Approches de la programmation DC et DCA en data mining : modélisation parcimonieuse de données

Mamadou Thiao

THÈSE

présentée par

Mamadou THIAO

pour obtenir le titre de

DOCTEUR DE L'INSTITUT NATIONAL

DES SCIENCES APPLIQUÉES DE ROUEN

(arrêté ministériel du 7 août 2006)

Spécialité : MATHÉMATIQUES APPLIQUÉES

Approches de la programmation DC et DCA en data mining.

Modélisation parcimonieuse de données.

Soutenue le 28 Octobre 2011 devant le jury composé de

Rapporteurs :	
Yann GUERMEUR	Directeur de Recherche CNRS, Université Nancy 2
Michèle SEBAG	Directrice de Recherche CNRS, Université Paris Sud
Examinateurs :	
Alexandre d'ASPREMONT	Professeur, Université Princeton-Ecole Polytechnique Paris
Francis BACH	Directeur de Recherche INRIA, ENS Paris
Stéphane CANU	Professeur, INSA Rouen
Hoai An LE THI	Professeur, Université Metz
Directeur de thèse :	
Tao PHAM DINH	Professeur, INSA Rouen

THÈSE PRÉPARÉE AU LABORATOIRE DE MATHÉMATIQUES DE L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE ROUEN, FRANCE

Remerciements

Cette thèse a été réalisée au sein du Laboratoire de Mathématique (LMI) de l'Institut National des Sciences Appliquées (INSA) de Rouen, France, sous la direction Monsieur Pham Dinh Tao, Professeur à l'INSA de Rouen. Je voudrais lui exprimer ici toute ma profonde gratitude pour son soutien, sa disponibilité, ses encouragements et ses conseils tout au long de ce travail.

Je tiens à remercier vivement Monsieur Yann Guermeur, Directeur de Recherche CNRS à l'Université de Nancy 2 et Madame Michèle Sebag, Directrice de Recherche CNRS à l'Université Paris-Sud, qui m'ont fait l'honneur d'être rapporteur de cette thèse.

Mes remerciement vont aussi à l'endroit de Monsieur Alexandre d'Aspremont, Professeur à l'Université de Princeton, Monsieur Francis Bach, Directeur de Recherche INRIA à l'Ecole Normale Supérieure de Paris, Monsieur Stéphane Canu, Professeur à l'INSA de Rouen et Madame Hoai An Le Thi, Professeur à l'Université de Metz, qui m'ont fait le plaisir d'accepter d'être membres de mon jury. En particulier, je voudrais témoigner ma sincère gratitude à Madame Le Thi pour ses conseils et les discussions très fructueuses que nous avons eues tout au long de cette thèse.

Je voudrais aussi adresser mes remerciements à mes collègues, aux membres du LMI et du département Génie Mathématique de l'INSA pour leurs soutiens, leurs conseils et l'excellent cadre de travail dont j'ai profité tout au long de ces années de travaux de thèse et aussi au cours de ma formation.

Je ne saurais terminer sans exprimer au plus profond de mon coeur, mes remerciements à mes parents, mes frères, mes soeurs, mes amis et tous les membres de ma famille pour leurs soutiens et sacrifices tout au long de mes années d'études.

Résumé

Nous abordons dans cette thèse les approches de la Programmation DC et DCA en Data Mining (fouille de données). Plus particulièrement, nous nous intéressons aux problèmes de parcimonie en modélisation parcimonieuse de données. Le travail porte sur des recherches théoriques et algorithmiques et la principale approche utilisée est la programmation DC et DCA.

Nous avons établi des propriétés intéressantes, des reformulations DC, voire quadratiques, équivalentes pour ces problèmes grâce à de nouvelles techniques de pénalité exacte développées durant cette thèse. Ces résultats donnent une nouvelle facette et une nouvelle manière de voir ces problèmes de parcimonie afin de permettre une meilleure compréhension et prise en main de ces problèmes. Ces nouvelles techniques ont été appliquées dans le cadre de la modélisation parcimonieuse pour le problème de la valeur propre maximale et dans le cadre de la modélisation parcimonieuse dans les modèles de régression linéaire.

La structure simple des reformulations obtenues se prête bien à la programmation DC et DCA pour la résolution. Les simulations numériques, obtenues avec DCA et un algorithme combiné DCA et la procédure Séparation et Evaluation pour l'optimisation globale, sont très intéressantes et très prometteuses et illustrent bien le potentiel de cette nouvelle approche.

Abstract

In this thesis, we investigate the DC Programming and DCA approaches in Data Mining. More precisely, we are interested in the sparse approximation problems in sparse modelling. The work focuses on theoretical and algorithmic studies, mainly based on DC Programming and DCA.

We established interesting properties concerning DC and quadratic reformulations for these problems with the help of new exact penalty techniques in DC programming. These results give new insights on these sparse approximation problems and so allow a better understanding and a better handling of these problems. These novel techniques were applied in both contexts of sparse eigenvalue problem and sparse approximation in linear models.

The simple and nice structure of the obtained reformulations are suitably adapted to DC programming and DCA. Computational experiments are very interesting and promising, illustrating the potential of the novel approach.

Table des matières

1 Introduction **15**
1.1 Contexte de l'étude 15
1.2 Vue d'ensemble du travail effectué 16
1.2.1 La Programmation DC et DCA en fouille de données 16
1.2.2 Programmation DC et DCA 16
1.2.3 Technique de Séparation et Evaluation (SE) pour l'optimisation globale 17
1.2.4 Modélisation parcimonieuse 17
1.2.5 Modélisation parcimonieuse pour le problème de la valeur propre maximale 17
1.2.6 Modélisation parcimonieuse dans les modèles de régression linéaire 18
1.3 Quelques activités de recherches effectuées durant la thèse 18

2 La Programmation DC et DCA en fouille de données **21**
2.1 Apprentissage supervisé 23
2.2 Apprentissage non-supervisé 24
2.3 Apprentissage semi-supervisé 24
2.4 Projection 24
2.5 Conclusion 25

3 Programmation DC et DCA **27**
3.1 Quelques rappels d'analyse convexe 28
3.2 Fonctions DC 33
3.3 Programmation DC 35
3.3.1 Programme DC 35
3.3.2 Dualité en programmation DC 36
3.4 Conditions d'optimalité en programmation DC 37
3.5 DCA (DC Algorithm) 39
3.5.1 Convergence de DCA 40
3.5.2 Illustration géométrique de DCA 43
3.6 Techniques de pénalité 45
3.6.1 Exemple 46
3.6.2 Quelques propriétés 47

4 Technique de Séparation et Evaluation pour l'optimisation globale 49
4.1 Introduction . . . 49
4.2 La Méthode de Séparation et Evaluation . . . 50
4.2.1 Phase de séparation . . . 50
4.2.2 Phase d'évaluation . . . 51
4.2.3 Un premier prototype . . . 51
4.2.4 Un deuxième prototype . . . 54
4.2.5 Réalisation . . . 57
4.2.5.1 Stratégie de division . . . 57
4.2.5.2 Règle de sélection . . . 58
4.2.5.3 Estimation de borne . . . 59
4.3 Relaxation DC . . . 59
4.3.1 Enveloppe convexe de fonctions non convexe . . . 60
4.3.2 Enveloppe convexe de fonction concaves sur un polyèdre convexe borné . . . 61
4.3.3 Enveloppe convexe de fonction séparable . . . 61
4.3.4 Minorante convexe de fonction DC sur un ensemble compact convexe . . . 62

5 Modélisation parcimonieuse 63
5.1 Introduction . . . 63
5.2 Apprentissage . . . 65
5.3 Traitement de signaux et d'images . . . 65
5.4 Analyse en composantes principales . . . 66
5.5 Classes de problèmes de parcimonie . . . 67
5.6 Quelques approximations de $\|.\|_0$ . . . 67
5.6.1 Approximation ℓ_p, $0 < p \leq 1$ . . . 68
5.6.2 Approximation exponentielle . . . 68
5.6.3 Approximation linéaire par morceaux . . . 68
5.6.4 Approximation logarithmique . . . 68
5.7 Solution parcimonieuse d'un systéme linéaire . . . 70
5.8 Problème des moindres carrés ℓ_0-régularisé . . . 72
5.9 Modèle parcimonieux pour le problème de la valeur propre maximale 73
5.10 Conclusion . . . 76

6 Modélisation parcimonieuse pour le problème de la valeur propre maximale 77
6.1 Introduction . . . 77
6.2 Notations . . . 78
6.3 Reformulations DC . . . 78
6.3.1 Problème avec contrainte ℓ_0 . . . 79
6.3.2 Problème ℓ_0-régularisé . . . 82
6.3.3 Commentaire sur le paramètre de pénalité . . . 83
6.4 DCA appliqué aux reformulations . . . 84
6.4.1 DCA pour la reformulation du problème avec contrainte ℓ_0 . . 84

6.4.2 DCA pour la reformulation du problème ℓ_0-régularisé 85
6.5 Expérimentation 86
6.5.1 ACP 86
6.5.2 ACP parcimonieuse 86
6.5.3 Tests numériques 86
6.5.3.1 Données artificielles 87
6.5.3.2 Données pit props 87
6.5.3.3 Données colon cancer 88
6.6 Conclusion 89

7 Modélisation parcimonieuse dans les modéles de régression linéaire 91
7.1 Introduction 91
7.2 Liens entre (P), (P^λ) et (Q^k) 92
7.3 Reformulations DC 94
7.3.1 Problème moindres carrés ℓ_0-régularisé (P^λ) 95
7.3.2 Problème moindres carrés avec contrainte ℓ_0 (Q^k) 97
7.3.3 Commentaires sur les paramètres de pénalité 100
7.3.4 Problème de minimisation ℓ_0 (P) 102
7.3.5 Quand est-ce que $(0, 0)$ est un point critique ? 103
7.3.6 Versions quadratiques 103
7.4 Formulations dans le cas avec contrainte de positivité 104
7.4.1 Cas de la minimisation ℓ_0 $(PosP)$ 106
7.4.2 Cas des moindres carrés ℓ_0-régularisés $(PosP^\lambda)$ 109
7.4.3 Cas des moindres carrés avec contrainte ℓ_0 $(PosQ^k)$ 111
7.5 Extensions 115
7.5.1 Cas ℓ_0-régularisé $(\bar{P}^\lambda)$ 116
7.5.2 Cas avec contrainte ℓ_0 $(\bar{Q}^k)$ 118
7.6 DCA pour les différentes reformulations DC 121
7.6.1 DCA pour (P_τ^λ) 122
7.6.2 DCA pour (Q_t^k) 123
7.7 Une technique combinée DCA et S.E. via la relaxation DC pour (P_τ^λ) 125
7.7.1 Relaxation DC de $(\hat{P}_\tau^\lambda)$ 126
7.7.2 Processus de séparation 126
7.7.3 Processus d'évaluation 127
7.7.3.1 Bornes inférieures 128
7.7.3.2 Amélioration de la borne supérieure 129
7.7.3.3 Algorithme combiné DCA-SE 129
7.8 Tests numériques préliminaires 129
7.9 Conclusion 132

Conclusion 132

Bibliographie 149

Notations

$\Gamma_0(X)$	représente la classe des fonctions $f : X \to]-\infty, +\infty]$ convexes, semi-continues inférieurement et propres.
$DC(C)$	la classe des fonctions DC sur C.
χ_C	la fonction indicatrice de l'nsemble C, $\chi_C(x) = 0$, si $x \in C$ et $\chi_C(x) = +\infty$ sinon.
P_C	la projection sur l'ensemble du sous ensemble C,
$\arg\min\ f$	ensemble des minimiseurs de la fonction f
∇f	gradient de f.
∂f	sous-différentiel de f.
$\lambda_{max}(A)$	la valeur propre maximale de la matrice A.
$e^1, ..., e^n$	les éléments de la base canonique de $\mathbb{R}^n$.
e	le vecteur de $\mathbb{R}^n$ dont toutes les composantes sont égales à 1.
$sign$	fonction signum.
$supp(.)$	fonction support.
$\langle ., . \rangle$	produit scalaire usuel de $\mathbb{R}^n$.
$\|.\|$	fonction valeur absolue.
$\|.\|_0$	norme ℓ_0.
$\|.\|_p$	norme ℓ_p.
$\mathbb{M}_{m,n}(\mathbb{R})$	ensemble des matrices réelles de m lignes et n colonnes.
$\mathbb{M}_n(\mathbb{R})$	ensemble des matrices carrées réelles d'ordre n.
$\mathbb{S}^n$	ensemble des matrices symétriques réelle d'ordre n.
$\mathbb{S}^n_+$	ensemble des matrices symétriques réelle semi-définie positive d'ordre n.
I	matrice identité d'ordre n.

A^T	matrice transposée de la matrice A.
J^c	complémentaire de l'ensemble J.
Card E	cardinal de l'ensemble E.

Chapitre 1

Introduction

« *We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances.* »

Isaac Newton

« *Everything should be made as simple as possible, but not simpler.* »

Albert Einstein

1.1 Contexte de l'étude

La modélisation parcimonieuse de données s'est développée au fil des années et devient de plus en plus incontournable dans les domaines où interviennent de très grandes quantités de données notamment en fouille de données et traitement de signaux et d'images, etc. Elle permet de supprimer les redondances sur les données et de ne considérer que les caractéristiques les plus pertinentes en supprimant celles qui sont superflues. Cette modélisation donne naisssance à des problèmes mathématiques notamment des problèmes d'optimisation contenant un terme de parcimonie discontinu.

A première vue certains des problèmes d'optimisation rencontrés ont des formulations avec des structures pour lesquelles les méthodes classiques en optimisation sont généralement inapplicables ou inefficaces. Face à cette situation des méthodes d'approximation ont été proposées en remplaçant le terme de parcimonie dans le problème d'optimisation original par des approximations afin d'obtenir un problème plus adapté pour la résolution. Nous distinguons généralement deux types d'approximations :

– approximations convexes afin d'utiliser des méthodes d'optimisation convexe,

– approximations continues non convexes permettant d'utiliser des méthodes de la programmation non convexe.

La question qui se pose immédiatement est de savoir si en résolvant le problème approché on résout le problème original. Autrement dit, on cherche à savoir quand est ce que les ensembles de solutions du problème approché et du problème original coincident ? quels sont les liens ?

Sauf dans des cas très spéciaux, les approximations proposées manquent de liens rigoureusement établis entre le problème original et le problème approché. Une des grandes difficultés est la reformulation équivalente de ces problèmes afin d'obtenir des problèmes avec des structures plus adaptées aux outils d'optimisation mathématique pour la résolution. Dans ce travail de thèse nous nous intéressons à la reformulation exacte et à la résolution de ces problèmes d'optimisation et la principale approche utilisée est la programmation DC (Difference of Convex functions) et DCA (DC Algorithms).

1.2 Vue d'ensemble du travail effectué

1.2.1 La Programmation DC et DCA en fouille de données

La fouille de données est un domaine émergent qui s'est développé depuis quelques décennies avec l'appartition de grandes bases de données et a de nombreuses applications dans différents domaines des sciences appliquées. Elle consiste à rechercher et extraire de l'information (utile et inconnue) stockée dans de grandes bases de données.

Nous parlerons dans ce chapitre du rôle central de la programmation DC et DCA dans la résolution des problèmes d'optimisation non convexes rencontrés dans différentes branches de la fouille de données (Apprentissage, classification, SVM, Méthodes à noyaux, reconnaissance de formes, tomographie binaire,...) et nous présenterons aussi l'état de l'art des méthodes DC pour les problèmes de fouille de données.

1.2.2 Programmation DC et DCA

La programmation DC (Difference of Convex functions) et DCA (DC Algorithms) ont été introduits par Pham Dinh Tao à l'état préliminaire en 1985, et ensuite intensivement développés par Pham Dinh Tao et Le Thi Hoai An depuis 1994 pour devenir maintenant classiques et de plus en plus utilisés par des chercheurs et praticiens de par le monde, dans différents domaines des sciences appliquées. Leur popularité réside dans leur robustesse et performance comparées à des méthodes existantes, leur adaptation aux structures des problèmes traités et leur capacité de résoudre des problèmes industriels de grande dimension. La programmation DC et DCA ont été appliqués avec grand succés à de nombreux et divers problèmes d'optimisation non convexe (différentiables ou non) et dans divers domaines : transport-logistique, télécommunication, génomique, finance, fouille de

données, cryptologie, mécanique, traitement d'image, robotique et vision par ordinateur, pétrochimie, contrôle optimal et automatique, problèmes inverses et problèmes mal posés, programmation multiobjectif, problèmes d'inégalités variationnelles, ... pour ne citer qu'eux.

Nous détaillerons dans ce chapitre les bases théoriques et algorithmiques de la programmation DC et DCA. Nous commencerons par quelques rappels d'analyse convexe, ensuite 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.

1.2.3 Technique de Séparation et Evaluation (SE) pour l'optimisation globale

La méthode de Séparation et Evaluation est une méthode d'énumération implicite intelligente pour résoudre de manière globale des problèmes d'optimisation non convexes continus ou combinatoires.

Nous détaillerons dans ce chapitre les bases théoriques et algorithmiques de la technique de séparation et évaluation dans le cas de la programmation non convexe continue. Nous y rappellerons différents résultats de convergence et de garantie d'optimalité globale. Nous présenterons aussi les différentes étapes pour la réalisation d'une procédure SE et la technique de relaxation DC.

1.2.4 Modélisation parcimonieuse

Dans ce chapitre, nous introduirons différents types de problèmes d'optimisation rencontrés en modélisation parcimonieuse. Nous y présenterons la formalisation mathématique de la notion de parcimonie et y détaillerons les méthodes d'approximation utilisées. Nous verrons les différentes classes de problèmes d'optimisation, incluant cette notion, et aussi différents résultats et approximations concernant ces classes de problèmes de parcimonie rencontrés dans la littérature.

1.2.5 Modélisation parcimonieuse pour le problème de la valeur propre maximale

Le problème de la valeur propre maximale est un problème bien connu et a de nombreuses applications en sciences appliquées et en ingénierie. Une des applications est l'analyse en composantes principales (ACP) en statistiques qui est un puissant outil pour l'analyse de facteur et la modélisation de données. L'ACP consiste à transformer des variables d'entrées liées entre elles en de nouvelles variables indépendantes les unes des autres. Ces nouvelles variables sont nommées composantes principales ou axes. Chaque composante étant une combinaison linéaire des variables d'entrées.

Dans la pratique, les axes obtenus prennent toute la dimension du problème c'est à dire presque toutes leurs composantes sont non nulles. Ce qui pose des problèmes

d'interprétation et de redondance de l'information obtenue à partir de ces axes la plupart des cas. Plusieurs techniques utilisant la notion de parcimonie ont été développées dans la littérature afin de contourner ce problème.

Dans ce chapitre nous considérerons deux formulations avec le terme de parcimonie et nous proposerons des reformulations DC exactes de ces deux formulations via de nouvelles techniques de pénalisation exacte. Nous présenterons également des résultats numériques préliminaires avec DCA, dans le contexte de l'analyse en composantes principales, qui illustrent bien le potentiel de cette nouvelle approche.

1.2.6 Modélisation parcimonieuse dans les modèles de régression linéaire

La modélisation parcimonieuse a émergé ces dernières années dans les modèles de regression en traitement de données notamment en traitement des signaux, des images et fouille de données. La parcimonie permet de prendre en compte la compressibilité des données à l'aide d'une représentation bien choisie. Elle est au coeur de l'échantillonnage compressé (compressed sensing, sparse signal recovery,...).

Nous considérerons dans ce chapitre plusieurs formulations permettant de prendre en compte cette notion de parcimonie dans les modèles de régression linéaire. Nous y discuterons aussi des liens entre ces différentes formulations. Ensuite nous proposerons des reformulations DC et des reformulations quadratiques exactes à l'aide de nouvelles techniques de pénalisation (différentes de celles de la section précédente). Ces techniques nous ont permis d'obtenir des caractéristiques intéressantes et une nouvelle approche concernant ces problèmes de parcimonie. Nous terminerons par des tests numériques préliminaires avec DCA et une méthode combinée DCA-SE pour l'optimisation globale.

1.3 Quelques activités de recherches effectuées durant la thèse

1. Publications
   - Mamadou Thiao, Tao Pham Dinh and Hoai An Le Thi, A Feature Selection Method by DC Programming and DCA. Submitted (2011).
   - Mamadou Thiao, Tao Pham Dinh and Hoai An Le Thi, Sparse Approximation by DC Programming. Submitted (2011).
   - Mamadou Thiao, Tao Pham Dinh and Hoai An Le Thi, A DC Programming Approach for Sparse Eigenvalue Problem. In proceedings of the 27th International Conference on Machine Learning (ICML 2010), pp. 1063-1070 (2010).
   - Mamadou Thiao, Tao Pham Dinh and Hoai An Le Thi, DC Programming Approach for a class of Nonconvex Programs Involving L0 Norm. Communications in Computer and Information Science, Springer, Vol. 14, pp. 348-357 (2008).

2. Conférences Internationales
   - Une méthode D.C. pour la Régression Logistique L0-régularisée. Roadef 2011 : 12e congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la décision, 2-3-4 Mars 2011, Saint-Etienne, France.
   - Mamadou Thiao, Sparse Fisher Discriminant Analysis by exact penalty techniques in DC Programming. 24th European Conference on Operational Research (EURO'10), July 11-14, 2010, Lisbonne-Portugal.
   - Mamadou Thiao, A DC Programming approach for Sparse Eigenvalue Problem. 27th International Conference on Machine Learning (ICML 2010), June 21-24, 2010, Haifa-Israel.
   - Mamadou Thiao, A DC Programming approach for feature selection in Support Vector Machines learning. 23rd European Conference on Operational Research (EURO'09), July 5-8, 2009, Bonn-Allemagne.
   - Mamadou Thiao, DC Programming Approach for a class of Nonconvex Programs Involving L0 Norm MCO'08. Modelling, Computation and Optimization in Information Systems and Management Sciences (MCO'08), September 8-10, 2008, Metz, France - Luxembourg.
3. Présentations en workshop et journées scientifiques
   - Sparse Approximation by DC Programming. Workshop : Optimization and learning : Theory, Algorithms and Applications, Metz, May 23-24, 2011.
   - A DC Programming approach for sparse Eigenvalue Problem and L0 norm reformulations. Workshop : Optimization and learning : Theory, Algorithms and Applications, Metz, June 17-18, 2010.
   - A DC Programming approach for feature selection in Support Vector Machines learning. Journées des doctorants de l'école doctorale SPMII Février 2009, Rouen.

Chapitre 2

La Programmation DC et DCA en fouille de données

La fouille de donnée est un domaine émergent qui s'est développé depuis quelques décennies avec l'appartition de grandes bases de données. Elle consiste à rechercher et extraire de l'information (utile et inconnue) stockée dans de grandes bases de données. Ce développement est dû à plusieurs facteurs : augmentation de la puissance de calculs des ordinateurs ; augmentation du volume des bases de données ; l'accès aux réseaux de taille mondiale, ces réseaux ayant un débit sans cesse croissant, qui rendent la distribution d'information sur un réseau d'échelle mondiale viable ; la prise de conscience de l'intérêt commercial pour l'optimisation des processus de fabrication, vente, gestion, logistique, ...

Le processus de fouille de données comprend plusieurs étapes :

1. collecte des informations et organisation de ces informations dans une base de données ;
2. nettoyage de la base de données : attributs sans valeur, ayant une valeur invalide (bruit), ... ; normalisation ;
3. sélection des attributs utiles ;
4. analyse statistique des données (réduction de la dimension, projection, etc...) ;
5. identification du type de problèmes (discrimination, clustering, etc...) et choisir un algorithme ;
6. évaluation des résultats de l'extraction de connaissance
7. déploiement de l'application.

