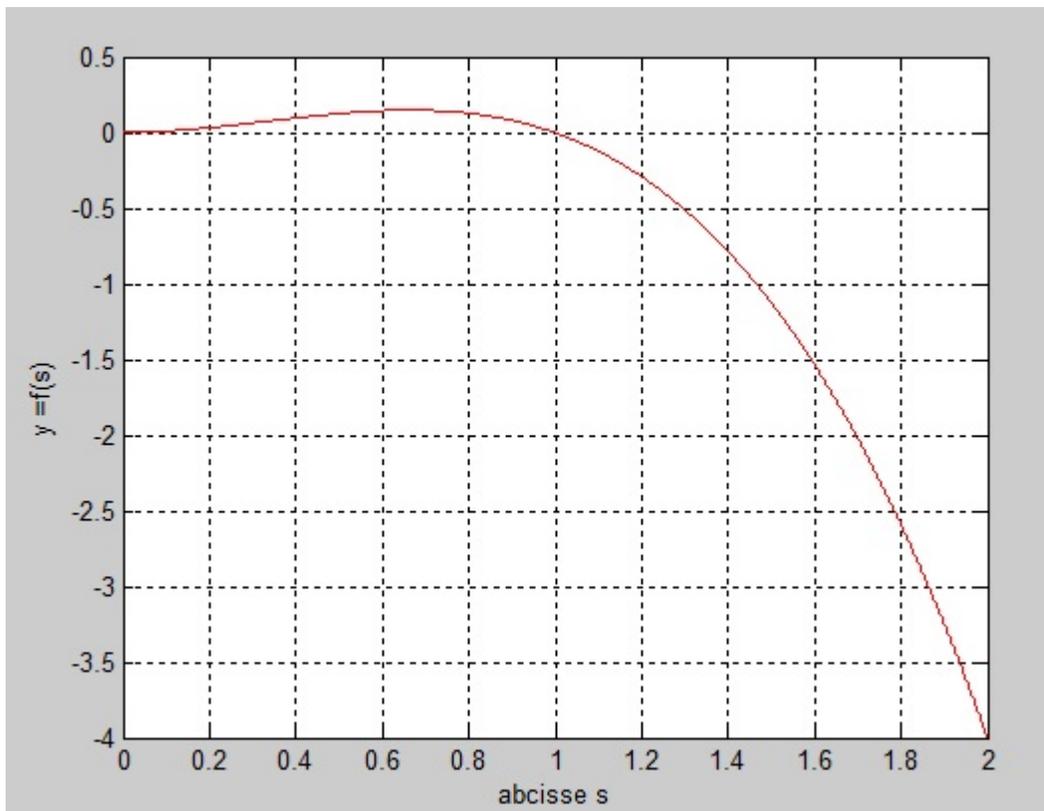
***/ Exhiber de nouvelles conditions nécessaires et suffisantes avec des paramètres flous et stochastiques dans \mathbb{R}^n .

****/Nous prévoyons d'améliorer le travail dans \mathbb{R}^n avec des espaces mixtes ou discrets.

Illustrations d'itérations pour des problèmes Tests :

Problèmes	Objectif	$[s_0, s_1]$
f0	$b(s) = +s^2 - s^3$	$[0, 2]$
f1	$b(s) = \sin(s) + \sin((3 * s)/10) + \log(s) - 0.84 * s$	$[2.5, 7.5]$
f2	$b(s) = 0.75 * \sin(s) + 0.25 * \cos(s)$	$[0, 1]$
f3	$b(s) = \sin(s) + \sin((2 * s)/3)$	$[3.1, 20.4]$
f4	$b(s) = (3 * s - 1.4) * \sin(18 * s)$	$[0, 1]$
f5	$b(s) = 2 * (s - 3)^2 + \exp((s^2)/2)$	$[-3, 3]$
f6	$b(s) = s^6 - 15 * s^4 + 27 * s^2 + 250$	$[-4, 4]$
f7	$b(s) = (s^2 - 5 * s + 6)/(s^2 + 1)$	$[-5, 5]$
f8	$b(s) = (s + \sin(s)) * \exp(-s^2)$	$[-10, 10]$
f9	$b(s) = s^4 - 3 * s^3 - 1.5 * s^2 + 10 * s$	$[-5, 5]$
f10	$b(s) = - \sum_{n=2}^6 (n * \sin((n) * s + (n - 1)))$	$[-10, 10]$

Problème Test f0 : $b(s) = +s^2 - s^3, \forall s \in [0, 2]$.

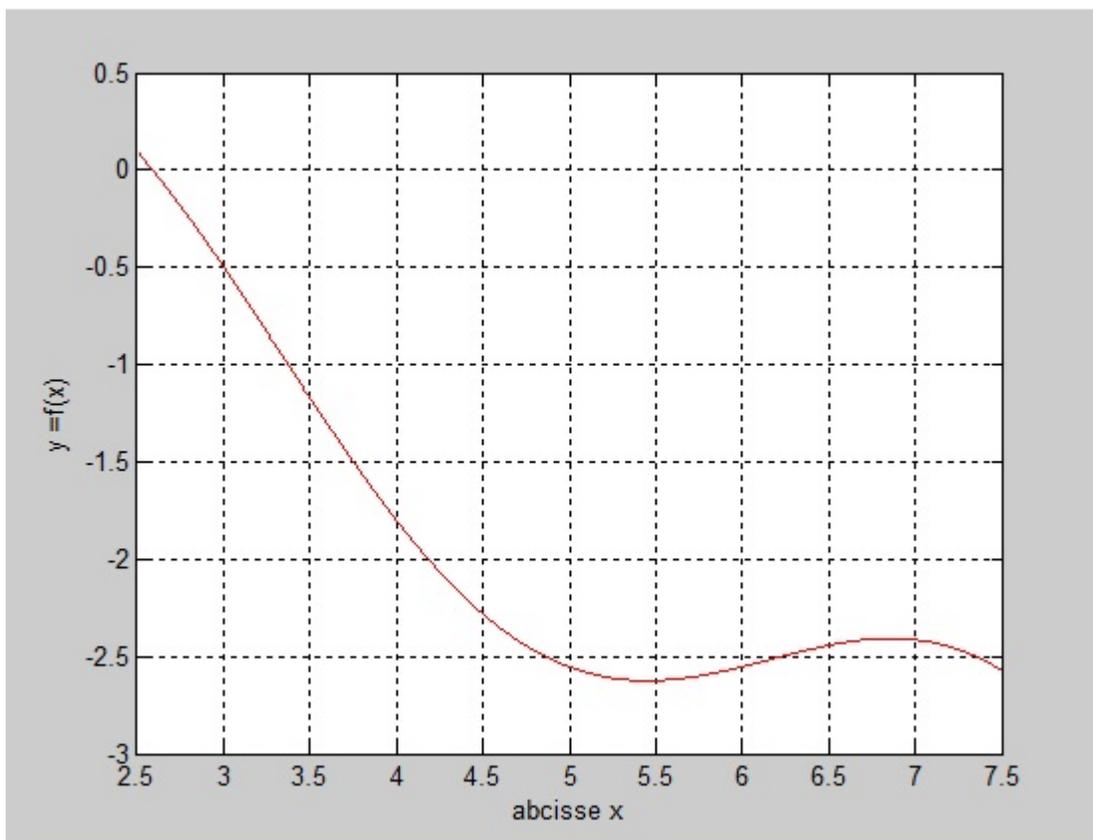


Progression des itérations

Intervalle Intitulé	Intervalle	Intervalle réduit	s^* (Optimal)
T_0	[0 , 2]	[2.00 ,2.00]	2.00

Solution en 1 Itération.

Problème Test f1 : $b(s) = \sin(s) + \sin((3 * s)/10) + \log(s) - 0.84 * s$, $\forall s \in [2.5 , 7.5]$.

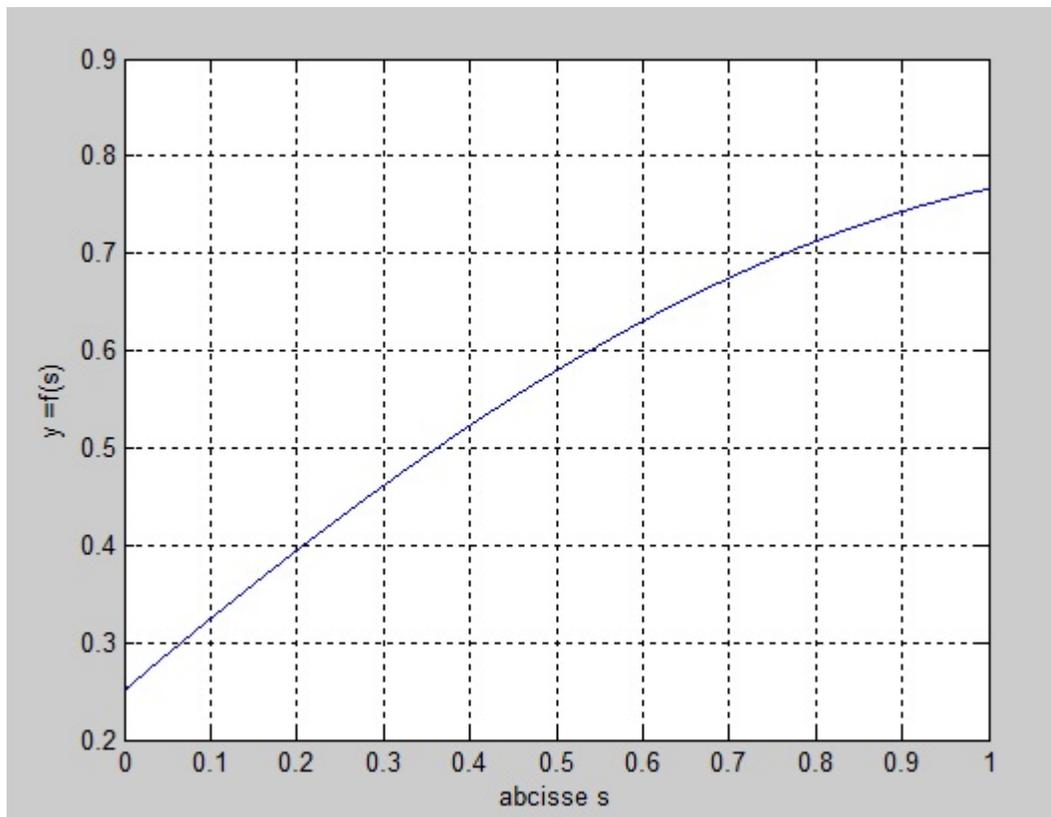


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle reduit	s^* (Optimal)
T_0	[2.5 , 7.5]	[5.0545 , 7.5]	-
T_{11}	[5.0545 , 6.0069]	[5.0545 , 6.0069]	5.455267
T_{12}	[6.0069 , 7.5]	[7.5 7.5]	-

Solution en 02 Itérations.

Problème Test f2 : $b(s) = 0.75 * \sin(s) + 0.25 * \cos(s), \forall s \in [0 , 1]$.

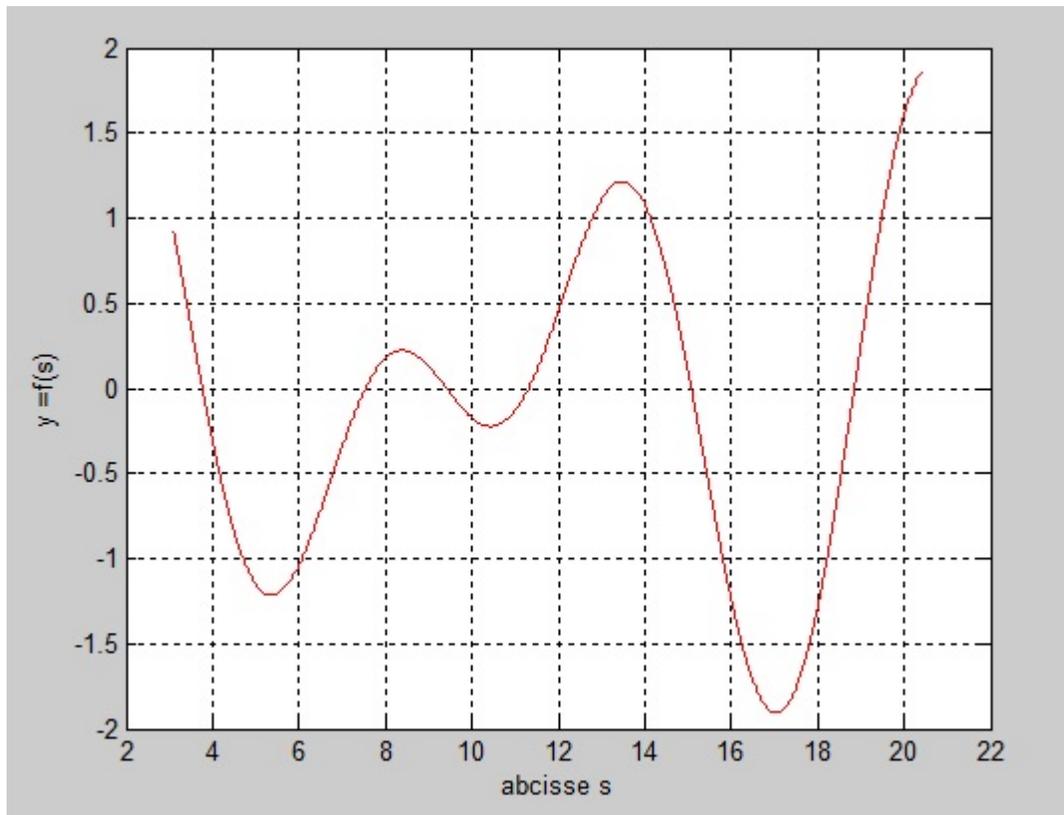


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle réduit	s^* (Optimal)
T_0	[0 , 1]	[0 , 1]	0

Solution en 1 Itération.

Problème Test f3 : $b(s) = \sin(s) + \sin((2 * s)/3), \forall s \in [3.1 , 20.4]$.

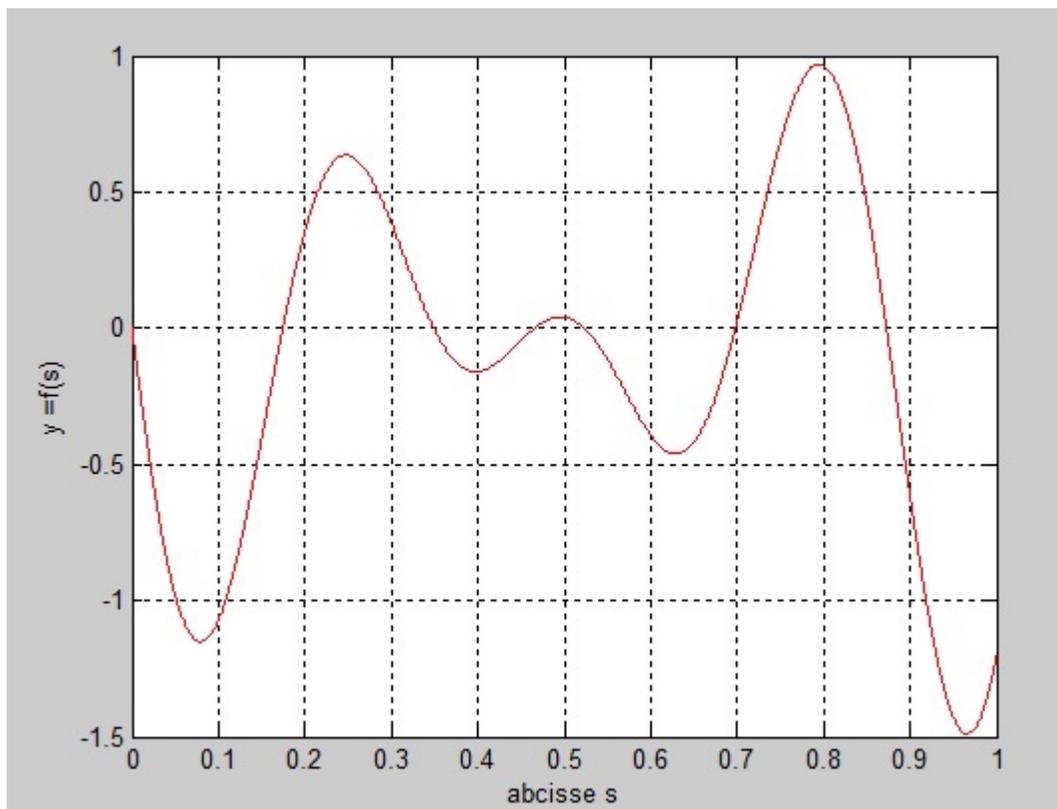


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle réduit	s^* (Optimal)
T_0	[3.1 , 20.4]	[3.1 , 19.426]	-
T_{11}	[3.1 , 11.263]	[3.7872 , 11.2634]	-
T_{12}	[11.263 , 19.426]	[11.263 , 18.835]	-
T_{21}	[3.7872 , 7.4956]	[3.7872 , 7.4956]	-
T_{22}	[7.4956 , 11.263]	[7.4956 , 11.263]	-
T_{31}	[3.7872 , 5.6414]	[3.7872 , 5.6414]	-
T_{32}	[5.6414 , 7.4956]	[5.6413 , 6.040]	-
T_{41}	[7.4956 , 9.4948]	[7.4956 , 9.4948]	-
T_{42}	[9.4948 , 11.263]	[9.4948 , 11.263]	-
T_{51}	[11.263 , 15.0969]	[9.848 , 11.263]	-
T_{52}	[15.0969 , 18.835]	[15.0969 , 18.835]	-
T_{61}	[15.0969 , 16.9661]	[16.560 , 16.9660]	-
T_{62}	[16.9661 , 18.835]	[15.0969 , 18.835]	17.0392131

Solution en 07 Itérations.

Problème Test f4 : $b(s) = (3 * s - 1.4) * \sin(18 * s)$, $\forall s \in [0 , 1]$.

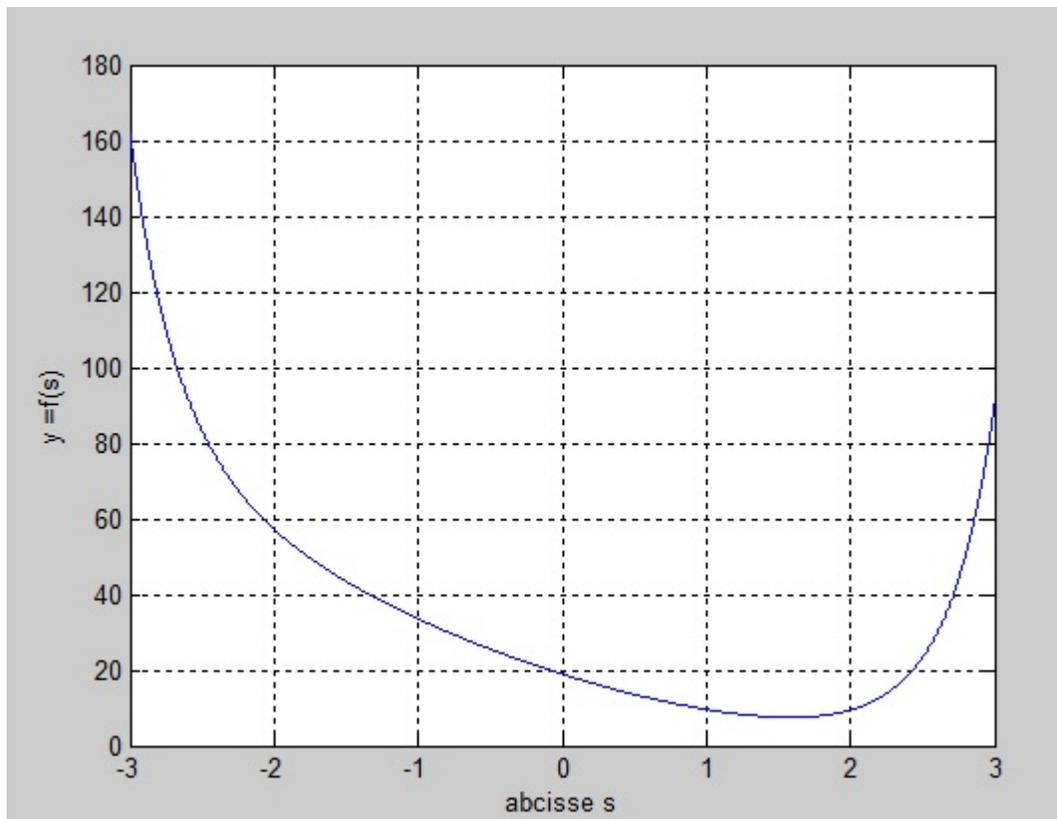


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle réduit	s^* (Optimal)
T_0	[0 , 1]	[0.89125 , 1]	0.966086

Solution en 1 Itération.

Problème Test f5 : $b(s) = 2 * (s - 3)^2 + \exp((s^2)/2), \forall s \in [-3 , 3]$.

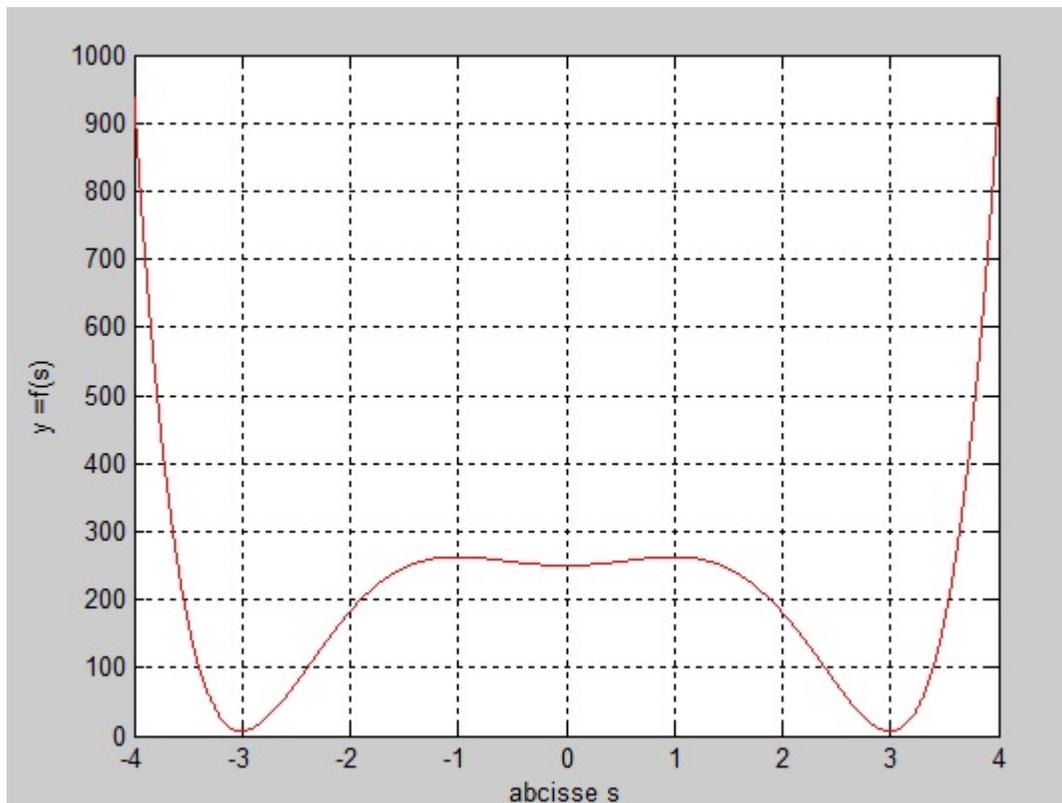


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle réduit	s^* (Optimal)
T_0	$[-3, 3]$	$[-3, 3]$	1.5907040

Solution en 1 Itération.

