REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE AMAR TELIDJI LAGHOUAT
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

Mémoire de Master

Domaine : Mathématiques et Informatique
Filière : Mathématiques
Option : Analyse Mathématique

Présenté par :

Belgharbi Ibrahim

Thème

Résolution des problèmes d'optimisation quadratique non-convexe par la programmation DC

Soutenu publiquement devant le jury composé de :

Mr. A. Belacel	M.C.A. Université de Laghouat	**Président**
Mr. A. Mokhtari	Prof. Université de Laghouat	**Examinateur**
Mme. Y.Boukhatem	M.C.A. Université de Laghouat	**Examinateur**
Mr. M. Bentobache	M.C.A. Université de Laghouat	**Encadreur**
Mr. S. Achour	Doctorant. Université de Laghouat	**Co-encadreur**

Année universitaire 2018/2019

Remerciement

Je remercie Allah le tout puissant de m'avoir donné le courage, la volonté et la patience de mener à terme ce présent travail.

Je tiens tout d'abord à exprimer mes sincères remerciements à mon encadreur Dr. Mohand Bentobache pour ses orientations et tous ses efforts consentis dans la préparation de ce travail.

Je remercie vivement mon co-encadreur M. Saadi Achour pour toute la documentation qu'il m'a fournie, ses conseils judicieux, son aide précieuse et les efforts qu'il a déployés pour l'accomplissement de ce travail.

Je remercie vivement Professeur Abdelkader Mokhtari pour avoir accepté d'examiner ce travail et aussi pour ses encouragements et ses précieux conseils. Je tiens également à remercier Dr. Amar Belacel d'avoir accepter de présider le jury de soutenance ainsi que Dr. Yamna Boukhatem pour avoir accepter d'examiner ce mémoire.

Un très grand merci à mes parents et toute ma famille sans oublier tous mes chers enseignants pour les efforts qu'ils ont fournis pour avoir participer à notre formation.

Enfin, mes meilleurs et vifs remerciements s'adressent à tous mes amis et tous ceux qui m'on aidé de près ou de loin à la réalisation de ce modeste travail.

الملخص

في هذه الرسالة ، نحن مهتمون بتطبيق خوارزمية فرق دالتين محدبتين لإيجاد القيمة الدنيا لدالة تربيعية غير محدبة على مجموعة محددة بواسطة قيود خطية. لهذا الغرض، قمنا بتطوير تطبيق خوارزمية فرق دالتين محدبتين باستخدام لغة برمجة الماطلاب. لإظهار أهمية اختيار النقطة الأولية لخوارزمية فرق دالتين محدبتين، قمنا بتزويد هذه الخوارزمية بنقطتين أوليتين: النقطة المتمثلة في المبدأ والنقطة المتمثلة في حل مشكلة إيجاد القيمة الدنيا لدالة الجزء الخطي. تظهر النتائج العددية لتطبيق هذه الخوارزمية لحل 40 مشكل رياضي تربيعي بعدد من المتغيرات تتراوح من 4 إلى 1000، التي تم إنشاؤها عشوائيا، أنه من المستحسن اختيار النقطة الأولية كحل لمشكلة إيجاد القيمة الدنيا لدالة الجزء الخطي.

Abstarct

In this work, we are interested in the application of the DCA algorithm to optimize non-convex quadratic functions subject to linear constraints. For this purpose, we have developed an implementation of the DCA algorithm with the MATLAB2018a programming language. In order to show the importance of the choice of the initial point for the DCA algorithm, we initialized this algorithm with two initial points: the origin and the point corresponding to the solution of the problem of minimization of the linear part. Numerical results on test problems randomly generated with a number of variables varying from 4 to 1000 show that it is more efficient to choose the initial point as the solution of the minimization problem of the linear part of the quadratic form subject to the linear constraints of the original problem.

Résumé

Dans ce mémoire, nous nous sommes intéressés à l'application de l'algorithme DCA pour optimiser les fonctions quadratiques non-convexes sur un domaine délimité par des contraintes linéaires. A cet effet, nous avons développé une implémentation de l'algorithme DCA avec le langage de programmation MATLAB2018a. Afin de montrer l'importance du choix du point initial pour l'algorithme DCA, nous avons initialisé cet algorithme avec deux points initiaux: l'origine et le point correspondant à la solution du problème de minimisation de la partie linéaire. Les résultats numériques sur des problèmes-test générés aléatoirement avec un nombre de variables variant de 4 à 1000 montrent qu'il est judicieux de choisir le point initial comme étant la solution du problème d'optimisation minimisant la partie linéaire de la forme quadratique sous les contraintes du problème original.

Table des matières

Introduction

La programmation quadratique est une branche d'optimisation non linéaire qui consiste à optimiser (minimiser ou maximiser) une fonction objectif quadratique sous des contraintes qui définissent le domaine des solutions réalisables. La programmation quadratique est une discipline très importante et son importance réside dans ses applications dans différents domaines de la vie courante [20, 11, 9, 12, 13]. En effet, plusieurs problèmes réels et académiques peuvent être modélisés sous forme d'un programme quadratique.

Les nombreuses recherches dans le domaine de la résolution des programmes mathématiques ont montré une distinction majeure entre les problèmes convexes et les problèmes non-convexes. La notion de convexité joue un rôle majeur dans la programmation mathématique. Lorsque la fonction objectif et le domaine réalisable sont convexes, on parle de problèmes de programmation quadratique convexe qui sont des problèmes plus simples à résoudre car tout optimum local est global. Cependant, les problèmes de programmation quadratique non-convexe sont difficiles à résoudre, et surtout quand la dimension du problème devient grande.

En réalité, le grand tournant de l'optimisation ne réside pas entre la linéarité et la non-linéarité, mais entre la convexité et la non-convexité. Ainsi, les méthodes de résolution sont généralement différentes selon la convexité des programmes à résoudre.

La convexité d'une fonction quadratique f s'obtient par l'étude des valeurs propres de la matrice Hessienne $H = \nabla^2 f(x)$. Si la matrice H

est semi-définie positive (i.e. toutes ses valeurs propres sont positives ou nulles), alors la fonction f sera dite convexe. Dans le cas contraire, elle est dite non-convexe.

Le but de ce mémoire est d'étudier les programmes quadratiques non-convexe de la forme :

$$(PQ1) \quad \min\{f(x) = \frac{1}{2}x^T Qx + c^T x, x \in \Omega\},$$

où

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

Q est une matrice symétrique carrée d'ordre n semi-définie négative ou indéfinie, A est une matrice de dimension $(m \times n)$ et $b \in \mathbb{R}^m$. Ces problèmes sont non-convexes car la matrice Q n'est pas semi-définie positive.

Il existe plusieurs méthodes pour la résolution du problème de programmation quadratique non-convexe [14, 17, 18, 13, 6, 7, 8]. Cependant, nous allons nous intéresser dans ce mémoire à l'application des techniques de l'optimisation **DC** (Différence de deux fonctions Convexes) et de l'algorithme **DCA** [20, 19, 15, 2, 21, 3]. Cette approche est basée sur les outils d'analyse convexe, la dualité **DC** et l'optimalité locale en optimisation **DC**. Son principe consiste à construire deux suites $\{x^k\}$ et $\{y^k\}$ qui convergent respectivement vers une solution optimale du problème primal et une solution optimale du problème dual et à chaque itération de l'algorithme, un problème quadratique convexe est résolu.

Organisation du mémoire

Notre mémoire est organisé de la façon suivante :

1. Dans le premier chapitre, nous allons donner quelque généralités sur l'analyse convexe.
2. Dans le deuxième chapitre, nous allons définir les fonctions **DC** et leurs propriétés essentielles. Ensuite, nous allons aborder les fonctions localement **DC**.
3. Dans le troisième chapitre, après une brève introduction à la programmation **DC** (Différence de deux fonctions Convexes) et l'algorithme **DCA**, nous détaillerons les bases théoriques et algorithmiques de la programmation **DC** et **DCA**. Nous présenterons la programmation DC, la notion de la dualité et les conditions d'optimalité locale en programmation **DC** sur lesquelles est basé l'algorithme **DCA**. Nous terminerons par les résultats de convergence de **DCA** et la pénalisation exacte en programmation **DC**.
4. Dans le quatrième chapitre, nous allons effectuer une étude expérimentale qui permettra de comparer deux variantes de l'algorithme DCA pour la résolution de plusieurs problèmes-test générés alétoirement.

Chapitre 1

Introduction à l'analyse convexe

1.1 Éléments d'analyse convexe

Introduction

Dans cette section, nous allons rappeler quelques définitions et théorèmes d'analyse convexe qui fondent la base de la programmation DC. Pour plus de détails sur l'analyse convexe, on pourra se référer par exemple à l'ouvrage [21, 3].

Soit l'espace euclidien $\mathbb{R}^n$ muni du produit scalaire usuel

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i y_i = x^T y$$

et de la norme euclidienne associée $||x||_2 = \sqrt{\langle x, y \rangle}$, où x et y sont deux vecteurs dans $\mathbb{R}^n$ et x^T est le transposé du vecteur x.

L'analyse convexe permet aux fonctions de prendre les valeurs $\pm\infty$. On notera $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ qui est muni d'une structure algébrique déduite de manière naturelle, avec la convention $(+\infty) - (+\infty) = +\infty$.

1.1.1 Ensemble convexe

Définition 1.1.1. *Soit C une partie de $\mathbb{R}^n$. On dit que C est convexe si :*

$$\forall x, y \in C, \forall \lambda \in [0,1] : \lambda x + (1-\lambda)y \in C.$$

C'est à dire, pour tout x et y de C, le segment $[x,y]$ est tout entier contenu dans C.

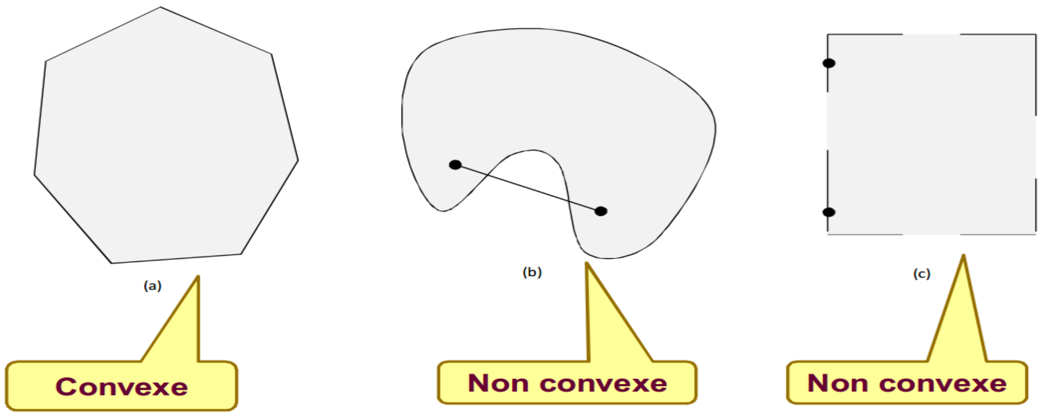

FIGURE 1.1 – Ensembles convexes et ensembles non-convexes

Géometriquement, le point $z(\lambda) = \lambda x + (1-\lambda)y$ est le barycentre des points x et y affectés des coefficients respectifs λ et $1-\lambda$. Lorsque λ décrit l'intervalle $[0,1]$, $z(\lambda)$ décrit le segment fermé d'extrémités x et y. Dire que C est convexe, c'est donc dire que C contient un segment dès qu'elle contient ses extrémités.

Propriété 1.1.1.