On s'intéresse aux étapes 4, 5 et 6 où l'on utilise des méthodes mathématiques appelées aussi méthodes d'apprentissage statistique. Nous distinguons principalement trois types de méthodes d'apprentissage : l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage semi-supervisé.

Les méthodes d'apprentissage statistique se sont développées sur la base de la théorie de l'apprentissage statistique de Vapnik-Chervonenkis (Vapnik [150, 151]). L'une de ces méthodes, appelée Machine à Vecteur de Support ou SVM (Support Vector Machine) Cortes et al.[31], permet de réaliser des estimations en classification

(à deux classes ou plus) Burges [16] ou en régression Smola et Schölkopf [134]. De telles méthodes permettent généralement de s'affranchir de contraintes statistiques sur les données étudiées comme la normalité de la distribution.

La mise en œuvre de ces méthodes d'apprentissage donne naissance à des problèmes mathématiques. Ces problèmes consistant le plus souvent à minimiser des fonctionnelles, c'est à dire à résoudre des problèmes d'optimisation. On peut distinguer deux phases d'évolution concernant ces problèmes d'optimisation

- une première phase pionnière durant laquelle une très grande partie de ces problèmes d'optimisation dans les modèles utilisés étaient convexes voire linéaires,
- une deuxième phase d'amélioration des modèles existants et de développement de nouveaux modèles généralement non linéaires et/ou non convexes, afin de pallier aux faiblesses des premiers.

Les limites des modèles convexes, voire linéaires, dans la modélisation de problèmes de fouille de données de la vie courante sont le plus souvent à l'origine de ces faiblesses dont l'une des plus connues est le sur-apprentissage dans le cas de l'apprentissage. Dans le contexte de classification par exemple, le classifieur obtenu est très performant sur les échantillons utilisés, mais aura de la peine à généraliser les caractéristiques des données. Il se comporte alors comme une table contenant tous les échantillons utilisés lors de l'apprentissage et perd ses pouvoirs de prédiction sur de nouveaux échantillons. Pour remédier à ces faiblesses, des modèles non-linéaires et le plus souvent non convexes ont été développés dans la littérature. Ces derniers modèles, plus réalistes que les premiers, nécessitent la résolution de problèmes mathématiques ayant des structures pour lesquelles les méthodes de résolution classiques sont généralement inefficaces, voire inapplicables. Une grande partie de ces problèmes peuvent être formuler comme des problèmes d'optimisation non convexes et donc nécessitent des outils de l'optimisation non convexe (programmation non convexe) pour leur résolution.

La structure de ces problèmes non convexes se prête bien à la programmation DC et DCA qui jouent ainsi un rôle central pour leur résolution. La programmation DC (Difference of Convex functions) et DCA sont introduits en 1985 par Pham Dinh Tao et intensivement développés par Le Thi Hoai An et Pham Dinh Tao depuis 1994. Ils ont été utilisés avec succès, par des chercheurs et praticiens de par le monde, dans différentes branches des sciences appliquées. Ce succès étant dû à leur robustesse et performance comparées à des méthodes existantes, leur adaptation aux structures des problémes traités et leur capacité de résoudre des problémes industriels de grande dimension.

Plusieurs algorithmes utilisés pour la résolution de problèmes d'optimisation non convexe rencontrés dans le domaine de l'apprentissage se révèlent être des cas spéciaux de DCA : l'algorithme SLA (Successive Linearization Algorithm) ([14]) utilisé dans le cas de la minimisation d'une fonction concave ainsi que l'algorithme CCCP (Concave-Convex Procedure) ([164]).

Nous allons présenter dans la suite plusieurs travaux utilisant la programmation DC et DCA au niveau des différents types d'apprentissage rencontrés en fouille de données.

2.1 Apprentissage supervisé

Etant donné un ensemble X de n observations $x_i, i = 1, ..., n$ et un ensemble Y de n mesures (labels) $y_i, i = 1, ..., n$. L'apprentissage supervisé à pour objectif d'estimer les dépendances entre X et Y.

Dans ce contexte, la programmation DC et DCA ont été appliqués dans divers cadres : sélection de variables, sélection de noyaux,...

Une des premières applications de la programmation DC et DCA en fouille de données concerne le problème de sélection de variables qui est un problème très important en apprentissage et plus particulièrement en apprentissage supervisé. Il consiste à sélectionner un sous-ensemble de m variables parmi p ($p > m$) variables disponibles par rapport à un critère de qualité donné. La sélection permet de supprimer les redondances sur les caractéristiques des données, d'en extraire les plus pertinentes et d'éviter le sur-apprentissage. Ces méthodes de sélection sont classées en trois grandes catégories ("filter", "wrapper", "embedded") suivant le type du critère de sélection et la façon dont il est pris en compte dans la procédure d'apprentissage (Kovahi et John[76], Guyon et Elisseeff[57]). Les méthodes de sélection de la catégorie "filter" évaluent l'importance des variables en utilisant un critère statistique indépendant a priori du classifieur. Tandis que celles de la catégorie "wrapper", intègrent les performances prédictives du classifieur dans la procédure de recherche et d'évaluation de la qualité des sous-ensembles de variables. Quant aux méthodes de la catégorie "embedded", elles combinent la sélection de variables et l'estimation du modèle en une seule tâche.

Les approches de la programmation DC pour la sélection de variables concernent généralement cette dernière catégorie. Une des premières méthodes utilisant une approche par la programmation DC est due à Bradley et Mangasarian [14] qui ont proposé l'algorithme SLA (cas spécial de DCA) pour résoudre le problème de minimisation concave dans leur modèle de sélection. Plutard, Weston et al.[158] ont proposé une nouvelle approche non convexe basée sur une fonction logarithmique et le problème d'optimisation obtenu est un programme DC. Récemment, les travaux de Neumann et al.[113], Le Thi et al.[87, 92] et Wu et Liu [161] ont montré l'efficacité et la robustesse de la programmation DC et DCA pour résoudre des problèmes de sélection de variables.

La programmation DC et DCA ont été aussi très performants dans divers types de problèmes de classification : en classification biclasse dans l'approche "ψ-learning" de Shen et al.[133] et aussi dans les modèles de séparation sphérique de données par Astorino et al.[7] et Le Thi et al.[90] ; et en classification multiclasse, ils ont été appliqués dans l'approche "ψ-learning" de Liu et Shen [103], Liu et al.[104].

Ils ont été utilisés dans le modèle de régression présenté par Quadrianto et al.[129] et dans le modèle de sélection de noyaux de Ying et al.[163]. Il existe aussi une approche de sélection de noyaux utilisant une méthode cutting plane en programmation DC Argyriou et al.[6].

2.2 Apprentissage non-supervisé

Dans ce cas on ne dispose que d'un ensemble X de n observations $x_i, i = 1, ..., n$ et l'objectif est de décrire comment les données sont organisées et d'en extraire des sous-ensembles homogènes. On distingue principalement deux catégories de méthodes en apprentissage non-supervisé : les méthodes de partitionnement et les méthodes hiérarchiques. La première catégorie consiste à construire plusieurs partitions puis les évaluer selon certains critères tandis que la deuxième consiste à créer une décomposition hiérarchique des objets selon certains critères.

La programmation DC et DCA ont été très fructueux dans cette catégorie, et d'importants travaux ont été effectués aussi bien pour les méthodes de partitionnement que pour les méthodes hiérarchiques Belghiti et al.[10], Le Thi et al.[88, 84, 89, 86, 99].

2.3 Apprentissage semi-supervisé

Parmi les n observations $x_i, i = 1, ..., n$, seul un petit nombre d'entre eux est labellisé. L'objectif est le même que pour l'apprentissage supervisé mais on aimerait tirer profit des observations non labellisées. Les méthodes les plus utilisées sont, sans doute, les méthodes transductives à vecteur support ($TSVM$,S^3VM) Vapnik [150], Joachims [72], Bennet et Demiriz [11], Fung et Mangasarian [49], Chapelle et al. [25].

Ces méthodes consistent, le plus souvent, à minimiser des fonctions non convexes ayant des structures très adaptées à la programmation DC et DCA. De ce fait, la programmation DC et DCA ont été intensivement utilisés pour la résolution de ces problèmes non convexes dans plusieurs travaux : Bennet et Demiriz [11], Fung et Mangasarian [49], Collobert et al.[30, 29], Chapelle et al. [25] et Wang et al.[153, 154].

2.4 Projection

Les méthodes de projection (aussi appelées méthodes de réduction de la dimensionnalité) consistent à effectuer la projection des données depuis leur espace d'origine dans un espace plus petit. Le but est de visualiser ensuite ces projections afin de mieux comprendre un jeux de données et éventuellement d'en extraire des configurations. La méthode de projection la plus utilisée est sans doute l'analyse en composantes principales (ACP). C'est une méthode d'analyse de données classique, très utilisée et très connue en statistique et dans les sciences expérimentales. L'ACP consiste à transformer des variables liées entre elles en de nouvelles variables indépendantes les unes des autres et ces nouvelles variables sont nommées composantes principales.

Dans la pratique, les composantes principales obtenues prennent toute la dimension de l'espace d'origine ce qui pose, le plus souvent, des problèmes d'interprétabilité. Pour pallier à ces problèmes, des modèles parcimonieux ont été mis en œuvre afin de promouvoir des composantes principales parcimonieuses. Cependant,

les problèmes de parcimonie considérés ont des structures pour lesquelles les méthodes d'optimisation classiques sont généralement inefficaces, voire inapplicables. Une première approche utilisant la programmation DC et DCA a été proposée par Sriperumbudur et al. [135] et récemment, Thiao et al.[139] ont montré que ces problèmes de parcimonie peuvent être reformulés sous forme DC et d'intéressants résultats ont été obtenus avec DCA.

2.5 Conclusion

Dans ce chapitre, nous avons présenté brièvement la richesse des applications de la programmation DC et DCA (que nous allons détailler dans prochain chapitre 3) en fouille de données. Ils ont été appliqués avec grand succès sur presque tous les types de méthodes d'apprentissage statistique. Les différents résultats obtenus montrent l'efficacité et la robustesse de la programmation DC et DCA pour résoudre des problèmes d'optimisation non convexe obtenus lors de la modélisation de problèmes de la vie courante. Cette approche est aussi très adaptée pour la modélisation parcimonieuse de données que nous aborderons dans les chapitres 5, 6 et 7.

Chapitre 3

Programmation DC et DCA

Nous détaillerons dans ce chapitre les bases théoriques et algorithmiques de la programmation DC et DCA. Nous commencerons par quelques rappels d'analyse convexe, ensuite 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.

La Programmation DC (Difference of Convex functions) est une des branches de l'optimisation non convexe qui exploite la structure de fonctions qui s'écrivent comme différence de fonctions convexes (fonctions DC) pour résoudre des problèmes non convexes. Elle utilise les puissants outils de l'analyse convexe pour établir d'importantes propriétés et des conditions d'optimalité afin de résoudre ces problèmes. Elle constitue une généralisation de la programmation convexe dans le sens où la programmation convexe est un cas particulier de la programmation DC et que l'ensemble des fonctions DC est stable par rapport aux opérations usuelles rencontrées en optimisation. Nous distinguons principalement deux approches de résolution en programmation DC : l'approche globale dans l'esprit de l'optimisation combinatoire et l'approche convexe qui utilise une dualité dite DC et des conditions d'optimalité locale.

1. La première approche s'intéresse uniquement à la conception d'algorithmes permettant de calculer des solutions optimales globales dans l'esprit des procédures de Séparation et Evaluation, de coupes,... de l'optimisation combinatoire. Même si cette approche permet d'obtenir des solutions globales en théorie, elle ne permet pas d'atteindre la solution pour des problèmes concrets de grande dimension.
2. La deuxième quant à elle se base sur les outils de l'analyse convexe, la dualité DC et les conditions d'optimalité locale. Cette deuxième plus connue sous le nom Programmation DC et DCA (DC Algorithm) a été introduite par Pham Dinh Tao en 1985 et intensivement développée par Le Thi Hoai An et Pham Dinh Tao depuis 1994 pour devenir maintenant classique et de plus en plus utilisée de par le monde. DCA est une méthode de descente primale duale très

simple à mettre en oeuvre, sans recherche de pas linéaire et qui ne travaille que sur la décomposition DC. Il est basé sur les conditions d'optimalité locale et a une convergence linéaire dans le cas général. DCA a été utilisé sur beaucoup de problèmes d'optimisation non convexes dans divers domaines des sciences appliquées où il s'est montré très efficace comparer aux méthodes standard. On pourra se référer à [1] pour voir toute la richesse et toute l'étendue de la programmation DC et DCA.

Dans le premier paragraphe nous rappellerons quelques outils d'analyse convexe indispensables pour la programmation DC. Nous définirons ensuite la classe des fonctions DC et présenterons quelques propriétés de cette classe. Nous parlerons ensuite de la programmation DC et de l'algorithme DCA. Et nous terminerons par les techniques de pénalité exacte en programmation DC.

3.1 Quelques rappels d'analyse convexe

Dans ce paragraphe, nous allons rappeler quelques notions et outils d'analyse convexe fondamentaux pour la programmation DC. Nous abordons par exemple les notions d'indicatrice d'un ensemble, très utile pour transformer un problème d'optimisation avec contraintes en un problème sans contraintes, d'ensemble convexe et de projection sur un convexe. Nous parlons aussi des fonctions convexes et la notion de sous-différentiel très utile pour établir des conditions d'optimalités. On pourra se référer aux ouvrages Rockafellar [130] et Hiriart-Urruty et Lemarechal[63] pour plus de détails en analyse convexe. Dans la suite, X désigne 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 associée $\|x\|_2 = \sqrt{\langle x, x\rangle}$. Y désignera l'espace dual de X relatif au produit scalaire qui peut être identifié à X. L'analyse convexe moderne permet aux fonctions de prendre des 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 $(+\infty) - (+\infty) = (+\infty)$.

Définition 3.1.1 (Fonction indicatrice d'un ensemble) *Soit $C \subset X$ un ensemble. La fonction indicatrice de cet ensemble est notée χ_C et se définit par :*

$$(\forall x \in X),\ \chi_C(x) := \begin{cases} 0, & \text{si } x \in C \\ +\infty, & \text{si } x \notin C. \end{cases}$$

Définition 3.1.2 (Distance à un ensemble) *Soit $C \subset X$ un ensemble non vide. La distance à cet ensemble est notée d_C et se définit par :*

$$\begin{aligned} d_C : \quad & X \to [0, +\infty[\\ & u \mapsto \inf_{v \in C} \|u - v\| . \end{aligned}$$

Définition 3.1.3 (Ensemble convexe) *Soit $C \subset X$ un ensemble. Cet ensemble est dit convexe si*

$$\forall((u,v) \in C^2)\ \ (\forall \lambda \in]0,1[)\ \ \lambda u + (1-\lambda)v \in C.$$

Définition 3.1.4 (Polyédre convexe) *On dit qu'un sous ensemble $C \subset \mathbb{R}^n$, est un polyédre convexe s'il est l'intersection d'un nombre fini de demi-espaces de $\mathbb{R}^n$, i.e.,*

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

Théorème 3.1.5 (Projection sur un convexe) *Soit $C \subset X$ un ensemble convexe fermé non vide et soit $u \in X$. Il existe alors un point unique noté $P_C(u) \in C$ tel que*

$$d_C(u) = \|u - P_C(u)\|.$$

On appelle $P_C(u)$ la projection de u sur C.

Définition 3.1.6 (Enveloppe convexe) *Soit $S \subset X$. L'enveloppe convexe de S, notée $co(S)$, est l'intersection de tous les sous ensembles convexes de X contenant S. En d'autes termes c'est le plus petit sous ensemble convexe de X contenant S.*

Définition 3.1.7 (Combinaison convexe) *Soient $x^1, ..., x^k$ des points de X. Un point $x \in X$ est dit combinaison convexe des points x^i $(1 \leq i \leq k)$ s'il peut s'écrire sous la forme*

$$x = \sum_{i=1}^{k} \lambda_i x^i,$$

les λ_i étant des réels vérifiant

$$\lambda_i \geq 0,\ (1 \leq i \leq k),\ \sum_{i=1}^{k} \lambda_i = 1.$$

On vérifie aisément que l'ensemble des combinaisons convexes des points x^i, $1 \leq i \leq k$ est convexe et fermé. C'est le plus petit convexe contenant les x^i, c'est à dire l'enveloppe convexe de la famille de points x^i, $1 \leq i \leq k$. On montre que c'est un polyèdre convexe.

Définition 3.1.8 (Point extrémal et sommet) *Soit A une partie convexe et fermée de X. Un point x de A est dit extrémal si l'égalité*

$$x = ty + (1-t)z,\ \ y \text{ et } z \in A,\ \ 0 < t < 1,$$

n'a pas d'autres solutions que $x = y = z$. Géométriquement cela signifie qu'il n'existe pas de segment de droite non réduit à un point, contenu dans A et contenant le point

x à son intérieur (c'est à dire de manière telle que x ne soit pas une des extrémités de ce segment).

Lorsque A est un polyédre convexe, un point extrémal est aussi appelé sommet.

Théorème 3.1.9 *Lorsque X est de dimension finie, $co(S)$ est l'ensemble des points de $x \in X$ s'écrivant comme combinaison convexe finie d'éléments de S, i.e.,*

$$co(S) = \left\{ x = \sum_i^m \lambda_i x^i : \ x^i \in S, \ \lambda_i \geq 0 \ i = 1, ..., m; \ \sum_i^m \lambda_i = 1 \right\}.$$

Définition 3.1.10 (Variété affine) *Soit $C \subset X$ un ensemble convexe non vide. La variété affine engendrée par C est notée aff C et se définit par*

$$\text{aff } C = \left\{ \sum_i \lambda_i x^i : \ x^i \in C, \ \sum_i \lambda_i = 1, \lambda_i \in \mathbb{R} \right\},$$

où seules les sommes finies sont prises en compte.

Définition 3.1.11 (Intérieur et intérieur relatif d'un convexe) *Soit $C \subset X$ un ensemble convexe non vide. L'intérieur du convexe C est noté int C et se définit par*

$$\text{int } C = \{u \in C : \exists \rho > 0, \ \ \mathcal{B}(u; \rho) \subset C\},$$

où $\mathcal{B}(u; \rho)$ désigne la boule de rayon ρ centrée en u. L'intérieur relatif de C est noté ir C et se définit par

$$\text{ir } C = \{u \in C : \exists \rho > 0, \ \ \mathcal{B}(u; \rho) \cap \text{aff } C \subset C\},$$

où aff désigne la variété affine de C.

On peut remarquer que int $C \subset$ ir $C \subset C$.

Définition 3.1.12 (Domaine d'une fonction) *Soit $f : X \to]-\infty, +\infty]$. Le domaine de la fonction f est l'ensemble, noté dom f, défini par*

$$\text{dom } f = \{x \in X : \ f(x) < +\infty\}.$$

On dit que f est propre si dom $f \neq \varnothing$ *(i.e. elle n'est pas identiquement égale à l'infini).*

Théorème 3.1.13 (Fonction coercive) *Une fonction $f : X \to]-\infty, +\infty[$ est dite coercive si elle vérifie :*

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

Définition 3.1.14 (Fonction convexe) *Une fonction $f : X \to]-\infty, +\infty]$ est dite convexe si elle vérifie :*

$$\forall ((x, y) \in X^2) \ \ (\forall \lambda \in]0, 1[) \ \ f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y). \qquad (3.1)$$

La fonction f est strictement convexe si l'inégalité (3.1) est stricte dès que $(x, y) \in (dom\ f)^2$ et $x \neq y$.

Définition 3.1.15 (Épigraphe) *L'épigraphe d'une fonction $f : X \to]-\infty, +\infty]$ est noté Epi f et se définit par :*

$$\text{Epi } f = \{(x, \alpha) \in X \times \mathbb{R} : \ f(x) \leq \alpha\}.$$

La fonction f est convexe si et seulement si Epi f est un sous ensemble convexe de $X \times \mathbb{R}$.

Définition 3.1.16 (Fonction fortement convexe) *Une fonction $f : X \to]-\infty, +\infty]$ est dite fortement convexe de module ρ (on dit aussi ρ-convexe) si*

$$\forall((x, y) \in X^2) \ \ (\forall \lambda \in]0, 1[) \ \ f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \lambda(1-\lambda)\frac{\rho}{2} \|x - y\|^2,$$

en d'autres termes la fonction $f - \frac{\rho}{2} \|.\|^2$ est convexe sur X.
Le *module de forte convexité* de f sur X, noté $\rho(f, X)$, est défini par

$$\rho(f, X) = \sup \left\{ \rho > 0 : \ \ f - \frac{\rho}{2} \|.\|^2 \text{ est convexe sur } X \right\}.$$

Ainsi la fonction f est fortement convexe si et seulement si $\rho(f, X) > 0$. On peut aussi remarquer que toute fonction fortement convexe est strictement convexe.

Définition 3.1.17 (Sous-différentiel et sous-gradient) *Soit $f : X \to]-\infty, +\infty]$ une fonction propre et soit $x^0 \in X$. Le sous-différentiel de f en x^0 est noté $\partial f(x^0)$ et se définit par*

$$\partial f(x^0) = \left\{ y \in X : \ (\forall x \in X) \ \ f(x) \geq f(x^0) + y^T (x - x^0) \right\}.$$

Un élément de $\partial f(x^0)$ est appelé sous-gradient de f en x^0. Le domaine du sous-différentiel de f, noté dom ∂f, est défini par

$$\text{dom } \partial f = \{x \in X : \ \partial f(x) \neq \varnothing\}.$$

Théorème 3.1.18
- *$\partial f(x)$ est un sous ensemble convexe fermé de X.*
- $\text{ir}(\text{dom}(f)) \subset \text{dom } \partial f \subset \text{dom } f$.

Définition 3.1.19 (ϵ-sous-différentiel et ϵ-sous-gradient) *Soit ϵ un réel positif, soit $f : X \to]-\infty, +\infty]$ une fonction propre et soit $x^0 \in X$. Le ϵ-sous-différentiel de f en x^0 est noté $\partial_\epsilon f(x^0)$ et se définit par*

$$\partial_\epsilon f(x^0) = \left\{ y \in X : \ (\forall x \in X) \ \ f(x) \geq f(x^0) + y^T (x - x^0) - \epsilon \right\}.$$

Un élément de $\partial_\epsilon f(x^0)$ est appelé ϵ-sous-gradient de f en x^0.

Définition 3.1.20 (Fonction s.c.i.) *Soit* $f : X \to]-\infty, +\infty]$ *et soit* $x^0 \in X$. *La fonction* f *est semi-continue inférieurement au point* x^0 *si*

$$\liminf_{x \to x^0} f(x) \geq f(x^0).$$

La fonction f *est dite semi-continue inférieurement sur* X *si elle l'est en tout point* x^0 *de* X.
Soit f *une fonction propre sur* X, *on appelle fermeture de* f, *notée* $\bar{f}$, *la plus grande fonction semi-continue inférieurement sur* X *et minorante de* f.

Définition 3.1.21 (Classe $\Gamma_0(X)$) $\Gamma_0(X)$ *représente la classe des fonctions* $f : X \to]-\infty, +\infty]$ *convexes, semi-continue inférieurement et propres.*

Théorème 3.1.22
- *Si* $f \in \Gamma_0(X)$, *alors*
  - f *est différentiable en* x^0 *si et seulement si* $\partial f(x^0) = \{\nabla f(x^0)\}$.
  - $x^0 \in \arg\min \{f(x) : x \in X\}$ *si et seulement si* $0 \in \partial f(x^0)$.
- *Soit* $\alpha \geq 0$, $f \in \Gamma_0(X)$ *et* $x \in X$. *Alors*

$$\partial(\alpha f)(x) = \alpha \partial f(x).$$

- *Soient* $f_1, ..., f_p \in \Gamma_0(X)$ *et* $x \in X$. *Alors*

$$\partial(f_1 + ... + f_p)(x) \supset \partial f_1(x) + ... + \partial f_p(x),$$

avec égalité si

$$\bigcap_{i=1,...,p} \text{ir}(\text{dom } f_i) \neq \varnothing.$$

Dans cette dernière condition, on peut remplacer ir(dom f_i*) par dom* f_i *si* f_i *est polyédrale,* $i = 1, ..., p$.