Problème Test f6 : $b(s) = s^6 - 15 * s^4 + 27 * s^2 + 250, \forall s \in [-4, 4]$.

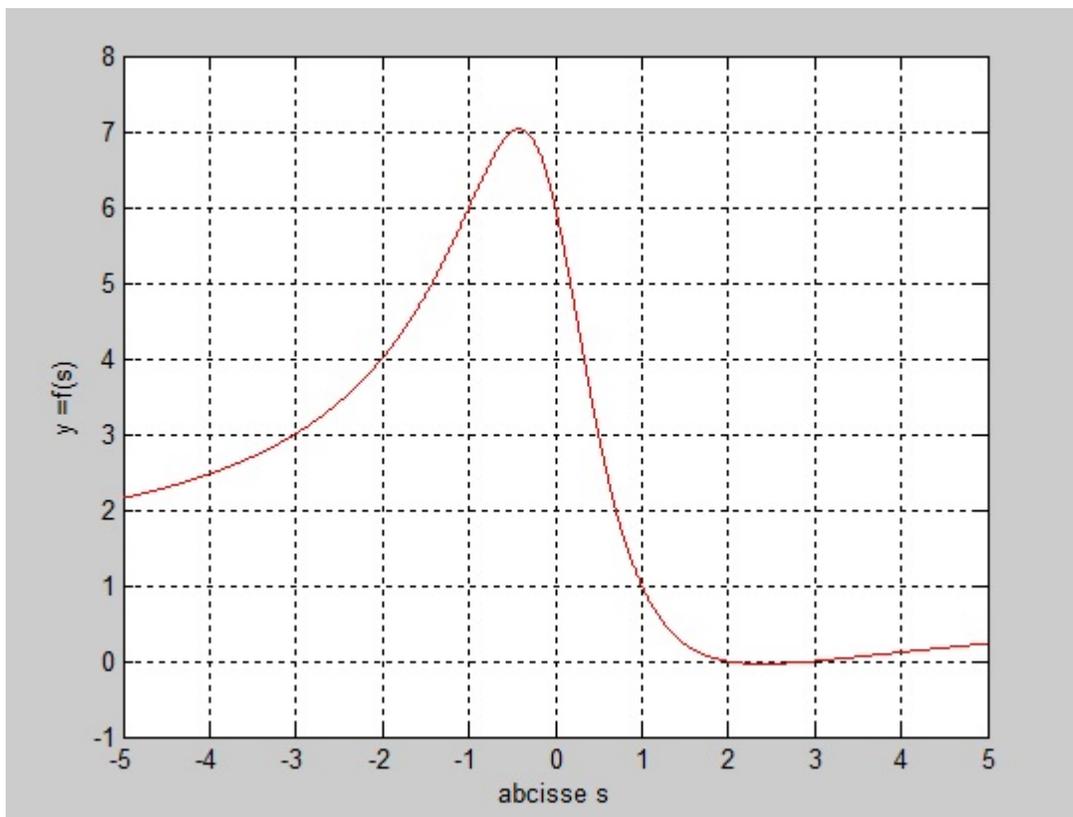


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle reduit	s^* (Optimal)
T_0	$[-3, 3]$	$[-3, 3]$	-
T_{11}	$[-3, 2.7651]$	$[-3.00, -3.000]$	-3.0000
T_{12}	$[2.7651, 3]$	$[2.7651, 3]$	2.99993421

Solution en 02 Itérations.

Problème Test f7 : $b(s) = (s^2 - 5 * s + 6)/(s^2 + 1), \forall s \in [-5, 5]$.

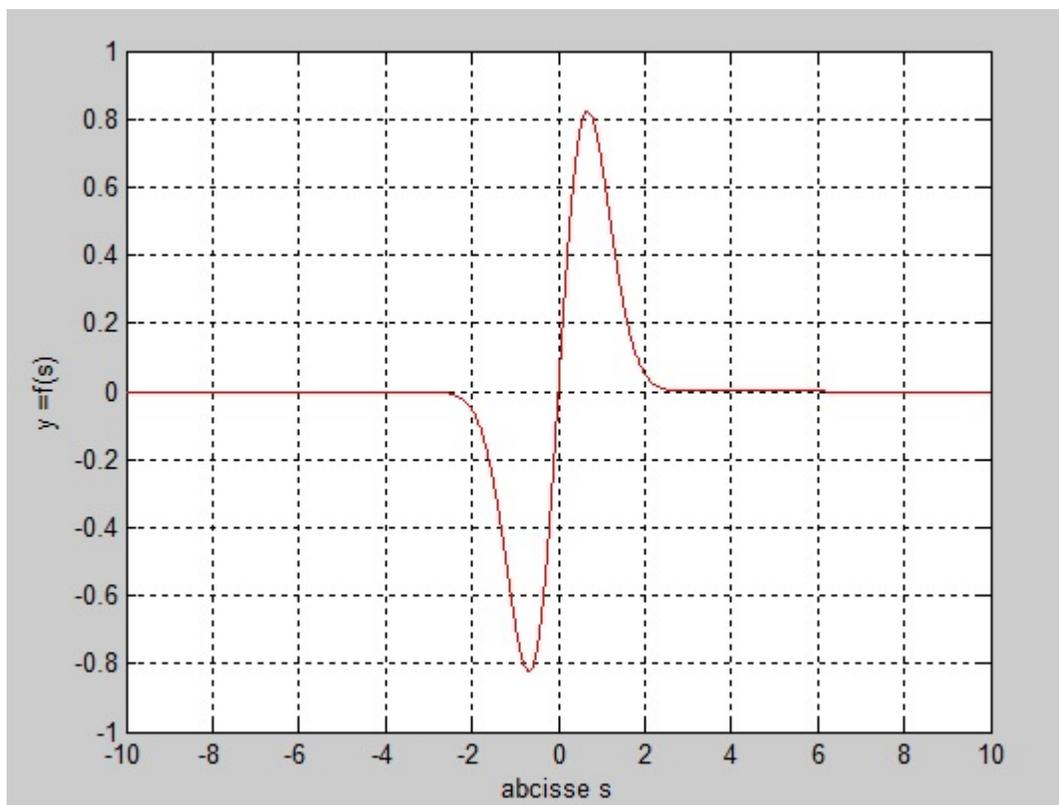


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle reduit	s^* (Optimal)
T_0	$[-5, 5]$	$[1.5, 5]$	-
T_{11}	$[1.5, 3.25]$	$[1.5, 3.25]$	2.414197
T_{12}	$[3.25, 5]$	$[0.6184, 3.2499]$	-

Solution en 02 Itérations.

Problème Test f8 : $b(s) = b(s) = (s + \sin(s)) * \exp(-s^2), \forall s \in [-10, 10]$.

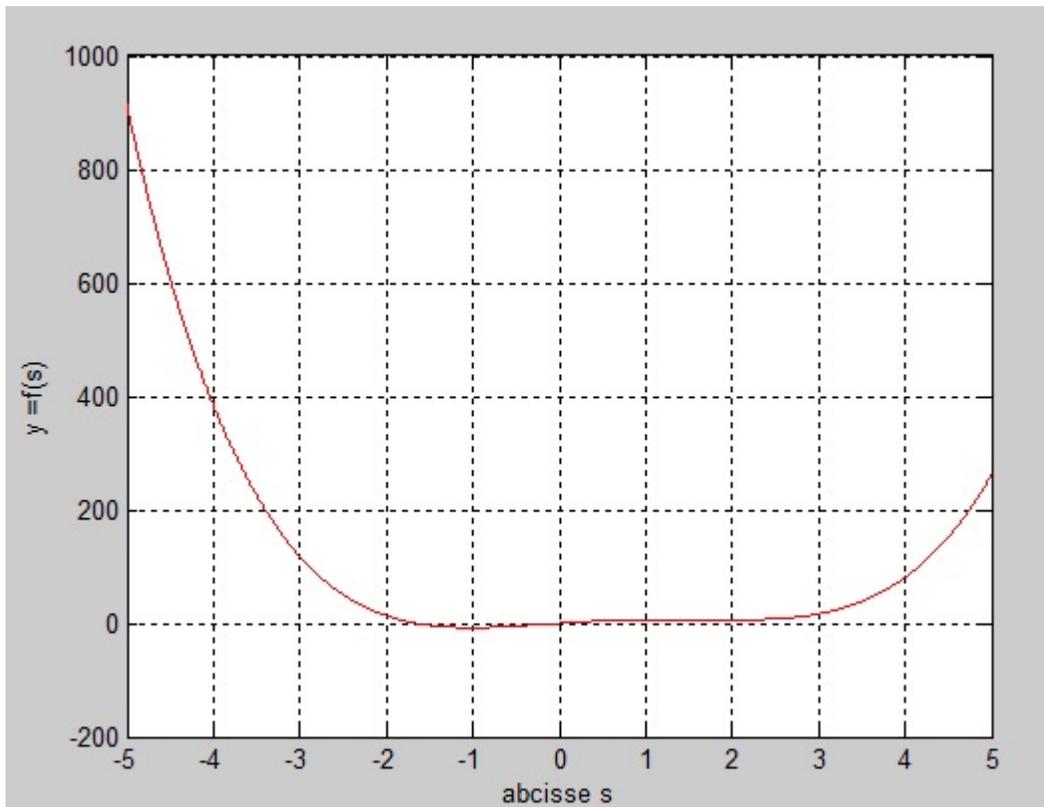


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle réduit	s^* (Optimal)
T_0	$[-10, 10]$	$[-10, 10]$	-
T_{11}	$[-10, 0]$	$[-10, 0]$	-
T_{12}	$[0, 10]$	$[0, 10]$	-
T_{21}	$[0, 5]$	$[0, 5]$	-
T_{22}	$[5, 10]$	$[0, 5]$	-
T_{31}	$[0, 2.5241]$	$[-1.2224, -0.2612]$	-
T_{32}	$[2.5241, 5]$	$[2.5241, 5]$	-
T_{41}	$[-1.2224, -0.74183]$	$[-0.789, -0.7418]$	-
T_{42}	$[-0.74183, -0.2612]$	$[-0.7418, -0.2612]$	-0.679576

Solution en 05 Itérations.

Problème Test f9 : $b(s) = s^4 - 3 * s^3 - 1.5 * s^2 + 10 * s, \forall s \in [-5, 5]$.

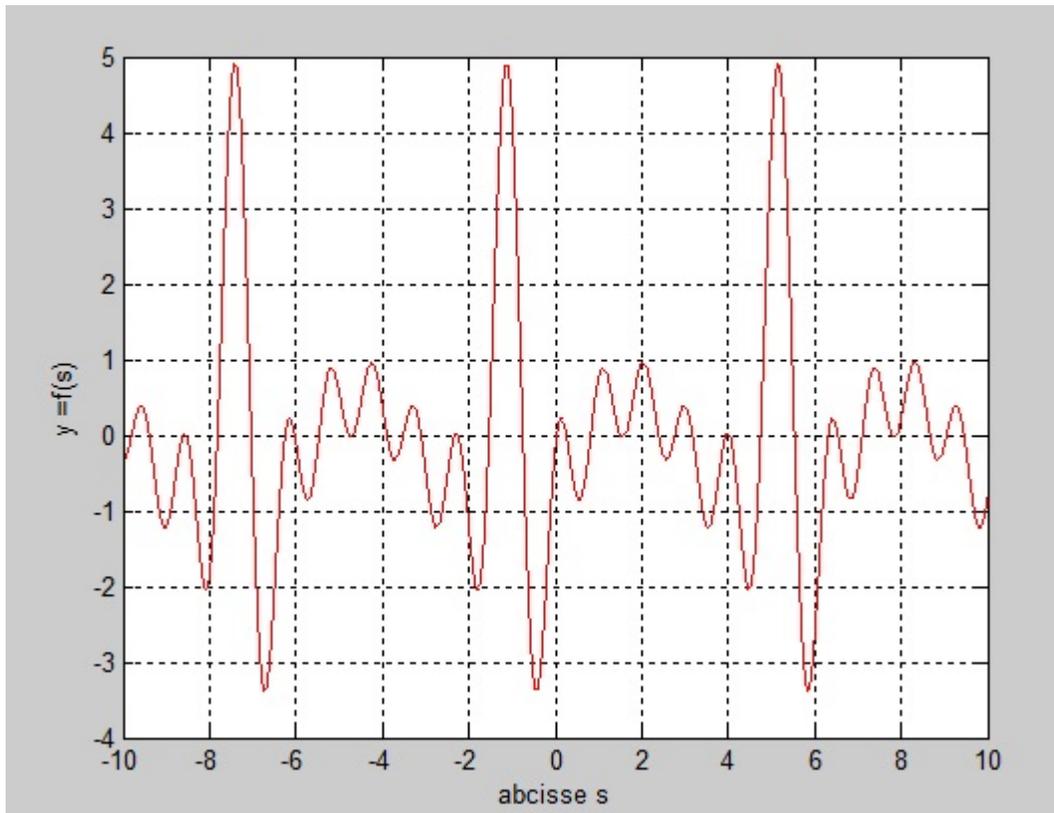


Progression des itérations

Intervalle Intitule	Intervalle	Intervalle reduit	s^* (Optimal)
T_0	$[-5, 5]$	$[-3.635, 5]$	-
T_{11}	$[-3.635, 0.6825]$	$[-1.8069, 0.6825]$	-
T_{12}	$[0.6825, 5]$	$[0.6825, 1.6243]$	-
T_{21}	$[-1.8069, -0.5622]$	$[-1.8069, -0.5622]$	-1.0000
T_{22}	$[-0.5622, 0.6825]$	$[-0.9157, -0.5622]$	-

Solution en 03 Itérations.

Problème Test f10 : $b(s) = -\sin((2) * s + 1) - \sin((3) * s + 2) - \sin((4) * s + 3) - \sin((5) * s + 4) - \sin((6) * s + 5), \forall s \in [-10, 10]$.



Progression des itérations

Iterations	Intervalle reduit	LB_k	UB_k	s^* (Optimal)	Obs
1	[-9.2032 , -8.8311]	-1.2168	-0.7511	-9.0276	Convexe
2	[-8.1341 , -8.1102]	-2.0114	-1.9602	-8.1102	Convexe
3	[-8.1102 , -7.8800]	-2.0354	-0.8251	-8.0804	Convexe
4	[-6.9879 , -6.6800]	-3.3729	-0.8253	-6.7201	Convexe
5	[-6.6800 , -6.6265]	-3.3179	-3.0890	-6.6799	Convexe
6	[-6.3721 , -6.0605]	-0.8244	0.1006	-6.3721	Concave
7	[-5.9017 , -5.6907]	-0.8454	-0.4738	-5.7290	Convexe
8	[-2.9200 , -2.5489]	-1.2168	-0.7550	-2.7444	Convexe
9	[-1.8509 , -1.8270]	-2.0114	-1.9601	-1.8270	Convexe
10	[-1.8270 , -1.5968]	-2.0354	-0.8251	-1.7972	Convexe
11	[-0.7047 , -0.3968]	-3.3729	-0.8253	-0.4369	Convexe
12	[-0.3968 , -0.3434]	-3.3178	-3.0891	-0.3967	Convexe
13	[0.5157 , 0.7332]	-0.8454	-0.4184	0.5542	Convexe
14	[4.3521 , 4.4708]	-2.0290	-1.6198	4.4708	Convexe
15	[4.4709 , 4.6235]	-2.0354	-1.4683	4.4860	Convexe
16	[5.5785 , 5.8864]	-3.3729	-0.8253	5.8463	Convexe
17	[5.8864 , 5.9398]	-3.3177	-3.0893	5.8865	Convexe
18	[6.1943 , 6.5058]	-0.8244	0.1009	6.1943	Concave
19	[6.6646 , 6.8756]	-0.8454	-0.4737	6.8374	Convexe
20	[9.5854 , 10.0000]	-1.2168	-0.5577	9.8220	Convexe

Solution en 20 Itérations.

TABLEAU GLOBAL RÉCAPITULATIF (comparatif) DES RÉSULTATS SELON LES MÉTHODES [51]

Functions	Eff.B.B	B.B	S-LLB	N-LLB	O-LLB	Zil1	Zil2	Strong	Pijav	Brent	Batish
f1	02	9	12	12	19	33	29	45	462	25	120
f3	07	18	18	18	24	37	38	442	448	45	158
f8	05	29	73	73	94	125	165	150	3817	161	816
f10	20	34	13	19	35	35	34	98	376	229	83
Average	8.50	22.50	29.00	30.50	43.00	57.50	66.5	158.75	1275.75	112.50	294.25

3.11 Notre deuxième (2°) contribution pour l'optimisation globale.

Consulter [90].

Remarque 3.11.1. Pour notre deuxième (2°) contribution (le texte original soumis et publié).

Combination of two underestimators for univariate global optimization.

RAIRO-Oper. Res. : :Volume 52, Number 1, January March

2018 : :Page(s)177-186 : :DOI <https://doi.org/10.1051/ro/2018013> : :Published online 23 April 2018

Ouanes¹mohand CHEBBAH¹Mohammed Zidna²ahmed

ouanes_mohand@yahoo.fr¹ chbbhea@yahoo.fr¹ ahmed.zidna@univ – lorraine.fr²

LITA UFR Metz, Univ Paul Verlaine - Metz, Ile du Saulcy, 57045 Metz, France²

Laboratoire LAROMAD University Tizi ouzou Algeria DZ¹.

On considère le problème suivant :

$$(P) \begin{cases} \min b(s) \\ s \in [s^0, s^1] \subset \mathbb{R}. \end{cases}$$

Avec b une fonction continue non convexe de classe C^2 sur un réel intervalle $[s^0, s^1]$. Multitudes de méthodes ont été étudiées en littérature pour des problèmes d'optimisation globale unidimensionnels (voir [42] et références). Citons aussi l'importance de l'optimisation globale unidimensionnel.

Nous proposons une nouvelle borne inférieure qui combine les deux bornes inférieures données dans [11][51]. Nos principales contributions pour notre travail sont : i) Construction de la nouvelle borne inférieure qui est meilleure que les deux bornes inférieures (voir [11][51]), ii) Test convexité/concavité est utilisé afin d'accélérer la convergence de notre algorithme branch and bound. iii) Comparaison avec la méthode présentée dans [45] en termes de sous intervalles.

3.11.1 Motivations quant à l'utilisation de l'optimisation globale unidimensionnelle

1. Calculs divers comme pour le cas en optimisation, quant à l'évaluation des paramètres Wolf (première et deuxième condition).
2. Horst et Tuy [56], [57] et bien d'autres utilisent l'optimisation unidimensionnelle comme procédure intermédiaire dans le processus de résolution, cas de programmation anti-convexe par exemple.
3. Les travaux de Patrick Siarry et Rachid Chelouah ou la méthode Aliénor utilisent l'optimisation unidimensionnelle pour résoudre des problèmes dans \mathbb{R}^n à travers des fonctions réductrices pour le cas Aliénor par exemple.
4. Floudas dans ses récents articles utilise l'optimisation unidimensionnelle pour résoudre des problèmes dans \mathbb{R}^n .
5. Les problèmes factorables/séparables et conséquences [75], [74].
6. ...etc.

Exemple 3.11.1. Voici un exemple sur les variables séparables.

$$\left\{ \begin{array}{l} f(x) = x_1x_2 + x_3^4 - 10x_3^3 + \cos(\exp(x_4)) \rightarrow \min \\ -1 \leq x_1, x_2 \leq +1 \\ -5 \leq x_3 \leq +10 \\ 0 \leq x_4 \leq +10. \end{array} \right. \quad (3.33)$$

On propose le principe de séparation (à variables séparables) suivant :

$$f_1(x_1, x_2) = x_1x_2, f_2(x_3) = x_3^4 - 10x_3^3, f_3(x_4) = \cos(\exp(x_4)).$$

Exemple 3.11.2. Voici un exemple (optimisation sur un Ω) avec la factorisation de Smith Standard.

$$\left\{ \begin{array}{l} f(x) = \log(x_1x_2) \exp(x_1 + x_2) + 2x_1 \rightarrow \min \\ w_1 = x_1x_2 \\ w_2 = x_1 + x_2 \\ w_3 = \log(w_1) \\ w_4 = \exp(w_2) \\ w_5 = w_3w_4 \\ f(w_5, x_1) = w_5 + 2x_1. \end{array} \right. \quad (3.34)$$

Remarque 3.11.2. - Ces techniques introduites dans le but de conversions (en guise de contributions).

- La factorisation et la séparation pour prendre le maximum de domaines d'intérêt d'application en optimisation non linéaire non convexe par exemple pour notre cas.
- Les exemples pour la suite, ont été choisis, on pouvait puiser dans les problèmes Tests.
- On a présenté une variante de notre algorithme, cette dernière améliorée et améliorabile.
- La reformulation de problèmes d'optimisation dans \mathbb{R}^n (**les problèmes Test de la Bibliothèque Global Optimization**) peuvent être pris en charge par notre algorithme ci-dessous ou du moins ce dernier peut être étape intermédiaire de résolution.
- Ajout de procédures d'accélération de la convergence pour l'algorithme (pour performance).

3.11.2 La borne inférieure dans la méthode α BB [11]

Soit K_α un nombre réel positif tel que $K_\alpha \geq \max\{0, -b''(s)\}$, pour tout s dans l'intervalle $[s^0, s^1]$. La borne inférieure dans la méthode α BB sur l'intervalle $[s^0, s^1]$ est donnée par

$$LB_\alpha(s) = b(s) - \frac{K_\alpha}{2}(s - s^0)(s^1 - s). \quad (3.35)$$

Cette borne inférieure a les propriétés suivantes :

1. Elle est convexe (i.e. $LB_\alpha''(s) = b''(s) + K_\alpha \geq 0$ car $K_\alpha \geq -b''(s), \forall s \in [s^0, s^1]$).
2. Elle coïncide avec la fonction b aux extrémités de l'intervalle $[s^0, s^1]$ (i.e. par construction de $LB_\alpha(s)$).
3. Elle est une fonction borne inférieure de $b(s)$ (i.e. $b(s) - LB_\alpha(s) = \frac{K_\alpha}{2}(s - s^0)(s^1 - s) \geq 0, \forall s \in [s^0, s^1]$).