1. *Soient C_1 et C_2 deux ensembles convexes de $\mathbb{R}^n$, alors* $C_1 + C_2 = \{x \in \mathbb{R}^n / x = x_1 + x_2, x_1 \in C_1, x_2 \in C_2\}$ *est convexe.*
2. *Soit C un ensemble convexe de $\mathbb{R}^n$, alors l'ensemble αC est convexe, tel que :*

$$\alpha C = \{\alpha x / x \in C\}, \forall \alpha \in \mathbb{R}.$$

3. *Soient $C_1, C_2, ..., C_n$ des ensembles convexes de $\mathbb{R}^n$, alors C est convexe où :*
$$C = \bigcap_{i=1}^{n} C_i.$$
4. *En général, une réunion quelconque des convexes n'est pas convexe.*

1.1.2 Combinaison convexe

Définition 1.1.2. *Soient $x_1, ..., x_n$ des vecteurs de $\mathbb{R}^n$. Le vecteur x est appelé une combinaison linéaire convexe des vecteurs $x_i, i = 1, ..., n$ si :*
$$x = \sum_{i=1}^{n} \lambda_i x_i,$$
tel que :
$$\sum_{i=1}^{n} \lambda_i = 1, \lambda_i \in \mathbb{R}_+, \forall i = 1, ..., n.$$

Proposition 1.1.1. *Soit C un convexe de $\mathbb{R}^n$, pour toute famille finie $x_1, ..., x_n$ de vecteurs de C et tout système $\lambda_1, ..., \lambda_n$ des réels positifs ou nuls avec $\sum\limits_{i=1}^{n} \lambda_i = 1$, le point $x = \sum\limits_{i=1}^{n} \lambda_i x_i$ appartient à C.*

Démonstration. Montrons par récurrence sur $\mathbb{N}^*$ que $x \in C$, c'est évident si $n = 1$.
Si $n \neq 1$, si $\lambda_1 = 1$, les autres coefficients sont nuls et il n'y a rien à montrer.
Si $\lambda_1 < 1$, alors $\sum\limits_{i=2}^{n} \lambda_i = 1 - \lambda_1 > 0$.
Posons $\mu_i = \frac{\lambda_i}{1-\lambda_1}$, on a $\sum\limits_{i=2}^{n} \mu_i = 1$, d'où $y := \sum\limits_{i=2}^{n} \mu_i x_i \in C$ (hypothèse de récurrence) et donc $x = \lambda_1 x_1 + (1 - \lambda_1) y \in C$. □

1.1.3 Projection sur un ensemble convexe

Définition 1.1.3. *Soit $C \subset \mathbb{R}^n$ un ensemble convexe fermé non vide et soit $x \in \mathbb{R}^n$.*

Il existe alors un point unique $P_c(x) \in C$ appelé projection de x sur C, défini par :

$$d_c(x) = \| x - P_c(x) \| = min\{\| x - y \| : y \in C\}.$$

1.1.4 Enveloppe convexe

Définition 1.1.4. *Soit S une sous ensemble de $\mathbb{R}^n$. L'enveloppe convexe de S noté $Co(S)$, est l'intersection de tous les sous-ensembles convexes de $\mathbb{R}^n$ qui contiennent S et aussi est le plus petit ensemble convexe de $\mathbb{R}^n$ qui contient S $(S \subset Co(S))$.*

Si C est un convexe contenant S, il contient $Co(S)$.

Proposition 1.1.2. *L'enveloppe convexe de S, est l'ensemble de toutes les combinaisons convexes finies d'éléments de S :*

$$Co(S) = \{x = \sum_{i=1}^{n} \lambda_i x_i : x_i \in S, \lambda_i \geq 0, \forall i = 1, ..., n; \sum_{i=1}^{n} \lambda_i = 1\}.$$

Démonstration. Comme $Co(S)$ est convexe, cet ensemble contient les combinaisons convexes finies de ses éléments (Propostion 1.1.1), donc des éléments de S.

Il est facile de voir que l'ensemble

$$C = \{\sum_{i=1}^{n} \lambda_i x_i : n \geq 1, x_i \in S, \lambda_i \geq 0, \sum_{i=1}^{m} \lambda_i = 1\}$$

est convexe ; il contient S, donc aussi $Co(S)$. □

Proposition 1.1.3.

1. *L'enveloppe convexe fermé d'une partie S de $\mathbb{R}^n$ est l'adhérence de son enveloppe convexe. On la note $\bar{C}o(S)$.*
2. *$\bar{C}o(S)$ est le plus petit convexe fermé de $\mathbb{R}^n$ contenant S.*
3. *Si S est un ensemble fini, $Co(S)$ est compact.*
4. *Plus généralement si $K_1, ..., K_p$ sont des convexes compacts, $Co(K_1 \cup ... \cup K_p)$ est compact.*

Dans la suite, soit C un ensemble convexe de $\mathbb{R}^n$, et soit la fonction $f : C \to \mathbb{R}$.

1.1.5 Fonction convexe

Définition 1.1.5. *On dit que la fonction f est convexe sur C si pour tout x et y de C et pour tout $\lambda \in [0,1]$:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y).$$

Si l'inégalité est stricte pour tout $\lambda \in]0,1[$ et pour tout x et y de C avec $x \neq y$, alors f est dite strictement convexe.

Cette définition est aussi valable pour $f : C \to \mathbb{R} \cup \{+\infty\}$.

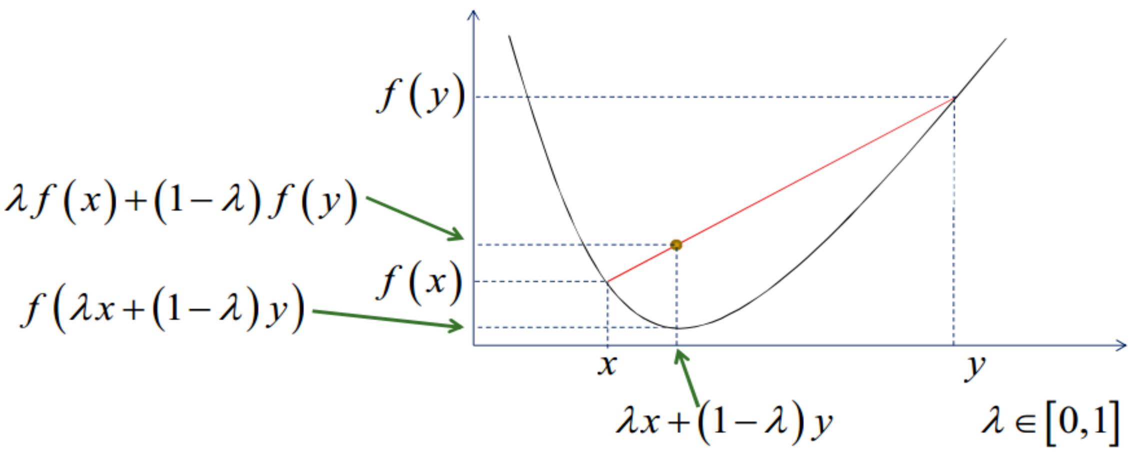

FIGURE 1.2 – Fonction convexe

1.1.6 Fonction concave

Définition 1.1.6. *On dit que la fonction f est concave sur C si pour tout x et y de C et pour tout $\lambda \in [0,1]$:*

$$f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y).$$

1.1.7 Fonction fortement convexe

Définition 1.1.7. *On dit que la fonction f est fortement convexe de constante (module) ρ sur l'ensemble convexe C (autrement dit ρ-convexe) si :* $\exists \rho > 0, \forall x, y \in C, \forall \lambda \in [0,1]$:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \lambda(1-\lambda)\frac{\rho}{2} \parallel x - y \parallel^2 .$$

Cela revient à dire que la fonction $f - \frac{\rho}{2} \parallel . \parallel^2$ est convexe sur C. Le module de forte convexité de f sur C, noté $\rho(f, C)$, est défini par :

$$\rho(f, C) = sup\left\{\rho > 0 : f - \tfrac{\rho}{2} \parallel . \parallel^2 \ \textit{est convexe sur } C\right\}.$$

On dit que f est fortement convexe sur C si $\rho(f, C) > 0$.

Remarque 1.1. *Toute fonction fortement convexe est strictement convexe et toute fonction strictement convexe est convexe.*
Une fonction linéaire est à la fois convexe et concave.

1.1.8 Épigraphe

Définition 1.1.8. *Soit $f : \mathbb{R}^n \to \mathbb{R}$. L'épigraphe de la fonction f noté $epi(f)$ est l'ensemble*

$$epi(f) = \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} : f(x) \leq \alpha\}.$$

La fonction f est convexe si et seulement si $epi(f)$ est un sous-ensemble convexe de $\mathbb{R}^{n+1}$.

1.1.9 Domaine effectif

Définition 1.1.9. *Soit la fonction $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, le domaine effectif de f noté $dom(f)$, est l'ensemble :*

$$dom(f) = \{x \in \mathbb{R}^n, f(x) < +\infty\} = \{x \in \mathbb{R}^n : \exists t \in \mathbb{R}, (x, t) \in epi(f)\}.$$

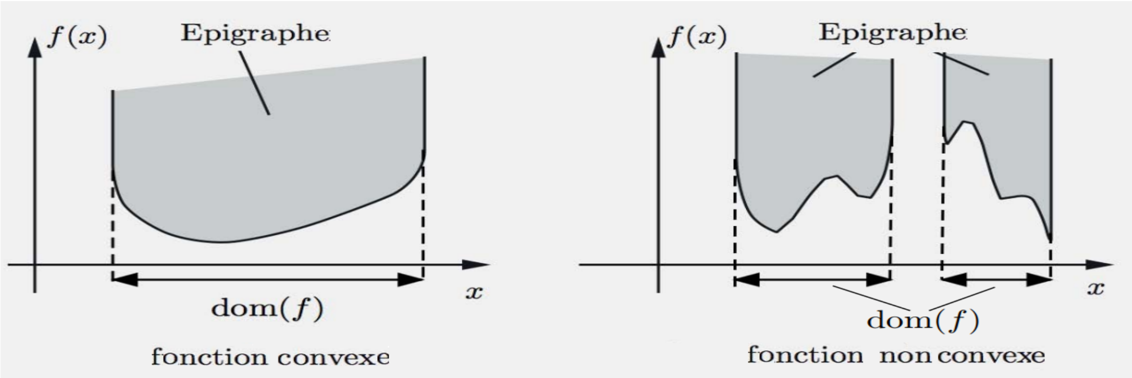

FIGURE 1.3 – épigraphe d'une fonction

- *Si f est concave alors son domaine effectif est défini comme celui de $-f$:*
$$dom(f) = \{x \in \mathbb{R}^n : f(x) > -\infty\}.$$
- *Si f est convexe alors le domaine effectif de f est convexe.*

1.1.10 Fonction propre

Définition 1.1.10. *La fonction f est dite propre si elle ne prend jamais la valeur $-\infty$ et si elle n'est pas identiquement égale à $+\infty$.*

1.1.11 Fonction coercive

Définition 1.1.11. *Une fonction $f : \mathbb{R}^n \to]-\infty, +\infty[$ est dite coercive si elle vérifie :*
$$\lim_{\|x\| \to \infty} f(x) = +\infty.$$

1.1.12 Fonction indicatrice

Définition 1.1.12. *On appelle fonction indicatrice de C, notée $\chi_C(x)$, définie par :*
$$\chi_C(x) = \begin{cases} 0 & si\ x \in C; \\ +\infty & sinon. \end{cases}$$

La fonction indicatrice d'un ensemble convexe est une fonction convexe.

Propriété 1.1.2. *Les propriétés suivantes sont équivalentes :*
1) La fonction f est convexe
2) L'épigraphe de f est convexe.
3) Le domaine effectif de f est convexe .
4) La restriction de f à son domaine effectif $f\mid_{dom(f)}$ est convexe.
5) $-f$ est concave.
6) La fonction $f+\chi_C$ est convexe sur $\mathbb{R}^n$ à valeurs dans $\mathbb{R}\cup\{+\infty\}$.

1.1.13 Fonction semi-continue inférieurement

Définition 1.1.13. *La fonction f est dite semi-continue inférieurement (s.c.i) au point $x \in \mathbb{R}^n$ si*

$$\lim_{y\to x}\inf f(y) \geq f(x).$$

On note $\Gamma_0(\mathbb{R}^n)$ l'ensemble des fonctions convexes, s.c.i. et propres sur $\mathbb{R}^n$.

Définition 1.1.14. *On dit que la fonction numérique $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ est semi-continue inférieurement (s.c.i) si pour tout $a \in \bar{\mathbb{R}}$, l'ensemble $\{x \in \mathbb{R}^n/f(x) > a\}$ est un ouvert de $\mathbb{R}^n$.Il revient au même de dire que pour tout $a \in \bar{\mathbb{R}}$, l'ensemble $\{x \in \mathbb{R}^n/f(x) \geq a\}$ est un fermé de $\mathbb{R}^n$.*

Exemple 1.1.1.

1. *Un sous-ensemble A de $\mathbb{R}^n$ est fermé si et seulement si sa fonction caractéristique χ_A est s.c.i.*
2. *Toute fonction continue est s.c.i.*
3. *Une fonction $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ est continue si et seulement si f et $-f$ sont s.c.i.*

Proposition 1.1.4. *La fonction numérique $f : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ est s.c.i si et seulement si son épigraphe est fermé, aussi si et seulement si pour tout $x \in \mathbb{R}^n$ on a $\lim_{y\to x}\inf f(y) = f(x)$.*

1.1.14 Sous-espaces affines

Définition 1.1.15. *Un sous-ensemble $A \subset \mathbb{R}^n$ est un sous-espace affine (en bref s.e.a) de $\mathbb{R}^n$ si :*

$$\forall x, y \in A, \forall \lambda \in \mathbb{R} : \lambda x + (1-\lambda)y \in A.$$

Géométriquement, cela signifie que A contient une droite D dès qu'il contient deux points (distincts) de D, un s.e.a de $\mathbb{R}^n$ est clairement un convexe.