Définition 3.1.23 (Fonction conjuguée) *La fonction conjuguée d'une fonction* $f \in \Gamma_0(X)$ *est notée* f^* *et se définit par*

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

Théorème 3.1.24
- $f \in \Gamma_0(X)$ *si et seulement si* $f^* \in \Gamma_0(X)$ *et* $(f^*)^* = f$.
- *Si* $f \in \Gamma_0(X)$, *alors*
  - $y \in \partial f(x)$ *si et seulement si* $x \in \partial f^*(y)$.
  - $y \in \partial f(x)$ *si et seulement si* $f(x) + f^*(y) = \langle x, y \rangle$.

Définition 3.1.25 (Fonction convexe polyédrale) *Une fonction* $f : \mathbb{R}^n \to]-\infty, +\infty]$ *est dite convexe polyédrale si son épigraphe est un polyédre convexe de* $\mathbb{R}^{n+1}$.

Théorème 3.1.26 *Soit $f \in \Gamma_0(\mathbb{R}^n)$. ALors f est une fonction convexe polyédrale si et seulement si dom f est un polyédre convexe et*

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

Théorème 3.1.27

- *Si f_1 et f_2 sont convexes polyédrales, alors il en est de même pour les fonctions $f_1 + f_2$ et $\max(f_1, f_2)$.*
- *Si $f \in \Gamma_0(X)$ est convexe polyédrale, alors*
  - *$\partial f(x)$ est un polyédre convexe non vide pour tout $x \in$ dom f.*
  - *f^* l'est aussi et dom $\partial f =$ dom f. De plus si f est finie partout, alors*

$$\texttt{dom } f^* = co\{a_i : \ i = 1, ..., m\},$$

$$f^*(y) = \min\left\{\sum_{i=1}^m \lambda_i b_i : \ y = \sum_{i=1}^m \lambda_i a_i, \sum_{i=1}^m \lambda_i = 1, \lambda_i \geq 0, \ i = 1, ..., m\right\}.$$

3.2 Fonctions DC

Dans ce paragraphe, nous allons donner la définition d'une fonction DC et des propriétés essentielles de l'ensemble des fonctions DC notamment sa stabilité par rapport aux opérations usuelles en optimisation.

Définition 3.2.1 (Fonction DC) *Soit C un sous ensemble convexe fermé non vide de X. Une fonction $f : C \to]-\infty, +\infty]$ est dite DC sur C si elle peut s'écrire sous la forme*

$$f(x) = g(x) - h(x), \ \ \forall x \in C,$$

où g et h sont deux fonctions convexes sur C. On dit alors que $g - h$ est une décomposition DC de f.

Cette décomposition n'est pas unique. En effet étant donné une fonction ϕ convexe finie sur C, nous avons $(g + \phi) - (h + \phi)$ qui est aussi une décomposition DC de f.

Théorème 3.2.2 *Toute fonction DC admet une infinité de décompositions DC.*

Définition 3.2.3 (Ensemble des fonctions DC) *Soit Conv(C) l'ensemble des fonctions convexes sur C. L'ensemble des fonctions DC sur C est noté $DC(C)$ et se définit par*

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

Une caractéristique très importante de l'ensemble $DC(C)$ est sa stabilité par rapport aux opérations usuelles rencontrées en optimisation [3, 59, 147, 67].

Théorème 3.2.4 *Si $f = g - h$, $f_i = g_i - h_i \in DC(C)$, $i = 1, ..., m$, alors*

– *Toute combinaison linéaire finie de fonctions DC est une fonction DC, i.e.,*

$$\sum_{i=1}^{m} \lambda_i f_i \in DC(C), \forall \lambda_i \in \mathbb{R}.$$

– $\max_{i=1,\ldots,m} f_i \in DC(C)$ et $\min_{i=1,\ldots,m} f_i \in DC(C)$, *car*

$$\max_{i=1,\ldots,m} f_i = \max_{i=1,\ldots,m} [g_i + \sum_{j=1,j\neq i}^{m} h_j] - \sum_{j=1}^{m} h_j,$$

et

$$\min_{i=1,\ldots,m} f_i = \sum_{j=1}^{m} g_j - \max_{i=1,\ldots,m} [h_i + \sum_{j=1,j\neq i}^{m} g_j].$$

– $|f| \in DC(C)$, *car*

$$|f| = 2\max(g,h) - (g+h).$$

– $f^+, f^- \in DC(C)$ *où* $f^+(x) = \max(0, f(x))$ *et* $f^-(x) = \max(0, -f(x))$, *car*

$$f^+ = \max(g,h) - h,$$

$$f^- = \max(g,h) - g.$$

– $\prod_{i=1}^{m} f_i \in DC(C)$.

Une deuxième caractéristique importante est que $DC(C)$ est l'espace vectoriel engendré par $\mathrm{Conv}(C)$, il contient l'ensemble des fonctions convexes sur C, l'ensemble des fonctions concaves sur C, ainsi beaucoup de fonctions à la fois ni convexes et ni concaves sur C. Il contient aussi la quasi-totalité des fonctions rencontrées dans les problèmes concrets en optimisation non convexe. Et joue ainsi un rôle clé en optimisation non convexe.

Théorème 3.2.5 [62, 43, 147, 67]

– *Toute fonction polynôme sur $\mathbb{R}^n$ est DC sur $\mathbb{R}^n$.*
– *$DC(C)$ contient l'ensemble des fonctions sous-$\mathcal{C}^2$ sur C (une fonction f est dite sous-$\mathcal{C}^2$ sur C si elle est localement l'enveloppe supérieure d'une famille de fonctions $\mathcal{C}^2$ sur C).*
– *$DC(C)$ contient aussi l'ensemble $\mathcal{C}^{1,1}(C)$ des fonctions dont le gradient est localement lipschitzien sur C.*
– *Si C est un sous ensemble compact convexe non vide de $\mathbb{R}^n$, alors $DC(C)$ est dense dans $\mathcal{C}^0(C)$, l'ensemble des fonctions continues sur C.*

3.3 Programmation DC

3.3.1 Programme DC

Définition 3.3.1.1 (Programme DC) *On appelle programme DC tout problème d'optimisation de la forme*

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

où g et h sont des éléments de $\Gamma_0(\mathbb{R}^n)$.

(P_{dc}) est un problème d'optimisation sans contrainte. Cependant un problème d'optimisation avec des contraintes convexes fermées C de la forme

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

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

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

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

Ces dernières années on note des recherches très actives en optimisation concernant les classes de problèmes d'optimisation suivantes :

1. $\sup\{f(x) : \ x \in C\}$, où f et C sont convexes.
2. $\inf\{g(x) - h(x) : \ x \in \mathbb{R}^n\}$, où g et h sont convexes.
3. $\inf\{g(x) - h(x) : \ x \in C, \ g_1(x) - h_1(x) \leq 0\}$, où g, h, g_1, h_1 et C sont convexes.

La raison est bien simple : 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.

Le problème 1. est un cas special du problème 2. dans lequel g est la fonction indicatrice de C et $h = f$ et le problème 2. (resp. problème 3.) peut être ramené à la forme 1. (resp. 2.) en ajoutant une variable t,

$$\inf\{g(x) - h(x) : \ x \in \mathbb{R}^n\} \Leftrightarrow \sup\{h(x) - t : \ g(x) - t \leq 0\}$$

(resp. via la technique de pénalité exacte relative à la contrainte $g_1(x) - h_1(x) \leq 0$). Il est clair que la complexité de ces problèmes croît de 1. à 3. et que la résolution de l'un d'entre eux implique la résolution des deux autres. Le problème 2. est le programme DC présenté ci-dessus. Il présente une structure très intéressante permettant d'importants développements tant sur le plan théorique que sur le plan pratique (algorithmique).

Cette structure a permis le développement d'une élégante théorie de dualité Toland [142, 143] et de conditions d'optimalité et aussi d'intéressants algorithmes de résolution dont le plus remarquable est, sans doute, l'algorithme DCA qui a

été introduit par Pham Dinh Tao [119, 120, 121] à l'état préliminaire en 1985, et ensuite intensivement développé par Pham Dinh Tao et Le Thi Hoai An depuis 1994 [122, 80, 136, 123, 81, 93, 124, 101, 82, 96, 125, 83, 97, 98, 127]. Dans la suite du chapitre, nous allons présenter la dualité, les conditions d'optimalité et l'algorithme DCA.

3.3.2 Dualité en programmation DC

Dans ce paragraphe, nous allons introduire brièvement le concept de dualité en programmation DC. Bien qu'établie depuis longtemps en analyse convexe, la notion de dualité dans le cas non convexe a été proposée récemment avec les travaux [8, 117] dans les cas Quasi-convexe et Anti-convexe et introduite par Toland [142, 143] dans le cadre de la programmation DC comme une généralisation des travaux sur la maximisation convexe de Pham Dinh Tao [118]. Cette théorie de dualité DC est très riche et on pourra se référer aux travaux de Let Thi Hoai An [80] pour plus de détails.

Considérons le programme DC

$$(P_{dc}) \quad \alpha = \inf \{ f(x) = g(x) - h(x) : \quad x \in X \},$$

où $g, h \in \Gamma_0(X)$.
Comme $h \in \Gamma_0(X)$, on a $(h^*)^* = h$ et donc

$$h(x) = (h^*)^*(x) = \sup \{ \langle x, y \rangle - h^*(y) : \ y \in Y \}, \ \forall x \in X.$$

Ainsi

$$\begin{aligned} \alpha &= \inf \{ f(x) = g(x) - \sup \{ \langle x, y \rangle - h^*(y) : y \in Y \} : x \in X \} \\ &= \inf \{ \alpha(y) : y \in Y \} \end{aligned}$$

avec

$$\begin{aligned} \alpha(y) &= \inf \{ g(x) - [\langle x, y \rangle - h^*(y)] : x \in X \} \\ &= \begin{cases} h^*(y) - g^*(y), & \text{si } y \in \texttt{dom}\, h^* \\ +\infty, & \text{sinon .} \end{cases} \end{aligned}$$

Avec la convention $(+\infty) - (+\infty) = (+\infty)$, nous obtenons le problème dual de (P_{dc}), noté (D_{dc})

$$(D_{dc}) \quad \alpha = \inf \{ h^*(y) - g^*(y) : \quad y \in Y \}.$$

Comme g^* et h^* appartiennent à $\Gamma_0(Y)$, le problème (D_{dc}) est un programme DC. De plus, (P_{dc}) et (D_{dc}) ont la même valeur optimale α. En outre, il existe une parfaite symétrie entre (P_{dc}) et (D_{dc}), le dual de (D_{dc}) est exactement (P_{dc}).
Les résultats suivants donnent quelques propriétés concernant les solutions des problèmes (P_{dc}) et (D_{dc}).

Théorème 3.3.2

– *x^* est une solution optimale globale de (P_{dc}) si et seulement si*

$$\alpha = (g - h)(x^*) \leq (h^* - g^*)(y), \ \forall y \in Y.$$

– *y^* est une solution optimale globale de (D_{dc}) si et seulement si*

$$\alpha = (h^* - g^*)(y^*) \leq (g - h)(x),\ \forall x \in X.$$

Théorème 3.3.3 *Soient $g, h \in \Gamma_0(X)$. Alors*

–
$$\inf\{g(x) - h(x) :\ \ x \in \text{dom } g\} = \inf\{h^*(y) - g^*(y) :\ \ y \in \text{dom } h^*\}.$$

– *Si y^* est un minimum de $h^* - g^*$ sur Y, alors tout point $x^* \in \partial g^*(x^*)$ est un minimum de $g - h$ sur X.*
– *Si x^* est un minimum de $g - h$ sur X, alors tout point $y^* \in \partial h(y^*)$ est un minimum de $h^* - g^*$ sur Y.*

Le théorème précédent donne une relation entre les solutions du problème primal P_{dc} et de son dual D_{dc}. Il montre que la résolution de l'un implique la résolution de l'autre. Ainsi pour résoudre P_{dc}, on pourra d'abord résoudre D_{dc} et vice versa, suivant que l'un ou autre est plus « simple »à résoudre. Il existe même des cas particuliers où le problème dual est convexe tandis que le problème primal est non convexe. L'exemple suivant illustre un tel cas.

Exemple 3.3.4 Soient $n > 1$, $g(x) = e^x$ et $h(x) = ne^{\frac{x}{n}}$, pour tout $x \in \mathbb{R}$. Le problème primal

$$(P)\ \ \min\left\{g(x) - h(x) = e^x - ne^{\frac{x}{n}} :\ \ x \in \mathbb{R}\right\}$$

est non convexe tandis que son dual

$$(D)\ \ \min\{h^*(y) - g^*(y) = (n-1)g^*(y) :\ \ y \in \mathbb{R}\}$$

est convexe. En effet, on a $h(x) = ng(\frac{x}{n})$ et

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

Donc

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

3.4 Conditions d'optimalité en programmation DC

Il est bien connu en analyse convexe qu'une condition nécessaire et suffisante pour que x minimise une fonction convexe f est

$$0 \in \partial f(x).$$

Cette condition bien que nécessaire et suffisante dans le cas convexe ne l'est plus dans le cas non convexe. En programmation DC, une condition nécessaire et suffisante d'optimalité globale utilisant le ϵ-sous différentiel est donnée par le théorème suivant.

Théorème 3.4.1 (Condition d'optimalité globale) *Soient $g, h \in \Gamma_0(X)$ et $f = g - h$. Alors x^* est un minimum global de $g - h$ si et seulement si*

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

Cette caractérisation est une traduction directe du théorème 3.3.2 et est invérifiable en pratique.

Remarque 3.4.2

1. Si $f \in \Gamma_0(X)$, on peut écrire $g = f$ et $h = 0$. La condition d'optimalité précédente est identique à celle dans le cas convexe du fait que $\partial_\epsilon h(x^*) = \{0\}\ \forall \epsilon > 0$.
2. D'une manière plus générale, considérons les décompositions DC de $f \in \Gamma_0(X)$ de la forme $f = g - h$ avec $g = f + h$ et $h \in \Gamma_0(X)$ finie partout sur X. Dans ce cas, la condition d'optimalité $0 \in \partial f(x^*)$ équivaut à $\partial h(x^*) \subset \partial g(x^*)$.

Notation On note par $\mathcal{P}$ et $\mathcal{D}$ les ensembles de solution des problèmes P_{dc} et D_{dc} respectivement.

Définition 3.4.3 (Programme DC polyédral) *Un programme DC est dit polyédrale si l'une de ses composantes convexes g ou h est une fonction convexe polyédrale.*

Définition 3.4.4 *On appelle point critique ou point KKT généralisé, tout point $x^* \in X$ vérifiant*

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

Définition 3.4.5 *Soient g et h deux fonctions de $\Gamma_0(X)$. Un point $x^* \in \mathrm{dom}(g) \cap \mathrm{dom}(h)$ est un minimum local de $g - h$ sur X si et seulement si*

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

où $V(x^*)$ désigne un voisinage de x^*.

Théorème 3.4.6 (Condition nécessaire d'optimalité locale) *Si x^* est un minimum local de $g - h$, alors*

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

Cette dernière condition étant suffisante dès que h est polyédrale.

Théorème 3.4.7 (Condition suffisante d'optimalité locale) *Si x^* admet un vosinage V tel que*

$$\partial h(x) \cap \partial g(x^*) \neq \varnothing,\ \forall x \in V \cap \mathrm{dom}\ g,$$

alors x^* est un minimum local de $g - h$.

Théorème 3.4.8 (Condition suffisante d'optimalité locale stricte) *Si $x^* \in \mathrm{int}(\mathrm{dom}\ h)$ vérifie*

$$\partial h(x^*) \subset \mathrm{int}(\partial g(x^*)),$$

alors x^* est un minimum local strict de $g - h$.

Théorème 3.4.9

1. $\partial h(x^*) \subset \partial g(x^*),\ \forall x^* \in \mathcal{P}$ *et* $\partial g^*(y^*) \subset \partial h^*(y^*),\ \forall y^* \in \mathcal{D}$.
2. *Transport de minima globaux :*

$$\left[\bigcup_{x^* \in \mathcal{P}} \partial h(x^*)\right] \subseteq \mathcal{D} \subset \operatorname{dom} h^*.$$

La première inclusion devient égalité dès que g^* est sous-différentiable dans $\mathcal{D}$ (en particulier si $\mathcal{D} \subset \operatorname{ir}(\operatorname{dom} g^*)$ ou si g^* est sous-différentiable dans $\operatorname{dom} h^*$*) et dans ce dernier cas $\mathcal{D} \subset \operatorname{dom} \partial g^* \cap \operatorname{dom} \partial h^*$.*

$$\left[\bigcup_{y^* \in \mathcal{D}} \partial g^*(y^*)\right] \subseteq \mathcal{P} \subset \operatorname{dom} g.$$

La première inclusion devient égalité dès que h est sous-différentiable dans $\mathcal{P}$ (en particulier si $\mathcal{P} \subset \operatorname{ir}(\operatorname{dom} h)$ ou si h est sous-différentiable dans $\operatorname{dom} g$*) et dans ce dernier cas $\mathcal{P} \subset \operatorname{dom} \partial g \cap \operatorname{dom} \partial h$.*

3. *Transport de minima locaux : Soit $x^* \in \operatorname{dom} \partial h$ un minimum local de $g - h$. Soient $y^* \in \partial h(x^*)$ et $V(x^*)$ un voisinage de x^* tel que $g(x) - h(x) \geq g(x^*) - h(x^*),\ \forall x \in V(x^*) \cap \operatorname{dom} g$. Si $x^* \in \operatorname{int}(\operatorname{dom} h)$ vérifie*

$$y^* \in \operatorname{int}(\operatorname{dom} g^*), \text{ et } \partial g^*(y^*) \subset V(x^*),$$

alors y^* est un minimum local de $h^* - g^*$.

3.5 DCA (DC Algorithm)

Basé sur la dualité DC et les conditions d'optimalité locale, l'algorithme DCA consiste à construire deux suites $\{x^l\}$ et $\{y^l\}$ candidates pour être solutions des problèmes primal et dual respectivement, vérifiant

- les suites $\{g(x^l) - h(x^l)\}$ et $\{h^*(y^l) - g^*(y^l)\}$ sont décroissantes,
- la suite $\{x^l\}$ (resp.$\{y^l\}$) converge vers un point admissible du problème primal $\tilde{x}$ (resp. un point admissible du problème dual $\tilde{y}$) vérifiant la condition d'optimalité locale et

$$\tilde{x} \in \partial g^*(\tilde{y}), \qquad \tilde{y} \in \partial h(\tilde{x}).$$

Étant donné un point $x^0 \in \operatorname{dom} g$, appelé *point initial*, et un point $y^0 \in \partial h(x^0)$, les suites $\{x^l\}$ et $\{y^l\}$ sont déterminées de sorte que x^{l+1} (resp. y^{l+1}) est une solution du problème convexe (P_l) (resp. (D_{l+1})) défini par

$$(P_l) \quad \min\left\{g(x) - h(x^l) - \langle x - x^l, y^l\rangle :\ x \in \mathbb{R}^n\right\},$$

Algorithm 1 DC Algorithm (DCA)

$x^0 \in \text{dom } g$, $l \leftarrow 0$.

Répéter

1. Calculer $y^l \in \partial h(x^l)$,
2. Calculer $x^{l+1} \in \partial g^*(y^l)$,

Jusqu'à la convergence.

$$(D_{l+1}) \quad \min\left\{h^*(y) - g^*(y^l) - \langle y - y^l, x^{l+1}\rangle : \ y \in \mathbb{R}^n\right\}.$$

L'interprétation de DCA est simple : à chaque itération, la seconde composante h dans le problème primal (P_{dc}) est remplacée par sa minorante affine $h_l(x) := h(x^l) + \langle x - x^l, y^l\rangle$ au voisinage de x^l, définie par un sous-gradient y^l de h en x^l, pour obtenir le problème primal convexe (P_l), dont l'ensemble des solutions n'est rien d'autre que $\partial g^*(y^l)$. Par dualité, une solution x^{l+1} de (P_l) est alors utilisée pour définir le problème convexe dual (D_{l+1}), obtenu à partir (D_{dc}) en remplaçant sa seconde composante DC g^* par sa minorante affine $g_l^*(y) := g^*(y^l) + \langle y - y^l, x^{l+1}\rangle$ au voisinage de y^l, dont l'ensemble des solutions est exactement $\partial h(x^{l+1})$. Le processus est alors réitéré jusqu'à la convergence. Ainsi DCA utilise une double linéarisation à l'aide des sous-gradients de h et g^*. En résumé, on peut décrire l'algorithme DCA comme suit : partant d'un point $x^0 \in \text{dom } g$,

$$y^l \in \partial h(x^l); \ \ x^{l+1} \in \partial g^*(y^l), \ \forall\, l \geq 0.$$

De façon schématique

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

Sur l'algorithme 1, nous résumons l'algorithme DCA.

3.5.1 Convergence de DCA

Lemme 3.5.1.1 (Existence des suites) *Les deux propriétés suivantes sont équivalentes :*

1. *Les suites* $\left\{x^l\right\}$ *et* $\left\{y^l\right\}$ *sont bien définies*
2. dom $\partial g \subset$ dom ∂h *et* dom $\partial g \subset$ dom ∂h.

Lemme 3.5.1.2 (Bornitude des suites)

1. *Si* $g - h$ *est coercive, alors la suite* $\left\{x^l\right\}$ *est bornée. Si de plus* $\left\{x^l\right\} \subset$ int(dom h), *alors la suite* $\left\{y^l\right\}$ *est bornée.*

2. *Si $h^* - g^*$ est coercive, alors la suite $\{y^l\}$ est bornée. Si de plus $\{y^l\} \subset \text{int}(\text{dom } g^*)$, alors la suite $\{x^l\}$ est bornée.*

Théorème 3.5.1.3 (Convergence de DCA)

1. *Le suites $\{g(x^l) - h(x^l)\}$ et $\{h^*(y^l) - g^*(y^l)\}$ sont décroissantes et*
   - *$g(x^{l+1}) - h(x^{l+1}) = g(x^l) - h(x^l)$, si et seulement si $y^l \in \partial g(x^l) \cap \partial h(x^l)$, $y^l \in \partial g(x^{l+1}) \cap \partial h(x^{l+1})$ et $[\rho(g) + \rho(h)] \left\|x^{l+1} - x^l\right\| = 0$. De plus si g et h sont strictement convexes sur X, alors $x^l = x^{l+1}$. Dans ce cas, DCA se termine à la l-ième itération (convergence finie de DCA).*
   - *$h^*(y^{l+1}) - g^*(y^{l+1}) = h^*(y^l) - g^*(y^l)$, si et seulement si $x^{l+1} \in \partial h^*(y^l) \cap \partial g^*(y^l)$, $x^{l+1} \in \partial h^*(y^{l+1}) \cap \partial g^*(y^{l+1})$ et $[\rho(h^*) + \rho(g^*)] \left\|y^{l+1} - y^l\right\| = 0$. De plus si h^* et g^* sont strictement convexes sur Y, alors $y^l = y^{l+1}$. Dans ce cas, DCA se termine à la l-ième itération (convergence finie de DCA).*
2. Si $[\rho(g) + \rho(h)] > 0$ (resp. $[\rho(h^*) + \rho(g^*)] > 0$), alors la série $\left\{\left\|x^{l+1} - x^l\right\|^2\right\}$ (resp. $\left\{\left\|y^{l+1} - y^l\right\|^2\right\}$) converge.
3. Si la valeur optimale α du problème (P_{dc}) est finie, alors
   - *les suites $\{g(x^l) - h(x^l)\}$ et $\{h^*(y^l) - g^*(y^l)\}$ convergent,*
   - *si de plus les suites $\{x^l\}$ et $\{y^l\}$ sont bornées, alors toute valeur d'adhérence $\tilde{x}$(resp. $\tilde{y}$) de la suite $\{x^l\}$ (resp. $\{y^l\}$) est un point critique de $g - h$ (resp. $h^* - g^*$).*
4. *DCA a une convergence linéaire pour un programme DC dans le cas général. De plus cette convergence est finie dans le cas polyédral.*

Le théorème précédent donne la convergence de la suite $\{x^l\}$ (resp. $\{y^l\}$) générée par DCA vers un point critique du programme P_{dc} (resp. D_{dc}). Récemment, de nouveaux résultats de raffinement de la qualité de convergence de toute la suite générée vers un point critique ont été établis dans le cas de fonctions subanalytiques [85]. Cependant, nous allons voir dans ce qui suit des conditions suffisantes pour que toute valeur d'adhérence de $\{x^l\}$ (resp. $\{y^l\}$) soit solution de P_{dc} (resp. D_{dc}).