Pour plus de détails voir [11].

3.11.3 La borne inférieure quadratique [51]

Soit K un nombre réel positif tel que $K \geq |b''(s)|$, pour tout s dans l'intervalle $[s^0, s^1]$. La borne inférieure quadratique développée dans [51] sur l'intervalle $[s^0, s^1]$ est donnée par

$$LB_q(s) = b(s^0) \frac{s^1 - s}{s^1 - s^0} + b(s^1) \frac{s - s^0}{s^1 - s^0} - \frac{K}{2}(s - s^0)(s^1 - s). \quad (3.36)$$

Cette borne inférieure a les propriétés suivantes :

1. Elle est convexe et quadratique (i.e. $LB_q''(s) = K \geq 0$).
2. Elle coïncide avec la fonction b aux extrémités de l'intervalle $[s^0, s^1]$ (i.e. par construction de $LB_q(s)$).
3. Elle est une fonction borne inférieure de $b(s)$ (i.e. $(b(s) - LB_q(s))'' = (b''(s) - K) \leq 0, \forall s \in [s^0, s^1]$ qui implique que $(b(s) - LB_q(s))$ est concave sur $[s^0, s^1]$, les minimums aux extrémités de $[s^0, s^1]$ alors $(b(s) - LB_q(s)) \geq 0, \forall s \in [s^0, s^1]$).

Pour plus de détails voir [51].

3.12 Nouvelle fonction borne inférieure

Soit K_q un réel nombre positif tel que $K_q \geq \max\{0, b''(s)\}$, pour tout s dans l'intervalle $[s^0, s^1]$ et soit $L_h b(s)$ l'interpolation linéaire de b sur cet intervalle donnée par

$$L_h b(s) = b(s^0) \frac{s^1 - s}{s^1 - s^0} + b(s^1) \frac{s - s^0}{s^1 - s^0}.$$

La nouvelle borne inférieure sur cet intervalle $[s^0, s^1]$ est donnée par

$$LB(s) = \frac{K_q b(s) + K_\alpha L_h b(s)}{K_\alpha + K_q} - \frac{K_\alpha K_q}{2(K_\alpha + K_q)} (s - s^0)(s^1 - s). \quad (3.37)$$

Proposition 3.12.1. i) $LB(s)$ coïncide avec $b(s)$ aux extrémités de $[s^0, s^1]$.

ii) $LB(s)$ est convexe sur l'intervalle $[s^0, s^1]$.

Démonstration. **i)** Elle est par construction de $LB(s)$.

ii) Pour tout s dans l'intervalle $[s^0, s^1]$, nous avons $LB''(s) = \frac{K_q b''(s)}{K_\alpha + K_q} + \frac{K_\alpha K_q}{K_\alpha + K_q} = \frac{K_q(K_\alpha + b''(s))}{K_\alpha + K_q} \geq 0$. En effet K_α, K_q sont des nombres positifs et pour tout s dans $[s^0, s^1]$, nous avons $(K_\alpha + b''(s)) \geq 0$. Alors LB est convexe. □

Dans le théorème suivant nous montrons que $LB(s)$ est une borne inférieure de $b(s)$.

Théorème 3.12.1. $LB(s) \leq b(s), \forall s \in [s^0, s^1]$.

Démonstration. Par calcul de la seconde dérivée de $LB(s) - b(s)$, nous avons $(LB(s) - b(s))'' = \frac{K_q(K_\alpha + b''(s))}{K_\alpha + K_q} - b''(s) = \frac{K_\alpha(K_q - b''(s))}{K_\alpha + K_q} \geq 0$. Comme K_α et K_q sont des nombres positifs et $K_q - b''(s) \geq 0$ sur $[s^0, s^1]$, alors $LB(s) - b(s)$ est convexe. Donc $LB(s) - b(s)$ est négatif, qui implique que $LB(s) \leq b(s)$ dans $[s^0, s^1]$. □

Nous allons montrer dans les deux prochains théorèmes que la nouvelle borne inférieure est meilleure que la borne inférieure classique développée dans la méthode α BB [11] et la borne inférieure quadratique développée dans [51] respectivement.

Théorème 3.12.2. $LB(s) \geq LB_\alpha(s), \forall s \in [s^0, s^1]$.

Démonstration. De la même manière comme ci-dessus, pour tout s dans $[s^0, s^1]$, nous avons $(LB(s) - LB_\alpha(s))'' = \frac{K_\alpha(K_q - b''(s))}{K_\alpha + K_q} - K_\alpha = -\frac{K_\alpha(K_\alpha + b''(s))}{K_\alpha + K_q} \leq 0$. puisque K_α et K_q sont des nombres positifs et $(K_\alpha + b''(s))$ est positif sur l'intervalle $[s^0, s^1]$; alors $(LB(s) - LB_\alpha(s))$ est concave. Les minimums aux extrémités de cet intervalle, donc $(LB(s) - LB_\alpha(s)) \geq 0$ qui implique que $LB(s)$ est meilleure que $LB_\alpha(s)$ sur l'intervalle $[s^0, s^1]$. □

Théorème 3.12.3. $LB(s) \geq LB_q(s), \forall s \in [s^0, s^1]$.

Démonstration. Par calcul de la seconde dérivée de $LB(s) - LB_q(s)$, nous avons

$$(LB(s) - LB_q(s))'' = \frac{K_q b''(s)}{K_\alpha + K_q} + \frac{K_\alpha K_q}{K_\alpha + K_q} - K.$$

Premier cas : $K = K_\alpha \geq K_q$

$$(LB(s) - LB_q(s))'' = \frac{K_q b''(s) - K_\alpha^2}{K_\alpha + K_q} \leq \frac{K_q b''(s) - K_q^2}{K_\alpha + K_q} = (b''(s) - K_q) \frac{K_q}{K_\alpha + K_q}.$$

Comme K_q, K_α sont des nombres positifs, $(b''(s) - K_q) \leq 0, \forall s \in [s^0, s^1]$, alors $(LB(s) - LB_q(s))$ est concave sur $[s^0, s^1]$. Les minimums aux extrémités de l'intervalle, donc $LB(s) - LB_q(s) \geq 0, \forall s \in [s^0, s^1]$ qui implique que $LB(s)$ est meilleure que $LB_q(s)$ sur l'intervalle $[s^0, s^1]$.

Second cas : $K = K_q \geq K_\alpha$

$$(LB(s) - LB_q(s))'' = \frac{K_q b''(s) + K_\alpha K_q - K_q K_\alpha - K_q K_q}{K_\alpha + K_q} = (b''(s) - K_q) \frac{K_q}{K_\alpha + K_q}$$

de la même manière comme ci-dessus, nous prouvons que $LB(s)$ est meilleure que $LB_q(s)$. □

Exemple 3.12.1. On considère un simple exemple, $b(s) = \sin s, s \in [0, 2\pi]$ tel montré dans figure1. Pour cet exemple, nous comparons $LB(s)$ avec $LB_\alpha(s)$ et $LB_q(s)$. Nous avons

$$LB_\alpha(s) = \sin s - \frac{1}{2}s(2\pi - s), \quad LB_q(s) = -\frac{1}{2}s(2\pi - s) \text{ et } LB(s) = \frac{1}{2}LB_\alpha(s).$$

Pour tout s dans $[0, 2\pi]$, $LB''_\alpha(s) = -\sin s + 1$ est positif alors est convexe sur cet intervalle, donc $LB_\alpha(s) \leq 0, \forall s \in [0, 2\pi]$. $LB''_q(s) = 1$, alors $LB_q(s)$ est convexe, donc $LB_q(s) \leq 0, \forall s \in [0, 2\pi]$.

- Nous avons $LB(s) = \frac{1}{2}LB_\alpha(s) \geq LB_\alpha(s)$. Comme $LB_\alpha(s) \leq 0$ alors $LB(s)$ est meilleure que $LB_\alpha(s)$ sur $[0, 2\pi]$.
- $(LB(s) - LB_q(s))'' = \frac{1}{2}(-\sin s + 1) - 1 = \frac{1}{2}(-\sin s - 1) \leq 0, \forall s \in [0, 2\pi]$ alors $(LB(s) - LB_q(s))$ est concave sur $[0, 2\pi]$ alors $(LB(s) - LB_q(s)) \geq 0, \forall s \in [0, 2\pi]$, donc $LB(s) \geq LB_q(s), \forall s \in [0, 2\pi]$ i.e. $LB(s)$ est meilleure que $LB_q(s)$.

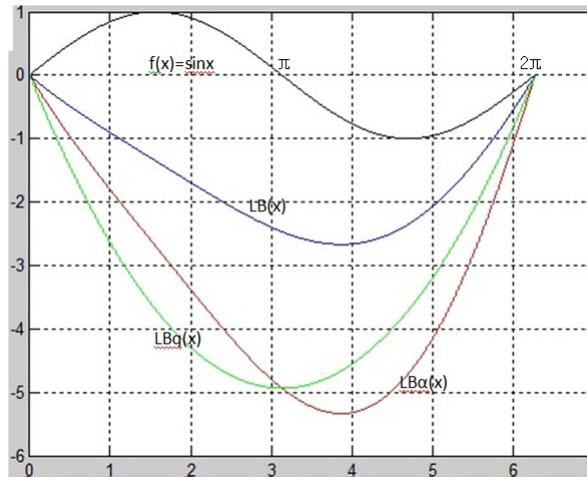


Figure1 : Comparaison de la nouvelle fonction borne inférieure $LB(x)$ (bleue) avec les deux fonctions bornes inférieures $LB_\alpha(x)$ (rouge) et $LB_q(x)$ (vert) pour la fonction $\sin(x)$ (noir).

3.12.1 Test convexité/concavité

Afin d'accélérer la convergence de notre algorithme branch and bound, le test convexité/concavité est utilisé. Nous donnerons ici une description du test Convexité/concavité. A l'itération k nous calculons K_α^k et K_q^k sur l'intervalle $[a_k, b_k]$ par utilisation des inégalités suivantes $K_\alpha^k \geq \max\{0, -b''(s)\}$, et $K_q^k \geq \max\{0, b''(s)\}$ respectivement.

- i) Si $K_\alpha^k = 0$, alors $-b''(s) \leq 0, \forall s \in [a_k, b_k]$, et b est convexe sur l'intervalle $[a_k, b_k]$, toute recherche locale donne le minimum global sur cet intervalle.
- ii) Si $K_q^k = 0$, alors $b''(s) \leq 0, \forall s \in [a_k, b_k]$ alors b est concave sur l'intervalle $[a_k, b_k]$ et ses minimums globaux sont rattachés aux extrémités de cet intervalle.
- iii) Si $K_\alpha^k = K_q^k = 0$ alors $b(s)$ est affine sur $[a_k, b_k]$ et ses minimums globaux sont rattachés aux extrémités de cet intervalle.

Remarque 3.12.1. Si le test convexité/concavité est satisfait pour tous les sous intervalles alors l'algorithme s'arrête.

Exemple 3.12.2. Nous considérons de nouveau le même exemple, $b(s) = \sin s, s \in [0, \pi]$.

On a $K_\alpha = 1, K = 1$, et $K_q = 0$.

$K_q = 0$ alors b est concave sur $[0, \pi]$, donc ses minimums globaux rattachés aux extrémités de cet intervalle (i.e. à 0 et π).

Pour cet exemple le test convexité/concavité nous permet d'arrêter l'algorithme à la première itération et de trouver le minimum global alors que les deux méthodes dans [11] et [51] ne peuvent le faire à la première itération (voir figure2).

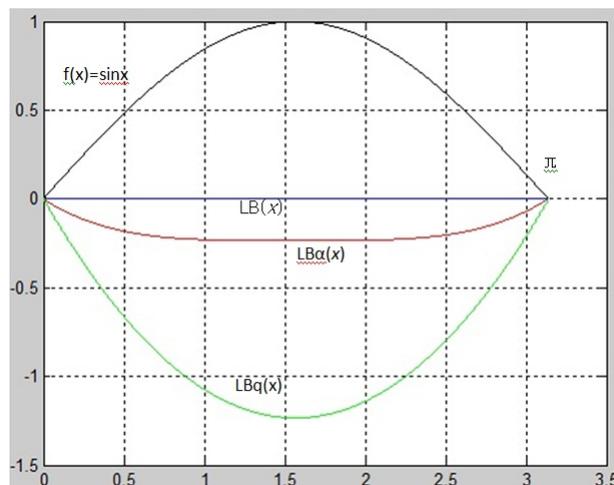


Figure2 : La nouvelle fonction borne inférieure $LB(x)$ (bleue), la αBB fonction borne inférieure $LB_\alpha(x)$ (rouge), la quadratique fonction borne inférieure $LB_q(x)$ (vert) et la fonction $\sin(x)$ (noir).

3.13 Algorithmes et sa convergence

Nous présentons maintenant notre algorithme branch and bound

3.13.1 Algorithme

1. 1.Étape Initialisation :

- i) Soit ε un nombre donné suffisamment petit, soit $[s^0, s^1]$ l'intervalle initial, calculons K_α^0 et K_q^0 tel que :
 $K_\alpha^0 \geq \max\{0, -b''(s)\}, \forall s \in [s^0, s^1]$, et $K_q^0 \geq \max\{0, b''(s)\}, \forall s \in [s^0, s^1]$ avec la méthode d'analyse des intervalles par exemple.
- ii) Test convexité/concavité
 Si $K_\alpha^0 = 0$ stop b est convexe, toute recherche locale donne la solution optimale.
 Si $K_q^0 = 0$ stop b est concave, la solution optimale est rattachée aux extrémités de $[s^0, s^1]$
- iii) Soit $k := 0; T^0 = [a_0, b_0]; M := T^0$
- iv) résolvons le problème convexe $\min \{LB_0(s), s \in T^0\}$ pour obtenir la solution optimale s_0^* .
- v) Soit $UB_0 := \min \{b(a_0), b(b_0), b(s_0^*)\} = b(\bar{s}^0)$, $LB_0 = LB(T^0) := LB^0(s_0^*)$.

2. Étape Itération. Tant que $UB_k - LB_k \geq \varepsilon$ faire

2.1 Soit $T^k = [a_k, b_k] \in M$ l'intervalle tel que $LB_k = LB(T^k)$

2.2 Scindons T^k en deux intervalles $T_1^k = [a_k, s_k^*]; T_2^k = [s_k^*, b_k]$

Soit $T_1^k := [a_k^1, b_k^1]$ et $T_2^k := [a_k^2, b_k^2]$

2.3 Pour $i = 1, 2$ faire

a Test Convexité/concavité

- Calculons K_α^{ki} et K_q^{ki} sur T_i^k .
- Si $K_\alpha^{ki} = 0$, b est convexe, toute recherche locale donne la solution optimale s_{ki}^* sur T_i^k , alors mise à jour $LB(T_i^k) = UB(T_i^k) = b(s_{ki}^*)$ et aller à c.)
- Si $K_q^{ki} = 0$, b est concave sur T_i^k , alors mise à jour $LB(T_i^k) = UB(T_i^k) = \min\{b(a_k^i), b(b_k^i)\}$ et aller à c.)

b. Soit $T_i^k = [a_k^i, b_k^i]$ Calculons $LB^{ki}(s)$

Soit s_{ki}^* la solution du problème convexe $\min \{LB^{ki}(s), s \in T_i^k\}$

c. Remplir M par les intervalles $T_i^k : M \leftarrow M \cup \{T_i^k : UB_k - LB(T_i^k) \geq \varepsilon, i = 1, 2\} \setminus \{T^k\}$

d. mise à jour $UB_{k+1} := \min\{UB_k, b(a_k^i), b(b_k^i), b(s_{ki}^*)\} := b(\bar{s}^k)$

2.4 mise à jour $LB_{k+1} := \min\{LB(T) : T \in M\}$

2.5 Supprimer de M tous les intervalles T tel que $LB(T) > UB_k - \varepsilon$.

2.6 Soit $k := k + 1$

2.7 Fin tant que

3. Sortie : \bar{s}^k est solution ε -optimale de (P) .

3.13.2 Convergence

Le théorème suivant montre la convergence de notre branch and bound algorithme.

Théorème 3.13.1. *La séquence $\{\bar{s}^k\}$ générée par l'algorithme converge vers la solution optimale du problème (P).*

Démonstration. Si l'algorithme s'arrête à l'itération k soit par le test convexité/concavité ou la condition d'arrêt $UB_k - LB_k < \varepsilon$ alors on obtient une exacte ou ε - solution optimale.

Si l'algorithme est infini alors il génère une séquence infinie $\{T^k\}$ d'intervalles dont les longueurs h_k décroissent vers zéro, alors cette séquence $\{T^k\}$ réduite à un singleton, nous devons voir que $\lim_{k \rightarrow \infty} (UB_k - LB_k) = 0$.

On a

$$0 \leq UB_k - LB_k = b(\bar{s}^k) - LB(s_k^*) =$$

$$b(\bar{s}^k) - \frac{K_q b(s_k^*) + K_\alpha L_h b(s_k^*)}{K_\alpha + K_q} + \frac{K_\alpha K_q}{2(K_\alpha + K_q)} (s_k^* - a_k)(b_k - s_k^*) =$$

$$\frac{K_q(b(\bar{s}^k) - b(s_k^*)) + K_\alpha(b(\bar{s}^k) - L_h b(s_k^*)) + \frac{K_\alpha K_q}{2}(s_k^* - a_k)(b_k - s_k^*)}{K_\alpha + K_q}.$$

Nous avons les inégalités suivantes :

- i) $K_q(b(\bar{s}^k) - b(s_k^*)) = K_q b'(\xi_1^k)(\bar{s}^k - s_k^*) \leq K_q C_1 (b_k - a_k)$, with $C_1 \geq |b'(\xi_1^k)| \geq 0$, ξ_1^k entre \bar{s}^k et s_k^* (i.e. par théorème de la moyenne).
- ii) $K_\alpha(b(\bar{s}^k) - L_h b(s_k^*)) \leq K_\alpha |b(\bar{s}^k) - L_h b(s_k^*)| \leq K_\alpha C_2 (b_k - a_k)$ (i.e. voir [35]) avec $C_2 > 0$.
- iii) $\frac{K_\alpha K_q}{2}(s_k^* - a_k)(b_k - s_k^*) \leq \frac{K_\alpha K_q}{8}(b_k - a_k)^2$

alors

$$0 \leq UB_k - LB_k \leq (b_k - a_k) \frac{K_q C_1 + K_\alpha C_2 + \frac{K_\alpha K_q}{8}(b_k - a_k)}{K_\alpha + K_q} \rightarrow 0 \text{ quand } k \rightarrow \infty \text{ (i.e. } \{T^k\} = \{[a_k, b_k]\} \text{ réduit à un singleton), donc la séquence } \{\bar{s}^k\} \text{ converge vers la solution optimale du problème (P).}$$

□

3.14 Résultats numériques

Nous résolvons un ensemble de problèmes tests trouvé dans [45] et comparons notre méthode avec la méthode présentée dans [45] en termes de nombre de sous intervalles. L'environnement expérimental est implémenté autour du programme MATLAB et exécuté sur un portable Dell avec la configuration : Intel Core I3 CPU M370 at 2.40 GHz et 4GB RAM.

Dans (Table 3.2, Table 3.3), nous reportons les résultats de la performante comparaison de l'algorithme proposé BB et l'algorithme donné dans [45].

- **NbF** est le nombre de sous intervalles utilisé dans [45].
- **GoF** est l'optimum global trouvé dans [45].
- **NbO** est le nombre de sous intervalles utilisé dans notre branch and bound algorithme.
- **GoO** est l'optimum global trouvé par notre branch and bound algorithme.

- **GsO** est les solutions globales trouvées par notre branch and bound algorithme.

Avec notre branch and bound algorithme proposé, cela nous a permis de trouver toutes les solutions optimales pour tous les problèmes et le nombre de sous intervalles dans notre algorithme utilisé moins important que celui utilisé dans [45] pour chaque fonction, de plus pour le problème 36, la solution trouvée dans [45] est fautive, alors qu'avec notre algorithme, nous trouvons la solution optimale.

3.15 Conclusion

Nous avons proposé dans ce travail une nouvelle fonction borne inférieure dans l'algorithme branch and bound pour l'optimisation globale de problèmes unidimensionnels. Cette nouvelle fonction borne inférieure est meilleure que la fonction borne inférieure classique développée dans la méthode α BB et la fonction borne inférieure quadratique développée dans [51]. Les résultats numériques expérimentaux montrent l'efficacité de notre branch and bound algorithme proposé. Les extensions pour notre algorithme pour le cas multidimensionnel sont en cours de progressions et réalisations.

TABLE 3.2 – Comparaison en termes de sous intervalles entre notre BB algorithme avec l’algorithme donné dans [45].

Fct	NbF	GoF	NbO	GoO	GsO	Obs
					7.853981522644390	
1	16	-1	4	-1	14.137166777212814	
					20.420352051661489	
2	8	-1	2	-1	6.283185305508640	
3	1	0	1	0	0.99999157232	
4	64	-17.58287	14	-17.582871815180280	6.894525986331936	
5	1024	-0.020903	1	-0.020903	0.06781274	
6	8	-0.952897	7	-0.952896792547349	2.839347106361496	
	8	-6.262872	1	-6.262872094177839	6.920063241795233	
8	16	0.077590	2	0.077589685038973	0.902209774288853	
9	16	0.211315	1	0.211314612658343	0.224882434878652	
10	16	-0.478362	3	-0.478361865691453	0.724896896650575	
11	4	-5.815675	2	-5.815674542978210	5.872866367472354	
12	8	-7.047444	3	-7.047444074140848	5.134338369777209	
13	8	-4.60138	3	-4.601307546066266	5.199786838870376	
14	4	-0.14110	2	-0.141100487712490	0.408237030060961	
15	16	-0.870885	6	-0.870885482641672	4.858047188917565	
16	16	-9.031249	7	-9.031249441398060	5.791796698749343	
17	1	0.475689	1	0.475688626380756	-0.787891419247242	
					3.141584237231236	
18	8	0	3	0	6.283180830887630	
19	64	-1	2	-1	-0.000044732677344	
20	1	1	1	1	0.000000000000000	
21	16	1	2	1	0.000055473416505	
22	4	-0.918397	2	-0.918397340885203	3.251079686033197	
23	64	-0.824239	8	-0.824239398456395	-0.679575338135825	
24	4	-0.027864	3	-0.027864070194811	3.926986265091935	
					2.094394271217541	
25	8	3.5	3	3.5	4.188791036091182	
					5.759584452678632	
26	8	0.367879	4	0.3678794411	3.665190847312067	
					1.570799741097562	
					5.006390855245968	
27	8	-0.451388	4	-0.45138781886	1.864797813097123	

TABLE 3.3 – Comparaison en termes de sous intervalles entre notre BB algorithme avec l’algorithme donné dans [45].