Proposition 1.1.5. *Soient S et S_1 deux parties non vides de $\mathbb{R}^n$*

1. *Le sous-espace affine engendré par S est le plus petit s.e.a contenant S (l'intersection des s.e.a contenant S). On le notera $Aff(S)$.*
2. $Aff(S) = \{\sum_{i=1}^{n} \lambda_i x_i : x_i \in S, \lambda_i \in \mathbb{R}, \sum_{i=1}^{n} \lambda_i = 1\}$.
3. *$Aff(S)$ est égal à l'ensemble des barycentres des points de S.*
4. *Si S est convexe, $Aff(S) = \{\lambda x + (1-\lambda)y : \lambda \in \mathbb{R}, x, y \in S\}$.*
5. *S et son enveloppe convexe engendrent le même sous-espace affine.*
6. *$S \subset Aff(S), S \subset S_1 \Rightarrow Aff(S) \subset Aff(S_1)$ et $Aff(Aff(S)) = Aff(S)$.*

1.1.15 Hyperplan affine

Définition 1.1.16. *Un hyperplan affine est un sous-espace affine propre maximal.*
Autrement dit M est un hyperplan affine si et seulement si :

i) $M \neq \mathbb{R}^n$.

ii) *$\mathbb{R}^n$ est le seul s.e.a contenant M.*

Lemme 1.1.1. *Les hyperplans affines de $\mathbb{R}^n$ sont exactement les ensembles de la forme $\{x \in \mathbb{R}^n : f(x) = a\}$, ou f est une forme linéaire non nulle sur $\mathbb{R}^n$ et a un réel.*

1.1.16 Fonction affine

Une fonction est dite affine si elle est la somme d'une fonction linéaire et une constante.

1.1.17 Intérieur relatif

Définition 1.1.17. *Soit C un convexe non vide de $\mathbb{R}^n$. L'intérieur relatif de C, noté $ir(C)$, est l'intérieur de C relatif au sous-espace affine $Aff(C)$, c'est-à-dire :*

$$ir(C) = \{x \in C : \exists r > 0, B(x, r) \cap Aff(C) \subset C\}.$$

Remarque 1.2.

- *Si $x \in ir(C)$, alors x est un point relativement intérieur à C.*
- *$ir(C)$ est vide si et seulement si C l'est aussi.*

1.1.18 Fonction conjuguée

Définition 1.1.18. *Notons $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, la fonction conjuguée de la fonction $f \in \Gamma_0(\mathbb{R}^n)$ définie par :*

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

Notons aussi $f^{**} = (f^*)^*$ la bi-conjuguée de f définie par :

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

1.1.19 Polyèdre convexe

Définition 1.1.19. *Un ensemble convexe C est dit polyèdre convexe s'il est l'intersection de nombre fini de demi-espaces de $\mathbb{R}^n$:*

$$C = \bigcap_{i=1}^{m} \{x \in \mathbb{R}^n : \langle a_i, x\rangle - b_i \leq 0, a_i \in \mathbb{R}^n, b_i \in \mathbb{R}\}.$$

1.1.20 Fonction polyédrale

Définition 1.1.20. *La fonction* $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *est dite polyédrale si son épigraphe est un polyèdre convexe de* $\mathbb{R}^{n+1}$.
Notons que toute fonction polyédrale est convexe, propre et s.c.i.
Inversement si $f \in \Gamma_0(\mathbb{R}^n)$, *alors* f *est polyédrale si et seulement si* $dom(f)$ *est un polyèdre convexe et* f *peut être écrite sous la forme :*

$$f(x) = sup\{\langle a_i, x\rangle - b_i : i = 1, ..., l\} + \chi_{dom(f)}(x),$$

où $a_i \in \mathbb{R}^n, b_i \in \mathbb{R}$ *pour tout* $i = 1, ..., n$ *et* $\chi_{dom(f)}(x)$ *est la fonction indicatrice de* $dom(f)$.

1.2 Sous-différentiabilité

1.2.1 Sous-gradient

Définition 1.2.1. *Soit* f *une fonction convexe propre sur* $\mathbb{R}^n$.
Un vecteur y *dans* $\mathbb{R}^n$ *est appelé sous-gradient de* f *au point* $x_0 \in dom(f)$ *si :*

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

Interprétation géométrique :

Le graphe de la fonction affine $\psi(x) = f(x_0) + \langle y, x - x_0\rangle$ est un hyperplan de support non vertical à epi(f) au point $(x_0; f(x_0))$.
Si f est différentiable, cet hyperplan est vertical au point x_0.

1.2.2 Sous-différentiel

Définition 1.2.2. *L'ensemble de tous les sous-gradients de* f *au point* $x_0 \in dom(f)$ *est le sous-différentiel de* f *au point* x_0, *noté* $\partial f(x_0)$.

$$\partial f(x_0) = \{y \in \mathbb{R}^n / f(x) \geq f(x_0) + \langle y, x - x_0\rangle, \forall x \in \mathbb{R}^n\}.$$

Le domaine du sous-différentiel de f est

$$dom(\partial f) = \{x \in \mathbb{R}^n : \partial f(x) \neq \emptyset\}.$$

Proposition 1.2.1. *Si f est une fonction convexe sur $\mathbb{R}^n$ et $x \in dom(f)$:*

1. *$\partial f(x)$ est une partie convexe fermée de $\mathbb{R}^n$.*
2. *$ir(dom(f)) \subset dom(\partial f) \subset dom(f)$.*

1.2.3 ε-sous-gradient

Définition 1.2.3. *Soit f une fonction convexe sur $\mathbb{R}^n$, et soit ε un réel positif.*
Un vecteur $y \in \mathbb{R}^n$ est appelé ε-sous gradient de f au point $x_0 \in dom(f)$ si :

$$f(x) \geq f(x_0) + \langle y, x - x_0 \rangle - \varepsilon, \forall x \in \mathbb{R}^n.$$

1.2.4 ε-sous-différentiel

Le ε-sous-différentiel de f au point x_0, noté $\partial_\varepsilon f(x_0)$, est l'ensemble de tout les ε-sous-gradient de f au point x_0.

Propriété 1.2.1.

- $f \in \Gamma_0(\mathbb{R}^n) \Longleftrightarrow f^* \in \Gamma_0(\mathbb{R}^n)$ *et* $(f^*)^* = f$.
- $f \in \Gamma_0(\mathbb{R}^n)$ *est différentiable en x si et seulement si $\partial f(x)$ se réduit au singleton $\{\nabla f(x)\}$.*
- *Si $f \in \Gamma_0(\mathbb{R}^n)$, alors $y \in \partial f(x) \Longleftrightarrow x \in \partial f^*(y)$.*
- $f(x) + f^*(y) \geq \langle x, y \rangle, \forall x, y \in \mathbb{R}^n$.
- $y \in \partial f(x) \Longleftrightarrow f(x) + f^*(y) = \langle x, y \rangle$.
- $y \in \partial_\epsilon f(x) \Longleftrightarrow x \in \partial_\epsilon f^*(y) \Longleftrightarrow f(x) + f^*(y) \leq \langle x, y \rangle + \epsilon, \forall \epsilon > 0$.
- $x^0 \in argmin\Big\{f(x) : x \in \mathbb{R}^n\Big\} \Longleftrightarrow 0 \in \partial f(x^0)$.
- *Si f est s.c.i, alors $\partial f(x) \neq \emptyset$.*

Proposition 1.2.2. *Pour les fonctions convexes polyédrales, on a les propriétés suivantes :*

1. *Si f_1 et f_2 sont convexes polyédrales, alors $f_1 + f_2$ est convexe polyédrale.*
2. *Si f est convexe polyédrale alors f^* l'est aussi, et si f est finie partout alors :*
$$dom(f^*) = co\{a_i : i = 1, ..., l\},$$
$$f^*(y) = \min\left\{\sum_{i=1}^{l} \lambda_i b_i : y = \sum_{i=1}^{l} \lambda_i a_i, \sum_{i=1}^{l} \lambda_i = 1, \lambda_i \geq 0, i = 1, ..., l\right\}.$$
3. *Si f est polyédrale alors $\partial f(x)$ est une partie convexe polyédrale non vide pour tout $x \in dom(f)$.*
4. *Soient $f_1, ..., f_l$ des fonctions convexes polyédrales sur $\mathbb{R}^n$ telles que les ensembles convexes $dom(f_i), i = 1, ..., l$ aient un point commun. Alors*
$$\partial(f_1 + \cdots + f_l)(x) = \partial f_1(x) + \cdots + \partial f_l(x), \forall x.$$

Chapitre 2

Fonctions DC

2.1 Introduction

La convexité est une propriété très importante, elle joue un rôle très fondamental pour résoudre des problèmes d'optimisation. Pour faciliter la résolution des problèmes d'optimisation non convexes, on essaie de trouver une méthode pour transformer cette dernière à des problèmes convexes, c'est donc le but de la programmation **DC**.

Dans ce chapitre, nous présenterons la définition des fonctions **DC** et leurs structure (nommée structure convexe complémentaire), à savoir la structure mathématique sous-jacente de tous les problèmes d'optimisation non-convexe.

2.2 Fonctions DC

Soit C un ensemble convexe de $\mathbb{R}^n$. Notons par $Conv(C)$ l'ensemble des fonctions convexes sur C.

Définition 2.2.1. *On dit que la fonction $f : C \to \mathbb{R} \cup \{+\infty\}$ est **DC** sur C si elle peut s'écrire sous la forme :*

$$f(x) = g(x) - h(x), \forall x \in C, \tag{2.1}$$

où g et h sont deux fonctions de $\Gamma_0(C)$ et $g - h$ est dite une décomposition ***DC*** *de f.*

La première propriété fondamentale des fonctions **DC** est leur stabilité par rapport aux opérations fréquemment utilisées dans la théorie de l'optimisation.

Propriété 2.2.1.

- *Chaque fonction $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ dont les dérivées partielles sont continues est* ***DC****.*
- *Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$* ***DC*** *et $g : \mathbb{R} \longrightarrow \mathbb{R}$ convexe alors la fonction $g \circ f$ est* ***DC****.*

Remarque 2.1.

- *Si f est une fonction* ***DC*** *sur C, alors pour toute fonction convexe finie ψ sur C, la fonction $f = (g + \psi) - (h + \psi)$ représente aussi une décomposition* ***DC*** *de f.*
  Ainsi, toute fonction ***DC*** *possède une infinité de décompositions* ***DC****.*

Définition 2.2.2. *[15] L'ensemble des fonctions* ***DC*** *sur C noté* ***DC****(C), est l'espace vectoriel engendré par $Conv(C)$ défini par :*

$$\boldsymbol{DC}(C) = Conv(C) - Conv(C).$$

Proposition 2.2.1. *[21]*
Si $f_i(x), i = 1, ..., m$ sont des fonctions ***DC*** *sur C alors les fonctions suivantes sont aussi des fonctions* ***DC*** *sur C :*

i) $\sum\limits_{i=1}^{m} \alpha_i f_i(x)$, *pour tout nombre réel α_i ;*

ii) $g(x) = \max\{f_1(x), ..., f_m(x)\}$ *;*

iii) $h(x) = \min\{f_1(x), ..., f_m(x)\}$.

Démonstration. Nous prouvons seulement (ii), parce que (i) est un cas trivial et que la preuve de (iii) est similaire a celle de (ii).

Soit $f_i(x) = p_i(x) - q_i(x)$ avec p_i et q_i sont des fonctions convexes sur C. D'après l'égalité évidente :

$$f_i = p_i + \sum_{j \neq i} q_j - \sum_j q_j,$$

on obtient

$$\max\{f_1, ..., f_m\} = \max_i [p_i + \sum_{j \neq i} q_j] - \sum_j q_j$$

qui est une différence de deux fonctions convexes, puisque l'enveloppe supérieure et la somme finie de plusieurs fonctions convexes sont convexes. $\square$

Ainsi, l'ensemble des fonctions **DC** sur C forme le plus petit espace linéaire contenant toutes les fonctions convexes sur C. Cet espace linéaire, noté $\mathbf{DC}(C)$, est stable sous les opérations d'enveloppe supérieure et inférieure d'un nombre fini de fonctions.