Rappelons que h_l est la minorante affine de la fonction convexe h au voisinage de x^l et que

$$h_l(x) = h(x^l) + \langle x - x^l, y^l\rangle = \langle x, y^l\rangle - h^*(y^l), \forall x \in X.$$

On définit la fonction h^l, par

$$h^l(x) = \sup\{h_i(x) :\ i = 0, ..., l\} = \sup\left\{\langle x, y^i\rangle - h^*(y^i) :\ i = 0, ..., l\right\},\ \forall x \in X.$$

De manière analogue, on définit la fonction $(g^*)^l$ à partir de la fonction duale g_l^*

$$g_l^*(y) = g^*(y^{l-1}) + \langle y - y^{l-1}, x^l\rangle = \langle y, x^l\rangle - g(x^l), \forall y \in Y,$$

$$(g^*)^l(y) = \sup\{g_i^*(y) :\ i = 0, ..., l\} = \sup\left\{\langle y, x^i\rangle - g(y^i) :\ i = 0, ..., l\right\},\ \forall y \in Y.$$

Sachant que toute fonction convexe propre s.c.i θ est caractérisée comme le suprémum de ses minorantes affines, il s'avère donc plus judicieux d'utiliser h^l (resp. $(g^*)^l$) comme estimation minorante convexe de h (resp. g^*) sur X (resp. Y), plutôt que la minorante affine h_l (resp. g_l^*).

Considérons maintenant les deux programmes non convexes

$$(P^l) \quad \inf\left\{g(x) - h^l(x) : \ x \in X\right\},$$

$$(D^l) \quad \inf\left\{h^*(y) - (g^*)^l(y) : \ y \in Y\right\}.$$

Noter la différence par rapport aux sous problèmes convexes (P_l) et (D_l).

Théorème 3.5.1.4

1. $g(x^{l+1}) - h(x^{l+1}) = h^*(y^l) - g^*(y^l)$ *si et seulement si* $h_l(x^{l+1}) = h(x^{l+1})$.
2. $h^*(y^l) - g^*(y^l) = g(x^l) - h(x^l)$ *si et seulement si* $g^*(y^l) = g_l^*(y^l)$.
3. $g(x^{l+1}) - h(x^{l+1}) = g(x^l) - h(x^l)$ *si et seulement si* $h_l(x^{l+1}) = h(x^{l+1})$ et $g^*(y^l) = g_l^*(y^l)$.
   Parconséquent, si $g(x^{l+1}) - h(x^{l+1}) = g(x^l) - h(x^l)$*, alors les deux propositions suivantes sont vraies :*
   - x^{l+1} *(resp.* y^l*) est une solution optimale du problème* (P^l) *(resp.* (D^l)*).*
   - *Si* h *et* h^l *coincident en une solution de* (P_{dc}) *ou* g^* *et* $(g^*)^l$ *coincident en une solution de* (D_{dc})*, alors* x^{l+1} *(resp.* y^l*) est également une solution de* (P_{dc}) *(resp.* (D_{dc})*)).*

Supposons que la valeur optimale de (P_{dc}) est finie et que la suite $\left\{x^l\right\}$ générée par DCA est bornée, alors pour toute valeur d'adhérence x^∞, on a

$$g(x^\infty) - h(x^\infty) = \inf\left\{g(x^{i+1}) - h_i(x^{i+1}) : \ i = 0, ..., \infty\right\},$$

où h_∞ est la minorante affine de h :

$$h_\infty(x) = h(x^\infty) + \langle x - x^\infty, y^\infty\rangle = \langle x, y^\infty\rangle - h^*(y^\infty), \forall x \in X,$$

avec $y^\infty \in \partial h(x^\infty)$ un point limite de $\left\{y^l\right\}$.
Par conséquent, le point x^∞ est une solution du programme DC

$$(P^\infty) \quad \inf\left\{g(x) - h^\infty(x) : \ x \in X\right\},$$

où la fonction convexe h^∞ est définie par

$$h^\infty(x) := \sup\left\{h_\infty(x) : \ i = 0, ..., \infty\right\} = \sup\left\{\langle x, y^i\rangle - h^*(y^i) : \ i = 0, ..., \infty\right\}, \ \forall x \in X.$$

De manière analogue, pour le programme dual (D_{dc}), la minorante affine de g^* est donnée par

$$g_\infty^*(y) = g^*(y^\infty) + \langle y - y^\infty, x^\infty\rangle = \langle y, x^\infty\rangle - g(x^\infty), \forall y \in Y,$$

et $(g^*)^\infty$ est définie par

$$(g^*)^\infty(y) = \sup\{g_i^*(y) :\ i = 1, ..., \infty\} = \sup\left\{\langle y, x^i\rangle - g(y^i) :\ i = 1, ..., \infty\right\},\ \forall y \in Y.$$

Ainsi, y^∞ est solution du programme DC

$$(D^\infty)\quad \inf\{h^*(y) - (g^*)^\infty(y) :\ y \in Y\}.$$

Théorème 3.5.1.5 *Si la valeur optimale du problème (P_{dc}) (resp. (D_{dc})) est finie et la suite $\{x^l\}$ (resp. $\{y^l\}$) bornée, alors toute valeur d'adhérence x^∞ (resp. y^∞) est une solution du problème (P^∞) (resp. (D^∞)). De plus, les valeurs optimales sont égales, c'est à dire,*

$$g(x^\infty) - h^\infty(x^\infty) = h^*(y^\infty) - (g^*)^\infty(y^\infty).$$

Si l'une des conditions suivantes est vérifiée :
- *les fonctions h et h^∞ coincident en une solution optimale de (P_{dc}),*
- *les fonctions g^* et $(g^*)^\infty$ coincident en une solution optimale de (D_{dc}),*

alors x^∞ et y^∞ sont également des solutions optimales de (P_{dc}) et (D_{dc}) respectivement.

3.5.2 Illustration géométrique de DCA

Tout d'abord il faut noter que DCA ne travaille qu'avec les composantes DC g et h. A la l-ième itération, DCA remplace la composante h par sa minorante affine

$$h_l(x) = h(x^l) - \langle x - x^l, y^l\rangle$$

au voisinage de x^l. Comme h est une fonction convexe, on a alors

$$h(x) \geq h_l(x), \forall x \in X.$$

Parsuite,

$$f^l(x) := g(x) - \left[h(x^l) - \langle x - x^l, y^l\rangle\right] \geq g(x) - h(x), \forall x \in X.$$

Donc f^l est une fonction majorante de la fonction f et de plus lorsque g et h sont différentiables sur X, on vérifie facilement que

$$f^l(x^l) = f(x^l)\ \text{ et }\ \nabla f^l(x^l) = \nabla f(x^l).$$

On considère la figure 3.1 (Niu [114]) ci-dessous sur laquelle on cherche à minimiser une fonction DC f (courbe en noir) sur le domaine C (en rouge au bas de la figure). A la k-ième itération, la courbe de f^k (en bleu) peut être vue comme un bol se plaçant au-dessus de la courbe de f qu'elle touche au point $(x^k, f(x^k))$.

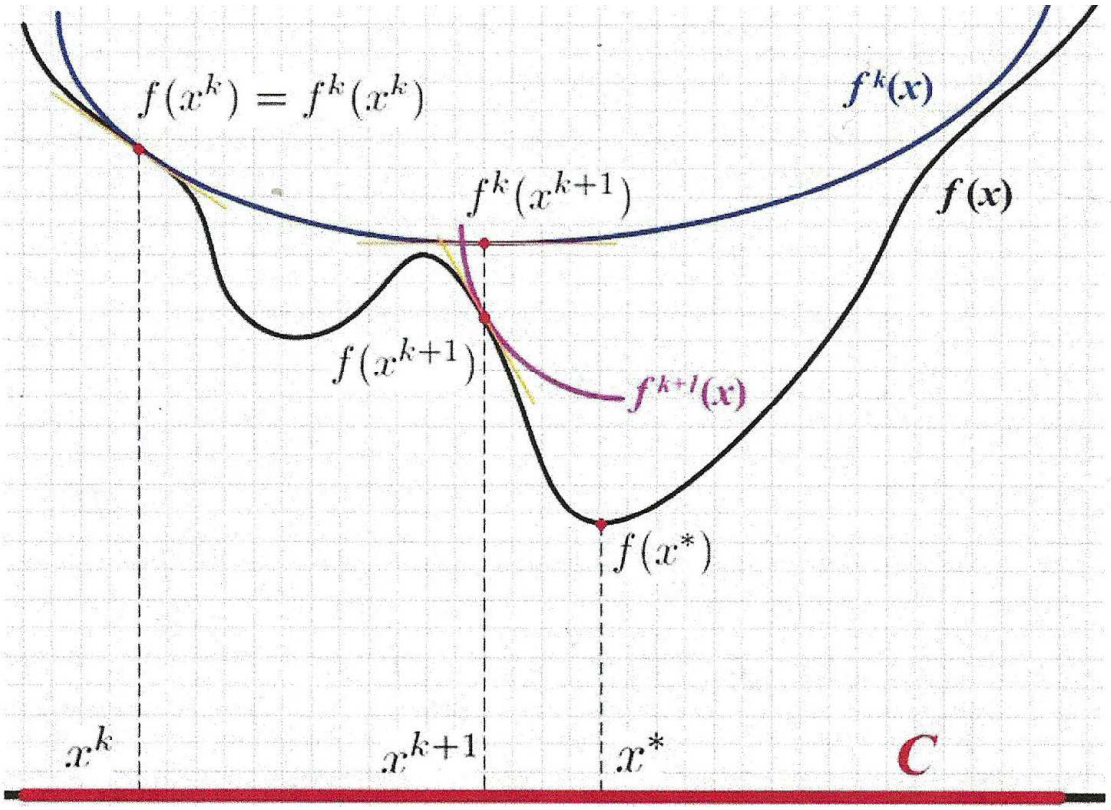

Figure 3.1 – Interprétation géométrique de DCA

On peut voir sur la figure que $f^k(x) \geq f(x), \forall x \in C$ et $f^k(x^k) = f(x^k)$. DCA construit d'abord la fonction f^k à l'aide de x^k, et ensuite minimise la fonction f^k sur C afin d'obtenir l'itéré suivant x^{k+1} et ainsi de suite pour obtenir x^{k+2}... Une des propriétés importantes de DCA qu'on peut observer ici est la décroissance $f(x^k) \geq f(x^{k+1})$. En réitérant le processus de construction de la suite x^k, x^{k+1}, x^{k+2},..., on observe la décroissance de la suite $f(x^k)$, $f(x^{k+1})$, $f(x^{k+2})$,... vers la valeur $f(x^*)$ et de plus la suite de point x^k, x^{k+1}, x^{k+2},... s'accumule sur le point x^*, ce qui illustre bien la propriété de convergence de DCA.

Bien que dans cette illustration DCA génère une suite de points convergent vers la solution optimale globale x^*, le théorème de convergence de DCA n'assure pas la convergence vers une solution globale mais vers un point vérifiant les conditions KKT généralisées (point critique). Cela va dépendre évidemment du point initial et de la décomposition DC utilisée. Par exemple, il est facile de voir que si l'on avait commencé DCA avec un point initial très proche du minimum local entre x^k et x^{k+1} (figure), la suite ne convergerait pas vers la solution x^* mais vers ce minimum. On dira qu'ici on a pu choisir une bonne décomposition et un bon point initial qui ont permis à DCA de sauter le mauvais minimum pour converger vers x^* (cette faculté de sauter est particulière à l'algorithme DCA contrairement à de nombreux algorithmes itératifs en optimisation qui s'engouffrent directement vers le mauvais minimum local en suivant la pente). Ce qui pose deux questions essentielles pour appliquer DCA :

1. Comment choisir le point initial ?
2. Comment choisir la décomposition DC ?

Actuellement, ces questions restent ouvertes. Cela va dépendre évidement de la structure du problème étudié. En pratique, on préfère souvent une décomposition dont les suites $\{x^k\}$ et $\{y^k\}$ générées par DCA sont plus facile à calculer (pas trop coûteux en temps de calcul par exemple), en d'autres termes,

- la fonction h est choisie de sorte que ∂h soit facilement calculable,
- le sous programme convexe $\min\{g(x) - \langle x, y^k\rangle\}$ soit facile à résoudre, cela est notamment très important pour pouvoir résoudre des problèmes de grande dimension,

et un point initial aussi proche que possible d'une solution optimale globale.

Il est important de noter la richesse de la programmation DC et DCA, en particulier sur le plan algorithmique où il a été démontré [125] qu'avec des décompositions DC convenables, on peut retrouver la plupart des algorithmes standards en programmation convexe et non convexe. En particulier, les algorithmes CCCP (Concave-Convex Procedure) [164] et SLA (Successive Linearization Algorithm) [14] sont des cas spéciaux de DCA.

On ne pourrait terminer ce chapitre sur la programmation DC sans parler des techniques de pénalisation plus particulièrement la pénalisation exacte qui est un outil très puissant en programmation DC.

3.6 Techniques de pénalité

Très souvent en optimisation, on est confronté à des problèmes avec contraintes dont une ou plusieurs contraintes sont complexes et difficiles à prendre en compte directement. La pénalisation est un concept simple qui permet de transformer un problème d'optimisation avec contraintes en un problème ou une suite de problèmes d'optimisation sans contraintes. Elle permet aussi de transformer un problème dont certaines contraintes sont difficiles à prendre en compte en un autre sans ces dernières.

Du point de vue théorique, l'approche par pénalisation est parfois utilisée pour étudier un problème d'optimisation dont certaines contraintes sont difficiles à prendre en compte, alors que le problème pénalisé a des propriétés mieux comprises ou plus simples à mettre en évidence. Si la pénalisation est bien choisie, si on a de la chance les propriétés recherchées du problème original peuvent être obtenues directement des propriétés du problème pénalisé, dans ce cas on parle de *pénalisation exacte*, sinon des passages à la limite parfois délicats permettent d'obtenir des propriétés du problème original, on parle de *pénalisation inexacte.*

Du point de vue numérique, cette transformation permet d'utiliser des algorithmes d'optimisation, très efficaces mieux adaptés à la structure du problème pénalisé, pour obtenir la solution de problèmes dont l'ensemble admissible peut avoir une structure complexe. Ce n'est cependant pas une technique passe partout, car elle a ses propres inconvénients : difficulté ou impossibilité de trouver une pénalisation exacte, non exactitude, nécessité de minimiser une suite de fonctions,...

Tout l'art de la pénalisation réside dans la recherche d'une fonction de pénalisation permettant d'obtenir une structure appropriée pour la résolution et possédant des propriétés d'exactitude ou de passage à la limite pas très difficiles. Cette recherche nécessite des outils mathématiques théoriques et numériques très approfondis : analyse convexe, analyse numérique, calcul différentiel, calcul sous-differentiel, algébre linéaire, géométrie,...

Les différentes techniques de pénalisation relèvent souvent du principe suivant. On remplace le problème

$$(P)\ \begin{cases} \min & f(x) \\ s.t. & x \in C, \\ & p(x) = 0, \end{cases}$$

où C est un sous ensemble fermé d'un espace vectoriel X et $p : X \to \mathbb{R}$ une fonction continue et positive ou nulle sur C avec $\{x \in C :\ p(x) = 0\} \neq \varnothing$, par un ou des problème(s) de la forme

$$(P_t)\ \begin{cases} \min & f_t(x) := f(x) + tp(x) \\ s.t. & x \in C. \end{cases}$$

Le but de ce terme additionnel est de pénaliser la violation de la contrainte $p(x) = 0$. Le scalaire $t > 0$ est appelé *paramètre de pénalisation*. Les propriétés de f_t vont dépendre de la grandeur de t. On peut alors résoudre (P_t) par une méthode d'optimisation adaptée.

Les questions qui se posent immédiatement sont

1. *en résolvant (P_t) résout-on (P) ? dans quelle condition sur C, f, p et t ?*
2. *le passage à la limite permet-elle d'obtenir une solution de (P) ?*

La réponse va dépendre évidemment de C, f, p et t.

3.6.1 Exemple

Voici un exemple de problème simple ayant une solution optimale au point $x = 0$

$$\begin{cases} \min & -x + 3x^3 \\ s.t. & x \in \{0, 1\}. \end{cases} \tag{3.2}$$

Ce problème peut être écrit sous les deux formes suivantes

$$\begin{cases} \min & -x + 3x^3 \\ s.t. & 0 \leq x \leq 1, \\ & q(x) := x(1 - x) = 0, \end{cases} \tag{3.3}$$

et

$$\begin{cases} \min & -x + 3x^3 \\ s.t. & 0 \leq x \leq 1, \\ & p(x) := x^2(1 - x) = 0. \end{cases} \tag{3.4}$$

Nous allons voir dans ce qui suit que la technique de pénalité appliquée à la première forme est exacte tandis que la deuxième ne l'est pas.

Considérons (3.3) auquel on associe le problème pénalisé

$$\begin{cases} \min & \Theta_t(x) := -x + 3x^3 + tq(x) \\ s.t. & 0 \leq x \leq 1, \end{cases} \tag{3.5}$$

Dans ce cas la pénalisation est exacte au moins pour $t \geq \frac{1}{1-\frac{1}{\sqrt{3}}}$ dans le sens où les deux ensembles de solutions sont identiques et se réduisent à $\{0\}$. En effet, pour $t \geq \frac{1}{1-\frac{1}{\sqrt{3}}}$ et $0 \leq x \leq 1$, on a

$$\Theta_t(0) = 0,$$

$$\Theta_t(x) = -x + 3x^3 + tx(1-x) \geq x(-1 + 3x^2 + (1 - \frac{1}{\sqrt{3}})t) > 0 \text{ si } 0 < x \leq \frac{1}{\sqrt{3}},$$

$$\Theta_t(x) = -x + 3x^3 + tx(1-x) \geq x(-1 + 3x^2) > 0 \text{ si } x > \frac{1}{\sqrt{3}}$$

et donc

$$\Theta_t(x) > \Theta_t(0) = 0, \ \ \forall 0 < x \leq 1,$$

c'est à dire 0 est l'unique solution du problème pénalisé (3.5).

Maintenant considérons (3.4) auquel on associe le problème pénalisé

$$\begin{cases} \min & \Theta_t(x) := -x + 3x^3 + tp(x) \\ s.t. & 0 \leq x \leq 1. \end{cases} \tag{3.6}$$

On a $\Theta_t(x) = -x + 3x^3 + tx^2(1-x) = x(-1 + 3x^2 + tx(1-x))$. Pour tout $t > 0$, on a

$$\lim_{x \mapsto 0^+} -1 + 3x^2 + tx(1-x) = -1,$$

donc il existe $x_t \in]0, 1[$ tel que $-1 + 3x_t^2 + tx_t(1-x_t) < 0$. Ainsi,

$$\Theta_t(x_t) = x_t(-1 + 3x_t^2 + tx_t(1-x_t)) < \Theta_t(0).$$

Ce qui montre que 0 n'est jamais solution de ce problème pénalisé et donc la pénalisation n'est pas exacte au sens que les ensembles de solutions ne sont pas identiques. Cependant, il faut noter que même si les ensembles de solutions ne sont pas identiques, les propriétés suivantes montrent que tout point d'adhérence de la suite de solutions des problèmes pénalisés est une solution du problème non pénalisé.

3.6.2 Quelques propriétés

Théorème 3.6.2.1 *On note $\bar{x}_t$ une solution de (P_t). Alors, lorsque $t > 0$ croît, $p(\bar{x}_t)$ décroît, $f(\bar{x}_t)$ et $f_t(\bar{x}_t)$ croissent.*

Théorème 3.6.2.2 *Supposons que f soit continue et coercive. Soit C un fermé, non vide. On suppose que $p : \mathbb{R}^n \to \mathbb{R}$ est continue et positive sur C. Alors, on a*

1. $\forall t > 0$, (P_t) *a au moins une solution* $\bar{x}_t$,
2. *la suite* $\{\bar{x}_t\}_{t>0}$ *est bornée,*
3. *tout point d'adhérence de la suite* $\{\bar{x}_t\}_{t\uparrow\infty}$ *est solution de* (P).

Théorème 3.6.2.3 (Pénalité exacte en programmation DC) *Soit* $\mathcal{P}$ *et* $\mathcal{P}_t$ *l'ensemble des solutions des problèmes* (P) *et* (P_t) *respectivement.*
On suppose C *un polyèdre convexe non vide et borné ;* f *et* p *deux fonctions concaves finies sur* C, p *positive ou nulle sur* C. *Alors il existe* $t_0 \geq 0$ *tel que pour tout* $t > t_0$, *les deux problèmes sont équivalents au sens que* $\mathcal{P}_t = \mathcal{P}$ *et* $\alpha(t) = \alpha$. *Le paramètre* t_0 *est 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* $t_0 = \max\left\{\frac{\alpha - f(x)}{p(x)} : \ x \in V(C), p(x) > 0\right\}$. *De plus*
  - $\alpha(t) = \alpha$ *si et seulement si* $t \geq t_0$
  - $\mathcal{P}_t \cap \{x \in C : \ p(x) \leq 0\} \neq \varnothing \Leftrightarrow \mathcal{P}_t \subset \mathcal{P} \Leftrightarrow t \geq t_0$
  - $\mathcal{P}_t = \mathcal{P}$ *si* $t > t_0$

Les théorèmes précédents sont très utiles pour transformer des problèmes ayant des contraintes non convexes en des problèmes avec des contraintes convexes. Ces résultats ont été très fructueux en optimisation et ont permis de transformer des problèmes combinatoires en des problèmes continus avec des contraintes convexes. Permettant ainsi d'utiliser de puissants outils de l'optimisation continue notamment la programmation DC qui est devenue incontournable au fil des années pour la résolution de nombreux problèmes d'optimisation non convexes.

De nouveaux résultats intéressants sur la pénalité exacte en programmation DC ont été établis dans [100]. Nous verrons dans les chapitres suivants d'autres nouveaux résultats de pénalité exacte, obtenus durant cette thèse, qui nous ont permis de tranformer des problèmes de parcimonie en des problèmes DC, voire quadratiques. Ces résultats nous ont permis ainsi d'établir plusieurs propriétés théoriques et numériques très importantes pour la compréhension et la résolution de ces problèmes de parcimonie.