Fct	NbF	GoF	NbO	GoO	GsO	Obs
					4.712397606983803	
28	8	-1	4	-1	10.995573997239797	
					17.278759613821912	
29	2	-0.410135	2	-0.410315206527357	3.862077972154709	
30	16	-0.718282	5	-0.718281789521809	2.617937458512833	
					0, 212206487; 0, 090945688	
32	1024	-1	8	-1	0, 057874517; 0, 042441322	
					0, 033506303; 0, 027679122	
					0, 023578509; 0, 020536123	
33	32	-12.03125	19	-12.031249441334111	-6.774573824849375	
34	4	-0.535534	3	-0.535533905932071	2.414211465942532	
36	1	-0.35	1	5.050000000093115	2.000012866129532	5.050000000093115
37	4	-32.78126	2	-32.781261290208491	0.713679797555900	
38	8	7	6	7	2 .9999917166	
					1.381976898751345	
39	16	-1	5	-1	3.618035944847370	
40	4	-89	3	-89	2.000058945943136	

Pour clore ce chapitre, on exhibe des exemples à résoudre en optimisation globale et des cas réels comme dans ([96]).

Exemple 3.15.1. Soit le problème suivant :

$$\left\{ \begin{array}{l}
 -8x_1 + 9x_2 + (x_1 \log(x_1) + x_2 \log(x_2) + (x_1 + x_2) \log(x_1 + x_2)) \rightarrow \min \\
 \exp(x_1) + x_1 + 3x_2 \leq 9 \\
 -x_1 + 2x_2 \leq 7 \\
 2x_1 - 3x_2 \leq 3 \\
 -2x_1 - 4x_2 \leq -5 \\
 2x_1^2 - x_1x_2 + 2x_2^2 - 8x_1 + \log(x_1 + x_2) \geq 0 \\
 0.4 \leq x_1 \leq 5.4 \\
 0.4 \leq x_2 \leq 5.4 \\
 (x_1, x_2)' \in \mathbb{R}^2.
 \end{array} \right.$$

(3.38)

Ce problème est assez intéressant à résoudre car celui-ci n’est pas convexe (très simple à vérifier). Donc nous pouvons le situer dans le cadre des problèmes en optimisation globale.

Le problème peut être pris en charge par la méthode DC mixée par Branch and Bound par exemple. En outre, on peut se ramener à sa solution optimale à l'aide de la méthode de l'annexion polyédrale ou la méthode d'approximation externe car ce problème possède la spécificité d'être un problème en optimisation globale type anti-convexe (si $\log(x_1 + x_2)$ et x_1x_2 convexifiés). Mais dans notre cas on peut aussi utiliser une méthode hybride (convexification + discrétisation) ([16],[71]). Ainsi on reformule le problème pour résolution, on utilise :

$$\left\{ \begin{array}{l} x_i = 0.4 + \tilde{x}_i \quad , i = 1..2 \\ \tilde{x}_i = \epsilon_i \sum_{\xi=1}^{\xi(i)} 2^{\xi-1} t_{i\xi} \quad , i = 1..2 \\ y_{ij} = 0.4x_j + 0.4\tilde{x}_i + \epsilon_i \sum_{\xi=1}^{\xi(i)} 2^{\xi-1} z_{ij\xi} \quad , 1 \leq i \leq j \leq 2 \\ z_{ij\xi} \leq 5t_{i\xi} \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \leq \tilde{x}_j \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \geq \tilde{x}_j - 5(1 - t_{i\xi}) \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \geq 0 \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ t_{i\xi} \in \{0, 1\} \quad , i = 1..2 \quad , \xi = 1..\xi(i) \\ \xi(i) = \lceil (\log_2(\frac{5}{\epsilon_i} + 1)) \rceil \quad , i = 1..2 \\ 0.4 \leq x_1 \leq 5.4 \\ 0.4 \leq x_2 \leq 5.4 \\ y_{ij} = x_i x_j \quad , 1 \leq i \leq j \leq 2. \end{array} \right.$$

(3.39)

Cela résout le problème et on peut le confirmer par les logiciels professionnels comme : Lingo, cplex, mapple, matlab ou autres ...etc.

Dans ([71], [57] P525) des cas résolubles de manière similaire.

Exemple 3.15.2. Soit le problème suivant aussi complexe et plus que le précédent :

$$\left\{ \begin{array}{l} 5x_1^2 - 3x_1x_2 - 3x_2^2 - x_1^4 - x_2^4 - \exp(x_1 + x_2) \rightarrow \max \\ -4x_1^2 - 3x_2^2 + 4x_1x_2 \leq 1.6 \\ 6x_1^2 + 7x_2^2 - 2x_1x_2 \leq 300 \\ -3x_1^2 - 2x_2^2 - 7x_1x_2 \leq -150 \\ 10x_1 - 6x_2 \leq 30 \\ 3 \leq x_1 \leq 9 \\ 3 \leq x_2 \leq 9 \\ (x_1, x_2)' \in \mathbb{R}^2. \end{array} \right.$$

(3.40)

Ainsi on reformule le problème à des fins de résolution, en utilisant :

$$\left\{ \begin{array}{l} x_i = 3 + \tilde{x}_i \quad , i = 1..2 \\ \tilde{x}_i = \epsilon_i \sum_{\xi=1}^{\xi(i)} 2^{\xi-1} t_{i\xi} \quad , i = 1..2 \\ y_{ij} = 3x_j + 3\tilde{x}_i + \epsilon_i \sum_{\xi=1}^{\xi(i)} 2^{\xi-1} z_{ij\xi} \quad , 1 \leq i \leq j \leq 2 \\ z_{ij\xi} \leq 6t_{i\xi} \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \leq \tilde{x}_j \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \geq \tilde{x}_j - 6(1 - t_{i\xi}) \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \geq 0 \quad , 1 \leq i \leq j \leq 2 \quad , \xi = 1..\xi(i) \\ t_{i\xi} \in \{0, 1\} \quad , i = 1..2 \quad , \xi = 1..\xi(i) \\ \xi(i) = \lceil (\log_2(\frac{6}{\epsilon_i} + 1)) \rceil \quad , i = 1..2 \\ 3 \leq x_1 \leq 9 \\ 3 \leq x_2 \leq 9 \\ y_{ij} = x_i x_j \quad , 1 \leq i \leq j \leq 2. \end{array} \right.$$

(3.41)

Cela résout le problème et on peut le confirmer par les logiciels professionnels comme : Lingo, cplex, mapple, matlab ou autres ...etc.

Exemple 3.15.3. Soit le problème suivant :

$$(3.42) \quad \left\{ \begin{array}{l} -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 - 6 \rightarrow \min \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \\ 0 \leq x_3 \leq 2 \\ (x_1, x_2, x_3)' \in \mathbb{R}_+^3. \end{array} \right.$$

Exemple 3.15.4. Et soit le problème suivant :

$$(3.43) \quad \left\{ \begin{array}{l} -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 - 6 \rightarrow \min \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \leq 5 \\ -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \geq 2 \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \\ 0 \leq x_3 \leq 2 \\ (x_1, x_2, x_3)' \in \mathbb{R}_+^3. \end{array} \right.$$

Ainsi on reformule à des fins de résolution, en utilisant :

$$\left\{ \begin{array}{l} x_i = \tilde{x}_i \quad , i = 1..3 \\ \tilde{x}_i = \epsilon_i \sum_{\xi=1}^{\xi(i)} 2^{\xi-1} t_{i\xi} \quad , i = 1..3 \\ y_{ij} = \epsilon_i \sum_{\xi=1}^{\xi(i)} 2^{\xi-1} z_{ij\xi} \quad , 1 \leq i \leq j \leq 3 \\ z_{ij\xi} \leq 2t_{i\xi} \quad , 1 \leq i \leq j \leq 3 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \leq \tilde{x}_j \quad , 1 \leq i \leq j \leq 3 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \geq \tilde{x}_j - 2(1 - t_{i\xi}) \quad , 1 \leq i \leq j \leq 3 \quad , \xi = 1..\xi(i) \\ z_{ij\xi} \geq 0 \quad , 1 \leq i \leq j \leq 3 \quad , \xi = 1..\xi(i) \\ t_{i\xi} \in \{0, 1\} \quad , i = 1..3 \quad , \xi = 1..\xi(i) \\ \xi(i) = \lceil (\log_2(\frac{2}{\epsilon_i} + 1)) \rceil \quad , i = 1..3 \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \\ 0 \leq x_3 \leq 2 \\ y_{ij} = x_i x_j \quad , 1 \leq i \leq j \leq 3. \end{array} \right.$$

(3.44)

Cela résout le problème et on peut le confirmer par les logiciels professionnels comme : Lingo, cplex, mapple, matlab ou autres ...etc.

Exemple 3.15.5. Soit le problème suivant :

$$\left\{ \begin{array}{l} 5x_1^{0.9} x_2^{-1.5} x_3^{-3} + 7x_4^{-0.3} x_5^{2.6} + 3x_6^{-1.8} x_7^{-0.5} x_8^1 + x_1^4 + x_2^4 + \exp(x_1 + x_2) + x_1^6 + x_2^6 \rightarrow \min \\ 8.2x_1^{-3.8} x_2^{2.2} x_3^{4.3} + 0.2x_4^{-0.7} x_5^{-1.6} + 2.2x_6^{4.3} x_7^{-1.9} x_8^{8.5} \leq 1 \\ 2x_1^{2.3} x_2^{1.7} x_3^{4.5} \leq 1 \\ 1.6x_4^{-2.1} x_5^{0.4} \leq 1 \\ 7.2x_6^{4.5} x_7^{-2.7} x_8^{-0.6} \leq 1 \\ 1.1x_1^{1.6} x_2^{0.4} x_3^{-3.8} \leq 1 \\ 1.7x_4^{5.4} x_5^{1.3} \leq 1 \\ 1.3x_6^{-1.1} x_7^{7.3} x_8^{-5.6} \leq 1 \\ (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)' \in \mathbb{R}_+^{*8}. \end{array} \right.$$

(3.45)

Ce problème est non convexe. Une manière d'opérer : est tout d'abord une convexification. Grâce aux transformations ([73]), puis on applique par exemples des itérations de l'algorithme d'annexion polyédrale ou autres. Dans ([73] P400), un problème similaire résolu.

Exemple 3.15.6. Soit le problème suivant :

$$\left\{ \begin{array}{l}
 3x_1^{-0.9}x_2^{-1.6}x_3^{-4} + 2x_4^{-0.3}x_5^{3.6} + 5x_6^{-1.9}x_7^{-0.6}x_8^{-1} + (x_1 \log(x_1) + x_2 \log(x_2) + (x_1 + x_2) \log(x_1 + x_2)) \rightarrow \min \\
 4.5x_1^{-3.8}x_2^{-2.2}x_3^8 + 3.5x_4^{-0.7}x_5^{-1.6} + 2.5x_6^{-4.5}x_7^{-1.9}x_8^{-1.2} \leq 2 \\
 5x_1^{-1.5}x_2^{-1.9}x_3^{5.5} \leq 2 \\
 6x_4^{-3.2}x_5^{-1.2} \leq 2 \\
 6.2x_6^{4.5}x_7^{-2.5}x_8^{-0.6} \leq 2 \\
 x_1^{5.5}x_2^{-0.5}x_3^{-3.5} \leq 2 \\
 2.5x_4^{5.5}x_5^{-1.4} \leq 2 \\
 7x_6^{-1.2}x_7^{-7.3}x_8^{-5.4} \leq 2 \\
 (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)' \in \mathbb{R}_+^{*8}.
 \end{array} \right.$$

(3.46)

Ce problème est convexe. Donc toute méthode locale citée plus haut est utilisable pour avoir l'optimum global.

Exemple 3.15.7. Soit le problème suivant :

$$\left\{ \begin{array}{l}
 \frac{-x_1^2+5x_1-2x_1x_2+6x_2-4x_1x_3+7x_3-x_2^2-4x_2x_3-x_3^2-6}{x_1^2+x_2^2+1+x_3^2+x_1x_2} \rightarrow \min \\
 -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \leq 5 \\
 -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \geq 2 \\
 0 \leq x_1 \leq 2 \\
 0 \leq x_2 \leq 2 \\
 0 \leq x_3 \leq 2 \\
 (x_1, x_2, x_3)' \in \mathbb{R}_+^3.
 \end{array} \right.$$

(3.47)

Remarque 3.15.1. Pour cet dernier exemple :

- Après convexification, On peut aussi résoudre les cas de sommes de ratios.
- On peut généraliser par exemples les indices de puissances dans notre cas.

- On peut étendre le nombre de variables.
- ►►► **Après convexification, pour les cas discrets**
- On peut prendre cela en charge par les algorithmes classiques, comme Gambini.
- On peut prétendre aux cas discrets et multi-Objectifs auxquels cas la méthode Abassi-Moulai.
- On peut prétendre aux cas de programmation bi-niveaux (multi-Objectifs) (sur des ensembles efficaces par exemples).

Et donc on peut en déduire la résolution :

Exemple 3.15.8. Du problème suivant :

$$\left\{ \begin{array}{l}
 \frac{-x_1^2+5x_1-2x_1x_2+6x_2-4x_1x_3+7x_3-x_2^2-4x_2x_3-x_3^2-6}{x_1^2+x_2^2+1+x_3^2+x_1x_2} + \frac{-x_1^2+5x_1-2x_1x_2+6x_2-4x_1x_3+7x_3-x_2^2-4x_2x_3-x_3^2-6}{x_1^2+x_2^2+1+x_3^2} \rightarrow \min \\
 -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \leq 5 \\
 -x_1^2 + 5x_1 - 2x_1x_2 + 6x_2 - 4x_1x_3 + 7x_3 - x_2^2 - 4x_2x_3 - x_3^2 \geq 2 \\
 0 \leq x_1 \leq 2 \\
 0 \leq x_2 \leq 2 \\
 0 \leq x_3 \leq 2 \\
 (x_1, x_2, x_3)' \in \mathbb{R}_+^3.
 \end{array} \right. \tag{3.48}$$

CHAPITRE 4

Optimisation de systèmes dynamiques en contrôle optimal.

Consulter ([8, 98, 48, 23, 87, 72, 76].)

Théorie, numérique et informatique (étude de cas)

4.1 Introduction

Nous avons bâti un logiciel ("Matrix") autour entre autres des résolutions pour les méthodes suivantes :

(**De plus, il s'agit donc de présenter le logiciel ("Matrix")**).

- ♣ La résolution de problème en contrôle optimal /cas discret/primal adaptée/support.
- ♣ La résolution de problème en contrôle optimal /cas continu/primal adaptée/support.
- ♣ La résolution de problème en contrôle optimal /cas discret/dual adaptée/support.
- ♣ La résolution de problème en contrôle optimal /cas continu/dual adaptée/support.
- ♣ La résolution de problème Min-Max (contraintes simples) /primal adaptée/support.
- ♣ La résolution de problème Min-Max (contraintes simples) /dual adaptée/support.
- ♣ La résolution de problème Min-Max (contraintes généralisées) /primal adaptée/support.
- ♣ La résolution de problème Min-Max (contraintes généralisées) /dual adaptée/support.
- ♣ La résolution de problème Min-Max /primal adaptée/support/grandes dimensions.
- ♣ La résolution de problème Min-Max /dual adaptée/support/grandes dimensions.
- ♣ La résolution de problème linéaire, forme standard (par l'algorithme Simplexe).
- ♣ La résolution de problème linéaire, forme généralisée (par l'algorithme Simplexe).
- ♣ La résolution de systèmes linéaires.
- ♣ La résolution de problème linéaire Multi-Objectifs discret.
- ♣ La résolution de problème quadratique convexe Multi-Objectifs discret.

♣ La résolution de problème quadratique non convexe Multi-Objectifs discret.

Remarque 4.1.1. Concernant ce logiciel :

♣ Peut être adapté pour le cas Multi-Objectifs.

♣ Présenté et sélectionné dans des conférences Nationales et Internationales (CMA2012 DZ/ATIM2012 Turkish - DZ/ISORAP2013 Maroc).

♣ Outil efficace pour la recherche et la pédagogie.

♣ Peut être développé et orienté vers plusieurs autres méthodes.

♣ Tous les résultats numériques présentés dans ce 4^{ème} chapitre sont issus (proviennent) exclusivement de ce Logiciel ("Matrix").

♣ Ce logiciel en Guise de 3^{ème} Contribution.

Une 4^{ème} Contribution en cours de rédaction (Optimisation Globale : Aspect Théorique Technique et Numérique) - Soumission à un Journal "Optimisation Globale".

Dans ce chapitre nous présentons une méthode de résolution pour un système dynamique à travers la méthode adaptée. Nous présentons principalement et essentiellement des résultats théoriques ainsi que des exemples d'application pour le cas continu et le cas discret.

4.2 Le cas discret

Pour cette partie de ce document, nous présentons une méthode de résolution d'un système dynamique pour le cas discret.

4.2.1 Position du problème

Considérons le système dynamique suivant :

$$J(u) = \min_{k \in K} (c'_k x(t^*) + \alpha_k) \longrightarrow \max_u \quad (4.1)$$

$$\dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = x_0, \quad (4.2)$$

$$Hx(t^*) = g, \quad (4.3)$$

$$f_* \leq u(t) \leq f^*, \quad t \in T = [t_0, t^*], t_0 = 0. \quad (4.4)$$

Où :

- ◇ $x(t)$ est un n -vecteur décrivant la trajectoire du système à l'instant t ,
- ◇ $u(t)$ la commande d'entrée à l'instant t du système,
- ◇ $f_*(t) \leq u(t) \leq f^*(t)$, $t \in [t_0; t^*]$,
- ◇ $f_*(t)$ et $f^*(t)$ des fonctions définies sur $T = [t_0; t^*]$,

- ◇ A une $n \times n$ matrice caractérisant l'état du système,
- ◇ b un n -vecteur,
- ◇ x_0 la position initiale du système à l'instant $t = t_0$,
- ◇ le système est caractérisé par un signal de sortie pour $t = t^*$:

$$Hx(t^*) = g,$$

- ◇ H est une $m \times n$ matrice avec $\text{rang}H = m \leq n$, g un m -vecteur,
- ◇ $J(u) = \min_{k \in K}(c'_k x(t^*) + \alpha_k) \rightarrow \max_u$ est un critère de qualité,
- ◇ $I = \{1 \dots m\}$: L'ensemble des indices des lignes,
- ◇ $J = \{1 \dots n\}$: L'ensemble des indices des colonnes,
- ◇ $K = \{1 \dots p\}$: L'ensemble des indices de la fonctionnelle,
- ◇ c_k des n vecteurs, α_k des scalaires, $k \in K$.

Définition 4.2.1. Une commande u est dite impulsive sur l'intervalle $[t_0, t^*]$ si :

$$u(t) = u_i = \text{constante}, \quad t \in [\tau_i, \tau_{i+1}], \quad i = (0, \dots, N), \quad \tau_0 = t_0, \quad \tau_{N+1} = t^*, \quad \tau_{i+1} - \tau_i = h > 0,$$

h est le pas de quantification.

En utilisant la formule de Cauchy, la solution du système dynamique (4.2) s'écrit sous la forme :

$$x(t) = F(t)[x_0 + \int_{t_0}^t F^{-1}(\tau)bu(\tau)d\tau], \quad (4.5)$$

où $F(t) = \exp(At)$, $t \in T$, est la solution du système

$$\begin{cases} \dot{F} = AF, \\ F(t_0) = Id, \quad t \in T. \end{cases}$$

Comme la commande u est impulsive, l'équation (4.5) prend la forme suivante :

$$x(i+1) = x(\tau_{i+1}) = F(h)[x(i) + \int_0^h F^{-1}(\tau)bu(i)d\tau],$$

où

$$u(i) = u(\tau_i), \quad \tau_i \in T, \quad i = (0, \dots, N), \quad T = \bigcup_{i=0}^N [\tau_i, \tau_{i+1}], \quad \tau_0 = t_0, \quad \tau_{N+1} = t^*, \quad \tau_{i+1} - \tau_i = h > 0.$$

En posant $D = \exp(Ah) = \sum_{n=0}^{\infty} \frac{(Ah)^n}{n!}$ et $d = D \cdot \int_0^h F^{-1}(\tau)b d\tau$, on aboutit au système suivant :

$$\begin{cases} \min_{k \in K}(c'_k x(t^*) + \alpha_k) \rightarrow \max_u, \\ x(t+h) = Dx(t) + du(t), & x(t_0) = x_0, \\ Hx(t^*) = g, \\ f_*(t) \leq u(t) \leq f^*(t), & t \in T = [t_0, t^*]. \end{cases} \quad (4.6)$$

Avec $x(t^*)$ vérifiant la propriété suivant :

$$x(t^*) = D^{\frac{t^*}{h}} x_0 + \sum_{t \in T} D^{\frac{t^*-t}{h}-1} du(t).$$

Donc cela va nous permettre d'établir une autre formulation équivalente du problème (4.6) comme suit :

$$\begin{cases} J(u) = (c'_k \sum_{t \in T} D^{\frac{t^*-t}{h}-1} du(t) + c'_k D^{\frac{t^*}{h}} .x_0 + \alpha_k) \longrightarrow \max_u, \\ H \sum_{t \in T} D^{\frac{t^*-t}{h}-1} du(t) = g - HD^{\frac{t^*}{h}} x_0, \\ f_*(t) \leq u(t) \leq f^*(t), \quad t \in T = [t_0, t^*]. \end{cases} \quad (4.7)$$

Où

$$x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0.$$

4.2.2 Notion de commandabilité

Un processus de contrôle décrit par l'équation (4.2) est dit commandable, si pour toute paire $x_0, x^* \in \mathbb{R}^n$, il existe une commande mesurable bornée $u(t)$ sur un intervalle fini $[t_0, t^*]$ qui ramène l'objet sur la trajectoire $x(t)$ du point $x(t_0) = x_0$ au point $x(t^*) = x^*$, (c'est-à-dire de la position initiale à la position finale).

Théorème 4.2.1. [23] *Un système de \mathbb{R}^n décrit par l'équation (4.2) est commandable si et seulement si le rang de la matrice $[b, Ab, A^2b, \dots, A^{n-1}b]$ est égal à n .*

Remarque 4.2.1. On impose au système linéaire de \mathbb{R}^n décrit par l'équation (4.2) une stabilité ($\lim A^v = 0$ quand $v \rightarrow \infty$).