Une inégalité de la forme $f(x) \leq 0$, où la fonction $f(x)$ est convexe, est appelée une inégalité convexe (car l'ensemble de tout les x satisfaisant cette inégalité est un ensemble convexe).

Si $f(x)$ est concave, alors l'inégalité est dite convexe complémentaire ou convexe inverse car son ensemble de solutions est le complémentaire d'un ensemble convexe. Ainsi, une inégalité convexe inverse est aussi une inégalité de la forme $f(x) \geq 0$, où $f(x)$ est convexe. Si $f(x)$ est une fonction **DC**, alors l'inégalité $f(x) \leq 0$ est appelée une inégalité **DC**. Bien entendu, l'inégalité $f(x) \geq 0$ est alors aussi une inégalité **DC** parce que $-f(x)$ est toujours **DC**.

Une conséquence importante de la proposition 2.2.1 est que tout système fini d'inégalités **DC**, qu'il soit conjonctif ou disjonctif, est équivalent à une seule inégalité **DC**. En effet, un système conjonctif :

$$g_i(x) \leq 0, \quad i = 1, ..., m$$

peut être écrit comme

$$g(x) := \max\{g_1(x), ..., g_m(x)\} \leq 0, \tag{2.2}$$

alors qu'un système disjonctif

$$g_i(x) \leq 0 \text{ pour au moins un } i = 1, ..., m$$

est équivalent à

$$g(x) := \min\{g_1(x), ..., g_m(x)\} \leq 0. \tag{2.3}$$

De plus, si $g(x) = p(x) - q(x)$, où $p(x)$ et $q(x)$ sont convexes, alors, en introduisant une variable supplémentaire t, l'inégalité de **DC** (2.2) [ou (2.3)] peut être séparée en deux inégalités :

$$p(x) - t \leq 0, \quad t - q(x) \leq 0, \tag{2.4}$$

où la première est une inégalité convexe et la seconde est une inégalité convexe inverse. Ainsi, tout système fini d'inégalités **DC** peut être réécrit comme un système d'une inégalité convexe et d'une inégalité convexe inverse. Cette propriété permet de décrire de manière compacte une large classe de problèmes d'optimisation non-convexes et constitue la base d'une théorie unifiée de l'optimisation globale.

La propriété suivante montre que pratiquement toutes les fonctions les plus fréquemment rencontrées sont en pratique **DC**. Notons par $C^k(\mathbb{R}^n)$ (ou simplement C^k) la classe de fonctions sur $\mathbb{R}^n$ continuellement différentiable jusqu'à l'ordre k.

Proposition 2.2.2. *[21]*
Une fonction $f : \mathbb{R} \longrightarrow \mathbb{R}$ et différentiable sur C est convexe si et seulement si sa dérivée f' est une fonction non décroissante. Une fonction f à valeurs réelles et deux fois différentiable est convexe si et seulement si sa dérivée seconde f'' est non négative tout au long de cet intervalle.

Proposition 2.2.3. *[21]*
Une fonction f à valeurs réelles deux fois différentiable sur un ensemble convexe et ouvert $C \subset \mathbb{R}^n$ est convexe si et seulement si pour

tout $x \in C$, sa matrice Hessiene

$$Q_x = (q_{ij}(x)), \quad q_{ij}(x) = \frac{\partial^2 f}{\partial x_i \partial x_j}(x_1, ..., x_n)$$

est définie positive, i.e., $\langle y, Q_x y\rangle \geq 0, \forall y \in \mathbb{R}^n$.

Démonstration.
La fonction f est convexe sur C si et seulement si pour tout $a \in C$ et tout $y \in \mathbb{R}^n$ la fonction $\varphi_{a,y}(t) = f(a + ty)$ est convexe sur l'intervalle réel ouvert $\{t|a+ty \in C\}$. La proposition découle alors de la proposition précédente puisqu'un calcul facile donne $\varphi''(t) = \langle y, Q_x y\rangle$ avec $x = a + ty$. □

En particulier, une fonction quadratique

$$f(x) = \frac{1}{2}\langle x, Qx\rangle + \langle x, a\rangle + \alpha,$$

où Q est une matrice symétrique carrée d'ordre n, est convexe sur $\mathbb{R}^n$ si et seulement si Q est semi-définie positive. Cette fonction est concave sur $\mathbb{R}^n$ si et seulement si sa matrice Q est semi-définie négative.

Théorème 2.2.1. *[21]*
Soit f une fonction propre convexe sur $\mathbb{R}^n$. Pour tout ensemble borné $C \subset int(domf)$ l'ensemble $\cup_{x\in C}\partial f(x)$ est non vide et borné.

Corollaire 2.2.1.
Soit $f : C \longrightarrow \mathbb{R}$ une fonction convexe définie et continue sur un ensemble convexe C d'intérieur non vide. Si l'ensemble $\cup\{\partial f(x)|x \in int(C)\}$ est borné, alors f peut être prolongéé à une fonction convexe finie sur $\mathbb{R}^n$.

Démonstration.
Pour chaque point $y \in int(C)$, prenons un vecteur $p_y \in \partial f(y)$ et considérons la fonction affine $h_y(x) = f(y) + \langle p_y, x - y\rangle$. La fonction $\tilde{f}(x) = \sup\{h_y(x)|y \in int(C)\}$ est convexe sur $\mathbb{R}^n$, ($\tilde{f}$ est l'enveloppe

supérieure d'une famille de fonctions affines). Si a est un point fixe quelconque de C alors

$$h_y(x) = f(y)+\langle p_y, a-y\rangle+\langle p_y, x-a\rangle \leq f(a)+\langle p_y, x-a\rangle \leq f(a)+\|p_y\|.\|x-a\|.$$

Puisque $\|p_y\|$ est borné sur $int(C)$, cette dernière inégalité montre que $-\infty < f(x) < +\infty, \forall x \in \mathbb{R}^n$. Ainsi $\tilde{f}(x)$ est une fonction convexe et finie sur $\mathbb{R}^n$. Enfin, puisque $\tilde{f}(x) = f(x)$ pour chaque $x \in int(C)$, il découle de la continuité des deux fonctions $f(x)$ et $\tilde{f}(x)$ sur C que $\tilde{f}(x) = f(x), \forall x \in C$. □

Proposition 2.2.4. *[21]*
*Toute fonction $f \in C^2(\mathbb{R}^n)$ est une fonction **DC** sur tout ensemble compact et convexe Ω de $\mathbb{R}^n$.*

Démonstration.
Montrons que, pour tout ensemble compact et convexe $\Omega \subset \mathbb{R}^n$, la fonction $g(x) = f(x) + \rho\|x\|^2$ devient convexe sur Ω si ρ est suffisamment grand (alors $f(x) = g(x) - \rho\|x\|^2$ est une représentation **DC** de f). En effet, puisque $\langle u, \nabla^2 g(x)u\rangle = \langle u, \nabla^2 f(x)u\rangle + \rho\|u\|^2$, si ρ est très grand, tel que

$$-\min\{\langle u, \nabla^2 f(x)u\rangle |\ x \in \Omega, \|u\| = 1\} \leq \rho,$$

alors $\langle u, \nabla^2 g(x)u\rangle \geq 0, \forall u$, donc $g(x)$ est convexe selon la proposition 2.2.3. □

Corollaire 2.2.2.
*Pour toute fonction continue $f(x)$ sur un ensemble compact et convexe Ω et pour tout $\varepsilon > 0$ il existe une fonction $g(x)$, **DC**, telle que*

$$\max_{x\in\Omega} |f(x) - g(x)| \leq \varepsilon.$$

Démonstration.
D'après le théorème de Weierstrass, il existe un polynôme $g(x)$ satisfaisant la condition requise, et bien évidemment $g(x) \in C^2$. □

Ainsi $\mathbf{DC}(\Omega)$ est dense dans $C(\Omega)$, l'espace de Banach des fonctions continues sur Ω, muni de la norme sup. Bien que les fonctions **DC** soient fréquentes dans la pratique, elles apparaissent souvent sous une forme cachée, non directement reconnaissable. Pour aider à identifier les fonctions **DC** dans diverses situations, nous prouvons quelques propriétés supplémentaires de ces fonctions.

2.3 Fonctions localement DC

Une fonction $f : C \longrightarrow \mathbb{R}$ défini sur un ensemble convexe $C \subset \mathbb{R}^n$ est dite localement **DC** si pour tout $x \in C$ il existe un voisinage convexe ouvert U de x et deux fonctions convexe g, h sur U tel que $f|_U = g|_U - h|_U$. Notons que puisque U peut être supposé borné, les ensembles $\cup\{\partial g(y)|y \in U\}$ et $\cup\{\partial h(y)|y \in U\}$ sont bornés d'après le théorème 2.2.1, donc par le corollaire 2.2.1, les fonctions g, h peuvent être supposées convexes finies sur tout $\mathbb{R}^n$.

Proposition 2.3.1. *[21]*
Soit f une fonction à valeurs réelles sur un ensemble convexe $C \subset \mathbb{R}^n$. Si pour tout $x \in C$, il existe un voisinage ouvert et convexe U_x de x tel que f est convexe sur $U_x \cap C$, alors f est convexe sur C.

Proposition 2.3.2. *[21]*
*Une fonction localement **DC** sur un ensemble convexe, ouvert ou fermé C est une fonction **DC** sur C.*

Démonstration.
Nous nous limiterons au cas où C est un ensemble convexe compact. À partir de l'hypothèse et de la compacité de C, on peut trouver un ensemble fini $\{x_1, ..., x_k\} \subset C$ ainsi que des voisinages convexes ouverts $U_1, ..., U_k$ de ces points couvrant C, et des fonctions convexes $h_i : \mathbb{R}^n \longrightarrow \mathbb{R}, (i = 1, ..., k)$ tel que $(f + h_i)|_{U_i}$ est convexe.
Soit $h = \sum\limits_{i=1}^{k} h_i$ et considérons la fonction $g = f + h$. Pour chaque

$i, (f+h_i)|_{U_i}$ est convexe, donc $g|_{U_i} = (f+h_i)|_{U_i} + (\sum_{j \neq i} h_j)|_{U_i}$ est convexe. Selon la proposition 2.4.4., g est convexe sur C, c'est-à-dire $f = g - h$ avec g, h convexes. $\square$

Chapitre 3

Programmation DC et DCA

3.1 Introduction

Durant les deux dernières décennies, d'énormes progrès ont été accomplis, particulièrement dans les aspects numériques. On peut maintenant résoudre globalement des programmes non convexes provenant de différentes applications, en particulier des programmes non convexes de grande taille. Parmi les algorithmes les plus efficace pour la résolution des problèmes d'optimisation non-convexe, on cite l'algorithme **DCA**.

Ce chapitre est une brève introduction à la programmation **DC** et l'algorithme **DCA**. Nous détaillerons les bases théoriques et algorithmiques de la programmation **DC** et **DCA**. Nous présenterons la programmation **DC**, la notion de la dualité et les conditions d'optimalité locale en programmation **DC**. Nous terminerons par les résultats de convergence de **DCA** et la pénalisation exacte en programmation **DC**.

3.2 Programmation DC

Définition 3.2.1. *(**Programme DC**) On appelle programme **DC** tout problème d'optimisation de la forme :*

$$(P_{DC})\ :\ \alpha = \inf\{f(x) = g(x) - h(x)\ : x \in \mathbb{R}^n\},$$

avec g et h $\in \Gamma_0(\mathbb{R}^n)$.

*(P_{DC}) est un problème d'optimisation sans contraintes. Cependant un problème d'optimisation **DC** avec des contraintes définissant un ensemble convexe fermée C de la forme :*

$$\inf\{f(x) = g(x) - h(x) : x \in C\}, \tag{3.1}$$

peut être ramené sous la forme (P_{DC}) en ajoutant la fonction indicatrice de C à la première composante de f, i.e.,

$$inf\{F(x) = \phi(x) - h(x) : x \in \mathbb{R}^n\},$$

$$\phi(x) := g(x) + \chi_C(x).$$

3.2.1 Les problèmes d'optimisation DC

La majorité des problèmes d'optimisation de la vie réelle sont de nature non convexes, de plus dans les modèles mathématiques industriels les approches convexes montrent leurs limites et sont de plus en plus remplacées par des approches non convexes qui sont plus réalistes.

Les problèmes d'optimisation non convexes et non différentiables rencontrés dans la littérature de la programmation mathématique ainsi que des applications du monde réel peuvent être divisés en trois classes suivantes :

1. $\sup\{f(x) : x \in C\}$, f et C sont convexes.
2. $\inf\{f_0(x) = g_0(x) - h_0(x) : x \in C\}$, g_0, h_0 et C sont convexes.
3. $\inf\{f_0(x) = g_0(x) - h_0(x) : x \in C, g_i(x) - h_i(x) \leq 0, i = 1, ..., m\}$, $g_i, h_i, i = 1, ..., m$ et C sont convexes.