Chapitre 4

Technique de Séparation et Evaluation pour l'optimisation globale

L'objet de ce chapitre est de présenter la technique de Séparation et Evaluation pour l'optimisation globale dans le cadre de l'optimisation continue non convexe. Après un bref rappel du principe, nous détaillerons les différentes étapes de cette technique et ensuite nous présenterons différents prototypes et les conditions de convergence. Enfin, nous aborderons la technique de relaxation DC pour le calcul de minorantes.

4.1 Introduction

Considérons le problème de minimisation d'une fonction f continue sur un compact $C \subset \mathbb{R}^n$

$$(P) \quad \begin{cases} \min_x & f(x) \\ s.t. & x \in C. \end{cases}$$

Il est bien connu en analyse mathématique que si $C \neq \varnothing$, le minimum et le maximum de f sont atteints en des points de C. Ainsi le problème (P) admet toujours au moins une solution. L'objectif est de trouver un point $x^* \in C$ tel que

$$f(x^*) \leq f(x),\ \forall x \in C. \tag{4.1}$$

Un tel point x^*, est appelé solution optimale globale ou minimum global. Il faut noter qu'il peut arriver qu'il existe plusieurs points de C vérifiant (4.1) et ici il s'agit d'en touver un et non pas tous.

Noter bien la différence avec la recherche de solution optimale locale qui consiste à déterminer un point $x^{loc} \in C$ tel qu'il existe un voisinage $V(x^{loc})$ de x^{loc} vérifiant

$$f(x^{loc}) \leq f(x),\ \forall x \in C \cap V(x^{loc}). \tag{4.2}$$

Toute solution optimale globale est aussi optimale locale. Cependant la réciproque est fausse dans le cas général.

Lorsque la fonction f et l'ensemble C sont convexes, on dit dans ce cas que le problème est convexe, les solutions optimales globales et optimales locales sont confondues et des conditions nécessaires et suffisantes d'optimalité de type KKT (Karush-Kuhn-Tucker) permettent la mise en oeuvre de méthodes algorithmiques très efficaces pour la résolution la plupart du temps.

Lorsque la fonction f est non convexe ou que l'ensemble C est non convexe, on dit que le problème est non convexe et dans ce cas l'absence de conditions d'optimalité utilisables, voire même l'absence de conditions d'optimalité (sauf dans certains cas par exemple la minimisation d'une fonction quadratique sur une boule euclidienne Pham Dinh et Le Thi [124] et la minimisation d'une fonction polynôme sous contraintes polynomiales Lasserre [78]), rend le problème difficile à résoudre. Au point de vue de la complexité, le problème est NP-difficile. Afin de contourner cette absence de conditions d'optimimalité globales, des méthodes génériques d'énumération implicite et intelligente ont été élaborées ces dernières décennies :

- Séparation et Evaluation (Branch and Bound)
- Approximation Extérieure (Outer Approximation)
- Annexion Polyédrale (Polyhedral Annexation).

Ici nous nous intéressons à la méthode Séparation et Evaluation (SE) très utile en programmation DC. Plusieurs algorithmes très efficaces combinant DCA, SE et la relaxation DC (que l'on présentera en fin de chapitre) ont été présentés dans [80, 94, 82, 128] par exemple.

4.2 La Méthode de Séparation et Evaluation

La méthode de Séparation et Evaluation est une méthode d'énumération implicite intelligente pour résoudre de manière globale des problèmes d'optimisation non convexes continus ou combinatoires

$$(P) \quad \begin{cases} \min_x & f(x) \\ s.t. & x \in C. \end{cases}$$

Elle consiste en une division succesive d'un ensemble contenant C en sous-ensembles de plus en plus petits. A chaque sous-ensemble contenant une partie de C, on associe une borne inférieure de la valeur de la fonction objectif sur cet ensemble afin d'éliminer les parties non prometteuses et de sélectionner un ensemble que l'on devrait diviser par la suite. Durant le processus nous distinguons deux phases cruciales et très importantes : la phase de séparation et la phase d'évaluation.

4.2.1 Phase de séparation

Elle consiste à diviser successivement l'ensemble C en sous-ensembles qui forment une partition de l'ensemble C. A chaque sous-ensemble est associé un sous-problème d'optimisation. Ainsi, en résolvant les sous-problèmes et en prenant la meilleure solution trouvée, on est assuré d'avoir résolu le problème initial.

4.2.2 Phase d'évaluation

Elle a pour but de déterminer l'optimum des sous-problèmes de la phase de séparation. Très souvent en pratique, les sous-problèmes sont difficiles à résoudre et donc on se contente de déterminer une bonne borne inférieure pour chacun des sous-problèmes par des techniques de relaxation (relaxation continue, relaxation lagrangienne, relaxation SDP, relaxation DC, etc). Si pour un sous-problème donné la borne inférieure obtenue est supérieure au coût de la meilleure solution réalisable (appelé borne supérieure) trouvée jusqu'à présent, alors on a l'assurance que le sous-ensemble ne contient pas l'optimum et donc on le supprime.

Nous allons présenter dans la suite deux prototypes pour la mise en œuvre d'une procédure de Séparation et Evaluation (SE).

4.2.3 Un premier prototype

Définition 4.2.3.1 *Soit M un compact dans $\mathbb{R}^n$ et soit I un ensemble fini d'indices. Un ensemble $\{M_i,\ i \in I\}$ de sous-ensembles compacts est dit une partition de M si*

$$M = \cup_{i\in I} M_i,\ M_i \cap M_j = \partial M_i \cap \partial M_j,\ i, j \in I,\ i \neq j$$

où ∂M_i représente la frontière relative à M de M_i.

Dans la suite du chapitre, on notera par $\min f(S)$ la valeur minimum de f sur l'ensemble S, c'est à dire,

$$\min f(S) := \min \{f(x) :\ x \in S\}.$$

Nous allons d'abord présenter un premier prototype (Horst [65]) résumant le schéma général de SE et ensuite nous donnerons des conditions garantissant la convergence du prototype.

Prototype 1
Initialisation

1. Choisir un compact $M_0 \supset C$, un ensemble fini d'indices I_0, une partition $\mathcal{M}_0 = \{M_{0,i} :\ i \in I_0\}$ de M_0 satisfaisant $M_{0,i} \cap C \neq \varnothing$, $i \in I_0$.
2. Pour chaque $i \in I_0$, déterminer

   $$C_{0,i} \subset M_{0,i} \cap C,\ C_{0,i} \neq \varnothing$$

   et

   $$\alpha_{0,i} = \alpha(M_{0,i}) := \min f(C_{0,i}),\ x^{0,i} \in \arg\min f(C_{0,i}).$$

3. Pour chaque $i \in I_0$, déterminer

   $$\beta_{0,i} = \beta(M_{0,i}) \leq \min f(C \cap M_{0,i}).$$

4. Calculer

$$\alpha_0 = \min_{i \in I_0} \alpha_{0,i} \tag{4.3}$$

$$x^0 \in \arg\min \left\{ f(x^{0,i}),\ i \in I_0 \right\} \tag{4.4}$$

$$\beta_0 = \min_{i \in I_0} \beta_{0,i} \tag{4.5}$$

Itération k=0,1,...

Etape 1 Supprimer tout $M_{k,i} \in \mathcal{M}_k$ vérifiant

$$\beta_{k,i} \geq \alpha_k$$

ou pour lequel on sait que $\min f(C)$ ne peut avoir lieu dans $M_{k,i}$. Soit $\mathcal{R}_k$ la collection des éléments restants $M_{k,i} \in \mathcal{M}_k$. Si $\mathcal{R}_k = \varnothing$, alors s'arreter : x^k est une solution.

Etape 2 Sélectionner $M_{k,i_k} \in \mathcal{R}_k$; choisir un ensemble fini d'indices J_{k+1} et construire une partition

$$M_{k,i_k} = \{M_{k+1,i} :\ i \in J_{k+1}\}$$

de M_{k,i_k} telle que $M_{k+1,i} \cap C \neq \varnothing$.

Etape 3 Pour chaque $i \in J_{k+1}$ déterminer

$$C_{k+1,i} \subset M_{k+1,i} \cap C,\ C_{k+1,i} \neq \varnothing$$

et

$$\alpha_{k+1,i} = \alpha(M_{k+1,i}) := \min f(C_{k+1,i}),\ x^{k+1,i} \in \arg\min f(C_{k+1,i}).$$

Etape 4 Pour chaque $i \in J_{k+1}$ déterminer $\beta_{k+1,i}$ tel que

$$\beta_{k,i_k} \leq \beta_{k+1,i} \leq \min f(S \cap M_{k+1,i}).$$

Etape 5 Poser

$$\mathcal{M}_{k+1} = (\mathcal{R}_k \setminus M_{k,i_k}) \cup \mathcal{M}_{k,i_k}.$$

Soit I_{k+1} l'ensemble d'indices tels que

$$\mathcal{M}_{k+1} = \{M_{k+1,i} :\ i \in I_{k+1}\}$$

est la partition actuelle.
Soient $\alpha_{k+1,i}, \beta_{k+1,i}, x^{k+1,i}$ les quantités correspondant à $M_{k+1,i}, i \in I_{k+1}$.

Etape 6 Calculer

$$\alpha_{k+1} = \min_{i \in I_{k+1}} \alpha_{k+1,i} \tag{4.6}$$

$$x^{k+1} \in \arg\min \left\{ f(x^{k+1,i}), i \in I_{k+1} \right\} \tag{4.7}$$

$$\beta_{k+1} = \min_{i \in I_{k+1}} \beta_{k+1,i} \tag{4.8}$$

poser k=k+1 et retourner à l'Etape 1.

Remarque 4.2.3.2

1. Les éléments $C_{k,i}, x^{k,i}$ et $\beta_{k,i}$, dans le prototype, doivent être choisis de sorte que les bornes obtenues soient les plus serrées possible, avec un effort de calcul raisonnable.
2. Pour chaque sous-ensemble $M_{k,i}$, $\alpha_{k,i}$ et $\beta_{k,i}$ représentent une borne supérieure et une borne inférieure de la valeur minimum de f sur $C \cap M_{k,i}$ ($\min f(C \cap M_{k,i})$) respectivement. La suite $\{\alpha_k\}$ représente une suite décroissante de bornes supérieures de la valeur minimum de f sur C ($\min f(C)$) et la suite $\{\beta_k\}$ forme une suite croissante de bornes inférieures de cette même valeur $\min f(C)$.
3. Si pour un sous-ensemble $M_{k,i}$ la borne inférieure $\beta_{k,i}$ obtenue est supérieure ou égale à la meilleure borne supérieure obtenue jusqu'a maintenant α_k ($\beta_{k,i} \geq \alpha_k$), alors la solution actuelle x^k ne peut pas être améliorée dans $M_{k,i}$ et donc cet élément est supprimé.

Conditions de la convergence
Par construction des suites $\left\{x^k\right\}$, $\left\{\beta^k\right\}$ et $\left\{\alpha^k\right\}$, on a

$$x^k \in C,\ k = 0, 1, ... \tag{4.9}$$

$$\alpha_k \geq \alpha_{k+1} \geq \min f(C) \geq \beta_{k+1} \geq \beta_k \tag{4.10}$$

$$f(x^k) \geq f(x^{k+1}),\ k = 0, 1, ... \tag{4.11}$$

On dira que la méthode SE converge lorsque tout point d'accumulation de la suite $\left\{x^k\right\}$ est une solution de (P).

Définition 4.2.3.3 *Une estimation de borne est dite cohérente si, pour une suite décroissante quelconque $M_{k_q,i_{k_q}}$ générée par la procédure de séparation, c'est à dire,*

$$M_{k_{q+1},i_{q+1}} \subset M_{k_q,i_q},$$

on a

$$\lim_{q \mapsto \infty} (\alpha_{k_q,i_{k_q}} - \beta_{k_q,i_{k_q}}) = 0. \tag{4.12}$$

Puisque $\beta_{k_q,i_{k_q}} \leq \alpha_{k_q} \leq \alpha_{k_q,i_{k_q}}$, la condition (4.12) peut s'écrire

$$\lim_{q \mapsto \infty} (\alpha_{k_q} - \beta_{k_q,i_{k_q}}) = 0,$$

$$\lim_{q \mapsto \infty} (\alpha_{k_q,i_{k_q}} - \alpha_{k_q}) = 0.$$

Par la monotonie et la bornitude des suites α_k et β_k, on a

$$(f(x^k) = \alpha_k) \to \alpha,\ \beta_k \to \beta,\ \alpha \geq \min f(C) \geq \beta.$$

Définition 4.2.3.4 *Une sélection est dite complète si pour chaque*

$$M \in \cup_{p=1}^{\infty} \cap_{k=p}^{\infty} \mathcal{R}_k$$

on a

$$\inf f(M \cap C) \geq \alpha.$$

Une sélection est dite borne-améliorante, si au moins après chaque nombre fini d'itérations, on a

$$M_{k,i_k} \in \arg\min \{\beta(M) : \ M \in \mathcal{R}_k\}.$$

Théorème 4.2.3.5 (Horst[68], Horst et Tuy[65]) *Supposons que l'opération d'estimation de borne est cohérente dans le Prototype 1. Alors*

1. *si la sélection est complète, alors*

$$\alpha = \lim_{k \mapsto \infty} \alpha_k = \lim_{k \mapsto \infty} f(x^k) = \min f(C).$$

2. *si la sélection est borne-améliorante, alors*

$$\beta := \lim_{k \mapsto \infty} \beta_k = \min f(C).$$

3. *si la sélection est complète, alors chaque point d'accumulation de* $\left\{x^k\right\}$ *résout le problème* (P).

Dans le Prototype 1, il est nécessaire que $M \cap C \neq \varnothing$ pour chaque élément M de la partition. Or, il s'avère que dans des cas concrets il n'y pas de régles standard pour savoir si $M \cap C$ est vide ou non. En effet, pour définir un ensemble M, souvent un polyèdre, il suffit de connaître l'ensemble des sommets de M. Mais très souvent, cela ne suffit pas pour savoir si $M \cap C$ est vide ou non, même pour des cas très simples. Cette difficulté limite l'applicabilité du Prototype 1. Dans ce qui suit, on va présenter un deuxième prototype qui ne nécessite pas de savoir $M \cap C \neq \varnothing$ pour tous les ensembles M, mais seulement une partie tout en assurant la convergence de l'algorithme.

4.2.4 Un deuxième prototype

Définition 4.2.4.1 *Un ensemble M est dit réalisable si $M \cap C \neq \varnothing$. Dans le cas contraire on dit qu'il est non-réalisable. Il est dit incertain si nous ne savons pas si il est réalisable ou non.*

Naturellement, un ensemble sera éliminé si on sait qu'il est non réalisable. Lorsque les ensembles incertains sont admis, on exige que

$$-\infty < \beta(M) \leq \min f(M \cap C), \quad \text{si M est réalisable}$$

$$-\infty < \beta(M) \leq \min f(M), \quad \text{si M est incertain.}$$

En général, $C_M \subset M \cap C$ peut être vide et il est possible que la borne $\alpha(M) = \infty$. La variante suivante du prototype ci-dessous sera appliquée lorsqu'on ne peut pas décider définitivement si M est réalisable ou non pour tous les ensembles de la partition donnée. Noter que dans ce cas, les bornes supérieures ne sont pas toujours disponibles. Nous allons présenter un prototype modifié avec la convention que le

minimum sur un ensemble vide prend la valeur infinie.

Prototype 2 (Horst[64])
Initialisation

1. Choisir un compact $M_0 \supset C$, un ensemble fini d'indices I_0, une partition $\mathcal{M}_0 = \{M_{0,i} : \ i \in I_0\}$ de M_0.
2. Pour chaque $i \in I_0$, déterminer

$$C_{0,i} \subset M_{0,i} \cap C, \ C_{0,i} \neq \varnothing$$

et

$$\alpha_{0,i} = \alpha(M_{0,i}) := \min f(C_{0,i}), \ x^{0,i} \in \arg\min f(C_{0,i}).$$

Si $C_{0,i}$ n'est pas disponible (par des efforts réalisables), on pose $C_{0,i} = \varnothing$.

3. Pour chaque $i \in I_0$, déterminer $\beta_{0,i} = \beta(M_{0,i})$ vérifiant

$$\beta(M_{0,i}) \leq \min f(C \cap M_{0,i}), \text{ si } M_{0,i} \text{ est réalisable}$$

$$\beta(M_{0,i}) \leq \min f(M_{0,i}), \text{ si } M_{0,i} \text{ est incertain.}$$

4. Calculer

$$\alpha_0 = \min_{i \in I_0} \alpha_{0,i} \tag{4.13}$$

$$x^0 \in \arg\min \left\{ f(x^{0,i}), \ i \in I_0 \right\} \tag{4.14}$$

$$\beta_0 = \min_{i \in I_0} \beta_{0,i} \tag{4.15}$$

Itération k=0,1,...

Etape 1 Supprimer tout $M_{k,i} \in \mathcal{M}_k$ vérifiant

$$\beta_{k,i} \geq \alpha_k$$

ou pour lequel on sait que $\min f(C)$ ne peut avoir lieu dans $M_{k,i}$. Soit $\mathcal{R}_k$ la collection des éléments restants $M_{k,i} \in \mathcal{M}_k$. Si $\mathcal{R}_k = \varnothing$, alors s'arreter : x^k est une solution.

Etape 2 Sélectionner $M_{k,i_k} \in \mathcal{R}_k$; choisir un ensemble fini d'indices J_{k+1} et construire une partition

$$M_{k,i_k} = \{M_{k+1,i} : \ i \in J_{k+1}\}$$

de M_{k,i_k}. Appliquer des règles pour éliminer les sous ensembles qu'on sait non réalisables.

Etape 3 Pour chaque $i \in J_{k+1}$ déterminer

$$C_{k+1,i} \subset M_{k+1,i} \cap C, \ C_{k+1,i} \neq \varnothing$$

et

$$\alpha_{k+1,i} = \alpha(M_{k+1,i}) := \min f(C_{k+1,i}), \ x^{k+1,i} \in \arg\min f(C_{k+1,i}).$$

Si $C_{k+1,i}$ n'est pas disponible, on pose $C_{k+1,i} = \varnothing$.

Etape 4 Pour chaque $i \in J_{k+1}$ déterminer $\beta_{k+1,i}$ tel que

$$\beta_{k,i_k} \leq \beta_{k+1,i} \leq \min f(C \cap M_{k+1,i}).$$

Etape 5 Poser

$$\mathcal{M}_{k+1} = (\mathcal{R}_k \setminus M_{k,i_k}) \cup \mathcal{M}_{k,i_k}.$$

Soit I_{k+1} l'ensemble d'indices tels que

$$\mathcal{M}_{k+1} = \{M_{k+1,i} : \ i \in I_{k+1}\}$$

est la partition actuelle.
Soient $\alpha_{k+1,i}, \beta_{k+1,i}, x^{k+1,i}$ les quantités correspondant à $M_{k+1,i}, i \in I_{k+1}$.

Etape 6 Calculer

$$\alpha_{k+1} = \min_{i \in I_{k+1}} \alpha_{k+1,i} \tag{4.16}$$

$$\beta_{k+1} = \min_{i \in I_{k+1}} \beta_{k+1,i} \tag{4.17}$$

Si $\alpha_{k+1} < \infty$, alors on prend $x^{k+1} \in C$ tel que $f(x^{k+1}) = \alpha_{k+1}$.
Poser k=k+1 et retourner à l'Etape 1.

Convergence

Définition 4.2.4.2 *Soit $\left\{M_{k_q}\right\}$ une suite d'ensembles générée par la procédure de division. Une procédure de division est dite exhaustive si la suite de diamètres $d(M_{k_q})$ associés à M_{k_q} vérifie*

$$\lim_{q \mapsto \infty} d(M_{k_q}) = 0.$$

Noter bien que pour une suite décroissante d'ensembles générés par une division exhaustive, on a

$$\lim_{q \mapsto \infty} M_{k_q} = \bigcap M_{k_q} = \{\bar{x}\}, \ \bar{x} \in \mathbb{R}^n.$$

Définition 4.2.4.3 *L'opération d'estimation de borne inférieure est dite fortement cohérente si pour n'importe quelle suite décroissante $\left\{M_{k_q}\right\}$ d'ensembles générée par une subdivision exhaustive telle que*

$$\lim_{q \mapsto \infty} M_{k_q} = \{\bar{x}\},$$

il existe une sous-suite $\left\{M_{k_q^*}\right\}$ de $\left\{M_{k_q}\right\}$ pour laquelle

$$\lim_{q \mapsto \infty} \beta(M_{k_q^*}) = f(\bar{x}).$$

Définition 4.2.4.4 *L'élimination-par-non réalisabilité est dite certaine à la limite si pour n'importe quelle suite décroissante $\left\{M_{k_q}\right\}$ d'ensembles générée par une division exhaustive telle que*

$$\lim_{q \mapsto \infty} M_{k_q} = \{\bar{x}\},$$

on a

$$\bar{x} \in C.$$

On désigne Y^α l'ensemble des points d'accumulation de la suite y^k de points correspondant à β_k. Soit $X^* = \arg\min f(C)$ l'ensemble de solutions optimales de (P).

Théorème 4.2.4.5 (Horst[64]) *Soit le Prototype 2 vérifiant les conditions suivantes :*

1. *la division est exhaustive ;*
2. *la sélection est borne améliorante ;*
3. *la borne inférieure est fortement cohérente ;*
4. *l'élimination est certaine à la limite.*

Alors, on a

$$\beta := \lim \beta_k = \min f(C)$$

et

$$Y^\alpha \subset X^*.$$

4.2.5 Réalisation

La réalisation d'un algorithme SE dépend du choix des opérations suivantes :
- Diviser M_{k,i_k}.
- Sélectionner M_{k,i_k}.
- Estimer les bornes inférieures $\beta_{k,i}$.

4.2.5.1 Stratégie de division

Elle consiste à chercher des éléments de partition M_k aussi simples que possible et très facile à manipuler durant la procédure d'estimation de bornes inférieures. Ces éléments doivent être choisis de sorte que la procédure de division soit exhaustive afin d'assurer la convergence de la méthode SE. On utilise généralement des polyèdres pour définir les éléments M_k plus particulièrement des *simplexes*, des *rectangles*, des *cones polyédraux*, des *prismes*,... qui sont plus simples à munipuler et à diviser.
Subdivision simpliciale : Avec cette subdivision, M_0 et tous les éléments M_k de subdivision sont des n-simplexes. Le premier simplexe M_0 contenant l'ensemble C est facile à construire. Etant donné un simplexe $M = conv\,\{v^0, ..., v^n\}$ dont v^0,...,v^n représentent l'ensemble des sommets, il est bien connu qu'un point quelconque $s \in M$ s'écrit

$$s = \sum_{i=0}^{n} \lambda_i v^i,\ \lambda_i \geq 0,\ \sum_{i=0}^{n} \lambda_i = 1.$$

Soit $s \neq v^i, \forall i = 0, ..., n$. Posons $J = \{j :\ \lambda_j > 0\}$. On peut construire un n-simplexe

$$M_j = \text{conv}\left\{v^0, ..., v^{j-1}, v^{j+1}, ..., v^n\right\}$$

en remplaçant un sommet $v^j, j \in J$ par s. On construit ainsi une subdivision de M. Très souvent, on choisit s comme le milieu de la plus longue arête de M et on divise

M en deux n-simplexes.
Subdivision rectangulaire : Avec cette subdivision, M_0 et tous les éléments M_k de subdivision sont des rectangles. On choisit le rectangle

$$M_0 = \prod_{i=1}^{n} [a_i, b_i]$$

le plus serré contenant C en résolvant $2n$ problèmes convexes

$$a_i = \min \{x_i : \ x \in C\}, \ b_i = \max \{x_i : \ x \in C\}, \ i = 1, ..., n.$$

Un rectangle M est divisé en deux rectangles ayant le même volume en utilisant le milieu de l'arête la plus longue de M.
Subdivision conique : Elle consiste à construire des cônes centrés en un point intérieur à C. Supposons que C possède un point intérieur w et soit M_0 un n-simplexe, $M_0 \supset C$. M_0 possède $n+1$ faces, notées $F_{0,i}$, $i = 1, ..., n+1$ de dimension $n-1$ qui sont des $(n-1)$-simplexes. Les cônes polyédraux $S_{0,i}$, centrés en w et engendrés par $F_{0,i}$, constituent une division de $\mathbb{R}^n$. Ainsi, une division de la face F en un ensemble de simplexes $\{F_j\}$ donne une division du cône S en un ensemble de cônes $\{S_j\}$.
Subdivision prismatique : Elle constitue une extension de la subdivision conique dans laquelle le centre w est considéré comme un point fictif à l'infini, formant ainsi des prismes dans $\mathbb{R}^n \times \mathbb{R}$. Chaque simplexe ou rectangle M définit un prisme $T = \{(x, t) : \ x \in M\}$.