Dans toute la suite, on fera hypothèse de la stabilité du système linéaire (4.2) pour $v=n$.

Définition 4.2.2. La commande $u(t)$ et sa trajectoire $x(t)$ sont admissibles, s'ils vérifient les contraintes du problème (4.7).

La commande admissible u^0 est optimal si $\max J(u) = J(u^0)$.

La commande admissible u^ε est ε -optimal si $J(u^0) - J(u^\varepsilon) < \varepsilon$.

DEFINITIONS

♣ L'ensemble $T_B \subset T / |T_B| = m$ est appelé support des contraintes et la matrice :

$D_B = (HD^{\frac{t^*-t}{h}-1}d, t \in T_B)$ matrice du support si D_B est inversible.

Avec le support T_B , on définit la matrice $\Delta(K, T)$.

$$\Delta(K, T) = (c'_k D^{\frac{t^*-t}{h}-1}d, t \in T_B) D_B^{-1} (HD^{\frac{t^*-t}{h}-1}d, t \in T) - (c'_k D^{\frac{t^*-t}{h}-1}d, t \in T).$$

✓ On construit la matrice $\Delta_f = (\Delta(K_f, T_f), e(K_f))$ avec $e(K) = 1, \forall \in K$.

$T_f \subset T - T_B$ et $K_f \subset K$ tel que $|K_f| = |T_f| + 1$. $K_H \subset K - K_f$.

✓ L'ensemble $\{K_f, T_f\} = Q_f$ est appelé support de la fonctionnelle si la matrice Δ_f est inversible.

✓ L'ensemble $Q_p = \{T_B, Q_f\}$ formé du support des Contraintes et du support de la fonctionnelle est appelé support du problème.

✓ La paire $\{ u, Q_p \}$ formée de la commande u et du support Q_p est appelée support - contrôle du problème.

Nous avons les relations suivantes :

- ♣ $\gamma'(K_f) = (0'(T_f), 1) \Delta_f^{-1}$.
- ♣ $\gamma'(K_f)e(K_f) = 1$.
- ♣ $\Delta(J_f) = 0(J_f)$.
- ♣ $\Delta(J_B) = 0(J_B)$.
- ♣ $\gamma(K_H) = 0(K_H)$.
- ♣ $\Delta(t) = \sum_{k \in K} \gamma_k \Delta_k(t)$, $\Delta(K, T) = \Delta_k(t)$, $t \in T, k \in K$.
- ♣ Le support de la fonctionnelle est régulier si $\gamma(K_f) \geq 0$.
- ♣ Le support Q_p du problème est régulier si le support Q_f de la fonctionnelle est régulier.
- ♣ Le support Q_p du problème avec $T_f = \phi$ est régulier.
- ♣ Il est aisé de déduire les cas de commande dégénérée non dégénérée.
- ♣ Dans ce qui suit nous considérons seulement des supports réguliers.

4.2.3 Accroissement de la fonctionnelle, critère d'optimalité et de suboptimalité.

On considère le support contrôle $\{u, Q_P\}$ et une autre commande admissible $\bar{u} = u + \Delta u$ et $\bar{x} = x + \Delta x$, sa trajectoire correspondante. L'accroissement de la fonctionnelle représentée par la valeur $\beta(u, Q_P)$:

$$\beta = \beta(u, Q_p) = \sum_{t \in T_H^+} \Delta(t)(u(t) - f_*(t)) + \sum_{t \in T_H^-} \Delta(t)(u(t) - f^*(t)) + \sum_{k \in K_f} \gamma_k \omega_k$$

est appelée valeur de suboptimalité du support - plan $\{u, Q_P\}$. Avec $T_H^+ = \{t \in T_H / \Delta(t) \geq 0\}$; $T_H^- = \{t \in T_H / \Delta(t) \leq 0\}$.

Théorème (critère d'optimalité)

Les Relations suivantes :

$$\begin{cases} u(t) = f^*(t) & \text{if } \Delta(t) < 0 \\ u(t) = f_*(t) & \text{if } \Delta(t) > 0 \\ f_*(t) \leq u(t) \leq f^*(t) & \text{if } \Delta(t) = 0, t \in T_H \\ \omega_k = 0 & \text{if } \gamma_k > 0, \\ \omega_k \geq 0 & \text{if } \gamma_k = 0, k \in K_f \end{cases}$$

sont suffisantes et dans le cas de la non dégénérescence, sont nécessaires pour l'optimalité du support- contrôle $\{u, Q_P\}$.

4.2.4 Critère de suboptimalité

Théorème 4.2.2. [23] Pour $\varepsilon > 0$ donné. Pour l' ε -optimalité de la commande u , il est suffisant de trouver un tel support $\{Q_P\}$ pour lequel la valeur de suboptimalité vérifie l'inégalité suivante :

$$\beta(u, Q_p) \leq \varepsilon.$$

4.2.5 Méthode de résolution

Dans ce cas, il suffit de résoudre le problème (4.7) par la méthode adaptée, une fois la commande optimale obtenue, on cherchera la trajectoire $x(t)$ correspondante à partir de l'expression suivante :

$$x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0.$$

Ce qui nous permet d'écrire l'algorithme suivante :

Algorithme de résolution

- ◇ Tester la commandabilité du système, dans le cas positif continuer.
- ◇ Sinon arrêter le processus le système n'est pas commandable.
- ◇ Tester la stabilité du système dans le cas positif, continuer.
- ◇ Sinon arrêter le processus le système n'est pas stable.
- ◇ Poser le problème (4.6) à résoudre.
- ◇ Écrire (4.6) sous forme (4.7).
- ◇ Résoudre (4.7) par la méthode adaptée.
- ◇ Calcul de la trajectoire $x(t)$ grâce à la solution obtenue et de la formule

$$x(t+h) = Dx(t) + du(t), \quad x(t_0) = x_0.$$

4.2.6 Exemple

Un exemple très détaillé, pour un cas général est résolu en [23].
Donc, en Guise de deuxième exemple.

Soit à résoudre : Avec réglage du logiciel("Matrix") à $p=2$, et donc les résultats suivants :

$$\left\{ \begin{array}{l} J(u) = \min_k (c'_k x(t^*)) \longrightarrow \max_u, \\ \dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = (1 \ 1 \ 0 \ 0 \ 1 \ 1), \\ Hx(t^*) = g, \\ |u(t)| \leq 3.2, \quad t \in T = [0, 6]. \end{array} \right.$$

Avec

$$c'_k = \begin{pmatrix} -1 & -1 & 3 & 1 & 0 & 4 \\ 1 & -2 & 3 & 2 & 0 & 1 \end{pmatrix},$$

$$\begin{aligned} \alpha'_k &= (2 \quad -2 \quad), \\ b' &= (0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad), \\ g' &= (0.16 \quad 0.1 \quad 0.2 \quad 0.3 \quad 0.6 \quad), \\ H &= \begin{pmatrix} -2 & 1 & -2 & 1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 1 & 1 \\ 1 & 1 & -2 & 1 & -2 & -2 \\ -2 & 2 & 3 & -2 & 3 & 1 \\ -3 & 0 & -2 & 1 & -1 & 0 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

On a le $\text{rang}[b, Ab, A^2b, \dots, A^{n-1}b] = 6$, donc on continue le processus. De plus $A^n = [0]$, avec $n = 6$.

Il en résulte que :

$$F(t) = \begin{pmatrix} 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 & \frac{1}{24}t^4 & \frac{1}{120}t^5 \\ 0 & 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 & \frac{1}{24}t^4 \\ 0 & 0 & 1 & t & \frac{1}{2}t^2 & \frac{1}{6}t^3 \\ 0 & 0 & 0 & 1 & t & \frac{1}{2}t^2 \\ 0 & 0 & 0 & 0 & 1 & t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$F^{-1} = \begin{pmatrix} 1 & -t & \frac{1}{2}t^2 & -\frac{1}{6}t^3 & \frac{1}{24}t^4 & -\frac{1}{120}t^5 \\ 0 & 1 & -t & \frac{1}{2}t^2 & -\frac{1}{6}t^3 & \frac{1}{24}t^4 \\ 0 & 0 & 1 & -t & \frac{1}{2}t^2 & -\frac{1}{6}t^3 \\ 0 & 0 & 0 & 1 & -t & \frac{1}{2}t^2 \\ 0 & 0 & 0 & 0 & 1 & -t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Avec un pas de quantification $h = 1$, alors

$$D = \exp(Ah) = \begin{pmatrix} 1 & 1 & 0.5 & 0.166 & 0.0416 & 0.0083333 \\ 0 & 1 & 1 & 0.5 & 0.166 & 0.0416 \\ 0 & 0 & 1 & 1 & 0.5 & 0.166 \\ 0 & 0 & 0 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

et

$$d = D \int_0^h F^{-1}(s) b ds = \begin{pmatrix} 0.2180 \\ 0.7166 \\ 1.7083 \\ 1.666 \\ 1.5 \\ 1 \end{pmatrix}.$$

Dans ce cas le problème devient :

$$\left\{ \begin{array}{l} (-37.823587483333u(0) + 6.17918411111112u(1) + 20.4319565388889u(2) + 20.1013964666667u(3) \\ +14.8541705611111u(4) + 9.85694548888889u(5) - 24.8 \\ 143.423599783333u(0) + 89.4208256888889u(1) + 53.9180507611111u(2) + 30.748608333333u(3) \\ +16.2458317388889u(4) + 8.2430543111111u(5) + 152.8) \rightarrow \max, \\ -240.163852266667u(0) - 116.991639911111u(1) - 51.0694258888889u(2) - 19.0638768666667u(3) \\ -5.6416595111111u(4) - 1.46944048888889u(5) = 244.76, \\ -226.5319183166667u(0) - 115.779146722222u(1) - 53.2763742944444u(2) - 20.8569343666667u(3) \\ -5.8541602722222u(4) - .101385344444444u(5) = 232.9, \\ 119.781919983333u(0) + 47.1958150555555u(1) + 11.8597092944444u(2) - 2.39306396666667u(3) \\ -6.22917139444445u(4) - 5.8152796555556u(5) = -116.6, \\ 50.8444711333334u(0) + 56.2666843222222u(1) + 44.6888991777778u(2) + 29.4444490333333u(3) \\ +16.8666672222222u(4) + 8.2888870777778u(5) = -66.1, \\ -459.0291039166667u(0) - 229.104119966667u(1) - 103.6791335166667u(2) - 41.2541445666667u(3) \\ -13.8291531166667u(4) - 3.9041591666667u(5) = 469, \\ -3.2 \leq u(0) \leq 3.2, \\ -3.2 \leq u(1) \leq 3.2, \\ -3.2 \leq u(2) \leq 3.2, \\ -3.2 \leq u(3) \leq 3.2, \\ -3.2 \leq u(4) \leq 3.2, \\ -3.2 \leq u(5) \leq 3.2. \end{array} \right.$$

La solution initiale est :

$$u' = \begin{pmatrix} -1.11548959967512 & 1.31095947328392 & -3.2 & 1.14896564320062 & 2.51677420660375 \\ -1.98128408649055 & & & & \end{pmatrix},$$

$$J_B = \{1, 2, 4, 5, 6\}, K_f = \{2\}.$$

La solution optimale est :

$$\bar{u} = \begin{pmatrix} -1.03458472456955 & 0.87152909073532 & -2.23007252185631 & 4.12026268108676E - 02 & 3.2 \\ -2.17908345970966 & & & & \end{pmatrix},$$

$$J_B = \{1, 2, 3, 4, 6\}, K_f = \{2\}.$$

La trajectoire $x(t)$ est donnée par la formule

$$x(t+h) = Dx(t) + du(t),$$

qui donnera :

x(0)	x(1)	x(2)	x(3)	x(4)	x(5)
1	1.8244	1.9119	1.9237	1.4045	.15856
1	.4668	-4.8178E-02	-.11307	-1.0006	-1.02608
0	-1.1007	.3821	-.76862	-1.24250	1.60587
0	-.2243	1.6590	8.1587E-02	-1.33363	1.20507
1	.4481	1.72083	-.787332	-2.11865	1.329417
1	-3.4584E-02	.83694	-1.39312	-1.3519	1.84807

4.3 Le cas continu [8, 98, 48, 72, 23, 87, 76]

Dans cette partie, nous présentons une méthode de résolution pour un système dynamique pour le cas continu.

4.3.1 Position du problème

Considérons le problème terminal de commande optimale suivant :

$$J(u) = \min_k (c'_k x(T_1) + \alpha_k) \rightarrow \max_u, \quad (4.8)$$

$$\dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = x_0, t_0 = 0. \quad (4.9)$$

$$Hx(T_1) = g, \quad (4.10)$$

$$f_* \leq u(t) \leq f^*, \quad t \in T = [t_0, T_1]. \quad (4.11)$$

Où :

- ◇ $J(u)$ est le critère de qualité,
- ◇ $A(i, j)$ est une $n \times n$ matrice caractérisant le système dynamique,
- ◇ $b(j)$ est un n -vecteur caractérisant le système dynamique,
- ◇ $x(t) \in \mathbb{R}^n$ est l'état du système,
- ◇ $H(i, j)$ est une matrice $m \times n$ de rang $m \leq n$,
- ◇ g est un m -vecteur représentant le signal de sortie au temps T_1 ,
- ◇ $I = \{1, \dots, n\}$ l'ensemble des indices lignes,
- ◇ $J = \{1, \dots, m\}$ l'ensemble des indices colonnes,
- ◇ $K = \{1, \dots, p\}$ l'ensemble des indices de la fonctionnelle,
- ◇ $u(t)$ est une commande constante par morceaux, bornée par $f_*, f^* \in \mathbb{R}$,
- ◇ $c_k = c_k(j)$ sont des n -vecteurs des coûts, α_k des scalaires, $k \in K$.

Notre objectif du problème :

$$J(u) = \min_k (c'_k x(T_1) + \alpha_k) \rightarrow \max_u.$$

4.3.2 Définitions

Dans les classes des commandes constantes par morceaux, on considère le problème suivant en contrôle optimal :

$$\begin{cases} J(u) = \min_k (c'_k x(T_1) + \alpha_k) \rightarrow \max_u \\ \frac{dx(t)}{dt} = Ax(t) + bu(t), x(0) = x_0 \\ H(x(T_1)) = g \\ d_* \leq u(t) \leq d^*; t \in [0, T_1] \end{cases}$$

$$d_* = f_*, d^* = f^*.$$

Le système dynamique sous l'effet de la commande u constante par morceaux, admet la solution provenant de la formule de Cauchy.

$$x(t) = F(t).F^{-1}(0)x(0) + \int_0^t F(t).F^{-1}(s)bu(s)ds, T_1 \geq t \text{ et } t \geq 0.$$

Avec ce contrôle (commande) constant par morceaux, nous obtenons l'expression suivante après quelques transformations :

$$\begin{cases} J(u) = \min_k (c'_k F(T_1)x_0 + \int_0^{T_1} c_k(t)u(t)dt + \alpha_k) \rightarrow \max_u \\ \int_0^{T_1} p(t)u(t)dt = g - HF(T_1)x_0 \\ d_* \leq u(t) \leq d^*, t \in [0, T_1]. \end{cases}$$

Avec $c_k(t) = c'_k F(T_1)F^{-1}(t)b$, $p(t) = HF(T_1)F^{-1}(t)b$, $q(t) = F(T_1)F^{-1}(t)b$ et $t \in [0, T_1]$.

Pour la suite nous ferons, hypothèses de stabilité et contrôlabilité de notre système dynamique.

4.3.3 Support et support-contrôle

En outre les définitions des contrôles admissibles, optimaux et ϵ -optimaux.

Dans un intervalle $[0, T_1]$, nous introduisons l'ensemble de m points isolés $T_c = \{t_j, j = 1..m\}$, T_c est appelé support des contraintes et $P_c = \{p(t), t \in T_c\}$ matrice des contraintes si P_c est inversible.

Nous construisons les ensembles des indices $T_f \subset T - T_c$, $K_f \subset K$ et tel que :

$|K_f| = |T_f| + 1$ tel que la matrice Δ_f .

Avec

$$\Delta_f = (\Delta[K_f, T_f], e[K_f])$$

$$\Delta[K_f, T_f] = (\Delta_k(t), t \in T_f, k \in K_f)$$

$$e[K_f] = (e_k = 1, k \in K_f)$$

$$\Delta_k(t) = y'_k p(t) - c_k(t), y'_k = (c_k(t), t \in T_c).P_c^{-1}, k \in K.$$

✓ L'ensemble $\{K_f, T_f\} = Q_f$ est appelé support de la fonctionnelle si la matrice Δ_f est inversible.

En usant de la dernière ligne de l'inverse de la matrice Δ_f^{-1} , nous construisons les vecteurs

$$\lambda'(K) = (\lambda'(K_f), \lambda'(K_H)).$$

$$\lambda'(K_f) = (0'(T_f); 1)\Delta_f^{-1}.$$

$$\lambda'(K_H) = 0'(K_H) \quad K_H = K - K_f.$$

✓ Le $Q_p = \{T_c, Q_f\}$. formé du support des contraintes et support de la fonctionnelle est appelé support du problème.

✓ La paire $\{u, Q_p\}$ formée de la commande u et du support Q_p est appelée support - contrôle du problème.

$$\Delta(t) = \sum_{k \in K_f} \lambda_k \Delta_k(t) \quad , \quad t \in T.$$

✓ En plus des définitions des commandes non dégénérées ou dégénérées et les propriétés de la dernière ligne de la matrice Δ_f^{-1} et $\Delta(t)$, cela nous permet d'établir la valeur de l'accroissement de la fonctionnelle à travers la valeur de suboptimalité $\beta(u, Q_p)$.

4.3.4 Accroissement de la fonctionnelle et critères d'optimalité et suboptimalité.

Nous considérons le support contrôle $\{u, Q_p\}$ et une autre commande admissible $\bar{u} = u + \Delta u$ et $\bar{x} = x + \Delta x$, sa trajectoire correspondante, l'accroissement de la fonctionnelle représenté par la valeur $\beta(u, Q_p)$:

$$\beta = \beta(u, Q_p) = \int_{t \in T^+} \Delta(t)(u(t) - d_*) + \int_{t \in T^-} \Delta(t)(u(t) - d^*) + \sum_{k \in K_f} \lambda_k \omega_k$$

est appelée valeur suboptimalité du support - plan $\{u, Q_p\}$. Avec $T^+ = \{t \in T / \Delta(t) \geq 0\}$, $T^- = \{t \in T / \Delta(t) \leq 0\}$.

4.3.5 Théorème (critère d'optimalité)

Les Relations suivantes :

$$\begin{cases} u(t) = d^* & \text{si } \Delta(t) < 0 \\ u(t) = d_* & \text{si } \Delta(t) > 0 \\ d_* \leq u(t) \leq d^* & \text{si } \Delta(t) = 0, t \in T \\ \omega_k = 0 & \text{si } \gamma_k > 0, \\ \omega_k \geq 0 & \text{si } \gamma_k = 0, k \in K_f \end{cases}$$

sont suffisantes et dans le cas de la non dégénérescence, sont nécessaires pour l'optimalité du support-contrôle $\{u, Q_p\}$.

4.3.6 Théorème (critère de sub-optimalité)

Pour ϵ donné. Pour ϵ -optimalité du contrôle u , il est nécessaire et suffisant de trouver un tel support Q_p pour qui la valeur de **suboptimalité** satisfait l'inégalité suivante :

$$\beta(u, Q_p) \leq \epsilon.$$

4.3.7 Méthode de résolution

Il existe différentes approches de résolutions. Nous choisissons celle-ci :

Supposons le support contrôle $\{u, Q_P\}$ qui ne satisfait pas le critère d'optimalité et suboptimalité. Le passage par les itérations de l'algorithme au vue de changer $\{u, Q_P\}$ vers $\{\bar{u}, \bar{Q}_P\}$ (Amélioré) tel que :

$$J(\bar{u}) \geq J(u).$$

Les itérations consistant en 03 procédures.

- ♣ Changement du contrôle u vers \bar{u} .
- ♣ Changement du support Q_P vers \bar{Q}_P .
- ♣ La procédure finale.

4.3.8 Changement du contrôle

La nouvelle commande \bar{u} est définie comme suit :

$$\bar{u}(t) = u(t) + \theta \Delta u(t), t \in T.$$

Et comme suit :

$$T_0 = \{t \in T / |\Delta(t)| \leq \alpha\}$$

$$T_1^* = \{t \in T / |\Delta(t)| > \alpha\}.$$

Nous subdivisons T_0 tel que :

$$T_0 = \bigcup_{j=1}^N [\tau_j, \bar{\tau}_j], \bar{\tau}_j - \tau_j \leq h, \text{ pour } j = 1..N.$$

$[\tau_j, \bar{\tau}_j] \cap [\tau_i, \bar{\tau}_i] = \phi, i \neq j$, pour $(i, j = 1..N)$, l'indice $N + 1$ correspondant à l'intervalle T_1^* .

Considérons $T_p = T_c \cup T_f \subset \{t_j, j = 1..N + 1\}$, les t_j sont choisis aux extrémités des intervalles $(T_0 \cup T_1^*)$ et $u(t) = u_j, t \in (T_0 \cup T_1^*), j = 1..N + 1$.

Posons donc :

$$l_j = \begin{cases} \theta \Delta u(t) & \text{if } t \in [\tau_j, \bar{\tau}_j], j = 1..N \\ \theta & \text{if } t \in T_1^*, j = N + 1. \end{cases}$$

Et Puis :

$$\Delta u(t) = \begin{cases} d^* - u(t) & \text{if } \Delta(t) < -\alpha \\ d_* - u(t) & \text{if } \Delta(t) > \alpha, t \in T_1^* . \end{cases}$$

Encore :

$$q_j = \int_{\tau_j}^{\bar{\tau}_j} p(t) dt, \quad c_k(j) = \int_{\tau_j}^{\bar{\tau}_j} c'_k q(t) dt, \quad k \in K.$$

$$q_{N+1} = \int_{T_1^*} p(t) \Delta u(t) dt, \quad c_k(N + 1) = \int_{T_1^*} c'_k q(t) \Delta u(t) dt, \quad k \in K.$$

$$d_{-j} = d_* - u_j, \quad d_j^+ = d^* - u_j \quad j = 1..N.$$

$$d_{-N+1} = 0, \quad d_{N+1}^+ = 1 \quad j = N + 1.$$

En utilisant les précédentes quantités, nous obtenons le modèle suivant :

$$\left\{ \begin{array}{l} \min_k \left(\sum_{j=1}^{N+1} c_k(j) l_j \right) \rightarrow \max_l \\ \sum_{j=1}^{N+1} q_j l_j = 0 \\ d_- \leq l_j \leq d^+; j = 1..N + 1. \end{array} \right.$$

On résout ce problème avec la méthode adaptée.