Le problème 1. est un cas spécial du problème 2. Il est équivalent à (P_{DC}). On peut transformer le problème 1 en (P_{DC}), avec $g = \chi_C$ (la fonction indicatrice de C) et $h = f$. Le problème 2 (un problème **DC** sous contraintes convexes) est aussi équivalent à (P_{DC}), avec
$g = g_0 + \chi_C$ et $h = h_0$.

Quant au problème 3, il peut être reformulé de manière équivalente sous la forme du problème 2 via une pénalité exacte relative à la contrainte $g_i(x) - h_i(x) \leq 0, i = 1, ..., m$.

3.2.2 La dualité en optimisation DC

Considérons le programme **DC** suivant :

$$(P_{DC})\ :\ \alpha = \inf\{f(x) = g(x) - h(x), x \in \mathbb{R}^n\},$$

avec g et $h\ \in \Gamma_0(\mathbb{R}^n)$.
Puisque $h \in \Gamma_0(\mathbb{R}^n)$, on a donc $h^{**} = h$ et

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

Donc on aura :

$$\begin{aligned} \alpha &= \inf\{g(x) - h(x)\ : x \in \mathbb{R}^n\} \\ &= \inf\{g(x) - \sup\{\langle x, y\rangle - h^*(y) : y \in \mathbb{R}^n\} : x \in \mathbb{R}^n\} \\ &= \inf\{\inf\{g(x) - [\langle x, y\rangle - h^*(y)] : x \in \mathbb{R}^n\} : y \in \mathbb{R}^n\} \\ &= \inf\{\alpha(y) : y \in \mathbb{R}^n\}, \end{aligned}$$

où

$$(Py)\ :\ \alpha(y) = \inf\{g(x) - [\langle x, y\rangle - h^*(y)] : x \in \mathbb{R}^n\},$$

$$\alpha(y) = \begin{cases} h^*(y) - g^*(y), & \text{si}\, x \in dom(h^*); \\ +\infty, & \text{sinon.} \end{cases}$$

Remarquons que (Py) est un programme convexe.
Finalement avec la convention $(+\infty) - (+\infty) = (+\infty)$, on obtient le problème dual de (P_{DC}), noté (D_{DC}) :

$$(D_{DC})\ :\ \alpha = \inf\{f(x) := h^*(x) - g^*(x) : y \in \mathbb{R}^n\}.$$

Remarque 3.1. *Comme g^* et h^* sont deux fonctions convexes dans $\Gamma_0(\mathbb{R}^n)$, alors le problème (D_{DC}) est aussi un programme **DC**. De plus, (P_{DC}) et (D_{DC}) ont la même valeur optimale et on peut observer la parfaite symétrie entre ces deux problèmes, le dual de (D_{DC}) est exactement (P_{DC}).*

Proposition 3.2.1. *[15] Soient $g, h \in \Gamma_0(\mathbb{R}^n)$.*

1. *x^* est une solution optimale globale de (P_{DC}) si et seulement si*

$$\alpha = (g - h)(x^*) \leq (h^* - g^*)(y), \forall y \in \mathbb{R}^n.$$

2. *y^* est une solution optimale globale de (D_{DC}) si et seulement si*

$$\alpha = (h^* - g^*)(y^*) \leq (g - h)(x), \forall x \in \mathbb{R}^n.$$

Théorème 3.2.1. *[15] Soient $g, h \in \Gamma_0(\mathbb{R}^n)$.*

1. $\inf\{g(x) - h(x) : x \in dom(g)\} = \inf\{h^*(y) - g^*(y) : y \in dom(h^*)\}$.
2. *Si y^* est un minimum de $h^* - g^*$ alors chaque $x^* \in \partial g^*(y^*)$ est un minimum de $g - h$.*
3. *Si x^* est un minimum de $g - h$ alors chaque $y^* \in \partial h(x^*)$ est un minimum de $h^* - g^*$.*

Ce dernier théorème montre que la résolution de l'un des deux problèmes (P_{DC}) et (D_{DC}) implique celle de l'autre.

3.3 Condition d'optimalité en programmation DC

Généralement en optimisation convexe, une fonction f possède le point $x^* \in \mathbb{R}^n$ comme un minimum si et seulement si $0 \in \partial f(x^*)$. En programmation **DC**, l'optimalité globale a une condition nécessaire et suffisante formulée à l'aide des ε-sous-différentiels de g et h.

3.3.1 Condition d'optimalité globale

Théorème 3.3.1. *[22] Soient* $g, h \in \Gamma_0(\mathbb{R}^n)$ *et* $f = g - h$*. Alors* x^* *est un minimum global de* $g - h$ *sur* $\mathbb{R}^n$ *si et seulement si*

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

Remarque 3.2. *On peut écrire* $g = f$ *et* $h = 0$*, si* $f \in \Gamma_0(\mathbb{R}^n)$*. Dans ce cas, la condition d'optimalité précédente est identique à celle de la programmation convexe :*

$$0 \in \partial f(x^*), \ \textit{du fait que } \partial_\varepsilon h(x^*) = \partial h(x^*) = \{0\}, \forall \varepsilon > 0.$$

3.3.2 Point critique

Définition 3.3.1. *On appelle point critique ou point KKT généralisé de* $g - h$*, tout point* $x^* \in \mathbb{R}^n$ *vérifiant*

$$\partial g(x^*) \cap \partial h(x^*) \neq \emptyset.$$

3.3.3 Minimum local

Définition 3.3.2. *Soient* $g, h \in \Gamma_0(\mathbb{R}^n)$*. Un point* $x^* \in dom(g) \cap dom(h)$ *est un minimum local de* $g - h$ *s'il existe un voisinage* $V(x^*)$ *de* x^* *tel que :*

$$(g - h)(x^*) \leq (g - h)(x), \forall x \in V(x^*).$$

3.3.4 Condition nécessaire d'optimalité locale

Théorème 3.3.2. *[22] Si* x^* *est un minimum local de* $g - h$ *alors*

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

Remarque 3.3. *Si* h *est convexe polyédrale et* f *est localement convexe au point* x^**, en particulier si* h *est polyédrale et différentiable en* x^**,* *alors* x^* *est un minimum local, et la condition nécessaire d'optimalité locale est également suffisante.*

Une condition nécessaire et suffisante pour que x^* *soit un minimum local de* $g - h$ *est :*

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

Proposition 3.3.1. *[15] Soit* C *un ensemble convexe non vide dans* $\mathbb{R}^n$ *; et soit* $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ *une fonction convexe. Tout minimum local de* f *sur* C *est également global. L'ensemble qui minimise la fonction* f $argmin_{x \in C} f(x)$ *est un sous-ensemble convexe de* C*.*

Démonstration. Soit $x^* \in C$ un minimum local de f et U_{x^*} un voisinage de x^* tel que $f(x^*) \leq f(x), \forall x \in C \cap U_{x^*}$. Pour tout $x \in C$, nous avons $x_0 := \lambda x + (1 - \lambda)x^* \in C \cap U_{x^*}$ pour $\lambda > 0$ suffisamment petit.

Alors $f(x^*) \leq f(x_0) \leq \lambda f(x) + (1 - \lambda) f(x^*)$. Par conséquent $f(x^*) \leq f(x)$; ce qui prouve la première partie de la proposition.

Si $\alpha = \min f(C)$, alors $argmin_{x \in C} f(x)$ coïncide avec l'ensemble $C \cap \{x|\ f(x) \leq \alpha\}$ qui est un ensemble convexe par la convexité de $f(x)$. □

3.3.5 Condition suffisante d'optimalité locale

Théorème 3.3.3. *[22] Si* x^* *admet un voisinage* V *tel que*

$$\partial h(x) \cap \partial g(x) \neq \emptyset, \forall x \in V \cap dom(g),$$

alors x^* *est un mnimum locale de* $g - h$*.*

Remarque 3.4. *La condition nécessaire d'optimalité locale est également suffisante si la fonction* f *est localement convexe au point* x^* *ou la fonction* h *est convexe polyédrale.*

Théorème 3.3.4. *[15] Soit $x^* \in dom(\partial h)$ un minimum local de $g-h$. Soient $y^* \in \partial h(x^*)$ et V_{x^*} un voisinage de x^*, tel que*

$$g(x) - h(x) \geq g^*(x) - h^*(x), \forall x \in V_{x^*} \cap dom(g).$$

Si $x^* \in int(dom(g^*))$ et $\partial g^*(y^*) \subset V_{x^*}$, alors y^* est un minimum local de $h^* - g^*$.

3.4 Algorithme d'optimisation DC (DCA)

DCA (**DC** Algorithm) est une méthode itérative d'optimisation locale basée sur l'optimalité locale et la dualité en programmation **DC**. Cette approche est complètement différente des méthodes classiques de sous-gradient en optimisation convexe. Elle permet de construire deux suites $\{x^k\}$ et $\{y^k\}$, telles que leurs limites x^* et y^* soient des points KKT généralisés des programmes **DC** primal et dual respectivement, et telles que les suites $(g-h)(x^k)$ et $(h^*-g^*)(y^k)$ soient décroissantes et tendent vers la même limite $\beta = (g-h)(x^*) = (h^*-g^*)(y^*)$. À chaque itération, ces suites sont améliorées d'une façon à vérifier les propriétés présentés dans la proposition suivante :

Proposition 3.4.1. *[22]*

1. *Les suites $\{g(x^k) - h(x^k)\}$ et $\{h^*(y^k) - g^*(y^k)\}$ décroissent et tendent vers la même limite β qui est supérieure ou égale à la valeur optimale globale α.*
2. *Si $(g-h)(x^{k+1}) = (g-h)(x^k)$, alors l'algorithme s'arrête à l'itération $k+1$, et le point x^k (resp. y^k) est un point critique de $g-h$ (resp. h^*-g^*).*
3. *Si la valeur optimale du problème (P_{DC}) est finie et si les suites $\{x^k\}$ et $\{y^k\}$ sont bornées, alors toute valeur d'adhérence x^* de la suite $\{x^k\}$ (resp. y^* de la suite $\{y^k\}$) est un point critique de $g-h$ (resp. h^*-g^*).*

3.4.1 Description de l'algorithme DC

Soit $x^* \in \mathbb{R}^n$ choisi à l'avance, les suites $\{x^k\}$ et $\{y^k\}$ sont définies par :

$$y^k \in \partial h(x^k), \quad x^{k+1} \in \partial g^*(y^k).$$

Pour construire les deux suites $\{x^k\}$ et $\{y^k\}$, on définit deux programmes convexes (D_k) et (P_k), pour $k \geq 1$, comme suit :

$$(D_k) \quad x^k \in \partial g^*(y^{k-1}) \longrightarrow y^k \in arg\min\{h^*(y) - [g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle] : y \in \mathbb{R}^n\} = \partial h(x^k).$$

$$(P_k) \quad y^k \in \partial h(x^k) \longrightarrow x^{k+1} \in arg\min\{g(x) - [h(x^k) + \langle y^k, x - x^k\rangle] : x \in \mathbb{R}^n\} = \partial g^*(y^k).$$

Alors le point x^{k+1} (resp. y^k) est une solution optimale du programme (P_k) (resp.(D_k)).

Le problème d'optimisation convexe (P_k) (resp. (D_k)) obtenu si en remplaçant h (resp. g^*) de (P_{DC}) (resp. (D_{DC})) par sa minorante affine $h_k(x) = h(x^k) + \langle y^k, x - x^k\rangle$ au voisinage de x^k avec $y^k \in \partial h(x^k)$ (resp. $g_k^*(y) = g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle$ au voisinage de y^k avec $x^k \in \partial g^*(y^{k-1})$).

On a ensuite le schéma simple suivant pour décrire l'algorithme **DC** [15] :

$$\begin{array}{ccc} x^k & \longrightarrow & y^k \in \partial h(x^k) \\ & \swarrow & \\ x^{k+1} \in \partial g^*(y^k) & \longrightarrow & y^{k+1} \in \partial h(x^{k+1}). \end{array}$$

Grâce à la proposition 3.4.1, l'algorithme **DC** s'arrête si au moins l'une des suites $\{g(x^k) - h(x^k)\}, \{h^*(y^k) - g^*(y^k)\}, \{x^k\}, \{y^k\}$ converge. En pratique, pour obtenir une solution $\varepsilon-$optimale, nous utilisons souvent les conditions d'arrêt suivantes :

- $|(g-h)(x^{k+1}) - (g-h)(x^k)| \leq \varepsilon,$

- $\|x^{k+1} - x^k\| \leq \varepsilon$.