4.2.5.2 Règle de sélection

Elle consiste à choisir un élément M_{k,i_k}, parmi les éléments de subdivision $M \in \mathcal{R}_k$, pour la division suivante. Ce choix s'effectue à l'aide de critères dont sans doute le plus simple et le plus naturel consiste à prendre l'élément de subdivision réalisant la plus petite borne inférieure, c'est à dire,

$$M_{k,i_k} \in \arg\min \{\beta(M) : \ M \in \mathcal{R}_k\}.$$

On peut aussi utiliser d'autres critères par exemple ceux proposés par Tuy et al.[148] :

1. choisir le plus vieil ensemble, c'est à dire,

$$M_{k,i_k} \in \arg\min \{\mathcal{G}(M) : \ M \in \mathcal{R}_k\},$$

où $\mathcal{G}(M)$ représente l'indice de l'itération où l'ensemble M a été créé.

2. choisir M suivant une mesure $\delta(M)$ donnée (exemple diamètre, volume,...),

$$M_{k,i_k} \in \arg\min \{\delta(M) : \ M \in \mathcal{R}_k\}.$$

4.2.5.3 Estimation de borne

Etant donné un ensemble M_k. L'objectif est d'estimer une borne inférieure $\beta(M_k)$ de $\min f(C \cap M_k)$ avec un coût de calcul raisonnable. Très souvent, le calcul de $\min f(C \cap M_k)$ est très difficile et donc on se contente de construire T_k, $C \cap M_k \subset T_k \subset M_k$ de sorte que la valeur (problème relaxé) $\min f(T_k)$ soit facilement calculable et ainsi prendre $\beta(M_k) = \min f(T_k)$.

L'estimation d'une borne supérieure, quand à elle, se fait à l'aide d'un point réalisable $x \in C$ qui une fois calculé nous donne une borne supérieure $f(x)$. Un tel point peut être trouvé à l'aide de méthodes d'optimisation locale comme DCA. Ces méthodes non seulement permettent d'obtenir un point x réalisable ($x \in C$) mais permettent aussi d'améliorer la borne supérieure courante par différentes techniques par exemple des techniques de réinitialisation, de perturbation, etc.

Lors de la réalisation d'un algorithme SE, l'estimation de borne présente toujours un dilemme entre la convergence et l'efficacité. L'algorihme convergera plus vite si on peut estimer les bornes d'une façon plus précise. Cette précision nécessite la résolution de sous-problèmes, qui sont généralement très couteux en temps de calcul par exemple, ce qui ralenti considérablement la convergence. L'utilisation de techniques de relaxation (construction de T_k par exemple) donne une certaine souplesse dans l'estimation de borne et permet dans certains cas la réalisation d'algorithme SE très performant en pratique.

4.3 Relaxation DC

Bien plus qu'un outil algorithmique, la relaxation est en fait une méthodologie générale, quasi incontournable dès qu'il s'agit de convexifier un problème d'optimisation. Etant donné un problème d'optimisation de la forme

$$(P) \quad \begin{cases} \min_x & f(x) \\ s.t. & x \in C, \end{cases}$$

les techniques de relaxation consistent le plus souvent à remplacer le problème (P) par un autre problème (P_{relax}) de la forme

$$(P_{\text{relax}}) \quad \begin{cases} \min_y & f_{\text{relax}}(y) \\ s.t. & y \in C_{\text{relax}}, \end{cases}$$

où f_{relax} et C_{relax} sont convexes, dont les propriétés sont mieux comprises et dont la résolution est plus simple. Ces techniques permettent d'obtenir des valeurs minorantes (bornes inférieures) de la valeur optimale du problème (P) et sont utilisées dans les procédures de séparation et évaluation, d'approximations extérieures,.... de l'optimisation globale.

On distingue plusieurs types de relaxation :
- relaxation lagrangienne,
- relaxation continue,
- relaxation semi-définie positive,

– relaxation DC,
– ...

Nous nous intéressons ici à la relaxation DC qui utilise la notion d'enveloppe convexe de fonctions non convexes sur un compact convexe et qui a été utilisée avec succés sur plusieurs problèmes non convexes (voir [128] et les références à l'intérieur).

Nous allons d'abord rappeler la notion d'enveloppe convexe de fonction non convexe et ensuite quelques résultats utilisés pour définir la relaxation DC.

4.3.1 Enveloppe convexe de fonctions non convexe

Définition 4.3.1.1 (Fonction minorante) *Etant donné une fonction $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ et un sous ensemble C non vide, on appelle fonction minorante de f sur C toute fonction ϕ vérifiant*

$$\phi(x) \leq f(x),\ \forall x \in C.$$

Etant donné un problème d'optimisation non convexe

$$\begin{cases} \inf_x & \psi(x) \\ s.t. & x \in \mathbb{R}^n, \end{cases} \tag{4.18}$$

où $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ est une fonction non convexe propre (dom $\psi \neq \varnothing$) possédant une minorante affine sur $\mathbb{R}^n$, l'*enveloppe convexe* de ψ, notée co ψ, est la plus grande fonction convexe minorante de f sur $\mathbb{R}^n$. Elle constitue la meilleure manière de formuler le problème en un problème convexe et est donnée par ([130, 63])

$$\text{co } \psi(x) := \inf \left\{ \sum_i \lambda_i \psi(x_i) : \quad \lambda_i \geq 0,\ \sum_i \lambda_i = 1, x_i \in \text{dom } \psi,\ x = \sum_i \lambda_i x_i \right\}, \tag{4.19}$$

où l'infimum est pris sur toutes les représentations de x comme combinaison finie et convexe d'éléments $x_i \in$ dom ψ. De plus

$$\text{dom co } \psi = \text{co dom } \psi, \tag{4.20}$$

$$\inf \{\text{co } \psi(x) : \quad x \in \mathbb{R}^n\} = \inf \{\overline{\text{co}}\ \psi(x) : \quad x \in \mathbb{R}^n\}. \tag{4.21}$$

Bien d'autres propriétés de l'enveloppe convexe ont été établies dans Hiriart-Urruty et Lemarechal[63]

1. $\arg\min \psi \ \subset\ \arg\min \text{co } \psi \ \subset\ \arg\min \overline{\text{co}}\ \psi$
2. $co(\arg\min \psi) \ \subset\ \bar{\text{co}}(\arg\min \ \psi) \ \subset\ \arg\min \overline{\text{co}}\ \psi$
3. $\overline{\text{co}}\ \psi = \psi^{**}$
4. si ψ est une fonction s.c.i. et 1-coercive ($\lim_{\|x\| \to +\infty} \frac{\psi(x)}{\|x\|}$), alors co $\psi = \overline{\text{co}}\ \psi = \psi^{**}$.

Remarque 4.3.1.2 Très souvent en analyse convexe, une fonction $\psi : C \subset \mathbb{R}^n \to \mathbb{R}$ est identifiée à son extension $\psi + \chi_C$ sur $\mathbb{R}^n$. Pour $C \subset \mathbb{R}^n$ et $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, avec $C \subset \text{dom } \psi$, on note par $\text{co}_C\psi$ l'enveloppe convexe de ψ sur C, c'est à dire, $\text{co}_C\psi := \text{co}(\psi + \chi_C)$. De même $\overline{\text{co}}_C\psi$ désigne $\overline{\text{co}}(\psi + \chi_C)$.

Le calcul explicite de l'enveloppe convexe d'une fonction non convexe donnée est très difficile en général. Cependant, le calcul est plus simple dans le cas d'une fonction concave sur un polyèdre convexe borné.

4.3.2 Enveloppe convexe de fonction concaves sur un polyèdre convexe borné

Soit K un polyèdre convexe borné non vide et soit $V(K) := \{v^1, ..., v^m\}$ l'ensemble de ses sommets. On a alors $K = \text{co } V(K)$. Les sommets $v^1, ..., v^m$ sont dits affinement indépendants ([130, 63]) s'il n'existe pas de $\lambda \in \mathbb{R}^m$ tel que

$$\lambda \neq 0, \ \sum_{i=1}^{m} \lambda_i = 0 \text{ et } \sum_{i=1}^{m} \lambda_i v^i = 0. \tag{4.22}$$

Dans ce cas, K est appelé un $(m-1)$-simplexe et tout point x de K est une combinaison convexe unique des sommets $v^1, ..., v^m$. Si ψ est finie concave sur K, alors l'expression de $\text{co}_K\psi$ devient plus simple à calculer ([69, 66, 137, 136, 128]).

Théorème 4.3.2.1 *Si ψ est une fonction concave et finie sur K, alors*

1. *$co_K\ \psi(x)$ est la fonction convexe polyédrale sur K donnée par*

$$\text{co}_K\ \psi(x) = \min \left\{ \sum_{i=1}^{m} \lambda_i \psi(v^i) : \ \ \lambda_i \geq 0, \ \sum_{i=1}^{m} \lambda_i = 1, \ x = \sum_{i=1}^{m} \lambda_i v^i \right\}. \tag{4.23}$$

   De plus, ψ et $co_K\ \psi(x)$ coincident sur les sommets $V(K)$.

2. *Si K est un $(m-1)$-simplexe, alors $co_K\ \psi(x)$ est la fonction affine donnée par*

$$\text{co}_K\ \psi(x) = \sum_{i=1}^{m} \lambda_i \psi(v^i) : \ \ \lambda_i \geq 0, \ \sum_{i=1}^{m} \lambda_i = 1, \ x = \sum_{i=1}^{m} \lambda_i v^i. \tag{4.24}$$

4.3.3 Enveloppe convexe de fonction séparable

Soit $\psi = (\psi_1, ..., \psi_m)$ une fonction séparable sur $C = C_1 \times C_2 \times ... \times C_m$, avec

$$C_i \subset \text{dom } \psi_i \subset \mathbb{R}^{n_i}, \ i = 1, ..., m,$$

$$\psi(x) = \sum_{i=1}^{m} \psi_i(x_i), \ \forall x = (x_1, ..., x_m) \in C, \tag{4.25}$$

alors $\mathrm{co}_C\ \psi$ peut être déterminer en fonction de $\mathrm{co}_{C_i}\ \psi_i$.

Proposition 4.3.3.1 *Si pour $i = 1, ..., m$, ψ_i est minorée sur C_i par une fonction affine, alors pour tout $x = (x_1, ..., x_m) \in C$*

$$\mathrm{co}_C\ \psi(x) \geq \sum_{i=1}^{m} \mathrm{co}_{C_i}\ \psi_i(x_i) \geq \sum_{i=1}^{m} \overline{\mathrm{co}}_{C_i}\ \psi_i(x_i) = \sum_{i=1}^{m} (\psi_i + \chi_{C_i})^{**}(x_i) = \overline{\mathrm{co}}_C\ \psi(x). \tag{4.26}$$

Après ce bref rappel, nous sommes maintenant en mesure de présenter la technique de relaxation DC.

4.3.4 Minorante convexe de fonction DC sur un ensemble compact convexe

Comme précisé plus haut, la meilleure manière de convexifier un problème d'optimisation non convexe est donnée par l'enveloppe convexe. Cependant, le calcul est très difficile dans la plupart des cas et très souvent dans la pratique on se contente de déterminer une minorante convexe qui soit le plus proche possible.

Etant donné un programme DC

$$\begin{cases} \inf_x & f(x) := \phi(x) - \psi(x) \\ s.t. & x \in C, \end{cases} \tag{4.27}$$

où $\phi, \psi \in \Gamma_0(\mathbb{R}^n)$ et C un convexe compact non vide tel que $C \subset \mathrm{dom}\ \phi \subset \mathrm{dom}\ \psi$, la technique de relaxation DC utilise la structure DC de la fonction objective $f = \phi - \psi$ pour construire une minorante convexe de f. Il faut noter qu'on a

$$f(x) \geq \mathrm{co}_C(\phi - \psi)(x) \geq \phi(x) + \mathrm{co}_C(-\psi)(x) \geq \phi(x) + \overline{\mathrm{co}}_C(-\psi)(x), \forall x \in C. \tag{4.28}$$

La relaxation DC consiste alors à remplacer la fonction f par la fonction convexe f_{relax} définie par

$$f_{relax} := \phi + \mathrm{co}_C(-\psi). \tag{4.29}$$

Ce qui donne le problème convexe relaxé suivant

$$\begin{cases} \inf_x & f_{relax}(x) = \phi(x) + \mathrm{co}_C(-\psi)(x) \\ s.t. & x \in C. \end{cases} \tag{4.30}$$

Conformément aux résultats précédents sur le calcul de l'enveloppe convexe, nous distinguons les minorantes convexes calculables sur C suivantes :

1. $\phi + \mathrm{co}_C(-\psi)$ si $V(C)$ est facile à calculer, par exemple, dans le cas où C est un ensemble convexe polyédral borné avec l'ensemble de sommets $V(C)$.
2. pour le cas général, $\mathrm{co}_C(-\psi)$ sera remplacé par
   (a) $\overline{\mathrm{co}}_L(-\psi)$ où L est un polyèdre contenant C, ($L_i := [a_i, b_i]$ est souvent utilisé en pratique), si ψ est séparable,
   ou
   (b) $\mathrm{co}_S(-\psi)$ avec S un simplexe contenant C.

Chapitre 5

Modélisation parcimonieuse

Sommaire

5.1 Introduction 63
5.2 Apprentissage 65
5.3 Traitement de signaux et d'images 65
5.4 Analyse en composantes principales 66
5.5 Classes de problèmes de parcimonie 67
5.6 Quelques approximations de $\|.\|_0$ 67
5.6.1 Approximation ℓ_p, $0 < p \leq 1$ 68
5.6.2 Approximation exponentielle 68
5.6.3 Approximation linéaire par morceaux 68
5.6.4 Approximation logarithmique 68
5.7 Solution parcimonieuse d'un systéme linéaire 70
5.8 Problème des moindres carrés ℓ_0-régularisé 72
5.9 Modèle parcimonieux pour le problème de la valeur propre maximale 73
5.10 Conclusion 76

5.1 Introduction

Depuis quelques années, on note un intérêt croissant pour la modélisation parcimonieuse dans les domaines des sciences appliquées où interviennent de grandes quantités de données, notamment en fouille de données et en traitement de signaux et d'images. Une formalisation mathématique de cette notion de parcimonie peut être énoncée comme suit :

Etant donné un vecteur $x \in \mathbb{R}^n$, on dira que x est parcimonieux (creux) lorsque la plupart de ces composantes sont nulles.

Définition 5.1.1 *On appelle support d'un vecteur $x \in \mathbb{R}^n$, noté $supp(x)$, l'ensemble des indices des composantes non nulles de x, i.e.,*

$$supp(x) = \{i \in \{1, ..., n\} : \ x_i \neq 0\}. \tag{5.1}$$

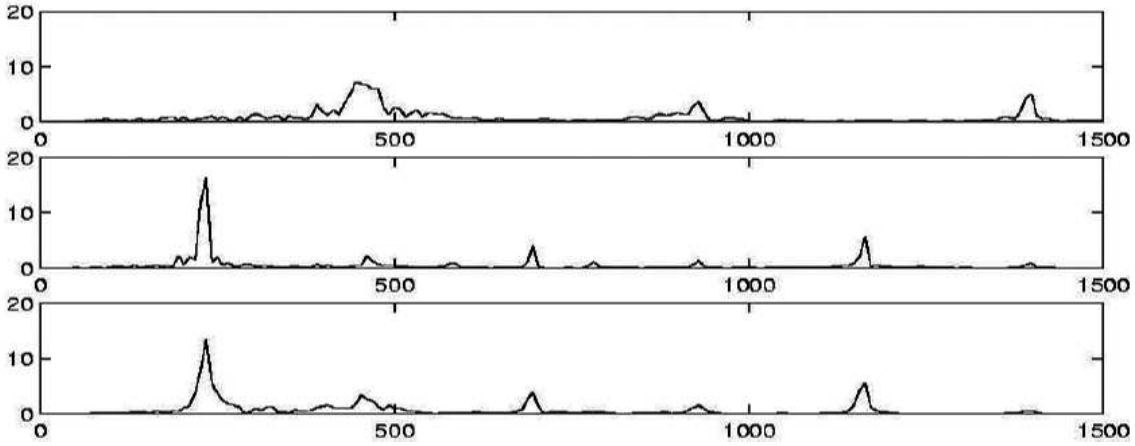

Figure 5.1 – Trois exemples de signaux parcimonieux. La plupart des valeurs des signaux sont nulles.

Définition 5.1.2 *Etant donné un vecteur $x \in \mathbb{R}^n$, on appelle norme ℓ_0 de x le nombre de composantes non nulles de x, i.e.,*

$$\|x\|_0 = card(supp(x)). \tag{5.2}$$

$\|.\|_0$ permet ainsi de quantifier cette notion de parcimonie en ce sens que plus $\|x\|_0$ est petit plus le vecteur x est parcimonieux. Même si on utilise le terme *norme* pour désigner le terme ℓ_0, il faut noter qu'elle n'en est pas une au sens mathématique. En effet, il est facile de voir que pour tout $x \in \mathbb{R}^n$ et tout $\lambda \neq 0$, on a l'égalité

$$\|\lambda x\|_0 = \|x\|_0 ,$$

cette dernière étant fausse pour une norme.

La figure 5.1 illustre cette notion dans un exemple en traitement de signaux [162].

Dans ce qui suit, nous allons détailler quelques modèles mathématiques permettant de prendre en compte cette notion dans différents problèmes considérés dans les domaines cités précédemment.

5.2 Apprentissage

Comme mentionné dans le chapitre précédent, une des faiblesses les plus courantes dans les méthodes d'apprentissage statistique est le sur-apprentissage (la méthode se comporte très bien sur l'échantillon de données utilisé, mais aura des difficultés à généraliser ses performances sur de nouvelles données). Dans le contexte de classification par exemple, le classifieur obtenu est très performant sur les échantillons utilisés, mais aura de la peine à généraliser les caractéristiques des données. Il se comporte alors comme une table contenant tous les échantillons utilisés lors de l'apprentissage et perd ses pouvoirs de prédiction sur de nouveaux échantillons.

Une des techniques les plus utilisées pour pallier au sur-apprentissage est la *régularisation*. Il en existe plusieurs types : régularisation ℓ_2, régularisation ℓ_1, régularisation ℓ_0,...

Ici nous nous intéressons à la régularisation ℓ_0 qui permet de promouvoir des solutions parcimonieuses afin d'éviter le sur-apprentissage.
Etant donné, un modèle basé sur la minimisation d'un critère f, i.e.,

$$\min_{(x,y)} \{f(x,y) : \ x \in \mathbb{R}^n\}, \tag{5.3}$$

on appelle problème ℓ_0-régularisé le problème suivant

$$\min_{(x,y)} \{f(x,y) + \lambda \|x\|_0 : \ x \in \mathbb{R}^n\}. \tag{5.4}$$

Le paramètre $\lambda > 0$, appelé paramètre de régularisation, permet de faire un compromis entre le critère f et la parcimonie des solutions.

5.3 Traitement de signaux et d'images

Considérons un ensemble d'observations représenté par une matrice $A \in \mathbb{M}_{m,n}(\mathbb{R})$, et leur réponse réponse $b \in \mathbb{R}^m$. Très souvent dans la pratique on se trouve dans le cas où m est plus petit que n et donc nous considérons ici $m < n$. Suivant que l'on souhaite une représentation exacte ou approchée nous distinguons les deux problèmes :

- représentation exacte : *trouver un signal x tel que $Ax = b$,*
- représentation ϵ-approchée : *trouver un signal x tel que $\|Ax - b\| \leq \epsilon$.*

Ces deux problèmes possèdent généralement une infinité de solutions dès lors que leur ensemble de solution n'est pas vide. Se pose alors les questions : *quelles solutions choisir ? suivant quel critère ?.*

Une technique très intéressante permettant d'éliminer les redondances est d'en prendre une parmi celles qui sont les plus parcimonieuses, c'est à dire

- représentation exacte : *trouver un signal x le plus parcimonieux possible tel que $Ax = b$,*
- représentation ϵ-approchée : *trouver un signal x le plus parcimonieux possible tel que $\|Ax - b\| \leq \epsilon$.*

Ce qui se traduit en d'autres termes

– représentation exacte : *minimiser* $\|x\|_0$ *sous la contrainte* $Ax = b$,
– représentation ϵ-approchée : *minimiser* $\|x\|_0$ *sous la contrainte* $\|Ax - b\|_2 \leq \epsilon$.

Une formulation alternative aux précédentes est de *minimiser* $\frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_0$, dans laquelle le paramètre $\lambda > 0$ permet de faire un compromis entre le terme d'erreur quadratique $\frac{1}{2}\|Ax - b\|_2^2$ et la parcimonie des solutions.

5.4 Analyse en composantes principales

Comme précédemment, nous allons présenter comment ce principe de parcimonie est incorporé en analyse en composantes principales (ACP). L' ACP est une méthode très importante de la statistique multivariée. Elle consiste à transformer des variables liées entre elles en de nouvelles variables indépendantes les unes des autres. Ces nouvelles variables permettent ainsi de réduire l'information en un nombre de composantes plus limité que le nombre initial de variables. L'ACP peut être reformulée comme des problèmes variationnels de la forme

$$\max \left\{ x^T \Sigma x : x^T x = 1 \right\}, \tag{5.5}$$

où $x \in \mathbb{R}^n$ et Σ est une *matrice variance-covariance* modélisant l'information disponible sur les données et leurs intéractions.

Dans la pratique, pour des données de grandes tailles (n grand), les solutions de (5.5) prennent toute la dimension du problème c'est à dire presque toutes les composantes x_i de la solution x sont non nulles. Ce qui pose des problèmes d'interprétation et de redondance de l'information obtenue à partir de x dans la plupart des cas. Pour pallier à ces problèmes, une contrainte sur la parcimonie des solutions est ajoutée au problème variationnel (5.5) comme suit

$$\begin{cases} \max_x & x^T \Sigma x \\ s.t. & x^T x = 1, \\ & \|x\|_0 \leq k, \end{cases} \tag{5.6}$$

où le paramètre k est un entier compris entre 1 et $n - 1$ et permet de contrôler la parcimonie des solutions.

La parcimonie peut être prise en compte aussi avec la version ℓ_0-régularisée de (5.6)

$$\begin{cases} \max_x & x^T \Sigma x - \rho \|x\|_0 \\ s.t. & x^T x = 1, \end{cases} \tag{5.7}$$

où ρ, appelé paramètre de régularisation, est un nombre réel strictement positif.