Après un nombre d'itérations, cela conduit à la solution optimale.

La nouvelle commande sera donnée comme suit :

$$\bar{u}(t) = \begin{cases} u(t) + \bar{l}_j & \text{si } t \in [\tau_j, \bar{\tau}_j], j = 1..N \\ u(t) + \bar{l}_{N+1} \Delta u(t) & \text{si } t \in T_1^*, j = N + 1. \end{cases}$$

Ce support- Contrôle $\{\bar{u}, \bar{Q}_P\}$ résultant du dernier problème après établissement de la procédure d'exclusion pour l'indice (N+1).

$\beta(\bar{u}, \bar{Q}_P) = 0$ alors le support contrôle est optimal.

$\beta(\bar{u}, \bar{Q}_P) \leq \epsilon$ alors le support contrôle est ϵ - optimal.

$\beta(\bar{u}, \bar{Q}_P) > \epsilon$ alors nous procédons au changement du support ou pour une nouvelle itération avec $\bar{\alpha} < \alpha$ et $\bar{h} < h$.

4.3.9 Changement du support

En utilisant le support \bar{Q}_P , on calcule le vecteur $\bar{\lambda}(K)$ et sa fonction $\bar{\Delta}(t)$, puis on calcule la quasi - commande $w(t)$, $t \in T$ définie comme suit :

$$w(t) = \begin{cases} d_* & \text{si } \bar{\Delta}(t) > 0 \\ d^* & \text{si } \bar{\Delta}(t) < 0 \\ \in [d_*, d^*] & \text{si } \bar{\Delta}(t) = 0 \end{cases}$$

et sa quasi- trajectoire $\chi(t)$, $t \in T$, solution de l'équation différentielle :

$$\frac{d\chi(t)}{dt} = A\chi(t) + bw(t), \chi(0) = x_0, \text{ avec le support } \bar{Q}_P.$$

Si $H(\chi(T_1)) = g$ et $J(w) = (c'_k \chi(T_1) + \alpha_k)$, avec $k \in \bar{K}_f$,

alors $w(t)$ est commande optimale pour notre problème initial.

Sinon on détermine le vecteur

$\xi = (\xi(\bar{T}_p), \xi(s + 1))$, avec

$$|\bar{T}_p| = |\bar{T}_c| + |\bar{T}_f| = s$$

$$\begin{pmatrix} \xi(\bar{T}_p) \\ \xi(s + 1) \end{pmatrix} = R^{-1} \begin{pmatrix} g - H(\chi(T_1)) \\ c'(\bar{K}_f)\chi(T_1) + \alpha(\bar{K}_f) - e(\bar{K}_f)J(w). \end{pmatrix}$$

Avec

$$\mathbf{R} = \begin{pmatrix} P(\bar{T}_c) & P(\bar{T}_f) & 0 \\ -c'(\bar{K}_f)q(\bar{T}_c) & -c'(\bar{K}_f)q(\bar{T}_f) & e(\bar{K}_f) \end{pmatrix}.$$

Nous évaluons les expressions suivantes :

$$\beta_k = \xi(s+1) + J(w) - c'_k \chi(T_1) - \alpha_k - \sum_{j=1}^{|\bar{T}_p|} c'_k q(t_j) \cdot \xi(t_j) \quad k \in \bar{K}_H.$$

Deux cas se présentent :

$$1^\circ / \|\xi(\bar{T}_p)\| < \mu. \quad (\mu \text{ est paramètre de la méthode})$$

$$2^\circ / \beta_k \leq 0. \quad (k \in \bar{K}_H)$$

Si 1°/ et 2°/ sont vérifiés, alors exécution de la procédure finale sinon considérons les deux cas suivants :

1° CAS

2°/ non satisfaite, ie : il existe $k_0 \in \bar{K}_H / \beta_{k_0} > 0$. On utilise la méthode duale pour exclusion.

La construction du dual du problème original, donc cela nous permet d'écrire :

$$\tilde{\Delta}(t) = \bar{\Delta}(t) + \sigma Z(t)$$

$$t \in T$$

et

$$\tilde{\lambda}(K) = \bar{\lambda}(K) + \sigma Z(K).$$

Avec $Z(t)$ est la direction de décroissance de la fonctionnelle Duale et σ le pas maximum le long de cette direction.

$$Z(t) = Z'(\bar{T}_c)P_c^{-1}p(t) + \Delta\bar{\lambda}(\bar{K})\bar{\Delta}(K, t), \quad t \in T.$$

$$Z(\bar{K}_f) = \Delta\bar{\lambda}(\bar{K}_f).$$

$$Z(\bar{K}_f) = -\Delta\bar{\lambda}(\bar{K}_H)(\bar{\Delta}(\bar{K}_H, \bar{T}_f); e(\bar{K}_H))\bar{\Delta}_f^{-1} + (Z(\bar{T}_f), 0)\bar{\Delta}_f^{-1} - Z'(\bar{T}_c)(P_c^{-1}P(\bar{T}_f); 0)\bar{\Delta}_f^{-1}.$$

Donc le pas maximum $\sigma = \sigma_0 = \min(\sigma(t^*), \sigma_{k^*})$.

$$\sigma_{k^*} = \min(-\bar{\lambda}_k / \Delta\bar{\lambda}_k; k \in \bar{K}_f).$$

$$\sigma(t^*) = \min(\sigma(t), /t \in T - T_p).$$

$$\sigma(t) = \begin{cases} -\bar{\Delta}(t)/Z(t) & \text{if } \bar{\Delta}(t)/Z(t) < 0 \\ 0 & \text{if } \bar{\Delta}(t) = 0, w(t) \neq d_*, Z(t) > 0; \text{ or } \bar{\Delta}(t) = 0, w(t) \neq d^*, Z(t) < 0, \\ \infty & \text{sinon.} \end{cases}$$

2° CAS

Supposons la condition 2° / vérifiée. Nous considérons **le nouveau support - contrôle** $\{\bar{u}, \bar{Q}p\}$ et test de la condition 1°/.

Si $\xi(\bar{T}_p) = 0$ alors la commande est optimale.

Si $\xi(\bar{T}_p) < \mu$ alors direction la procédure finale.

Si $\xi(\bar{T}_p) > \mu$ alors nous changeons le support selon la méthode adaptée.

Finalement le support - contrôle $\{\bar{u}, \bar{Q}p\}$ résultant de la procédure de changement du support, alors

si le nouveau support $\{\bar{u}, \bar{Q}p\}$ est admissible alors la commande est optimale

Sinon nous déclenchons la procédure finale (avec ξ approchant μ assez petit).

4.3.10 Procédure finale

La quasi commande $\bar{w}(t)$ et sa trajectoire $\bar{\chi}(t)$, $t \in T$ avec le support \bar{Q}_p , avec relations suivantes non vérifiées par hypothèses.

$$\begin{cases} g = H\bar{\chi}(T_1) \\ \bar{w}(\bar{K}_f) = 0. \end{cases}$$

L'objectif de la procédure finale est de trouver un tel support- contrôle $\{\bar{w}, \bar{Q}_p\}$ qui satisfassent les relations ci-dessus. Pour garantir cela soit :

$T^0 = \{t \in T, \bar{\Delta}(t) = 0\}$ formé par les points isolés t_j , $j=1..M$, $|\bar{T}_p| = |\bar{T}_c| + |\bar{T}_f| = M$ et $\bar{\Delta} \cdot (t_j) \neq 0$, $t_j \in \bar{T}_p$.

$$\begin{aligned} \text{Nous avons } g - H\bar{\chi}(T_1) &= \int_0^{T_1} p(t)\bar{w}(t)dt - g + HF(T_1)x_0. \\ &= \int_{T^0} p(t)\bar{w}(t)dt + \int_{T-T^0} p(t)\bar{w}(t)dt - g + HF(T_1)x_0. \end{aligned}$$

Avec $\|\xi(\bar{T}_p)\| < \mu$, cela nous permet d'écrire les relations Suivantes :

$$g - H\bar{\chi}(T_1) = \begin{cases} \sum_{j=1}^M \left(\int_{t_j}^{\tau_j} p(t)\bar{w}_j(t)dt - \int_{t_j}^{\tau_j} p(t)\bar{w}_j(t)dt \right) & \text{if } \bar{\Delta} \cdot (t_j) > 0 \\ \sum_{j=1}^M \left(\int_{t_j}^{\tau_j} p(t)\bar{w}_j(t)dt - \int_{t_j}^{\tau_j} p(t)\bar{w}_j(t)dt \right) & \text{if } \bar{\Delta} \cdot (t_j) < 0. \end{cases}$$

$$g - H\bar{\chi}(T_1) = \begin{cases} \sum_{j=1}^M (d^* - d_*) \int_{t_j}^{\tau_j} p(t)dt & \text{if } \bar{\Delta} \cdot (t_j) > 0 \\ \sum_{j=1}^M - (d^* - d_*) \int_{t_j}^{\tau_j} p(t)dt & \text{if } \bar{\Delta} \cdot (t_j) < 0. \end{cases}$$

Cela nous conduit aux approximations suivantes :

$$t_j^{k+1} = t_j^k + \text{signe} \bar{\Delta} \cdot (t_j^k) \xi(t_j^k) / (d^* - d_*).$$

4.3.11 Conclusions et perspectives

On a résolu des problèmes en contrôle optimal avec Implémentations et Production de Software - déploiement :

- Résolutions et Implémentations de la méthode adaptée et support, pour beaucoup de cas.
- Simulations et comparaisons.
- Résolutions et Implémentations de la méthode adaptée et support pour le contrôle optimal (Min-Max : discret /continu).
- Comme Perspectives : Résolutions et Implémentations de la méthode adaptée, support et Simplex pour le contrôle optimal, appliquées à des problèmes non linéaire par exemples ([108]) avec des caractères flous et stochastiques.

4.3.12 Exemple

Un exemple très détaillé, pour un cas général est résolu en [23].

Donc, en guise de deuxième exemple, l'exemple suivant présente l'idée de la discrétisation.

4.3.13 Exemple

Soit à résoudre : Avec réglage du logiciel("Matrix") à p =5, et donc les résultats suivants :

$$\begin{cases} J(u) = \min_k(c'_k x(t^*)) \longrightarrow \max_u, \\ \dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0), \\ Hx(t^*) = g, \\ |u(t)| \leq 21, \quad t \in T = [0, 8]. \end{cases}$$

Avec

$$c'_k = \begin{pmatrix} 1 & -1 & 1 & 1 & 0 & 0 & 1 \\ 1 & -2 & 1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 2 & -1 & 1 & 0 & 1 \\ -1 & 1 & -1 & -1 & 0 & 0 & 2 \\ -1 & 2 & -1 & -2 & 0 & -1 & -1 \end{pmatrix},$$

$$\alpha'_k = (0 \ 0 \ 0 \ 0 \ 0),$$

$$b' = (0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1),$$

$$g' = (.8535 \ .4282 \ 1.21 \ 1.6376 \ 2.3773 \ 3.75),$$

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 2 & 0 & 0 \\ 0 & 2 & 3 & 1 & 3 & 1 & 1 \\ 1 & 0 & 2 & 1 & 1 & 4 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Le $\text{rang}[b, Ab, A^2b, \dots, A^{n-1}] = 7$, donc on continue le processus, de plus le système est stable pour $n = 7$, $A^7 = [0]$.

Il en résulte que : les calculs de $F(t)$ et $F^{-1}(t)$ sont immédiats. Dans l'espace \mathbb{R}^{500} , on aboutit à une commande ϵ -optimale, réalisable, $\|(\xi \tilde{T}_p)\| < \mu = 0.40$ et avec l'appui suivant.

$$\tau_B = \{0.336, 1.296, 2.72, 4.384, 5.968, 7.28\}, T_c = \tau_B.$$

$$K_f = \{5\}.$$

Ce qui nous permet d'aborder la procédure finale.

La procédure finale

Avec $\tau_B = \{0.336, 1.296, 2.72, 4.384, 5.968, 7.28\}$, et $K_f = \{5\}$.

On aura

$$\tau_j^{k+1} = \tau_j^k + \frac{1}{d^* - d_*} \{ \text{sign} \tilde{\Delta}(t_j^k) \xi_j(\tau_j^k) \}, \quad j = \overline{1, M},$$

ce qui donne comme nouvelle commande

$$\tilde{\omega} = \begin{cases} 21, & 0 \leq t \leq .334945873505841, \\ -21, & .334945873505841 \leq t \leq 1.28931655773698, \\ 21, & 1.28931655773698 \leq t \leq 2.71618144128623, \\ -21, & 2.71618144128623 \leq t \leq 4.37496741769497, \\ 21, & 4.37496741769497 \leq t \leq 5.95827591838834, \\ -21, & 5.95827591838834 \leq t \leq 7.27615497203418, \\ 21, & 7.27615497203418 \leq t \leq 8, \end{cases}$$

la commande ci-dessus est optimale, avec le support

$$\tau_B^0 = \{.334945873505841, 1.28931655773698, 2.71618144128623, 4.37496741769497, 5.95827591838834, 7.27615497203418\}.$$

$$K_f = \{5\}.$$

4.3.14 Exemple

Soit à résoudre : Avec réglage du logiciel("Matrix") à $p = 1$, et donc les résultats suivants :

$$\begin{cases} J(u) = c'x(t^*) \longrightarrow \max_u, \\ \dot{x} = \frac{dx}{dt} = Ax + bu, \quad x(t_0) = (1 \ 0 \ 1), \\ Hx(t^*) = g, \\ |u(t)| \leq 3.5, \quad t \in T = [0, 3]. \end{cases}$$

Avec $(t^* = T_1)$

$$c' = (1 \ -1 \ 1),$$

$$g' = \begin{pmatrix} 0.1 & 0.5 \end{pmatrix},$$

$$b' = \begin{pmatrix} 0 & 2 & 1 \end{pmatrix}.$$

et

$$H = \begin{pmatrix} 1 & 3 & 2 \\ 1 & 1 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Le $\text{rang}[b, Ab, A^2b, \dots, A^{n-1}] = 3$, donc on continue le processus, de plus le système est stable pour $n = 3$, $A^3 = [0]$.

Il en résulte que :

$$F(t) = \begin{pmatrix} 1 & t & \frac{1}{2}t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{pmatrix}, \quad F^{-1}(t) = \begin{pmatrix} 1 & -t & \frac{1}{2}t^2 \\ 0 & 1 & -t \\ 0 & 0 & 1 \end{pmatrix}.$$

Soit la commande suivante :

$$u(t) = \begin{cases} .901886792452819, & 0 \leq t \leq 1, \\ -3.5, & 1 \leq t \leq 2, \\ 1.93018867924531, & 2 \leq t \leq 3. \end{cases}$$

Avec

$$p(t) = \phi(t) = HF(t^*)F^{-1}b = \begin{pmatrix} \frac{1}{2}t^2 - 8t + 27.5 \\ \frac{1}{2}t^2 - 6t + 16.5 \end{pmatrix},$$

$$c(t) = c'F(t^*)F^{-1}(t)b = \left(\frac{1}{2}t^2 - 4t + 6.5 \right),$$

la commande ainsi définie est admissible car :

$$\int_0^1 \phi(t)u(t)dt + \int_1^2 \phi(t)u(t)dt + \int_2^3 \phi(t)u(t)dt = g - HF(3)x(0), \quad |u(t)| \leq 3.5.$$

A présent considérons le support contrôle $\{u, \tau_B\}$ avec

$$u(t) = \begin{cases} .901886792452819, & 0 \leq t \leq 1, \\ -3.5, & 1 \leq t \leq 2, \\ 1.93018867924531, & 2 \leq t \leq 3. \end{cases}$$

Avec $\tau_B = \{1, 3\}$, $T_c = \tau_B$

$$\beta(u, \tau_B) = \sum_{t \in T^+} \Delta(t)\Delta u(t) + \sum_{t \in T^-} \Delta(t)\Delta u(t) = .506019766397131.$$

Soit $\tau_B = \{2, 3\}$, $T_c = \tau_B$

avec les paramètres de la méthode $h \leq 1$ et $\alpha = 0.5$, $|\Delta(t)| \leq 0.5$. Nous scindons l'intervalle $T = [0, 3] = T_0 \cup T_1$, comme suit :

$$T_0 = [0.05051, 0.525255] \cup [0.525255, 1] \cup [1, 1.25] \cup [1.25, 1.5] \cup [1.5, 1.75] \cup [1.75, 2] \cup [2, 2.25] \cup [2.25, 2.5] \cup [2.5, 2.75] \cup [2.75, 3],$$

$$T_1 = [0, 0.05051].$$

Cela nous permet d'aboutir au système suivant :

$$\left\{ \begin{array}{l} (2, 5632l_1 + 1.7801l_2 + 0.6588l_3 + 0.4869l_4 + 0.3307l_5 + 0.1901l_6 + 0.0651l_7 - 0.0442l_8 - \\ 0.1380l_9 - 0.2161l_{10} + .8397984l_{11}) \rightarrow \max_l, \\ 11, 9862l_1 + 10, 3015l_2 + 4, 7838l_3 + 4, 3619l_4 + 3, 9557l_5 + 3, 5651l_6 + 3, 1901l_7 + 2, 8307l_8 + \\ 2, 4869l_9 + 2, 1588l_{10} + 3.582386l_{11} = 0, \\ 7.0373l_1 + 5.8034l_2 + 2.5963l_3 + 2.2994l_4 + 2.0182l_5 + 1.7526l_6 + 1.5026l_7 + 1.2682l_8 + \\ 1.0494l_9 + 0.8463l_{10} + 2, 145476l_{11} = 0, \\ -4.40188679245282 \leq l_1 \leq 2.59811320754718, \\ -4.40188679245282 \leq l_2 \leq 2.59811320754718, \\ 0 \leq l_3 \leq 7, \\ 0 \leq l_4 \leq 7, \\ 0 \leq l_5 \leq 7, \\ 0 \leq l_6 \leq 7, \\ -5.43018867924531 \leq l_7 \leq 1.56981132075469, \\ -5.43018867924531 \leq l_8 \leq 1.56981132075469, \\ -5.43018867924531 \leq l_9 \leq 1.56981132075469, \\ -5.43018867924531 \leq l_{10} \leq 1.56981132075469, \\ 0 \leq l_{11} \leq 1, \end{array} \right.$$

la solution optimale est :

$$l' = \begin{pmatrix} 2.59811320754718 & -3.69249042372986 & 0 & 0 & 0 & 0 & 0 \\ & -2.64015161672731 & & & & & \\ & 1.56981132075469 & 1.56981132075469 & 1.56981132075469 & 1 & & \end{pmatrix},$$

donc la nouvelle commande \bar{u} est :

$$\bar{u}(t) = \begin{cases} 3.5, & 0 \leq t \leq 0.05051, \\ 3.5, & 0.05051 \leq t \leq 0.525255, \\ -2.79060363127704, & 0.525255 \leq t \leq 1, \\ -3.5, & 1 \leq t \leq 1.25, \\ -3.5, & 1.25 \leq t \leq 1.5, \\ -3.5, & 1.5 \leq t \leq 1.75, \\ -3.5, & 1.75 \leq t \leq 2, \\ -.709962937481998, & 2 \leq t \leq 2.25, \\ 3.5, & 2.25 \leq t \leq 2.5, \\ 3.5, & 2.5 \leq t \leq 2.75, \\ 3.5, & 2.75 \leq t \leq 3. \end{cases}$$

Le nouveau support contrôle est $\{\bar{u}, \tilde{\tau}_B\}$ avec

$$\beta(\bar{u}, \tilde{\tau}_B) = \sum_{t \in T^+} \tilde{\Delta}(t) \Delta u(t) + \sum_{t \in T^-} \tilde{\Delta}(t) \Delta u(t) = .110150609334363.$$

Passons au changement du support.

Changement du support

Posons $\mu = 4.0 \times 10^{-1}$ (paramètre de la méthode). Soit la quasi-commande :

$$\tilde{\omega}(t) = \begin{cases} 3.5, & \tilde{\Delta}(t) < 0, \\ -3.5, & \tilde{\Delta}(t) > 0, \\ \tilde{\omega}(t) \in [-3.5, 3.5], & \tilde{\Delta}(t) = 0. \end{cases}$$

Avec sa trajectoire correspondante $\chi(t)$ sur l'intervalle $[0, 3]$, avec $H\chi(t) \neq g$.

✓ Initialisation

Avec $\tau_B = \{0.05050, 1.75\}$,

alors la nouvelle commande aura l'allure suivante :

$$\tilde{\omega} = \begin{cases} 3.5, & 0 \leq t \leq 0.05050, \\ -3.5, & 0.05050 \leq t \leq 1.75, \\ 3.5, & 1.75 \leq t \leq 3, \end{cases}$$

toujours $H\chi(t^*) \neq g$.

✓ 1^{eme} itération

On choisit $t = 0.62$ car pour cette valeur, on a $\tilde{\Delta}(t)\delta(t) < 0$, ($\delta(t) = Z(t)$).

Alors la nouvelle commande sera la suivante :

$$\tilde{\omega} = \begin{cases} 3.5, & 0 \leq t \leq 0.62, \\ -3.5, & 0.62 \leq t \leq 1.75, \\ 3.5, & 1.75 \leq t \leq 3, \end{cases}$$

avec $\tau_B = \{0.62, 1.75\}$,

mais

$H\chi(t^*) \neq g$.

✓ 2^{eme} itération

$\bar{\Delta}(t) = 0 \Rightarrow$

on choisit $t = 2.14$ car pour cette valeur, on a $\tilde{\Delta}(t)\delta(t) < 0$.

$$\tilde{\omega} = \begin{cases} 3.5, & 0 \leq t \leq 0.62, \\ -3.5, & 0.62 \leq t \leq 2.14, \\ 3.5, & 2.14 \leq t \leq 3 \end{cases}$$

on a $\|(\xi \tilde{T}_p)\| < \mu$, donc on passe à la procédure finale.

La Procédure finale

Posons $\tau_B^0 = \{0.62, 2.14\}$.

On a

$$\tau_j^{k+1} = \tau_j^k + \frac{1}{d^* - d_*} \{sign \tilde{\Delta}(t_j^k) \xi_j(\tau_j^k)\}, \quad j = \overline{1, M},$$

ce qui donne comme nouvelle commande

$$\tilde{\omega} = \begin{cases} 3.5, & 0 \leq t \leq .568834174124978, \\ -3.5, & .568834174124978 \leq t \leq 2.14451888022972, \\ 3.5, & 2.14451888022972 \leq t \leq 3 \end{cases}$$

la commande ci-dessus est optimale, avec le support $\tau_B^0 = \{.568834174124978, 2.14451888022972\}$.

4.4 Conclusion

Donc, on a établi la théorie du contrôle optimal sur un système dynamique, avec des cas discrets et continus. Nous avons résolu ces cas, cela nous permet de résoudre des cas plus complexes, élargis aux types non linéaires et multi-objectifs (stochastiques ou/et flous).