Par conséquent, on peut décrire l'algorithme **DC** comme suit :
Algorithme DCA
Étape 0 : x^0 donné, $k = 0$ et soit $\varepsilon > 0$ une précision bien définie.
Étape 1 : On calcule $y^k \in \partial h(x^k)$.
Étape 2 : On détermine $x^{k+1} \in \partial g^*(y^k)$ (en général, en résolvant un problème d'optimisation convexe).
Étape 3 : Si les conditions d'arrêt sont vérifiées alors on s'arrête : $x^* = x^{k+1}$ est solution optimale du problème, sinon on pose $k = k + 1$ et on retourne à l'étape 1.

3.4.2 Convergence de DCA

La convergence des suites $\{x^k\}$ et $\{y^k\}$ dépend du choix du point initial $x^0 \in \mathbb{R}^n$.

Lemme 3.4.1. ***(existence des suites)***
Les suites $\{x^k\}$ et $\{y^k\}$ sont bien définies si et seulement si

$$dom(\partial g) \subset dom(\partial h) \text{ et } dom(\partial h^*) \subset dom(\partial g^*).$$

Démonstration.
Par construction, on a $x^{k+1} \in \partial g^*(y^k)$ et $y^k \in \partial h(x^k)$, $\forall k \geq 0$. D'où

$$\{x^k\} \subset R(\partial g^*) = dom(\partial g) \text{ et } \{y^k\} \subset R(\partial h) = dom(\partial h^*),$$

tel que $R(\partial z)$ est l'image de ∂z. □

Théorème 3.4.1. *(**convergence de DCA** [22]) séparons deux cas spéciaux :*

I) : Supposons que les suites $\{x^k\}$ et $\{y^k\}$ sont bien définies, alors on a les propriétés suivantes :

1. *Les suites $\{(g-h)(x^k)\}$ et $\{(h^*-g^*)(y^k)\}$ sont décroissantes, telles que :*

$$g(x^{k+1})-h(x^{k+1}) \leq h^*(y^k)-g^*(y^k) \leq g(x^k)-h(x^k) \leq h^*(y^{k-1})-g^*(y^{k-1}).$$

i) L'égalité $(g-h)(x^{k+1}) = (g-h)(x^k)$ est réalisée si et seulement si :

$$x^k \in \partial g^*(y^k) \quad et \quad y^k \in \partial h(x^{k+1}).$$

ii) L'égalité $(h^*-g^*)(y^{k+1}) = (h^*-g^*)(y^k)$ est réalisée si et seulement si :

$$x^{k+1} \in \partial g^*(y^{k+1}) \quad et \quad y^k \in \partial h(x^{k+1}).$$

2. *Si la valeur optimale α du problème primal (P_{DC}) est finie, alors les suites $\{(g-h)(x^k)\}$ et $\{(h^*-g^*)(y^k)\}$ convergent vers la même limite $\beta \geq \alpha$.*

3. *Notons par $\Omega(z^k)$ l'ensemble des valeurs d'adhérence de la suite $\{z^k\}$.*
Si les suites $\{x^k\}$ et $\{y^k\}$ sont bornées et α est finie, alors : pour tout $x^* \in \Omega(x^k)(resp.\, y^* \in \Omega(y^k))$ il existe un $y^* \in \Omega(y^k)(resp.\ x^* \in \Omega(x^k))$ tels que
i) $g(x^*) - h(x^*) = g^*(y^*) - h^*(y^*) = \beta \geq \alpha$.
ii) $\lim\limits_{k\to+\infty}\{g(x^k) - g^*(y^k)\} = g(x^*) - g^*(y^*) = \langle x^*, y^* \rangle$.
iii) $\lim\limits_{k\to+\infty}\{h(x^k) - h^*(y^k)\} = h(x^*) - h^*(y^*) = \langle x^*, y^* \rangle$.

II) : Supposons que les deux fonctions g et h sont fortement convexes et les suites $\{x^k\}$ et $\{y^k\}$ sont bien définies, alors on a les propriétés suivantes :

1. *i)* $g(x^{k+1}) - h(x^{k+1}) \leq h^*(y^k) - g^*(y^k) - \frac{1}{2}\rho_h \parallel x^{k+1} - x^k \parallel^2$
$\leq g(x^k) - h(x^k) - \frac{1}{2}(\rho_h + \rho_g) \parallel x^{k+1} - x^k \parallel^2$,
où ρ_h et ρ_g sont respectivement les coefficients de coercivité de h et g.
L'égalité $(g-h)(x^{k+1}) = (g-h)(x^k)$ est réalisée si et seulement si
$$y^k \in \partial h(x^{k+1}), x^k \in \partial g^*(y^k) \text{ et } x^k = x^{k+1},$$
dans ce cas on a :
$$y^k \in \partial g(x^k) \cap \partial h(x^k), \quad \text{et} \quad y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1}).$$
ii) $h^*(y^{k+1}) - g^*(y^{k+1}) \leq g(x^{k+1}) - h(x^{k+1}) - \frac{1}{2}\rho_{h^*} \parallel y^{k+1} - y^k \parallel^2$
$\leq h^*(y^k) - g^*(y^k) - \frac{1}{2}(\rho_{h^*} + \rho_{g^*}) \parallel y^{k+1} - y^k \parallel^2$,
où ρ_{h^*} et ρ_{g^*} sont respectivement les coefficients de coercivité de h^* et g^*.
Et l'égalité $(h^* - g^*)(y^{k+1}) = (h^* - g^*)(y^k)$ est réalisée si et seulement si : $x^{k+1} \in \partial g^*(y^{k+1}), y^k \in \partial h(x^{k+1})$ et $y^k = y^{k+1}$, dans ce cas on a
$$y^k \in \partial g(x^k) \cap \partial h(x^k), \quad \text{et} \quad y^k \in \partial g(x^{k+1}) \cap \partial h(x^{k+1}).$$
2. *Si la valeur optimale α du problème primal (P_{DC}) est finie, alors les suites décroissantes $\{(g-h)(x^k)\}$ et $\{(h^* - g^*)(y^k)\}$ convergent vers la même limite $\beta \geq \alpha$.*
3. *Si les suites $\{x^k\}$ et $\{y^k\}$ sont bornées et α est finie, alors : pour tout $x^* \in \Omega(x^k)(resp.\, y^* \in \Omega(y^k))$ il existe un $y^* \in \Omega(y^k)(resp.\, x^* \in \Omega(x^k))$ tels que :*
   i) $y^* \in \partial g(x^*) \cap \partial h(x^*)$ *et* $g(x^*) - h(x^*) = \beta$.
   ii) $x^* \in \partial g^*(y^*) \cap \partial h^*(y^*)$ *et* $h^*(y^*) - g^*(y^*) = \beta$.
   ii) $\lim\limits_{k \to +\infty} \|x^{k+1} - x^k\| = 0 \quad \text{et} \quad \lim\limits_{k \to +\infty} \|y^{k+1} - y^k\| = 0.$

*Le théorème 3.4.1 de convergence de **DCA** est valable si les deux suites $\{x^k\}$ et $\{y^k\}$ sont bien définies et bornées .*

Lemme 3.4.2. ***(bornitude des suites [15])*** *Si $g-h$ est coercive, i.e., ($\lim_{\|x\|\to\infty}(g-h)(x)=+\infty$), alors on a*

i) La suite $\{x^k\}$ est bornée.

ii) Si $\{x^k\}\subset int(dom(h))$, alors la suite $\{y^k\}$ est aussi bornée.
Similairement et par dualité, si h^*-g^* est coercive, alors

iii) La suite $\{y^k\}$ est bornée.

iv) Si $\{y^k\}\subset int(dom(g^*))$, alors la suite $\{x^k\}$ est aussi bornée.

Démonstration.
(i), est une conséquence immédiate de la croissance de la suite $\{(g-h)(x^k)\}$ (Théorème 3.4.1).
(ii) Par construction $\{y^k\}\subset\left(\cup\{\partial h(x^k):k\geq 0\}\right)$ qui est borné car $\{x^k:k\geq 0\}$ est un ensemble borné de $int(dom(h))$. □

3.5 Techniques de pénalisation

Les différentes techniques de pénalisation relèvent souvent du principe suivant. Considérons le problème d'optimisation (P) avec contraintes :

$$(P)\begin{cases} Minimiser\ f(x) \\ s.c. \\ g(x)\leq 0, \\ h(x)=0, \\ x\in C\subset\mathbb{R}^n, \end{cases}$$

où C est un sous-ensemble de $\mathbb{R}^n$ défini par des contraintes qui peuvent être facilement incorporées dans l'optimisation (par exemple contraintes d'égalité linéaire).
Une fonction $p:\mathbb{R}^n\longrightarrow\mathbb{R}$ est appelée fonction de pénalisation pour (P) si elle vérifie :

- $p(x) = 0$ si $g(x) \leq 0,\ h(x) = 0$.
- $p(x) \leq 0$ si $g(x) > 0$ ou $h(x) \neq 0$.

Le programme de pénalisation est donc

$$(P_t) \begin{cases} Minimiser\ f_t(x) = f(x) + tp(x) \\ \text{s.c. } x \in C. \end{cases}$$

Ici, t est un scalaire strictement positif, appelé facteur de pénalisation. On cherche à savoir quand les ensembles de solutions de (P) et (P_t) coïncident. Cela va dépendre du choix de la fonction p et du paramètre $t > 0$.

Proposition 3.5.1. ***(monotonie en pénalisation [15])*** *Supposons que, pour tout t considéré, (P_t) ait au moins une solution, noté $\bar{x}_t$. Alors, lorsque $t > 0$ croît,*

1. *$p(\bar{x}_t)$ décroît,*
2. *$f(\bar{x}_t)$ croît,*
3. *$f_t(\bar{x}_t)$ croît (si $p(.) \geq 0$).*

3.5.1 Pénalisation exacte en programmation DC

Les techniques de pénalité exacte sont souvent très utiles pour transformer un problème **DC** sous contraintes non convexes en un problème **DC** sous contraintes convexes. Ce dernier problème peut être résolu par **DCA**.

Considérons le problème d'optimisation :

$$(P) \quad \alpha = \inf\{f(x) : x \in C,\ p(x) \leq 0\},$$

où C est un polyèdre convexe non vide et borné dans $\mathbb{R}^n$, f, p sont deux fonctions concaves finies sur C, où p est une fonction à valeurs positives ou nulles sur C.

Le problème (P) est un programme non convexe. La non convexité du problème est due au fait que la fonction objectif f est concave et que la contrainte $p(x) \leq 0$ est non convexe. Une contrainte $p(x) \leq 0$ avec p concave est dite anti-convexe.

L'objectif du problème (P) est de minimiser une fonction concave f sous contraintes polyédrales (convexes) et une contrainte anti-convexe. Supposons que le problème (P), dont l'ensemble de solutions est noté par P, est réalisable. Étant donné un paramètre $t > 0$, on peut définir le problème de pénalité par :

$$(P_t) \quad \alpha = \inf\{f(x) + tp(x) : x \in C\}.$$

L'ensemble des solutions du problème (P_t) est noté par P_t. On obtient le théorème suivant :

Théorème 3.5.1. ***(théorème de pénalité exacte [22])*** *Soit C un polyèdre convexe borné non vide, f et p sont deux fonctions concaves finies sur C, où p est une fonction à valeurs positives ou nulles sur C. Alors il existe un nombre fini $t_0 \geq 0$ tel que pour tout $t \geq t_0$, les deux problèmes (P) et (P_t) sont équivalents au sens que $P = P_t$ et $\alpha(t) = \alpha$. Le paramètre t_0 peut être déterminé comme suit :*

- Si l'ensemble des sommets $V(C)$ de C est contenu dans l'ensemble $\{x \in C : p(x) \leq 0\}$, alors $t_0 = 0$ et $\alpha(0) = \alpha$.
- Si $\alpha(0) < \alpha$, alors $t_0 = \max\{\frac{\alpha - f(x)}{p(x)} : x \in V(C), p(x) > 0\}$.
  On a aussi les propriétés suivantes :
- $\alpha(t) = \alpha$ si et seulement si $t \geq t_0$.
- $P_t \cap \{x \in C : p(x) \leq 0\} \neq \emptyset \Leftrightarrow P_t \subset P \Leftrightarrow t \geq t_0$.
- $P = P_t$ si $t > t_0$.