5.5 Classes de problèmes de parcimonie

Les différents types de problèmes d'optimisation contenant la norme ℓ_0 peuvent se résumer aux deux classes de problèmes d'optimisation suivantes

$$\begin{cases} \min_{x,y} & f(x,y) \\ s.t. & (x,y) \in C, \\ & \|x\|_0 \leq k, \\ & (x,y) \in \mathbb{R}^n \times \mathbb{R}^m, \end{cases} \tag{5.8}$$

et

$$\begin{cases} \min_{x,y} & f(x,y) + \lambda \|x\|_0 \\ s.t. & (x,y) \in C, \\ & (x,y) \in \mathbb{R}^n \times \mathbb{R}^m, \end{cases} \tag{5.9}$$

où k est un entier compris entre 1 et $n-1$, λ est un réel positif, C un sous ensemble fermé de $\mathbb{R}^n \times \mathbb{R}^m$, f une fonction représentant un critère donné.

Le premier est appelé *problème avec contrainte* ℓ_0 dans le sens où la contrainte $\|x\|_0 \leq k$ permet de contrôler la parcimonie de x tout en minimisant le critère f, et le deuxième est appelé *problème* ℓ_0*-régularisé* dans lequel λ, appelé paramètre de régularisation, permet de faire un compromis entre le critère f et la parcimonie de x.

Théorème 5.5.1 (Existence de solution)

- *Le problème (5.9) admet au moins une solution dès lors que le minimum de f sur C est atteint.*
- *Le problème (5.8) admet au moins une solution dès lors que le minimum de f sur C est atteint et que son ensemble admissible est non vide.*

Très souvent la résolution du problème

$$\begin{cases} \min_{x,y} & f(x,y) \\ s.t. & (x,y) \in C, \\ & (x,y) \in \mathbb{R}^n \times \mathbb{R}^m, \end{cases} \tag{5.10}$$

ne présente pas de grandes difficultés (il existe des méthodes de résolution très efficaces) et ainsi toute la complexité des problèmes (5.8) et (5.9) réside sur le terme $\|x\|_0$ qui rend le problème d'optimisation non convexe, discontinu et difficile à résoudre.

Durant ces dernières années, on rencontre dans la littérature une recherche très active et un nombre très important de méthodes et algorithmes utilisant des approximations continues de $\|.\|_0$.

5.6 Quelques approximations de $\|.\|_0$

Dans ce paragraphe, nous présentons quelques approximations continues de $\|.\|_0$ utilisées dans la littérature.

5.6.1 Approximation ℓ_p, $0 < p \leq 1$

Elle consiste à remplacer le terme $\|.\|_0$ par le terme $\|.\|_p^p$ défini par

$$\|.\|_p^p = \sum_{i=1}^{n} |x_i|^p .$$

$\|.\|_p^p$ est une fonction non convexe si $0 < p < 1$ et possède la propriété suivante

$$\lim_{p \to 0} \|x\|_p^p = \|x\|_0 .$$

Voir figure 5.2 pour des représentations graphiques des fonctions $r \longmapsto |r|^p$, pour $p = 2, 1$ et 0.5.

5.6.2 Approximation exponentielle

Elle consiste à remplacer le terme $\|x\|_0$ par une fonction de la forme (Bradley et Mangasarian[14])

$$m(x, \alpha) := \sum_{i=1}^{n} (1 - \epsilon^{-\alpha |x_i|}),\ \alpha > 0,$$

ϵ représente la base de la fonction logarithmique.

$m(., \alpha)$ est une fonction non convexe et vérifie

$$m(x, \alpha) < \|x\|_0 , \quad \lim_{\alpha \to +\infty} m(x, \alpha) = \|x\|_0 .$$

Voir figure 5.2 pour une représentation graphique de la fonction $r \longmapsto 1 - \epsilon^{-\alpha |r|}$.

5.6.3 Approximation linéaire par morceaux

Elle consiste à remplacer le terme $\|x\|_0$ par la fonction linéaire par morceaux

$$z(x, \alpha) := \sum_{i=1}^{n} \min(\frac{|x_i|}{\alpha}, 1),\ \alpha > 0.$$

$z(., \alpha)$ est une fonction non convexe et possède les propriétés suivantes

$$z(x, \alpha) < \|x\|_0 , \quad \lim_{\alpha \to 0^+} z(x, \alpha) = \|x\|_0 .$$

Voir figure 5.2 pour une illustration de la fonction $r \longmapsto \min(\frac{|r|}{\alpha}, 1)$.

5.6.4 Approximation logarithmique

Dans ce cas, le terme $\|x\|_0$ est remplacé par (Weston et al.[158])

$$w(x, \alpha) := \sum_{i=1}^{n} log(1 + \frac{|x_i|}{\alpha}),\ \alpha > 0.$$

Voir figure 5.2 pour une représentation graphique de la fonction $r \longmapsto log(1 + \frac{|r|}{\alpha})$.

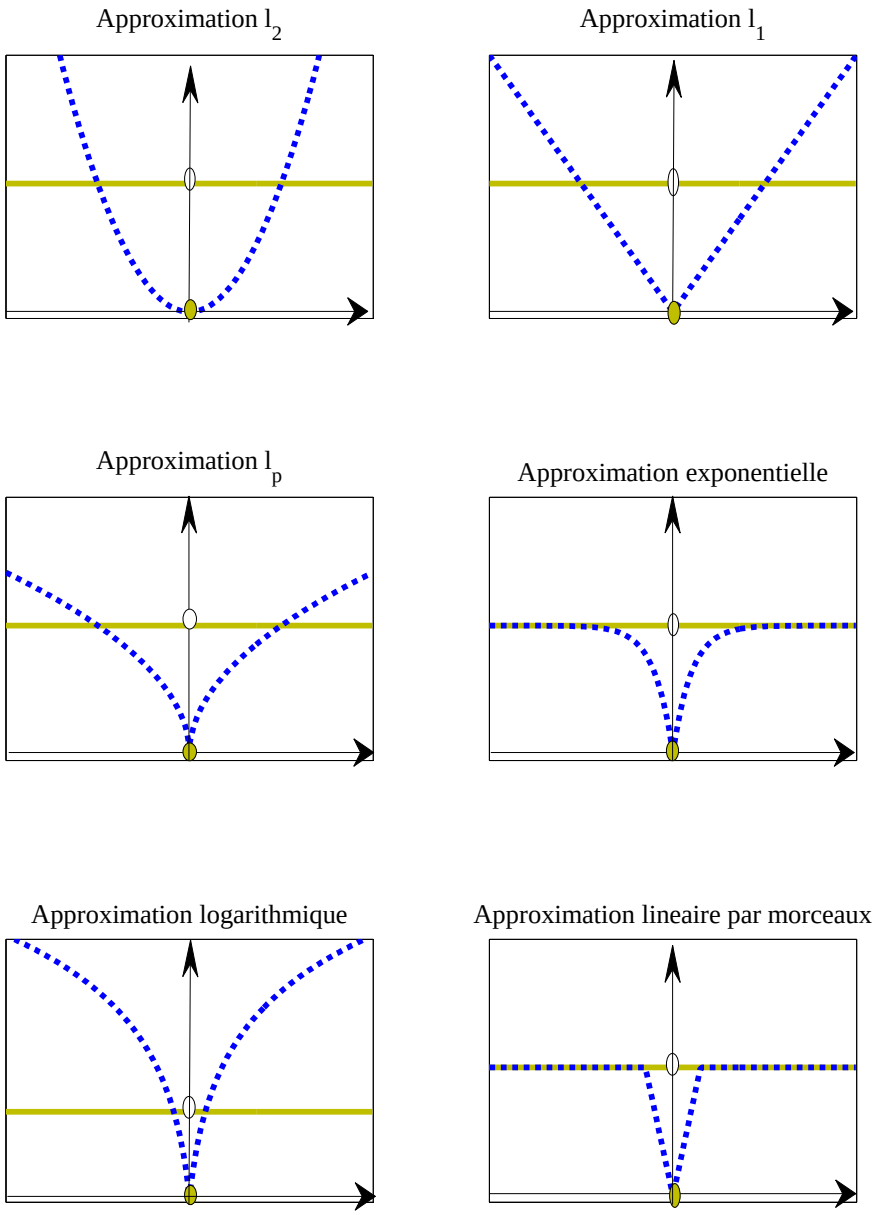

FIGURE 5.2 – Différents types d'approximations.

5.7 Solution parcimonieuse d'un systéme linéaire

Etant donné une matrice $A \in \mathbb{M}_{m,n}(\mathbb{R})$ et un vecteur $b \in \mathbb{R}^m$. On appelle solution parcimonieuse du sytème $Ax = b$ toute solution ayant le plus petit nombre de composantes non nulles, en d'autres termes toute solution du problème

$$\begin{cases} \min_x & \|x\|_0 \\ s.t. & Ax = b, \\ & x \in \mathbb{R}^n. \end{cases} \tag{5.11}$$

(5.11) est discontinu et NP-difficile [111, 37]. Pour contourner cette discontinuité, des méthodes d'approximation continues ont été proposées dans la littérature dont sans doute l'une des plus fructueuse et des plus intéressante est l'approche poursuite de bases (Basis pursuit BP), Chen et al. [27], qui utilise l'approximation ℓ_1

$$\begin{cases} \min_x & \|x\|_1 \\ s.t. & Ax = b, \\ & x \in \mathbb{R}^n. \end{cases} \tag{5.12}$$

(5.12) est un problème d'optimisation convexe et il existe une large variété de méthodes très éfficaces permettant d'en obtenir une solution : programmation linéaire [12], méthodes de points intérieurs [18, 27, 74], méthodes de gradient projeté [149, 45] et d'autres méthodes itératives [35, 58]. La solution est dans certains cas satisfaisante, sans toutefois atteindre le résultat qui aurait été obtenu avec la norme ℓ_0.

Il est démontré que si la solution est suffisament parcimonieuse et sous certaines conditions spéciales sur les données (A, b), l'approximation ℓ_1 (5.12) permet de retrouver une solution du problème ℓ_0 (5.11) [39, 38, 56, 20, 21, 19].

Nous présentons dans la suite quelques conditions suffisantes permettant de retrouver une solution de (5.11) à partir de (5.12).

Définition 5.7.1 (*spark*, Donoho et Elad[38]) *Le spark d'une matrice A, noté $spark(A)$, est le plus petit nombre de colonnes de A linéairement dépendantes.*

Noter bien la différence avec le rang de la matrice A ($rang(A)$) qui représente le plus grand nombre de colonnes de A linéairement indépendantes.

Théorème 5.7.2 (Gorodnitsky et Rao [54]) *Si x est une solution du système $Ax = b$ vérifiant*

$$\|x\|_0 < \frac{spark(A)}{2},$$

alors x est la solution du problème ℓ_0 (5.11).

Preuve Soit $y \neq x$ tel que $Ay = b$. On a

$$A(x - y) = 0 \text{ et } x - y \neq 0.$$

Donc $\|x-y\|_0 \geq spark(A)$. Comme $\|x\|_0 + \|y\|_0 \geq \|x-y\|_0$, nous obtenons

$$\|y\|_0 \geq \|x-y\|_0 - \|x\|_0 \geq spark(A) - \|x\|_0 > \frac{spark(A)}{2} > \|x\|_0 .$$

Ce qui termine la preuve. □

Définition 5.7.3 (Cohérence mutuelle, Mallat et Zhang [106]) *La cohérence mutuelle d'une matrice A, notée $\mu(A)$, est le nombre défini par*

$$\mu(A) := \max_{k,j,k\neq j} \frac{A_k^T A_j}{\|A_k\|_2 \|A_j\|_2}.$$

Théorème 5.7.4 (Donoho et Elad [38])

$$spark(A) \geq 1 + \mu(A)^{-1}.$$

Théorème 5.7.5 *Si x est une solution du système $Ax = b$ vérifiant*

$$\|x\|_0 < \frac{1+\mu(A)^{-1}}{2},$$

alors x est l'unique solution du problème ℓ_0 (5.11).

Théorème 5.7.6 (Donoho et Elad [38], Gribonval et Nielsen [56]) *Si les colonnes de A sont normalisées et si x est une solution du système $Ax = b$ vérifiant*

$$\|x\|_0 < \frac{1+\mu(A)^{-1}}{2},$$

alors x est l'unique solution du problème ℓ_0 (5.11) et l'unique solution du problème ℓ_1 (5.12).

Définition 5.7.7 (Restricted Isometry Constant, Candès et Tao [22]) *Soit s un entier compris entre 1 et m. On définit δ_s (Restricted Isometry Constant) comme étant le plus petit réel positif tel que*

$$(1-\delta_s)\|c\|_2^2 \leq \|A_S c\|_2^2 \leq (1+\delta_s)\|c\|_2^2 ,$$

pour tout sous-ensemble S de s indices et tout vecteur c de dimension s. A_S représentant la sous matrice de A constituée des colonnes de A d'indices dans S.

Théorème 5.7.8 (Candès et Tao [22]) *Soit $x \in \mathbb{R}^n$ tel que $\|x\|_0 \leq k$. Soit $b = Ax$. Si A vérifie*

$$\delta_k + \delta_{2k} + \delta_{3k} < 1,$$

alors x est l'unique solution du problème ℓ_1 (5.12).

Il est important de noter que ces conditions suffisantes bien que presque toujours vérifiées par certaines classes de matrices aléatoires [22, 19, 40, 108, 9], sont très difficilement vérifiables pour une matrice A donnée.

Une extension des travaux de Candès et Tao [22], dans le cadre de l'approximation ℓ_p, $0 < p < 1$, a été développée dans Chartrand et Staneva [26]. Très récemment, il est établi dans Fung et Mangasarian [50], que pour des valeurs de p suffisamment petit l'approximation ℓ_p permet de retrouver une solution du problème ℓ_0 (5.11).

Il existe aussi des algorithmes de type gloutons comme l'algorithme Poursuite adaptative (Matching pursuit MP) Mallat et Zhang [106] et son extension Poursuite orthogonale adaptative (Orthogonal matching pursuit OMP) Pati et al. [116] qui consistent à trouver itérativement dans le dictionnaire l'élément le plus corrélé avec le signal, d'ôter la contribution de cet élément au signal et de recommencer le processus jusqu'à obtention d'un nombre d'éléments égal à la dimension du signal. Il suffit ensuite de décomposer le signal sur la famille génératrice obtenue pour trouver les coefficients. Cet algorithme de type glouton n'est pas optimal [36, 27], mais il offre dans certains cas de bonnes propriétés de décroissance de l'erreur entre le signal original et le signal approché. Lorsque le signal original est suffisament parcimonieux et sous certaines conditions spéciales sur les données (A, b), l'OMP peut retrouver le signal Tropp [144]. D'importants travaux de développement et d'améliorations de ces méthodes peuvent être trouvés dans [152, 55, 145] ainsi qu'une étude bibliographique assez complète [15].

Il existe d'autres méthodes hybrides utilsant les méthodes ci-dessus et les approximations présentées dans la section précédente.

5.8 Problème des moindres carrés ℓ_0-régularisé

A et b étant définis comme précédemment. Une formulation du problème des moindres carrés est donnée par

$$\begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 \\ s.t. & x \in \mathbb{R}^n. \end{cases} \tag{5.13}$$

Le problème des moindres carrés ℓ_0-régularisé est défini par

$$\begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n, \end{cases} \tag{5.14}$$

où le paramètre de régularisation $\lambda > 0$ permet de faire le compromis entre le critère des moindres carrés et la parcimonie de la solution x.

C'est aussi un problème NP-difficile et combinatoire Natarajan [111]. Comme pour le cas précédent une des approximations continues les plus utilisées est l'approche Poursuite de Base appelé aussi LASSO qui utilise l'approximation ℓ_1 [141,

27, 115, 146]

$$\begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_1 \\ s.t. & x \in \mathbb{R}^n. \end{cases} \tag{5.15}$$

(5.15) est un problème convexe et il existe plusieurs méthodes très efficaces pour en obtenir une solution [74, 58] par exemple. Généralement, la résolution du problème ne permet pas d'obtenir une solution du problème (5.14).

Plusieurs méthodes permettant d'obtenir des solutions plus parcimonieuses que (5.15) ont été proposées dans la littérature. Ces méthodes sont le plus souvent des méthodes itératives utilisant des repondérations de problèmes de type LASSO à chaque itération [166, 52]. Une des premières est le SCAD (Smoothly Clipped Absolute Deviation) proposé dans [44]. Une deuxième approche utilisant l'approximation $\ell_p,\ 0 < p < 1$ a été présentée dans [48, 75, 70, 28]. Cependant, contrairement à la minimisation ℓ_1 de la section précédente, on ne dispose pas de résultats théoriques rigoureux concernant d'éventuelles liens entre ses solutions et les solutions du problème ℓ_0-régularisé (5.14).

Il existe aussi d'autres algorithmes gloutons utilisant des méthodes de poursuite et des méthodes hybrides avec les approximations ci-dessus, permettant d'améliorer les solutions obtenues avec les approximations ci-dessus.

Des résultats empiriques assez satisfaisants ont été obtenus avec ces différentes méthodes qui continuent d'être améliorées de jour en jour même si parfois les garanties théoriques n'existent pas concernant le lien entre les solutions obtenues et les solutions du problème ℓ_0 original. Il faut noter que l'approximation ℓ_1 est sans doute celle dont les résultats ont été les plus fructueux pour la prise en main et la compréhension du problème de minimisation ℓ_0 [39, 38, 56, 20, 21, 19].

5.9 Modèle parcimonieux pour le problème de la valeur propre maximale

Etant donné une matrice symétrique réelle d'ordre n, $A \in \mathbb{M}_n(\mathbb{R})$, une formulation variationnelle du problème de la valeur propre maximale est donnée par

$$\begin{cases} \max_x & x^T A x \\ s.t. & x^T x = 1, \\ & x \in \mathbb{R}^n. \end{cases} \tag{5.16}$$

Nous distinguons les deux types de formulations du problème de la valeur propre maximale avec parcimonie suivantes

$$\begin{cases} \max_x & x^T A x \\ s.t. & x^T x = 1, \\ & \|x\|_0 \leq k, \\ & x \in \mathbb{R}^n \end{cases} \tag{5.17}$$

et sa version régularisée

$$\begin{cases} \max_x & x^T A x - \rho \|x\|_0 \\ s.t. & x^T x = 1, \\ & x \in \mathbb{R}^n, \end{cases} \tag{5.18}$$

où k est un entier compris entre 1 et $n-1$ et ρ est un paramètre strictement positif.

Le problème (5.17) est appelé problème de la valeur propre maximale avec contrainte ℓ_0, dans lequel le paramètre k permet de contrôler la parcimonie de la solution. Le deuxième (5.18) est appelé problème de la valeur propre maximale ℓ_0-régularisé, dans lequel le paramètre de régularisation ρ permet de faire un compromis entre la parcimonie de x et le critère x^TAx.

Ces deux problèmes sont combinatoires et NP-difficiles [109]. Il existe plusieurs approches pour ces problèmes dans littérature : ℓ_1, semi-définie positive, gloutonne combinatoire et programmation DC par exemple.

Une des premières méthodes pour ces problèmes consistait à mettre à zéro tous les composantes dont le module était inférieur à un seuil donné [17]. Plus tard, dans [73] est proposé SCoTLASS dans laquelle la contrainte de cardinalité est remplacée par une contrainte avec la norme ℓ_1. Récemment, une méthode appelée SPCA, utilisant la régularisation ℓ_1 de [141], a été présentée dans [168]. Une méthode DC appelée DC-PCA utilisant une approximation logarithmique [158] a été proposée dans Sriperumbudur et al. [135] et des algorithmes combinatoires gloutons, ont été développés dans Moghaddam et al. [110] et d'Aspremont et al. [33]. Il existe aussi une méthode utilisant une relaxation SDP (DSPCA)[34]. D'intéressantes conditions suffisantes d'optimalité globale, permettant de confirmer la globalité dans certains cas, ont été développées par d'Aspremont et al. [33].

Nous détaillons dans la suite, ces conditions suffisantes d'optimalité globale. Elles ont été établies grâce à des techniques de la programmation semi-définie positive SDP. On pourra se référer aux travaux [4, 13, 160] pour plus de détails sur la programmation semi-définie.

Théorème 5.9.1 (d'Aspremont et al. [33]) *Soit $A \in \mathbb{M}_n(\mathbb{R})$, $\rho \geq 0$, $\Sigma = A^TA$ avec $a_1, ..., a_n$ les colonnes de A. Soit $J \subset \{1, ..., n\}$ un sous ensemble d'indices. Soit x un vecteur propre correspondant à la plus grande valeur propre de $\sum_{j\in J} a_j a_j^T$. S'il existe $\rho^* \geq 0$ tel que les conditions suivantes sont vérifiées*

$$\max_{j\in J^c}(a_j^Tx)^2 < \rho^* < \max_{j\in J}(a_j^Tx)^2 \text{ et } \lambda_{\max}\left(\sum_{i=1}^n Y_i\right) \leq \sum_{j\in J}((a_j^Tx)^2 - \rho^*)$$

avec

$$Y_j = \begin{cases} \max\left\{0, \rho\frac{a_j^Ta_j-\rho}{\rho-(a_j^Tx)^2}\right\}\frac{(I-xx^T)a_ja_j^T(I-xx^T)}{\|(I-xx^T)a_j\|^2} & \text{si } j \in J^c, \\ \frac{(a_ja_j^T-\rho I)xx^T(a_ja_j^T-\rho I)}{x^T(a_ja_j^T-\rho I)x} & \text{si } j \in J, \end{cases} \qquad j = 1, ..., n,$$

alors

$$\bar{z} \in \arg\max\left\{z^T\Sigma z : \|z\| = 1, z_{J^c} = 0\right\}$$

est une solution de (5.18) avec $\rho = \rho^*$.

D'après le théorème précédent, pour un ensemble J donné, il faudra déterminer ρ^* vérifiant les conditions du théorème pour confirmer la globalité de la solution $\bar{z}$. Le résultat suivant permet de trouver un tel ρ^*, s'il existe.

Lemme 5.9.2 (d'Aspremont et al. [33]) *Soit $A \in \mathbb{M}_n(\mathbb{R})$, $\rho \geq 0$, $\Sigma = A^T A$ avec $a_1, ..., a_n$ les colonnes de A. Soit $J \subset \{1, ..., n\}$ un sous ensemble d'indices. Soit x un vecteur propre correspondant à la plus grande valeur propre de $\sum_{j \in J} a_j a_j^T$ et*

$$Y_j = \begin{cases} \max\left\{0, \rho \frac{a_j^T a_j - \rho}{\rho - (a_j^T x)^2}\right\} \frac{(I - xx^T) a_j a_j^T (I - xx^T)}{\|(I - xx^T) a_j\|^2} & \text{si } j \in J^c, \\ \frac{(a_j a_j^T - \rho I) xx^T (a_j a_j^T - \rho I)}{x^T (a_j a_j^T - \rho I) x} & \text{si } j \in J, \end{cases} \qquad j = 1, ..., n.$$

Le saut de dualité dans (5.18), donné par

$$gap(\rho) := \lambda_{\max}\left(\sum_{i=1}^{n} Y_i\right) - \sum_{j \in J}((a_j^T x)^2 - \rho),$$

est une fonction convexe en la variable ρ lorsque

$$\max_{j \in J^c}(a_j^T x)^2 < \rho < \max_{j \in J}(a_j^T x)^2.$$

Ce résultat montre que l'ensemble des valeurs ρ^* devrait être un intervalle. Cet intervalle peut être vide et dans ce cas $\bar{z}$ n'est pas optimale globale. Il donne aussi une procédure (d'Aspremont et al. [33]) pour confirmer la globalité de $\bar{z}$ en minimisant la fonction convexe $gap(.)$ sur l'intervalle $[\rho_{\min}, \rho_{\max}]$,

$$\rho_{\min} := \max_{j \in J^c}(a_j^T x)^2, \ \rho_{\max} := \max_{j \in J}(a_j^T x)^2.$$

Soit

$$\rho^* \in \arg\min\left\{gap(\rho) : \rho \in [\rho_{\min}, \rho_{\max}]\right\}. \tag{5.19}$$

Si la valeur optimale $gap(\rho^*)$ est nulle, alors $\bar{z}$ est une solution optimale de (5.18) pour $\rho = \rho^*$.