Conclusion générale

Actuellement, la convexité est l'outil principal par excellence de l'optimisation globale, comme on a prouvé cela tout le long de notre travail. L'optimisation convexe est une sous-discipline de l'optimisation mathématique, dans laquelle le critère à minimiser (à optimiser) est convexe et l'ensemble admissible est convexe [55, 99, 54]. Dans ce cadre, une prise en charge de ces problèmes d'optimisation est vite faite. L'existence d'une panoplies de théorèmes et d'algorithmes et en général une théorie élégante facilitent donc cette tâche. Ces problèmes sont plus simples à analyser et à résoudre que les problèmes d'optimisation non convexes, bien qu'ils puissent être NP-difficiles (c'est le cas de l'optimisation convexe en général).

La théorie permettant d'analyser des problèmes ne requiert pas toujours la différentiabilité des fonctions. Cette généralité est motivée par le fait que certaines méthodes de construction de problèmes d'optimisation convexe conduisent à des problèmes non différentiables (fonction marginale, dualisation de contraintes, etc). Si cette généralité est un atout, permettant de prendre en compte davantage de problèmes, l'abord de la théorie est également plus difficile.

L'optimisation convexe repose sur l'analyse convexe. Dans l'optimisation globale en plus des méthodes propres à cette branche, l'analyse convexe, la théorie de la convexité et la dualité convexe sont d'une utilité très estimable. A noter aussi les nouvelles innovations de l'optimisation globale.

Actuellement l'optimisation globale connaît une nette progression en matière de théorie et application au niveau informatique [101, 95, 57] en particulier.

Pour les techniques d'optimisation globale, nous avons entre autres au moins les méthodes suivantes :

- ♣ La programmation anti- convexe en général.
- ♣ La méthode de Branch and Bound/Branch and cut/Branch and Prune.
- ♣ La méthode DC (Difference convex) et DCA (Difference convex Algorithm) [104, 79, 85].
- ♣ La méthode d'analyse des intervalles [78].
- ♣ La méthode d'analyse affine [86].
- ♣ La méthode d'annexion polyédrale [102].
- ♣ La méthode homotopique.
- ♣ La méthode dynamic programming.
- ♣ La méthode Branch and Bound (mixed integer programming) [16].

Toutes les méthodes de l'optimisation globale que ce soit déterministes ou non déterministes que

nous avons déjà citées plutôt, trouvent leurs places incontestablement dans divers domaines de la vie réelle (du spatial vers la médecine en transitant par les sciences humaines par exemple) comme on a prouvé cela plus haut.

Comme perspective, nous pouvons orienter les recherches vers :

- * / Amélioration de nos contributions publiées dans l'espace \mathbb{R}^n .
- * / Amélioration de nos contributions publiées dans l'espace \mathbb{R}^n et \mathbb{R} concernant les fonctions Bornes.
- * / Obtenir de nouveaux algorithmes dans tout espace en multi-Objectifs.
- * / Obtenir de nouvelles conditions nécessaires et suffisantes dans le cadre flou-stochastique.
- * / Amélioration de notre travail sur les espaces discrets.

ANNEXE A

Généralités sur les fonctions à plusieurs variables

A.1 Introduction

Dans cet annexe, nous allons présenter les fonctions à plusieurs variables notions de base et théorie. Il est recommandé d'aborder même sommairement les notions de normes, espace vectoriel normé, distance et quelques éléments de la topologie.

A.2 Généralités sur les fonctions à plusieurs variables et topologie. [16, 34, 38, 39, 70]

On désigne par $\mathbb{R}^n = \mathbb{R} \times \mathbb{R} \times \dots \times \mathbb{R}$, le produit cartésien (n fois) et donc \mathbb{R}^n est l'ensemble des n -uplets $(x_1, x_2, \dots, x_i, \dots, x_n)$, $i = 1, \dots, n$ avec les x_i étant des réels. \mathbb{R}^n possède une structure d'espace vectoriel sur le corps \mathbb{R} .

A.2.1 Éléments de la topologie

Soit E un espace vectoriel sur un corps K tel que $K = \mathbb{R}$ ou $K = \mathbb{C}$. Soit l'application produit scalaire définie ainsi :

Définition A.2.1. Un produit scalaire sur E est une application

$$\langle \cdot, \cdot \rangle : E \times E \longrightarrow K.$$

L'application $\langle \cdot, \cdot \rangle$ vérifie les conditions suivantes :

- (i) $\langle \cdot, \cdot \rangle$ est sesquilinéaire,

$$\forall x_1, x_2, y \in E, \forall \alpha, \beta \in K, \langle \alpha x_1 + \beta x_2, y \rangle = \bar{\alpha} \langle x_1, y \rangle + \bar{\beta} \langle x_2, y \rangle .$$

- (ii) $\langle \cdot, \cdot \rangle$ est semi-symétrique : $\forall x, y \in E \langle x, y \rangle = \overline{\langle y, x \rangle}$.
- (iii) $\langle \cdot, \cdot \rangle$ est définie positive : $\forall x \in E, x \neq 0 \implies \langle x, x \rangle > 0$.

Mathématiquement, un espace vectoriel muni d'un produit scalaire est appelé espace préhilbertien. Si l'espace vectoriel est réel, c'est un espace euclidien et espace hermitien dans le cas complexe.

Exemple A.2.1. */ E désignant l'espace vectoriel des fonctions réelles continues sur $[a, b]$. L'application ainsi définie désigne un produit scalaire.

$$\forall f, h \in E, \langle f, h \rangle = \int_a^b f(x)h(x)dx.$$

*/ E désignant l'espace vectoriel des matrices à éléments réels, m lignes et n colonnes

$$\forall A, B \in E, \langle A, B \rangle = \text{tr}(B^T A).$$

tr désignant l'opérateur trace, B^T transposée de B, est une application produit scalaire.

A.2.2 Les espaces normés

Soit E un espace vectoriel sur un corps K tel que $K = \mathbb{R}$ ou $K = \mathbb{C}$. Soit l'application norme sur E définie ainsi :

Définition A.2.2. Une norme sur E est une application

$$\|\cdot\| : E \longrightarrow \mathbb{R}, \quad x \longmapsto \|x\|$$

avec les conditions suivantes :

- (i) $\forall x \in E, \|x\| \geq 0$ et $\|x\| = 0 \iff x = 0$.
- (ii) $\forall x \in E, \forall \alpha \in K, \|\alpha x\| = |\alpha| \|x\|$.
- (iii) $\forall x, y \in E, \|x + y\| \leq \|x\| + \|y\|$. (inégalité triangulaire ou inégalité de Minkowski)

Un espace vectoriel muni d'une norme est dit espace vectoriel normé.

Sur E , l'application $x \longmapsto \|x\| = \sqrt{\langle x, x \rangle}$ est une norme associée au produit scalaire \langle, \rangle sur E .

Proposition A.2.1. Pour le produit scalaire \langle, \rangle sur E , on a l'inégalité de (Cauchy-Schwarz).

$$\forall x, y \in E, |\langle x, y \rangle| \leq \|x\| \cdot \|y\|.$$

Remarque A.2.1. */ $\forall x, y \in E, |\langle x, y \rangle| = \|x\| \cdot \|y\|$ si et seulement si x et y sont linéairement dépendants.

Exemple A.2.2. Soit $x = (x_1, \dots, x_i, \dots, x_n) \in \mathbb{R}^n, i = 1, \dots, n$. Posant

*/ $\|x\|_1 = |x_1| + |x_2| + \dots + |x_n|$.

*/ $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$.

*/ $\|x\|_p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p)^{\frac{1}{p}}, 1 \leq p < \infty$.

*/ $\|x\|_\infty = \max(|x_1|, |x_2|, \dots, |x_n|)$.

Les applications suivantes :

$$\|x\|_1, \|x\|_2, \|x\|_p, \|x\|_\infty : \mathbb{R}^n \mapsto \mathbb{R}_+$$

sont des normes sur \mathbb{R}^n .

A.2.3 Les espaces métriques

Soit E un ensemble (non vide). Soit l'application distance sur E définie ainsi :

Définition A.2.3. Une distance sur E est une application

$$d : E \times E \mapsto \mathbb{R}_+, (x, y) \mapsto d(x, y)$$

avec les conditions suivantes :

- (i) $\forall x, y \in E, d(x, y) \geq 0$ et $d(x, y) = 0 \iff x = y$.
- (ii) $\forall x, y \in E, d(x, y) = d(y, x)$.
- (iii) $\forall x, y, z \in E, d(x, z) \leq d(x, y) + d(y, z)$. (inégalité triangulaire)

Un ensemble muni d'une distance est dit espace métrique.

Si E est un espace vectoriel normé, alors l'application $(x, y) \mapsto \|x - y\|$ est une distance sur E .

Exemple A.2.3. Soit $\forall x = (x_1, \dots, x_i, \dots, x_n) \in \mathbb{R}^n, \forall y = (y_1, \dots, y_i, \dots, y_n) \in \mathbb{R}^n, i = 1, \dots, n$, et soient

$$*/ (x, y) \mapsto d_1(x, y) = \sum_{i=1}^n |x_i - y_i|.$$

$$*/ (x, y) \mapsto d_2(x, y) = \sqrt{\sum_{i=1}^n (|x_i - y_i|)^2}.$$

$$*/ (x, y) \mapsto d_\infty(x, y) = \max(|x_i - y_i|), i = 1, \dots, n.$$

Les applications suivantes :

$$d_1(x, y), d_2(x, y), d_\infty(x, y) : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}_+$$

sont des distances sur $\mathbb{R}^n \times \mathbb{R}^n$.

A.2.4 Éléments de topologies sur \mathbb{R}^n

On appelle boule ouverte de centre $a \in \mathbb{R}^n$ et de rayon $r \geq 0$, l'ensemble

$$B(a, r) = \{x \in E : \|x - a\| < r\}.$$

Lorsque $\|x - a\| \leq r$, on dira que la boule est fermée et on note $\bar{B}(a, r)$.

Définition A.2.4. On appelle voisinage d'un point $a \in \mathbb{R}^n$, tout ensemble $v(a)$ qui contient une boule ouverte $B(a, r)$.

Définition A.2.5. Soit $A \subset \mathbb{R}^n$. On dit que A est un ouvert si $A = \emptyset$ ou

$$\forall a \in A, \exists r > 0 : B(a, r) \subset A.$$

Autrement dit, A est ouvert si $\text{int}A = A$. ($\text{int}A$) désignant intérieur de A .

Définition A.2.6. A est fermé si son complémentaire est ouvert.

A.2.5 Les fonctions sur \mathbb{R}^n et théories

On s'intéressera aux fonctions de \mathbb{R}^n vers \mathbb{R}^p . Les résultats qu'on précisera ont un aspect général.

A.2.6 Limites et continuité

Soit Ω un ouvert de \mathbb{R}^n et soit $f : \Omega \rightarrow \mathbb{R}^p$, une application.

Définition A.2.7. On dit que f admet une limite $l \in \mathbb{R}^p$, lorsque x tend vers a dans Ω , si

$$\forall \varepsilon > 0, \exists \delta > 0 : \forall x \in \Omega, \|x - a\| < \delta \Rightarrow \|f(x) - l\| < \varepsilon.$$

On écrit $l = \lim_{x \rightarrow a} f(x)$.

Définition A.2.8. On dit que f est continue en a si $\lim_{x \rightarrow a} f(x) = f(a)$. On dit que f est continue sur Ω , si elle est continue en tout point de Ω .

A.2.7 Dérivées partielles et directionnelles

Soient $a \in \mathbb{R}^n$, Ω un voisinage de a , $f : \Omega \rightarrow \mathbb{R}^p$ et $u \in \mathbb{R}^n$ avec $u \neq 0$.

Définition A.2.9. Si la limite $\lim_{\substack{h \rightarrow 0 \\ h \neq 0}} \frac{f(a+hu) - f(a)}{h}$ existe alors on l'appelle dérivée directionnelle de f en a dans la direction u et on la note $\frac{\partial f}{\partial u}(a)$ ou $\partial_u f(a)$ ou $D_u f(a)$ ou encore $f'(a, u)$.

Un cas particulier de la dérivée directionnelle est la dérivée partielle. En effet la dérivée partielle de f au point a par rapport à la $i^{\text{ème}}$ variable est obtenue en prenant comme u le $i^{\text{ème}}$ vecteur $e_i = (0, \dots, 0, 1, 0, \dots, 0)$ de la base canonique (e_1, \dots, e_n) de \mathbb{R}^n . On la note $\frac{\partial f}{\partial x_i}(a)$ ou $f'_i(a)$ ou $D_{x_i} f(a)$.

D'après les définitions, on a

$$\frac{\partial f}{\partial x_i}(a) = \lim_{\substack{h \rightarrow 0 \\ h \neq 0}} \frac{f(a + he_i) - f(a)}{h}.$$

$$\frac{\partial f}{\partial x_i}(a) = \lim_{\substack{h \rightarrow 0 \\ h \neq 0}} \frac{f(a_1, \dots, a_i + h, \dots, a_n) - f(a_1, \dots, a_i, \dots, a_n)}{h}.$$

A.2.8 Différentiabilité des fonctions

Soit Ω un ouvert de \mathbb{R}^n et $f : \Omega \mapsto \mathbb{R}^p$, une application.

Définition A.2.10. On dit que f est différentiable au point $a \in \Omega$ s'il existe application linéaire $L : \mathbb{R}^n \mapsto \mathbb{R}^p$, telle que :

$$\forall h \in \mathbb{R}^n, f(a + h) = f(a) + L(h) + \epsilon(h),$$

avec $\lim_{\substack{h \rightarrow 0 \\ h \neq 0}} \frac{\epsilon(h)}{\|h\|} = 0$ (ou $\epsilon(h) = o(\|h\|)$). C'est à dire : $L : \mathbb{R}^n \mapsto \mathbb{R}^p$, telle que :

$$f(x) = f(a) + L(x - a) + \epsilon(x - a),$$

(avec un changement de variables)

avec $\lim_{\substack{x \rightarrow a \\ x \in \Omega - \{a\}}} \frac{\epsilon(x-a)}{\|x-a\|} = 0$. L'application L si elle existe, est unique et s'appelle la différentielle de f au point a . On la note $df(a)$ ou df_a .

Proposition A.2.2. [38, 39, 16, 34, 70] Si f est différentiable au point a , alors les dérivées partielles $\frac{\partial f}{\partial x_i}(a)$ existent et on a :

$$df : \mathbb{R}^n \mapsto \mathbb{R}^p, (dx_1, \dots, dx_n) \mapsto df = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i.$$

Remarque A.2.2. La seule existence des dérivées partielles ne suffit pas à assurer la différentiabilité.

Définition A.2.11. On dit que f est continument différentiable ou de classe C^1 sur Ω lorsque les dérivées partielles $\frac{\partial f}{\partial x_i}$ de f existent et sont continues sur Ω .

A.2.9 Matrice jacobienne

Considérons l'application $f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}^p$, $x \mapsto y = f(x)$. On a

$$y_1 = f_1(x_1, \dots, x_n), y_2 = f_2(x_1, \dots, x_n), \dots, y_p = f_p(x_1, \dots, x_n).$$

Avec hypothèse que les dérivées partielles $\frac{\partial f_i(a)}{\partial x_j}$, $1 \leq i \leq p$, $1 \leq j \leq n$, $a \in \Omega$ existent.

Définition A.2.12. On appelle matrice jacobienne de f en a la matrice d'ordre $p \times n$, définie comme suit :

$$J_f(a) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(a) & \cdots & \frac{\partial f_1}{\partial x_n}(a) \\ \frac{\partial f_2}{\partial x_1}(a) & \cdots & \frac{\partial f_2}{\partial x_n}(a) \\ \vdots & \cdots & \vdots \\ \frac{\partial f_p}{\partial x_1}(a) & \cdots & \frac{\partial f_p}{\partial x_n}(a) \end{pmatrix}.$$

Si $p = 1$, $J_f(a)$ se réduit à un vecteur de \mathbb{R}^n ,

$$\text{grad } f = \left(\frac{\partial f}{\partial x_1}(a), \dots, \frac{\partial f}{\partial x_n}(a) \right)$$

est appelé le gradient de la fonction f on le note aussi ∇f . Si par contre on se trouve dans les conditions $n = p$ alors le déterminant de la matrice $J_f(a)$ s'appelle le jacobien de f et de suite :

$$\det J_f(a) = \frac{\partial(f_1, \dots, f_n)}{\partial(x_1, \dots, x_n)}(a).$$

Proposition A.2.3. [38, 39, 16, 34, 70] On a le résultat, avec f et g différentiables

$$J_{g \circ f}(a) = J_g(f(a)) \cdot J_f(a).$$

Si f est bijective avec $g = f^{-1}$ et donc

$$\frac{\partial(f_1, \dots, f_n)}{\partial(x_1, \dots, x_n)} = \frac{1}{\frac{\partial(x_1, \dots, x_n)}{\partial(f_1, \dots, f_n)}}.$$

Exemple A.2.4. * / Si $w = f(u, v)$ avec $u = g(x, y)$, $v = h(x, y)$ et si f, g et h sont différentiables alors

$$\begin{cases} \frac{\partial(w)}{\partial(x)} = \frac{\partial(w)}{\partial(u)} \frac{\partial(u)}{\partial(x)} + \frac{\partial(w)}{\partial(v)} \frac{\partial(v)}{\partial(x)} \\ \frac{\partial(w)}{\partial(y)} = \frac{\partial(w)}{\partial(u)} \frac{\partial(u)}{\partial(y)} + \frac{\partial(w)}{\partial(v)} \frac{\partial(v)}{\partial(y)} \end{cases}$$

* / $w = r^3 + sv^2 + t^4, r = x^2 + y^2 + z^2, s = xyz, v = \exp(y)x, t = yz^2$; alors on utilise :

$$\frac{\partial(w)}{\partial(x)} = \frac{\partial(w)}{\partial(r)} \frac{\partial(r)}{\partial(x)} + \frac{\partial(w)}{\partial(s)} \frac{\partial(s)}{\partial(x)} + \frac{\partial(w)}{\partial(v)} \frac{\partial(v)}{\partial(x)} + \frac{\partial(w)}{\partial(t)} \frac{\partial(t)}{\partial(x)}$$

$$\frac{\partial(w)}{\partial(y)} = \frac{\partial(w)}{\partial(r)} \frac{\partial(r)}{\partial(y)} + \frac{\partial(w)}{\partial(s)} \frac{\partial(s)}{\partial(y)} + \frac{\partial(w)}{\partial(v)} \frac{\partial(v)}{\partial(y)} + \frac{\partial(w)}{\partial(t)} \frac{\partial(t)}{\partial(y)}$$

$$\frac{\partial(w)}{\partial(z)} = \frac{\partial(w)}{\partial(r)} \frac{\partial(r)}{\partial(z)} + \frac{\partial(w)}{\partial(s)} \frac{\partial(s)}{\partial(z)} + \frac{\partial(w)}{\partial(v)} \frac{\partial(v)}{\partial(z)} + \frac{\partial(w)}{\partial(t)} \frac{\partial(t)}{\partial(z)}.$$

A.2.10 Formule de Taylor

Il existe une manière d'approcher des fonctions de structures complexes par utilisation du développement de Taylor.

Proposition A.2.4. [38, 39, 16, 34, 70] Soit $a \in \mathbb{R}^n$, Ω un voisinage de a et $f : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$. Soit $h \in \mathbb{R}^n$, tel que le segment $[a, a + h]$, soit contenu dans Ω . On suppose que $f \in C^{r+1}$ sur Ω . Alors, il existe $\theta \in]0, 1[$ tel que :

$$\begin{aligned}
 f(a+h) &= f(a) + \sum_{i=1}^n \frac{\partial f}{\partial x_i}(a)h_i + \frac{1}{2} \sum_{i_1, i_2=1}^n \frac{\partial^2 f}{\partial x_{i_2} \partial x_{i_1}}(a)h_{i_1}h_{i_2} + \dots \\
 &\quad + \frac{1}{r!} \sum_{i_1, \dots, i_r=1}^n \frac{\partial^r f}{\partial x_{i_r} \dots \partial x_{i_1}}(a)h_{i_1} \dots h_{i_r} \\
 &\quad + \frac{1}{(r+1)!} \sum_{i_1, \dots, i_{r+1}=1}^n \frac{\partial^{r+1} f}{\partial x_{i_{r+1}} \dots \partial x_{i_1}}(a+\theta h)h_{i_1} \dots h_{i_{r+1}}.
 \end{aligned}$$

Exemple A.2.5. Le développement de Taylor au voisinage de l'origine pour la fonction $ch(x+y)$ tel que :

$$ch(v) = 1 + \frac{v^2}{2!} + \dots + \frac{v^{2k}}{(2k)!} + R_{2k}(v), \quad \lim_{\substack{v \rightarrow 0 \\ v \neq 0}} \frac{R_{2k}(v)}{v^{2k}} = 0$$

$$ch(x+y) = 1 + \frac{(x+y)^2}{2!} + \dots + \frac{(x+y)^{2k}}{(2k)!} + R_{2k}((x+y)), \quad \lim_{\substack{(x+y) \rightarrow 0 \\ (x+y) \neq 0}} \frac{R_{2k}((x+y))}{(x+y)^{2k}} = 0.$$

A.2.11 Points critiques et extremas

Dans ce cadre Ω est un ouvert de \mathbb{R}^n , f une fonction de Ω dans \mathbb{R} et $a \in \Omega$. On s'intéresse aux extremas de f .

Définition A.2.13. On dit que f admet un minimum local en a s'il existe un voisinage V (dans Ω) de a tel que :

$$f(a) \leq f(x), \forall x \in V.$$

On dit que f admet un minimum global ou absolu en a si

$$f(a) \leq f(x), \forall x \in \Omega.$$

Proposition A.2.5. (Condition nécessaire) : Si f est différentiable en a , est un extremum (un maximum ou un minimum), alors $df(a) = 0$.

Définition A.2.14. La matrice hessienne de f en $a \in \Omega$ un ouvert tel que

$f : \Omega \subset \mathbb{R}^n \mapsto \mathbb{R}$, f de classe C^2

$$\mathbb{H}(f, a) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(a) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(a) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(a) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(a) & \frac{\partial^2 f}{\partial x_2^2}(a) & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(a) \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(a) & \frac{\partial^2 f}{\partial x_n \partial x_2}(a) & \dots & \frac{\partial^2 f}{\partial x_n^2}(a) \end{pmatrix} = \left(\left(\frac{\partial^2 f}{\partial x_j \partial x_i}(a) \right) \right)_{1 \leq i, j \leq n}.$$

On associe à la matrice hessienne \mathbb{H} de f au point a la forme quadratique

$Q : \mathbb{R}^n \mapsto \mathbb{R}$ comme suit :

$$Q(h) = \sum_{i,j=1}^n \frac{\partial^2 f}{\partial x_j \partial x_i}(a) h_i h_j, \forall h = (h_1, \dots, h_n) \in \mathbb{R}^n.$$

On note cette forme quadratique $d^2 f(a)$.