Ce théorème est très utile en programmation non convexe, notamment en programmation **DC**, puisque beaucoup de problèmes d'optimisation sous contraintes non convexes peuvent être transformés en problème (P) à l'aide de ce théorème. Remarquons que l'estimation du paramètre t n'est pas toujours évidente. En pratique, on donne souvent une valeur assez grande a priori pour simplifier l'estimation de t. Pour s'assurer que la valeur de t est assez grand, on a le théorème suivant :

Théorème 3.5.2. *[22, 15] Soient C un polyèdre convexe non vide et borné, f et p deux fonctions concaves finies sur C, où p est une fonction à valeurs positives ou nulles sur C. Alors il existe un nombre fini $t^* > 0$ tel que pour tout $t \leq t^*$, les deux suites générées par* ***DCA*** *$\{f(x^k) + tp(x^k)\}$ et $\{p(x^k)\}$ sont décroissantes.*

Ce théorème montre qu'il existe une valeur de t assez grande telle que la suite $\{p(x^k)\}$ générée par ***DCA*** *soit décroissante. C'est-à-dire, si l'on trouve que la suite $\{p(x^k)\}$ n'est pas décroissante, alors t n'est pas assez grande.*

Algorithme DCA Modifié[15]

Étape 0 : x^0 donné, t une grande valeur donné, $k = 0$.
Étape 1 : On calcule $y^k \in \partial(-f - tp)(x^k)$.
Étape 2 : On détermine $x^{k+1} \in argmin\{-\langle x, y^k\rangle : x \in C\}$.
Étape 3 : Si $p(x^{k+1}) > p(x^k)$, alors on augmente la valeur de t (par exemple $t = t + 1000$).
Étape 4 : Si les conditions d'arrêt sont vérifiées alors on s'arrête : $x^* = x^{k+1}$ est solution optimale du problème, sinon on pose $k = k + 1$ et on retourne à l'étape 1.

Chapitre 4

Résolution des problèmes quadratiques par DCA

4.1 Introduction

La programmation quadratique est une discipline mathématique qui traite des techniques de résolution du problème d'optimisation d'une fonction quadratique quelconque sous un domaine délimité par des contraintes linéaires. Cette discipline revêt une importance théorique et pratique. En effet, elle est utilisée pour l'approximation des solutions des problèmes de programmation non linéaire. De plus, plusieurs problèmes pratiques peuvent êtres modélisés sous forme d'un programme quadratique convexe et non convexe, à savoir les problèmes de gestion de production, les problèmes des moindres carrées, le problème de gestion des portefeuilles en finance, les problèmes de SVM etc.

4.2 Position du problème

Un problème de minimisation d'une fonction quadratique avec contraintes linéaires se présente sous la forme standard suivante :

$$(P_m) \quad \begin{array}{c} \min f(x) = \frac{1}{2}x^T Qx + c^T x, \\ s.c.\ Ax \leq 0, x \geq 0, \end{array} \tag{4.1}$$

où Q est une matrice symétrique indéfinie, telle que $Q \neq 0$, c, x, sont des *vecteurs* de $\mathbb{R}^n$; A est une matrice de dimension $m \times n$, avec $rang(A) = m < n$ et $b \in \mathbb{R}^m_+$.
Il est clair qu'on peut sans restreindre la généralisation se limiter à l'étude du problème suivant :

$$(P_m) \quad \begin{array}{c} \min f(x) = \sum_{i=1}^n \alpha_i x_i + \beta_i x_i^2, \\ s.c.\ Ax \leq 0, x \geq 0. \end{array} \tag{4.2}$$

C-à-d la matrice Q pour le problème (P_m) sera une matrice diagonale. Cette écriture est dite la forme canonique d'une fonction quadratique, où $\alpha_i, \beta_i \in \mathbb{R}$. La méthode de transformation de (P_m) vers sa forme canonique est détaillé dans le section suivante.

4.3 Conditions d'optimalité

Rappelons les résultats importants suivants :

Proposition 4.3.1. *[3] Tout minimum local (resp, maximum) d'une fonction quadratique $f(x)$ sur un ensemble affine $E \subset \mathbb{R}^n$ est un minimum global (resp, maximum global).*

Démonstration.
Si x^* est un minimum local de $f(x)$ sur E, il existe une boule B centrée en x^* telle que $f(x^*) \leq f(x), \forall x \in B \cap E$. Pour tout $z \in E \backslash \{x^*\}$, soit L la ligne passant par x^* et z, i.e., $L = \{x \in \mathbb{R}^n | x = x^* + t(z - x^*),\ t \in \mathbb{R}\}$. Alors la restriction de f sur L est la fonction $\varphi(t) = f(x^* + t(z -$

$x^*)), t \in \mathbb{R}$. Puisque $f(x)$ est une fonction quadratique sur $\mathbb{R}^n$, $\varphi(t)$ est une fonction quadratique de la variable réelle t. Cette fonction atteint un minimum local à $t = 0$, qui doit alors être un minimum global. Alors $\varphi(0) \leq \varphi(t), \forall t \in \mathbb{R}$, en particulier $f(z) = \varphi(1) \geq \varphi(0) = f(x^*)$, c'est-à-dire, $f(z) \geq f(x^*), \forall z \in E$. □

Proposition 4.3.2. *[3] Une fonction quadratique bornée inférieurement sur $\mathbb{R}^n$ est convexe. Le minimum (ou le maximum) d'une fonction quadratique indéfinie sur un ensemble convexe compact C est toujours atteint en un point frontière (qui peut ne pas être un point extrême).*

Démonstration.
Si une fonction quadratique $f(x)$ n'est pas convexe, il existe deux points distincts $a, b \in \mathbb{R}^n$ et un nombre $t \in (0,1)$ tels que $f((1-t)a + tb) > (1-t)f(a) + tf(b)$. Donc, la fonction $\varphi(t) := f((1-t)a + tb), t \in \mathbb{R}$, qui est la restriction de f sur la droite passant par a, b, n'est pas convexe. Alors $\varphi(t)$ doit être concave car une fonction quadratique sur $\mathbb{R}$ est convexe ou concave. Par conséquent, $\varphi(t)$ est non bornée inférieurement sur $\mathbb{R}$, ce qui contredit le fait que $f(x)$ soit borné inférieurement sur $\mathbb{R}^n$. Cela prouve la première partie de la proposition.

Passons à la deuxième partie, sans perte de généralité, nous pouvons supposer que l'enveloppe affine de C est tout l'espace $\mathbb{R}^n$. Puisque $f(x)$ est continue et que C est compact convexe, $f(x)$ atteint un minimum sur C. Si ce minimum était un point intérieur de C, ce serait un minimum local de $f(x)$ sur $\mathbb{R}^n$, et par conséquent, par la proposition 4.3.1, c'est un minimum global. Donc, $f(x)$ serait bornée inférieurement sur $\mathbb{R}^n$, et serait convexe, ce qui contredit le fait qu'elle soit indéfinie. La démonstration du cas du maximum est analogue à celle du minimum. □

4.4 Ecriture d'une forme quadratique sous sa forme canonique

Certaines fonctions objectifs ont les deux propriétés que les coefficients de $x_i^2, i = 1, ..., n$ sont identiques et qu'il n'existe aucun terme impliquant $x_i x_j, i, j \in \{1, \ldots, n\}$. Par conséquent, les ensembles de niveaux pour chaque fonction objectif sont des boules qui peuvent être tracés par inspection. Il n'est pas immédiatement évident de savoir comment les ensembles de niveaux doivent être dessinés. Le but de cette section est de montrer une méthode par laquelle ces ensembles peuvent être dessinés facilement. Les outils mathématiques pertinents sont les vecteurs propres et les valeurs propres. Pour plus d'informations sur ces concepts, vous pouvez consulter de nombreuses références appropriées d'algèbre linéaire.
Considérons une fonction quadratique générale de n variables :

$$f(x) = \frac{1}{2} x^T Q x + c^T x, \tag{4.3}$$

où x et c sont des vecteurs de $\mathbb{R}^n$ et Q est une matrice carrée d'ordre n symétrique. Soit S la $n \times n$-matrice dont les colonnes sont les vecteurs propres de Q et soit D la $n \times n$-matrice diagonale des valeurs propres correspondantes. La propriété qui définit S et D est :

$$QS = SD. \tag{4.4}$$

Une propriété élémentaire des vecteurs propres est qu'ils sont orthogonaux pour les matrices symétriques réelles. La condition qu'ils soient également de norme unité est :

$$S^T S = I, \tag{4.5}$$

où I désigne la $n \times n$-matrice identité. Multiplier (4.4) à gauche par S^T et utiliser (4.5) donne :

$$S^T Q S = D. \tag{4.6}$$

Soit x^* le point qui minimise f. S'agissant d'un problème de minimisation quadratique sans contraintes, les conditions d'optimalité impliquent que le gradient de f en x^* doit être nul. Soit $g(x)$ le gradient de f. En écrivant $f(x)$ explicitement en fonction des composantes de x, il est facile de voir que :

$$g(x) = c + Qx. \tag{4.7}$$

Puisque $g(x^*) = 0,\ x^0$ est la solution des équations linéaires

$$Qx^* = -c. \tag{4.8}$$

Nous introduisons ensuite un changement de variable y lié à x par

$$x = Sy + x^*0. \tag{4.9}$$

La substitution de cette expression pour x dans (4.3) nous donne

$$f(x) = f(x^*0) + g^T(x^*)Sy + \frac{1}{2}y^T S^T QSy. \tag{4.10}$$

Nous avons choisi x^* tel que $g(x^*) = 0$. Avec ceci, (4.3) se simplifie :

$$f(x) = f(x^*) + \frac{1}{2}y^T Dy. \tag{4.11}$$

Puisque D est diagonale, (4.10) exprime les coordonnées de f en y uniquement en termes de $y_i^2, i = 1, ..., n$. En particulier, (10) ne comprend aucun terme linéaire dans y et aucun terme de produit mixtes $y_i y_j,\ i, j \in \{1, \dots, n\}$. En utilisant (8), il est facile de tracer des ensembles de niveaux pour les coordonnées f en y.

En résumant, les ensembles de niveaux pour f peuvent être dessinés en utilisant les étapes suivantes :

1. Calculer S et D en résolvant le problème de calcul des valeurs propres pour Q.
2. Résoudre le système $Qx^* = -c$.
3. Dessiner les axes $y_j, j = 1, ..., n$ dans l'espace $x_i, i = 1, ..., n$ en traçant les vecteurs colonnes de S centrés sur x^*.

4.5 DCA pour les problèmes quadratiques

Comme nous l'avons vu dans la section précédente, on peut considérer une forme quadratique sous sa forme canonique, et ce, sans restreindre la généralisation.

4.5.1 La décomposition DC

Pour faire une décomposition d'une forme quadratique écrite sous sa forme canonique sous une différence de deux fonctions convexes, on peut séparer les termes quadratiques par les signes des coefficients comme suit :

$$f(x) = \sum_{i=1}^{n} \alpha_i x_i + \beta_i x_i^2 = \left(\sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2 \right) + \left(\sum_{\beta_i < 0} \alpha_i x_i + \beta_i x_i^2 \right)$$

$$= \left(\sum_{\beta_i \geq 0} \alpha_i x_i + \beta_i x_i^2 \right) - \left(- \sum_{\beta_i < 0} \alpha_i x_i + \beta_i x_i^2 \right) = g(x) - h(x).$$

Cette méthode nous garantit l'obtention de la décomposition DC pour une forme quadratique.
Le principe de DCA est basé sur la résolution d'un problème quadratique convexe à chaque itération. Pour résoudre le problème convexe, il existe plusieurs méthodes efficaces (points intérieurs, méthode d'activation de contraintes, etc.)

4.6 Expériences et résultats numériques

Nous avons développé une implémentation de l'algorithme DCA appliqué au problème d'optimisation qudaratique sous contraintes linéaires en utilisant la décomposition décrite ci-dessus, avec le langage de programmation MATLAB 2018a. Nous avons exécuté notre solveur sur un PC doté d'un processeur Intel Pentium Intel (R) Core (TM) i5-6500 à 3,20 GHz (4 CPU), 3,2 GHz et 8 Go de RAM.

Nous avons appliqué DCA à 40 problèmes-test de différentes dimensions, avec un nombre de variables variant de 4 à 1000. Nous avons choisi une tolérance $\epsilon = 10^{-6}$. Nous avons initialisé DCA avec deux différents points initiaux l'un est le point d'origine $x^* = 0$ (DCA0), et l'autre correspond à la solution du problème d'optmisation de la partie linéaire de la forme quadratique sous les contraintes linéaires du probléme quadratique (DCA). Les résultats numériques obtenus sont présentés dans le tableau 1 et représentés graphiquement dans les figures 1 et 2, où It, CPU, f^* et M représententent respectivement le nombre d'itérations moyen et le tempts d'exécution moyen (en seconde), le minimum obtenu par les deux variantes DCA0 et DCAL et la moyenne des valeurs des différents paramètres.

			DCA0			DCAL		
No.	m	n	f^*	CPU	It	f^*	CPU	It
1	0	4	-4.9645	0.18	2	-4.9645	0.07	2
2	0	8	-64.0218	0.12	3	-64.0218	0.09	3
3	0	12	-62.9365	0.02	3	-62.9365	0.06	3
4	0	16	-110.2297	0.02	3	-110.2297	0.14	3
5	0	20	-325.1345	0.02	3	-325.1345	0.04	3
6	0	24	-79.8414	0.01	3	-79.8414	0.04	3
7	0	28	-115.2116	0.01	3	-115.2116	0.03	3
8	0	32	-290.1660	0.01	3	-290.1660	0.04	3
9	0	36	-345.0987	0.02	3	-345.0987	0.03	3
10	0	40	-310.5537	0.01	3	-310.5537	0.03	3
M				0.43	2.9		0.56	2.9
11	41	40	6.4092e+03	0.08	2	6.4092e+03	0.03	1
12	41	40	4.5296e+03	0.01	2	4.5296e+03	0.02	1
13	41	40	4.8002e+03	0.02	2	4.8002e+03	0.03	1
14	41	40	6.3272e+03	0.01	2	6.3272e+03	0.02	1
15	41	40	7.1455e+03	0.02	2	7.1455e+03	0.02	1
16	41	40	4.8923e+03	0.01	2	4.8923e+03	0.02	1
17	41	40	8.0125e+03	0.01	2	8.0125e+03	0.02	1
18	41	40	6.3100e+03	0.01	2	6.3100e+03	0.02	1
19	41	40	5.7916e+03	0.02	2	5.7916e+03	0.02	1
20	41	40	8.1005e+03	0.01	2	8.1005e+03	0.03	1
M				0.20	2		0.24	1

No.	m	n	f^*	CPU	It	f^*	CPU	It
21	1001	1000	1.4058e+05	7.56	2	1.4058e+05	4.33	1
22	1001	1000	5.4394e+05	7.68	2	5.4394e+05	4.28	1
23	1001	1000	3.2549e+05	7.94	2	3.2549e+05	4.25	1
24	1001	1000	6.1356e+05	7.66	2	6.1356e+05	4.03	1
25	1001	1000	3.2497e+05	7.25	2	3.2497e+05	4.04	1
26	1001	1000	4.8255e+05	7.64	2	4.8255e+05	4.22	1
27	1001	1000	4.6456e+04	8.08	2	4.6456e+04	4.44	1
28	1001	1000	1.3012e+05	7.61	2	1.3012e+05	4.27	1
29	1001	1000	1.2659e+05	7.78	2	1.2659e+05	4.30	1
30	1001	1000	1.4635e+05	8.08	2	1.4635e+05	4.30	1
M				77.27	2		42.47	1
31	800	1000	-1.6482e+06	25.39	6	-1.6482e+06	21.40	6
32	800	1000	-1.6890e+06	17.35	5	-1.6890e+06	17.17	5
33	800	1000	-1.7370e+06	22.22	6	-1.7370e+06	22.04	6
34	800	1000	-1.6024e+06	16.90	5	-1.6024e+06	16.82	5
35	800	1000	-1.7641e+06	18.57	5	-1.7641e+06	18.18	5
36	800	1000	-1.6770e+06	17.64	5	-1.6770e+06	17.32	5
37	800	1000	-1.6199e+06	18.96	5	-1.6199e+06	18.49	5
38	800	1000	-1.5604e+06	18.88	5	-1.5604e+06	18.43	5
39	800	1000	-1.4329e+06	18.86	5	-1.4329e+06	18.53	5
40	800	1000	-1.4295e+06	18.92	5	-1.4295e+06	17.82	5
M				193.71	5.2		186.19	5.2

À partir des différents tableaux et figures, nous remarquons que les temps d'exécutions des deux variantes sont presque identiques pour les problèmes de dimensions $4 \leq n \leq 40$, mais le temps CPU de DCAL commence à devenir meilleur que celui de DCA0 pour les problèmes

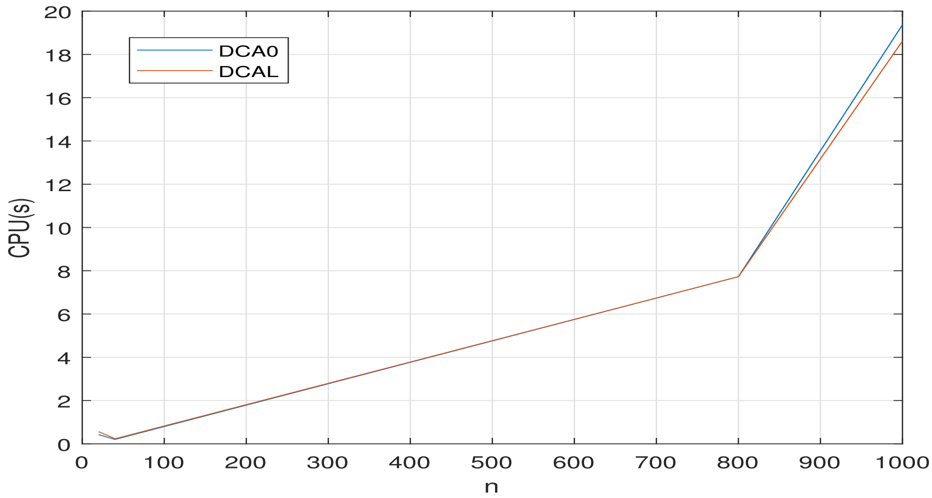

FIGURE 4.1 – Le temps CPU en fonction de la dimension du problème

avec $n = 1000$. De plus, le nombre d'itérations de DCA initialisé avec l'argument du minimum de la partie linéaire de la forme quadratique est inférieur ou identique à celui de DCA initialisé avec l'origine, et ce, pour tout les problèmes-test considéré.

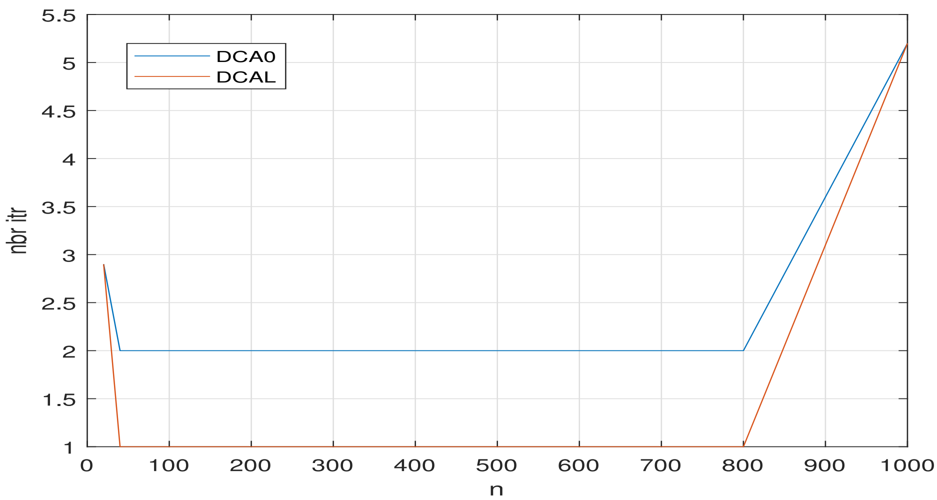

FIGURE 4.2 – Le nombre d'itération moyen en fonction de la dimension du problème

Conclusion

Dans ce mémoire, après avoir rappelé les principales notions de base de l'analyse convexe, nous sommes passés à l'exposé des notions de fonctions DC et leurs propriétés, par la suite nous avons présenté les fondements théoriques de la programmation DC et de l'algorithme DCA, et ce, pour optimiser des fonctions à plusieurs variables non-convexes et non-différentiables. Finalement, nous nous sommes intéressés à l'application de l'algorithme DCA pour optimiser les fonctions quadratiques écrites sous forme canonique sur un domaine délimité par des contraintes linéaires. A cet effet, nous avons développé une implémentation de l'algorithme DCA avec le langage de programmation MATLAB2018a. Afin de montrer l'importance du choix du point initial pour l'algorithme DCA, nous avons initialisé cet algorithme avec deux points initiaux : l'origine et le point correspondant à la solution du problème de minimisation de la partie linéaire. Les résultats numériques sur des problèmes-test générés aléatoirement avec un nombre de variables variant de 4 à 1000 montrent qu'il est judicieux de choisir le point initial comme étant la solution du problème d'optimisation minimisant la partie linéaire de la forme quadratique sous les contraintes du problème original.

Bibliographie

[1] S. Achour (2017) Formulation variationnelle d'un problème aux limites de Dirichlet sous un problème d'optimisation quadratique. Mémoire de Master en mathématiques, Université de Laghouat.

[2] F. Akoa (2005) Approches de points intérieurs et de la programmation DC en optimisation non convexe. Code et simulations numériques industrielles. Thèse de doctorat en Mathématiques, INSA de Rouen.

[3] An L.T.H., P.D. Tao (1998) A branch and bound method via dc optimization algorithms and ellipsoidal technique for box constrained nonconvex quadratic problems. Journal of Global Optimization 13(2) :171–206

[4] L.T.H. An, P.D. Tao (2018) DC programming and DCA : thirty years of developments. Math. Program., Ser. B 169 :5–68

[5] M. Bentobache, M.O. Bibi (2016) Numerical methods of linear and quadratic programming. French Academic Presses, Germany (in french)

[6] M. Bentobache, M. Telli, A. Mokhtari (2018) A global minimization algorithm for concave quadratic programming. In : proceedings of the 29th European Conference on Operational Research, EURO 2018, University of Valencia, July 08-11, pp 329

[7] M. Bentobache, M. Telli, A. Mokhtari (2018) A simplex algorithm with the smallest index rule for concave quadratic programming. In : proceedings of the Eighth International Conference on Advan-

ced Communications and Computation, INFOCOMP 2018, Barcelona, Spain, July 22-26, pp 88–93

[8] M. Bentobache, M. Telli, A. Mokhtari (2019) A sequential linear programming algorithm for continuous and mixed-integer nonconvex quadratic programming. To appear in Advances in Intelligent Systems and Computing, Springer

[9] I. Bieche, R. Maynard, R. Rammal, J.P. Uhry (1980) On the ground states of the frustration model of a spin glass by a matching method of graph theory. J. Phys. A : Math. Gen. 13 :2553–2576

[10] M.J. Best (2017) Quadratic programming with computer programs. CRC Press, London

[11] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, J. Vanthienen (2003) Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the operational research society 54(6) :627–635

[12] V. Bugera, H. Konno, S. Uryasev (2002) Credit cards scoring with quadratic utility functions. Journal of Multi-Criteria Decision Analysis 11(4-5) :197–211

[13] R. Enkhbat, Y. Bazarsad (2010) General quadratic programming and its applications in response surface analysis. In : Chinchuluun A et al. (eds.), Optimization and Optimal Control, Springer Optimization and Its Applications 39 :121–137

[14] K.L. Hoffman (1975) A Successive Underestimating Method for Concave Minimization Problems. Ph.D. Dissertation, The George Washington University

[15] M. Al Kharboutly (2018) Résolution d'un problème quadratique non convexe avec contraintes mixtes par les techniques de l'optimisation D.C.. Thèse de doctorat, Optimisation et contrôle. Normandie Université.

[16] Y.S. Niu (2010) DC programming and DCA combinatorial optimization and polynomial optimization via SDP techniques. Thèse de doctorat. PhD dessertation in General Mathematics. INSA de Rouen.

[17] P.M. Pardalos, G. Rodgers (1990) Computational aspects of a branch and bound algorithm for quadratic zero-one programming. Computing 45(2) :131–144

[18] A.I. Rusakov (2003) Concave programming under simplest linear constraints. Computational mathematics and mathematical physics 43(7) :908–917

[19] M. Thiao (2011) Approches de la programmation DC et DCA en data mining : modélisation parcimonieuse de données. Thèse de doctorat, INSA de Rouen.

[20] T.T. Tran (2017) DC programming and DCA for some classes of problems in Wireless Communication Systems. Networking and Internet Architecture, PhD dessertation, Université de Lorraine.

[21] H. Tuy (2016) Convex analysis and global optimization, Springer Optimization and Its Applications book series, vol 110. Springer, Berlin

[22] A. Yassine (1998) Méthode de région de confiance et optimisation D.C. Théorie, algorithmes et applications, HDR, Université Henri Poincaré, Nancy I.