Proposition A.2.6. (Conditions suffisantes) : Considérons une fonction f de classe C^2 sur un ouvert Ω de \mathbb{R}^n , $a \in \Omega$. $f : \Omega \rightarrow \mathbb{R}$, avec $df(a) = 0$ et donc :

- 1/ $d^2 f(a)$ une forme quadratique définie positive, alors f possède un minimum local en a .
- 2/ $d^2 f(a)$ une forme quadratique définie négative, alors f possède un maximum local en a .
- 3/ $d^2 f(a)$ une forme quadratique non définie, f n'admet pas d'optimum en a .

Proposition A.2.7. (Conditions suffisantes dans \mathbb{R}^2) : Considérons une fonction f de classe C^2 sur un ouvert Ω de \mathbb{R}^2 , $a \in \Omega$. $f : \Omega \rightarrow \mathbb{R}$, et tel que $\frac{\partial f}{\partial x}(a) = \frac{\partial f}{\partial y}(a) = 0$, (point critique).

Et avec les notations de Monge : $p = \frac{\partial^2 f}{\partial x^2}(a)$; $r = \frac{\partial^2 f}{\partial y^2}(a)$; $q = \frac{\partial^2 f}{\partial x \partial y}(a)$.

1/ $pr - q^2 > 0$; $p > 0$ alors a minimum local.

2/ $pr - q^2 > 0$; $p < 0$ alors a maximum local.

3/ $pr - q^2 < 0$ alors a point selle.

4/ $pr - q^2 = 0$ alors on ne peut pas conclure et on doit faire une étude approfondie.

Théorème A.2.1. Considérons une fonction f de classe C^1 sur un ouvert Ω de \mathbb{R}^n , $a \in \Omega$. $f : \Omega \rightarrow \mathbb{R}$ et une autre fonction g de classe C^1 sur Ω . $g : \Omega \rightarrow \mathbb{R}^p$. Soit a le point optimum pour f sous contraintes que $g(x) = 0$, et de plus la matrice jacobienne $J_g(a)$ de g au point a est de rang p . Et donc il existe des constantes $\lambda_1, \dots, \lambda_p$ appelées multiplicateurs de Lagrange avec :

$$\text{grad } f(a) = \sum_{k=1}^p \lambda_k \text{grad } g_k(a).$$

Ou bien

$$\frac{\partial f}{\partial x}(a) = \sum_{k=1}^p \lambda_k \frac{\partial g_k}{\partial x}(a).$$

Définition A.2.15. (Domaine d'une fonction)

Soit $f : D \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe. On appelle domaine de f noté $\text{dom}(f)$, l'ensemble

$$\text{dom}(f) = \{x \in D, / f(x) < +\infty\}.$$

C'est un ensemble convexe.

Si f concave alors son domaine

$$\text{dom}(f) = \{x \in D, / f(x) > -\infty\}.$$

Définition A.2.16. (Fonction coercitive)

Soit $f : D \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction, f est dite coercitive si elle vérifie

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

Bibliographie

- [1] E. Aart J. Korst, *Simulated annealing and Boltzman machines : A stochastic approach to combinatorial optimization and neural computing*, John Willey - Sons New York, (1997).
- [2] M. Achache, H. Roumili, A. Keraghel, *A numerical study of an infeasible primal-dual path-following algorithm for linear programming*, Applied Mathematics and Computation, Volume 186, 2, 1472-1479, (2007).
- [3] D. Aaid, Noui A, Ouanes M, *New technique for solving univariate global optimization*, Archivum Mathematicum, Vol. 53 No. 1, 19933(2017), (2017).
- [4] Djamel. Aaid, *Étude et mise en oeuvre d'algorithmes d'optimisation pour résoudre certaines classes de problèmes pratiques*, Doctorat Univ Constantine, (2017).
- [5] Mme Abassi Nacéra née Ikhenche, *Méthode de support pour la minimisation d'un fonction Quadratique convexe*, Thèse Magister Univ Bejaia Lab Lamos DZ, (2014).
- [6] Claire.S. Adjiman, Christodoulos A. Floudas, *Rigorous Convex Underestimators for General Twice-Differentiable Problems*, Journal of Global Optimization 9 : 23-40, (1996).
- [7] Farid. Aiche, *Sur la programmation linéaire multi-Objectifs floue Stochastique*, Doctorat Université de Tizi Ouzou, (2013).
- [8] M. Aidene, I. L. Vorobev, and B. Oukacha, *Algorithm for solving a linear optimal control problem with minimax*, Performance Index Computational Mathematics and Mathematical Physics 45. vol 10, pp. 1691-1700, (2005).
- [9] Mira Al Kharboutly, *Résolution d'un problème quadratique non convexe avec contraintes mixtes par les techniques de l'optimisation D.C*, Doctorat Université Le Havre, (2018).
- [10] Moustapha Alhadj Dibo, *Validation de données et diagnostic des systèmes incertains à l'aide de l'analyse par intervalle*, Doctorat Institut National Polytechnique de Lorraine Nancy, (2005).
- [11] I.P. Androulakis, C.D. Marinas, C.A. Floudas, *A global optimization method for general constrained nonconvex problems*, J. Glob. Optim, 7, 337-3637, (1995).
- [12] S. Bazarra, H.D. Sherali and C.M. Shetty, *Nonlinear programming, theory and algorithms*, Second edition, (1993).
- [13] M. Bardi and I. Capuzzo-Dolcetta, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Birkhäuser. Inc. Boston, (1997).
- [14] G. Barles, *Solutions de viscosité des équations de Hamilton-Jacobi*, Math. Appl. Springer-Verlag. 17, (1994).

- [15] C. Bennani, *Stabilisation et Estimation de l'état des Systèmes dynamiques non Linéaires et application*, Mémoire de Magister (UMMTO). **3**, pp. 41-50, (2011).
- [16] Alain Billionnet, *Optimisation Discrète*, Edition Dunod, (2007).
- [17] Michel Bierlaire, *Introduction à l'Optimisation différentiable*, Presse Polytechnique et universitaire romandes, (2006).
- [18] Mme Bekri Houria, *Optimisation Globale avec applications pour les fonctions Holderiennes*, Thèse Magister Univ UMMTO, (2012).
- [19] F Bonnans - JC Gilbert-C Lemarechal et C.Sagastizabal, *Optimisation numérique : aspects théoriques et pratiques*, Springer Berlin , (1997).
- [20] F Bonnans, *Optimisation Continue :cours et problèmes corrigés*, Dunod Paris, (2006).
- [21] D Bertsekas, *Nonlinear programming*, Athena Scientific Belmont MA, (1999).
- [22] Brahimi, *Résolution de problèmes Quadratiques convexes par les Méthodes Primale et Duale du Support ;Applications aux SVM*, Thèse Doctorat Univ Bejaia Lab Lamos DZ, (2011).
- [23] M. Chebbah, *Résolution et implémentation d'un problème min-max en contrôle optimal*, Thèse de magister en mathematique appliquées UMMTO, (2006).
- [24] Mohammed Chebbah, Ouanes Mohand, and Zidna Ahmed, *Simulations for efficient combination of two lower bound functions in univariate global optimization*, AIP Conference Proceedings 1863, 250004 (2017) ; <https://doi.org/10.1063/1.4992412>, (2017).
- [25] Y. Cherrault, Mora G, *Optimisation globale : Théorie des courbes des courbes alpha-denses*, Economica (2005), (2005).
- [26] Y. Cherrault, *Optimisation : méthodes locales et globales*, Presses Universitaires de France (P.U.F), (1999).
- [27] Y. Chitour, F. Jean, and E. Trélat, *Propriétés génériques des trajectoires singulières*, C R Math Acad Sci Paris. **337**, pp. 49-52, (2003).
- [28] Y. Chitour, F. Jean, and E. Trélat, *Genericity results for singular curves*, J Differential Geom. **73**, pp. 45-73, (2006).
- [29] Y. Chitour, F. Jean, and E. Trélat, *Singular trajectories of control affine systems*, Siam J Control Optim. **vol 2**, pp. 1078-1095, (2008).
- [30] Y Colette et P Siarry, *Optimisation multiobjectif*, Eyrolles Paris, (1994).
- [31] E Chong - S Zak, *An introduction to Optimisation*, John Willey - Sons New York, (1995).
- [32] P G Ciarlet, *Introduction à l'analyse numérique matricielle et à l'Optimisation*, journal= Masson Paris, (1994).
- [33] M.G. Crandall and P.L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Trans. Amer. Math. Soc. **277**, pp. 1-42, (1983).
- [34] Jean Christophe Culioli, *Introduction à l'Optimisation. 2ème Edition*, Edition ellipses, (2012).
- [35] C. De Boor, *Applied Mathematical Sciences*, Springer verlag (1978), (1978).

- [36] C De Boor, *A practical Guide to Splines Applied Mathematical Sciences*, Springer-Verlag, (1978).
- [37] J Dennis - R Schnabel, *Numerical methods for unconstrained Optimization and nonlinear equations*, Prentice Hall, Englewood cliffs - Sons Nj, (1983).
- [38] Jacques Douchet, *Analyse, Recueil d'exercices et aide mémoire vol 2*, Presses Polytechniques et universitaires romandes, (2004).
- [39] Jacques Douchet, *Tout Analyse de la licence*, Dunod, (2014).
- [40] G. Faraut et Y. Zylberberg, *Equation de Hamilton-Jacobi*, (2004).
- [41] R. Fletcher, *Practical methods of Optimization*, John Willey - Sons New York, (1987).
- [42] C.A. Floudas and C.E. Gounaris, *A review of recent advances in global optimization*, J Glob Optim., 45, 3-38(2009), (2009).
- [43] Christodoulos A. Floudas, *Deterministic Global Optimization : Advances in Theory and Applications*, Cours Princeton University USA, (2007).
- [44] C. A. Floudas, Gounaris, C. E, *A review of recent advances in global optimization*, J Glob Optim DOI 10.1007/s10898-008-9332-8, (2008).
- [45] C.E. Gounaris and C.A. Floudas, *Tight convex underestimator for C^2 -continuous problems : I. Univariate functions*, J Glob Optim, 45, 3-38(2009), (2009).
- [46] Chrysanthos E. Gounaris, Christodoulos A. Floudas, *Tight convex underestimators for C^2 -continuous problems : I. univariate functions*, Glob Optim (2008), (2008).
- [47] E. Gourdin, B. Jaumard and R. Ellaia, *Global Optimization of Hölder Functions*, J. Global Optim. 8 (1996) 3233348, (1996).
- [48] R. Gabasov and F. M. Kirillova, *Adaptive methods of solving linear programming problems*, Preprint Series of University of Karlsruhe. Institute for Statistics and Mathematics, (1994).
- [49] Le Thi H.A., Ouanes M., Zidna A, *Computing real zeros of a polynomial by branch and bound and branch and reduce algorithms*, Yugoslav Journal of Operations Research (2014)24,53-69, (2014).
- [50] Le Thi Hoai An, Mohand Ouanes, Ahmed Zidna, *An Adapted Branch and Bound Algorithm for Approximating Real Root of a Polynomial*, MCO 2008 : 182-189, (2008).
- [51] L.T. Hoai An and M. Ouanes, *Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint*, RAIRO Oper. Res., 40,285-302(2006), (2006).
- [52] Le Thi Hoai An and Mohand Ouanes, *Convex quadratic underestimation and Branch and Bound for univariate global optimization with one nonconvex constraint.*, RAIRO 2006, (2006).
- [53] Richard. J. Hillestad and Stephen E. Jacobsen, *Reverse Convex Programming*, Appl. Math. Optim. 6, 63-78, (1980).
- [54] Jean-Baptiste Hiriart-Urruty, *Bases, outils et principes pour l'analyse variationnelle*, Springer, (2012).

- [55] Jean-Baptiste Hiriart-Urruty, *Optimisation et analyse convexe : Exercices et problèmes corrigés, avec rappels de cours*, EDP, (2009).
- [56] Reiner Horst, *On solving general reverse convex programming problems by a sequence of linear programs and line searches*, Annals of Operations Research, 25 (1990) 1-18, (1990).
- [57] Professor Dr. Reiner Horst, Professor Dr. Hoang Tuy, *Global Optimization Deterministic Approaches*, Springer Berlin Heidelberg, (1990).
- [58] R. Ignat, *Equation de Hamilton-Jacobi et Contrôle Optimal*, Seminar on Fixed Point Theory Cluj-Napoca. **Vol 3**, pp. 239-248, (2002).
- [59] F.Kacher, *Concept d'équilibre pour un jeu non coopératif sous forme Normale avec Paramètres Indéterminés flous*, Thèse Doctorat Univ Tizi ousou DZ, (2006).
- [60] F. Kacher, *Sur l'équilibre pour un jeu non coopératif sous forme normale avec paramètres indéterminés flous*, Cours Univ Tizi Ouzou, (2006).
- [61] R.E. Kalman, *Mathematical description of linear dynamical systems*, J. Siam. control. pp. 152-192, (1963).
- [62] M. Abelkrim Kareghel, *Étude adaptive et comparative des principales variantes dans l'Algorithme de Karmarkar*, Thèse de Doctorat en mathématique appliquées Univ Joseph fourier Grenoble I, (1989).
- [63] Zakia. Kebbiche, *Étude et Extension d'Algorithmes de Points Interieurs pour la programmation non linéaire*, Doctorat Université Le Havre - Université Sétif Co-tutelle, (2007).
- [64] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica 4, 373-395, (2005).
- [65] Z. Kebbiche, A. Keraghel, A. Yassine, *Extension of a projective interior point method for linearly constrained convex programming*, International conference Applied Mathematics and Computation, (2007).
- [66] Z. Kebbiche, A. Keraghel A. Yassine, *An infeasible interior point method for the monotone linear complementarity problem*, International Journal of Mathematical Analysis, Vol. 1, 2007, No 17, 841-849, (2007).
- [67] Z. Kebbiche, A. Keraghel, A. Yassine, *Extension of a projective interior point method for linearly constrained convex programming*, Applied Mathematics and Computation (2007), (2007).
- [68] Hiroshi Konno, Phan Thien Thach, Hoang Tuy, *Optimization on Low Rank Nonconvex Structures*, Springer US (1997), (1997).
- [69] Arnaud Henry Labordere, *Cours de recherche Opérationnelle*, Presse des Pont des Chaussées, (1995).
- [70] Ahmed Lasfari, *Notions Fondamentales d'analyse Mathématiques*, Edition ellipses, (2014).
- [71] Han-Lin Li, Ching-Ter Chang, *An approximate approach of global optimization for polynomial programming problems*, European Journal of Operational Research 107 (1998) 625-632, (1998).

- [72] K. Louadj, *Résolution de problèmes paramétrés de contrôle optimal*, Thèse de doctorat UMMTO, pp. 25-50, (2012).
- [73] Andreas Lundell, Joakim Westerlund, Tapio Westerlund, *Some transformation techniques with applications in global optimization*, Glob Optim (2009) 43 :391-405, (2009).
- [74] G.P. McCormick, *Computability of global solutions to factorable nonconvex programs : Part I-Convex underestimating problems*, Mathematical Programming, 10 :147-175, (1976).
- [75] G.P. McCormick, *Converting general nonlinear programming problems to separable nonlinear programming problems*, Technical Report T-267, The George Washington University, Washington D.C., (1972).
- [76] A. Merakeb, *Optimisation multicritères en contrôle Optimal. Application au Voiture électrique*, Thèse de doctorat univ Tizi-Ouzou, (2014).
- [77] Frédéric Messine, *L'Optimisation Globale par Intervalles : de l'Etude Théorique aux Applications*, Habilitation à Diriger les Recherches UNV Toulouse, (2006).
- [78] Frederic Messine, *Méthodes d'Optimisation Globale basées sur l'analyse d'intervalles pour la résolution de problèmes avec contraintes*, Doctorat Université Toulouse Paul Sabatier, (1997).
- [79] Mahdi. Morini, *La programmation DC et DCA pour l'optimisation de portefeuille*, Doctorat Université Metz, (2008).
- [80] Nacima. Moussouni, *Contrôle optimal : Optimisation d'une production céréalière*, Doctorat Co-tutelle Univ Paris 6 Univ Tizi Ouzou, (2012).
- [81] Michel Minoux, *Programmation mathématique Théories et Algorithmes*, 2ème édition, TEC et DOC., (2009).
- [82] Khosrow Moshirvaziri, Mahyar A. Amouzegar, *A Cutting Plane Algorithm for Linear Reverse Convex Programs*, Annals of Operations Research 105, 201-212, 2001 / 2002 Kluwer Academic Publishers. Manufactured in The Netherlands, (2001).
- [83] Gill.W. Murray - M Wright, *Practical Optimization*, Academic Press New York, (1987).
- [84] A. Nemirovskii, K. Scheinberg, *Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems*, Mathematical Programming : Series A 72(3) : 273-289, (1996).
- [85] Pham Viet Nga, *Programmation DC et DCA pour l'optimisation non convexe/optimisation globale en variables mixtes entières. Codes et Applications*, Doctorat Université Metz, (2013).
- [86] Jordan Ninin, *Optimisation Globale basée sur l'analyse d'intervalles, Relaxation Affine et limitation de Mémoire*, Doctorat Université Toulouse Paul Sabatier, (2010).
- [87] B. Oukacha, *Résolution de problème de controle optimal*, Thèse de doctorat univ Tizi-Ouzou, (2005).
- [88] Mohand Ouanes, Hoai An Le Thi, Trong Phuc Nguyen, Ahmed Zidna, *New quadratic lower bound for multivariate functions in global optimization*, Mathematics and Computers in Simulation, (2015) 109, 197-211, (2015).

- [89] Mohand Ouanes, Hoai An Le Thi, Tronc Phuc Nguyen and Ahmed Zidna., *New Quadratic lower Bound for multivariate functions in Global Optimization.*, International conference, (2015).
- [90] Mohand. Ouanes, Chebbah Mohammed, and Zidna Ahmed, *Combination of two underestimators for univariate global optimization*, RAIRO 2018, (2018).
- [91] L.S. Pontrygin, V.G Boltyaki, R.V. Gamekrelidze and E.F Mishchenko, *The mathematical theory of optimal processes*, Interscience publishers, New york, (1962).
- [92] L. Pontryagin and al, *Mathematical theory of optimal processes*, Eds Mir Moscou, pp. 504-1974, (1974).
- [93] Hayet Roumili, *Méthodes de points Intérieurs non réalisables en optimisation : théories, algorithmes et applications*, Doctorat Cotutelle Univ Le Havre - Univ Sétif, (2007).
- [94] D.G. Sotiropoulos and T.N. Grapsa, *Optimal centers in branch-and-prune algorithms for univariate global optimization*, Applied Mathematics and Computation, 169 (2005), pp.247-277, (2005).
- [95] Springer-Verlag Berlin Heidelberg, *Inexactness and Optimization*, Springer-Verlag Berlin Heidelberg, (2008).
- [96] Xiaoling Sun, Ken McKinnon, Duan Li, *A Convexification Method for a Class of Global Optimization Problems with Applications to Reliability Optimization*, Research Grants Council of Hong Kong, the National Science Foundation of China, (2000).
- [97] N. Toutzi, *Optimisation Dynamique*, Ecole Nationale de la Statistique et de l'Administration Economique, (2003).
- [98] E. Trélat, *Contrôle optimal : théorie et applications*, Université Pierre et Marie Curie (Paris 6) et Institut Universitaire de France Laboratoire Jacques-Louis Lions CNRS, UMR 7598 4 place Jussieu, BC 187 75252 Paris cedex 05, (2013).
- [99] Jacques Teghem, *Recherche Opérationnelle Tome 2 Gestion de Production Modèles Aléatoires Aide Multicritère*, Ellipse, (2013).
- [100] J. Teghem, *Programmation linéaire*, Edition Ellipses SMA éditions de l'université de Bruxelles, (2005).
- [101] Hoang Tuy, *Convex Analysis and Global Optimization*, Springer US (1998), (1998).
- [102] H. TuY, *Convex Programs with an Additional Reverse Convex Constraint*, Journal of optimization theory and applications : Vol. 52, No, 3, MARCH 1987, (1987).
- [103] Hoang Tuy, *Canonical DC programming problem : Outer approximation methods revisited*, Operations Research Letters 18 (1995) 99-106, (1995).
- [104] Nguyen Quang Thuan, *Approches locales et globales basées sur la programmation DC et DCA pour des problèmes combinatoires en variables mixtes 0-1. Applications à la planification opérationnelle*, Doctorat Université Metz, (2010).
- [105] M.J. Todd, B. P. Burell, *An extension of Karmarkar's algorithm for linear programming using dual variables*, Algorithmica 1 : 409-424 (1986), (1986).

- [106] S.J. Wright, *Primal-dual interior point methods*”, Siam. Philadelphia, (1997).
- [107] Y. Ye E. Tse, *An extension of Karmarkar’s projective algorithm for convex quadratic programming*, Mathematical Programming 44 157-179, (1989).
- [108] Ouiza Zerdani épouse Bouarab, *L’Optimisation non Linéaire Multiobjectif*, Doctorat Univ Tizi Ouzou, (2013).

RÉSUMÉ

L'optimisation globale a reçu une grande attention de la part des chercheurs et elle a été étudiée intensivement ces dernières années dans la littérature. Ceci s'explique par ses applications intéressantes dans divers domaines à savoir le contrôle des flux routiers, télécommunication, raffinage pétrolier,...etc. Dans notre travail, nous avons effectué un état de l'art sur la théorie de l'optimisation globale (convexité, non convexité, α B.B, méthodes de résolutions numériques,...etc) tout en étudiant des méthodes linéaires (la linéarisation par exemple). Des exemples d'applications pour illustrations.

Mots clés : convexité, non convexité, branch and Bound, méthode D.C et D.C.A, méthodes de résolutions numériques, méthodes en multi-objectifs, commande optimale, cas discret et cas continu.

ABSTRACT

The Global Optimization theory has received much attention from researchers and it has been intensively studied in recent years. This is explained by its interesting applications in various fields namely the control of traffic flows, telecommunications, oil refining,...etc. In our work, we have conducted a state of the art in the field of Global Optimization (convexity, no convexity, α B.B, numerical resolution methods,...etc.) while studying a linear methods (The linearization for example). Examples of applications for illustrations.

Key words : convexity, no convexity, Branch and Bound, D.C and D.C.A Method, numerical resolution methods, multi-criteria methods, optimal command, discrete case, continuous case.