Le problème convexe (5.19) peut être résolu efficacement par une technique de recherche binaire avec une complexité $O(n^2 log((\rho_{\max} - \rho_{\min})/\epsilon))$, ϵ étant la précision souhaitée. La complexité de toute la procédure est de $O(n^3 + n^2 log((\rho_{\max} - \rho_{\min})/\epsilon))$ pour un sous-ensemble d'indice J donné.

Ces conditions suffisantes d'optimalité globale sont très intéressantes et permettent de confirmer l'optimalité dans certains cas pour un sous ensemble d'indices de colonnes J donné (J étant un sous-ensemble candidat fourni par un algorithme donné). Cependant, du fait du saut de dualité ces conditions ne sont pas nécessairement pas vérifiées par toute solution globale (il peut y avoir des solutions globales ne satisfaisant pas ces conditions).

5.10 Conclusion

Nous avons décrit dans ce chapitre différents types de problèmes rencontrés en modélisation parcimonieuse de données. Nous avons aussi présenté différentes approches et approximations utilisées ainsi que quelques résultats théoriques. Pour chacune des formulations ℓ_0 présentées, nous proposerons des reformulations DC voire quadratiques et de nouvelles propriétés grâce à de nouvelles techniques développées durant cette thèse. Le chapitre 6 est dédié à la modélisation parcimonieuse pour le problème de la valeur propre maximale tandis que le chapitre 7 concerne la modélisation parcimonieuse dans les modèles de régression linéaire.

Chapitre 6

Modélisation parcimonieuse pour le problème de la valeur propre maximale

Ce chapitre présente une nouvelle approche pour la modélisation parcimonieuse pour le problème de la valeur propre maximale. Après une présentation des problèmes de parcimonie rencontrés, nous proposerons de nouvelles reformulations et montrerons l'équivalence avec ces problèmes de parcimonie. Nous présenterons ensuite l'algorithme DCA appliqué à ces différentes reformulations et terminerons par des simulations numériques dans le cadre l'analyse en composantes principales.

6.1 Introduction

Etant donné une matrice symétrique réelle A d'ordre n, la formulation variationnelle du problème de la valeur propre maximale est donnée par

$$\begin{cases} \max & x^T Ax \\ s.t. & x^T x = 1, \\ & x \in \mathbb{R}^n. \end{cases} \tag{6.1}$$

On appelle modèle parcimonieux pour le problème de la valeur propre maximale tout problème ayant une des formes

$$\begin{cases} \max_x & x^T Ax \\ s.t. & x^T x = 1, \\ & \|x\|_0 \leq k, \\ & x \in \mathbb{R}^n, \end{cases} \tag{6.2}$$

ou

$$\begin{cases} \max_x & x^T Ax - \rho \|x\|_0 \\ s.t. & x^T x = 1, \\ & x \in \mathbb{R}^n, \end{cases} \tag{6.3}$$

où k est un entier compris entre 1 et $n-1$ et ρ est un nombre réel strictement positif.

La première formulation permet de maximiser l'information contenue sur les données (modélisée par $x^T Ax$) tout en contrôlant la parcimonie des solutions (modélisée par $\|x\|_0$) et la deuxième permet de faire un compromis entre l'information et la parcimonie des solutions.

Bien que le problème (6.1) soit non convexe, il existe des méthodes de résolution très efficaces. Il est clair que par un procédé énumératif fini les problèmes (6.2) et (6.3) peuvent être transformés en un nombre fini de sous problèmes de la forme (6.1). Cependant ce procédé n'est pas utilisable en pratique pour des problèmes avec des dimensions concrètes du fait de sa complexité exponentielle.

Les deux problèmes ℓ_0 (6.2) et (6.3) sont discontinus et NP-difficiles. Afin de contourner ces difficultés plusieurs approximations ont été proposées dans la littérature. A l'exception de l'approche gloutonne combinatoire Moghaddam et al. [110] et de l'approche semi-définie positive d'Aspremont et al. [34, 33], ces approximations sont très souvent systématiques et manquent de liens rigoureux entre le problème original et le problème approché. Dans ce travail nous proposons des reformulations DC de ces problèmes ℓ_0 par des techniques de pénalité et montrons l'exactitude des reformulations. Les résultats expérimentaux préliminaires illustrent bien le potentiel de cette nouvelle approche.

Le chapitre est organisé de la façon suivante. Dans le premier paragraphe, nous rappellerons quelques notations utilisées dans ce chapitre. Des reformulations DC équivalentes pour les deux problèmes de la modélisation parcimonieuse seront présentées dans le deuxième paragraphe. Dans le troisième paragraphe, nous décrirons les algorithmes DCA appliqués à ces reformulations DC. Finalement, nous discuterons de l'originalité de cette nouvelle approche et présenterons quelques résultats expérimentaux préliminaires.

6.2 Notations

$\Gamma_0(\mathbb{R}^p)$	représente la classe des fonctions $f : \mathbb{R}^p \to]-\infty, +\infty]$ convexes, semi-continues inférieurement et propres.
χ_C	la fonction indicatrice du sous ensemble C, $\chi_C(x) = 0$, si $x \in C$ et $\chi_C(x) = +\infty$ sinon.
$\lambda_{max}(A)$	la valeur propre maximale de la matrice A.
$e^1, ..., e^n$	les éléments de la base canonique de $\mathbb{R}^n$.
e	le vecteur de $\mathbb{R}^n$ dont toutes les composantes sont égales à 1.
A_{ij}	coefficient correspondant à la ligne i et colonne j de A.
I	matrice identité d'ordre n.

6.3 Reformulations DC

Rappelons qu'un programme DC est un problème d'optimisation de la forme

$$(P_{dc}) \quad \alpha = \inf \{f(z) := g(z) - h(z) : z \in \mathbb{R}^p\},$$

avec $g, h \in \Gamma_0(\mathbb{R}^p)$. Rappelons aussi qu'un problème d'optimisation avec contraintes de la forme

$$\inf \{g(z) - h(z) : z \in C, z \in \mathbb{R}^p\},$$

où $g, h \in \Gamma_0(\mathbb{R}^p)$ et C est un sous ensemble convexe, peut être mis sous la forme (P_{dc}) en ajoutant la fonction indicatrice de C, χ_C, à g ce qui donne

$$\inf \{G(z) - h(z) : z \in \mathbb{R}^p\},$$

avec $G := g + \chi_C \in \Gamma_0(\mathbb{R}^p)$.

Nous allons d'abord donner des reformulations continues des problèmes ℓ_0 (6.2) et (6.3), ensuite par une technique de pénalité adéquate nous pénaliserons la ou les contraintes qui ne seront pas convexes et montrerons l'équivalence entre le problème obtenu (dont toutes les contraintes sont convexes) et le problème ℓ_0 de départ.

6.3.1 Problème avec contrainte ℓ_0

Considérons le problème avec contrainte ℓ_0 (6.2). Sans perte de généralité, nous supposerons que A est semi-définie positive dans ce qui suit. En effet, si A ne l'est pas, nous choisirons $\mu > 0$ de sorte que $\mu I + A$ soit semi-définie positive et considérerons le problème équivalent

$$\begin{cases} \max_x & x^T(\mu I + A)x \\ s.t. & x^T x = 1, \\ & \|x\|_0 \leq k. \end{cases}$$

Ainsi, nous obtenons un problème équivalent en remplaçant la contrainte $x^T x = 1$ dans (6.2) par la contrainte $x^T x \leq 1$. Et finalement, nous considérons le problème équivalent

$$\begin{cases} \max_x & x^T A x \\ s.t. & x^T x \leq 1, \\ & \|x\|_0 \leq k. \end{cases} \tag{6.4}$$

Une formulation mixte $0-1$ de (6.4) est donnée par

$$\begin{cases} \max_{(x,u)} & x^T A x \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u \leq k, \\ & u \in \{0,1\}^n. \end{cases} \tag{6.5}$$

La variable binaire u nous permet de contrôler les composantes non nulles de x dans le sens

$$u_i = 0 \;\Rightarrow\; x_i = 0; \;\; x_i \neq 0 \;\Rightarrow\; u_i = 1.$$

Nous allons d'abord donner une version continue de (6.5). Pour cela considérons la fonction q, bien connue en programmation DC, définie par

$$q(u) := \sum_{j=1}^{n} u_j(1 - u_j) = e^T u - u^T u. \tag{6.6}$$

Elle possède les propriétés suivantes

– *q est une fonction concave et positive sur $[0,1]^n$.*
– *$u \in \{0,1\}^n$ si et seulement si $q(u) = 0$* et $u \in [0,1]^n$.

Le problème (6.5) devient alors

$$\begin{cases} \max_{(x,u)} & x^T A x \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u \leq k, \\ & q(u) = 0, \\ & u \in [0,1]^n, \end{cases}$$

qui est équivalent au problème de minimisation

$$\begin{cases} \min_{(x,u)} & -x^T A x \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u \leq k, \\ & q(u) = 0, \\ & u \in [0,1]^n. \end{cases} \tag{6.7}$$

Maintenant, dans (6.7) seule la contrainte $q(u) = 0$ est non convexe. C'est la contrainte que nous allons pénaliser dans la suite.

Pour $t > 0$, considérons le problème pénalisé

$$\begin{cases} \min_{(x,u)} & f_t(x,u) := -x^T A x + t q(u) \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^T u \leq k, \\ & u \in [0,1]^n. \end{cases} \tag{6.8}$$

On va montrer que (6.8) est équivalent à (6.4). Pour cela il faut et il suffit de montrer que (6.8) est équivalent à (6.7), c'est à dire $q(\bar{u}) = 0$ pour toute solution $(\bar{x}, \bar{u})$ de (6.8).

Proposition 6.3.1.1

$$\textit{Soit } t > t_1(A,k) := 2 \max \left\{ \max_{j=1,...,n} \left[2 \sum_{k=1, k \neq j}^{n} |A_{kj}| + A_{jj} \right], \lambda_{\max}(A) - \max_j A_{jj} \right\}.$$

Alors, (6.4) et (6.8) sont équivalent dans le sens :
– *Si $\bar{x}$ est une solution de (6.4), alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{u})$ est une solution de (6.8).*
– *Si $(\bar{x}, \bar{u})$ est une solution de (6.8), alors $\bar{x}$ est une solution de (6.4).*

Preuve Soit $(\bar{x}, \bar{u})$ une solution de (6.8). Nous procédons par contradiction.

Supposons que $q(\bar{u}) > 0$. Nous allons préciser comment calculer un point admissible (x, u) de (6.8) tel que $f_t(\bar{x}, \bar{u}) > f_t(x, u)$, en contradiction avec $(\bar{x}, \bar{u})$ solution de (6.8). Pour cela posons

$$J := \{j \in \{1, ...n\} : 0 < \bar{u}_j < 1 - \bar{u}_j\} \text{ et } I := \{j \in \{1, ...n\} : 0 < 1 - \bar{u}_j \leq \bar{u}_j\}.$$

Considérons les cas suivants :

- $J \neq \varnothing$. Soit $j_0 \in J$. Posons $x_j := \bar{x}_j, u_j := \bar{u}_j, \forall j \neq j_0$ et $x_{j_0} = u_{j_0} := 0$.

$$\begin{aligned} f_t(\bar{x}, \bar{u}) - f_t(x, u) &= -2\sum_{k=1, k\neq j_0}^{n} A_{kj_0}\bar{x}_k\bar{x}_{j_0} - A_{j_0j_0}\bar{x}_{j_0}^2 + t\bar{u}_{j_0}(1 - \bar{u}_{j_0}) \\ &\geq -|\bar{x}_{j_0}|\left[2\sum_{k=1, k\neq j_0}^{n}|A_{kj_0}| + A_{j_0j_0}\right] + \tfrac{1}{2}t\bar{u}_{j_0} \\ &\geq \bar{u}_{j_0}\left[\tfrac{1}{2}t - 2\sum_{k=1, k\neq j_0}^{n}|A_{kj_0}| - A_{j_0j_0}\right] \\ &> 0. \end{aligned}$$

- $J = \varnothing$ et $I \neq \varnothing$. Si $e^T\bar{u} < k$, alors soient $i_0 \in I$ et $\epsilon > 0$ tel que $e^T\bar{u} + \epsilon \leq k$ et $\bar{u}_{i_0} + \epsilon \leq 1$ et posons $x := \bar{x}$, $u_i := \bar{u}_i, \forall i \neq i_0$ et $u_{i_0} := \bar{u}_{i_0} + \epsilon$,

$$f_t(\bar{x}, \bar{u}) - f_t(x, u) = t\epsilon(-1 + \epsilon + 2\bar{u}_{i_0}) > 0,$$

sinon soit i tel que $A_{ii} = \max_j A_{jj}$, alors posons $(x, u) := (e^i, e^i)$.

$$\begin{aligned} f_t(\bar{x}, \bar{u}) - f_t(e^i, e^i) &= f_t(\bar{x}, \bar{u}) + A_{ii} \\ &\geq -\bar{x}^TA\bar{x} + \tfrac{1}{2}t + A_{ii} \\ &\geq -\lambda_{\max}(A)\bar{x}^T\bar{x} + \tfrac{1}{2}t + A_{ii} \\ &\geq -\lambda_{\max}(A) + \tfrac{1}{2}t + A_{ii} \\ &> 0. \end{aligned}$$

□

A partir de cette reformulation, nous tirons une deuxième reformulation en remplaçant la contrainte $e^Tu \leq k$ par $e^Tu = k$ et en enlevant e^Tu de la fonction objective, i.e.,

$$\begin{cases} \min_{(x,u)} & f_t^1(x, u) := -x^TAx - tu^Tu \\ s.t. & x^Tx \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & e^Tu = k, \\ & u \in [0, 1]^n, \end{cases} \tag{6.9}$$

avec $t > t_1(A, k)$.

Proposition 6.3.1.2 *(6.9) et (6.4) sont équivalents dans le sens :*

- *Si $\bar{x}$ est une solution de (6.4), alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{u})$ est une solution de (6.9).*
- *Si $(\bar{x}, \bar{u})$ est une solution de (6.9), alors $\bar{x}$ est une solution de (6.4).*

Preuve Il suffit de montrer que (6.8) possède une solution $(\bar{x}, \bar{u})$ tel que $e^T\bar{u} = k$.

Soit $(\tilde{x}, \tilde{u})$ une solution de (6.8). On a $e^T\tilde{u} \leq k$ et $\tilde{u} \in \{0, 1\}^n$. Pour construire cette solution $(\bar{x}, \bar{u})$, on pose $\bar{x} := \tilde{x}$, $\bar{u}_j := \tilde{u}_j, \forall j$ tel que $\tilde{u}_j = 1$ et on compléte le reste des composantes de $\bar{u}$ par 0 ou 1 pour avoir $e^T\bar{u} = k$. □

6.3.2 Problème ℓ_0-régularisé

Considérons le problème ℓ_0-régularisé (6.3). Pour les mêmes raisons que précédemment, nous supposons (sans perte de généralité) que A est semi-définie positive et nous considérons le problème

$$\begin{cases} \max_x & x^T Ax - \rho \|x\|_0 \\ s.t. & x^T x \leq 1. \end{cases} \tag{6.10}$$

Pour $t > 0$, considérons le problème pénalisé suivant

$$\begin{cases} \min_{(x,u)} & \phi_t(x,u) := -x^T Ax + \rho e^T u + tq(u) \\ s.t. & x^T x \leq 1, \\ & |x_j| \leq u_j, j = 1, ..., n, \\ & u \in [0,1]^n, \end{cases} \tag{6.11}$$

où q est définie comme précédemment (6.6).
La proposition suivante montre l'équivalence entre (6.10) et (6.11).

Proposition 6.3.2.1

$$\textit{Soit } t > t_2(A,\rho) := 2\max\left\{\rho, \max_{j=1,...,n}\left[2\sum_{k=1,k\neq j}^{n}|A_{kj}| + A_{jj}\right] - \rho\right\},$$

(6.10) et (6.11) sont équivalents dans le sens :

- *Si $\bar{x}$ est une solution de (6.10), alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{u})$ est une solution de (6.11).*
- *Si $(\bar{x}, \bar{u})$ est une solution de (6.11), alors $\bar{x}$ est une solution de (6.10).*

Preuve Soit $(\bar{x}, \bar{u})$ une solution de (6.11). Nous procédons par contradiction. Supposons que $q(\bar{u}) > 0$. Nous allons préciser comment calculer un point admissible (x, u) de (6.11) tel que $\phi_t(\bar{x}, \bar{u}) > \phi_t(x, u)$, en contradiction avec $(\bar{x}, \bar{u})$ solution de (6.11). Pour cela posons

$$J := \{j \in \{1, ...n\} : 0 < \bar{u}_j < 1 - \bar{u}_j\} \text{ et } I := \{j \in \{1, ...n\} : 0 < 1 - \bar{u}_j \leq \bar{u}_j\}.$$

Considérons les cas suivants :

- $J \neq \varnothing$. Soit $j_0 \in J$. Prendre $x_j := \bar{x}_j, u_j := \bar{u}_j, \forall j \neq j_0$ et $x_{j_0} = u_{j_0} := 0$.

$$\begin{aligned} \phi_t(\bar{x}, \bar{u}) - \phi_t(x, u) &= -2\textstyle\sum_{k=1,k\neq j_0}^{n} A_{kj_0}\bar{x}_k\bar{x}_{j_0} - A_{j_0j_0}\bar{x}_{j_0}^2 + \rho\bar{u}_{j_0} + t\bar{u}_{j_0}(1 - \bar{u}_{j_0}) \\ &\geq -|\bar{x}_{j_0}|\left[2\textstyle\sum_{k=1,k\neq j_0}^{n}|A_{kj_0}| + A_{j_0j_0}\right] + \rho\bar{u}_{j_0} + \tfrac{1}{2}t\bar{u}_{j_0} \\ &\geq \bar{u}_{j_0}\left[\tfrac{1}{2}t + \rho - 2\textstyle\sum_{k=1,k\neq j_0}^{n}|A_{kj_0}| - A_{j_0j_0}\right] \\ &> 0. \end{aligned}$$

- $J = \varnothing$ et $I \neq \varnothing$. Soit $i_0 \in I$. Prendre $x := \bar{x}$, $u_i := \bar{u}_i, \forall i \neq i_0$ et $u_{i_0} := 1$.

$$\begin{aligned} \phi_t(\bar{x}, \bar{u}) - \phi_t(x, u) &= \rho(\bar{u}_{i_0} - 1) + t\bar{u}_{i_0}(1 - \bar{u}_{i_0}) \\ &= (t\bar{u}_{i_0} - \rho)(1 - \bar{u}_{i_0}) \\ &\geq (\tfrac{1}{2}t - \rho)(1 - \bar{u}_{i_0}) \\ &> 0. \end{aligned}$$

□

6.3.3 Commentaire sur le paramètre de pénalité

Les valeurs $t_1(A,k)$ et $t_2(A,\rho)$ données précédemment, pour obtenir les reformulations équivalentes, représentent des valeurs majorantes des valeurs limites

$$t^k(A) := \inf\{t>0 : \quad (6.4) \Leftrightarrow (6.8)\}$$

et

$$t_\rho(A) := \inf\{t>0 : \quad (6.10) \Leftrightarrow (6.11)\}.$$

Les propositions 6.3.1.1 et 6.3.2.1 restent donc valides lorsque l'on remplace $t_1(A,k)$ par $t^k(A)$ et $t_2(A,\rho)$ par $t_\rho(A)$, respectivement. Cependant, il faut noter que ces valeurs limites $t^k(A)$ et $t_\rho(A)$ sont difficiles à calculer.
Sur le tableau 6.1 nous récapitulons les différentes reformulations.

TABLE 6.1 – Récapitulatif des différentes reformulations.

Problèmes ℓ_0	Reformulations
$\max_x \quad x^TAx$ $s.t. \quad x^Tx \le 1$ $\quad\quad \|x\|_0 \le k$	$\min_{(x,u)} \quad -x^TAx + tu^T(e-u)$ $s.t. \quad x^Tx \le 1,$ $\quad\quad |x_j| \le u_j, j=1,...,n,$ $\quad\quad e^Tu \le k,$ $\quad\quad u \in [0,1]^n$ pour $t > t_1(A,k)$
$\max_x \quad x^TAx$ $s.t. \quad x^Tx \le 1$ $\quad\quad \|x\|_0 \le k$	$\min_{(x,u)} \quad -x^TAx - tu^Tu$ $s.t. \quad x^Tx \le 1,$ $\quad\quad |x_j| \le u_j, j=1,...,n,$ $\quad\quad e^Tu = k,$ $\quad\quad u \in [0,1]^n$ pour $t > t_1(A,k)$.
$\max_x \quad x^TAx - \rho\|x\|_0$ $s.t. \quad x^Tx \le 1$	$\min_{(x,u)} \quad -x^TAx + \rho e^Tu + tq(u)$ $s.t. \quad x^Tx \le 1,$ $\quad\quad |x_j| \le u_j, j=1,...,n,$ $\quad\quad u \in [0,1]^n$ pour $t > t_2(A,\rho)$.

Nous venons de reformuler de manière équivalente les différents problèmes de parcimonie considérés en des problèmes de minimisation quadratique non convexe

sous contraintes convexes. Ces reformulations présentent des structures quadratiques simples et très adaptées à la Programmation DC et DCA. Dans ce qui suit, nous allons présenter l'algorithme DCA appliqué aux reformulations (6.9) et (6.11).

6.4 DCA appliqué aux reformulations

Rappelons qu'une fois le problème mis sous forme DC

$$\min \{G(x,u) - H(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{6.12}$$

où G et H sont des éléments de $\Gamma_0(\mathbb{R}^n \times \mathbb{R}^n)$, DCA consiste à calculer les deux suites primale et duale $\{(x^l, u^l)\}$ et $\{(X^l, U^l)\}$ définies par

$$(X^l, U^l) \in \partial H(x^l, u^l), (x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l). \tag{6.13}$$

De façon schématique

$$\begin{array}{lcl} (x^l, u^l) & \longrightarrow & (X^l, U^l) \in \partial H(x^l, u^l) \\ & \swarrow & \\ (x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l) & \longrightarrow & (X^{l+1}, U^{l+1}) \in \partial H(x^{l+1}, u^{l+1}). \end{array}$$

6.4.1 DCA pour la reformulation du problème avec contrainte ℓ_0

Une formulation DC de (6.9) est donnée par

$$\min \{G(x,u) - H(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{6.14}$$

où

$$G(x,u) := \chi_C(x,u); \quad H(x,u) := x^T A x + t u^T u;$$

$$C := \left\{(x,u) \in \mathbb{R}^n \times \mathbb{R}^n : \ x^T x \leq 1, \ |x_j| \leq u_j, j = 1, ..., n, \ e^T u = k, \ u \in [0,1]^n\right\}.$$

$$(X^l, U^l) \in \partial H(x^l, u^l) \Leftrightarrow X^l = 2Ax^l, U^l = 2tu^l, \tag{6.15}$$

et $\partial G^*(X^l, U^l)$ est l'ensemble des solutions du programme convexe suivant

$$\min \left\{-(X^l)^T x - (U^l)^T u : (x,u) \in C\right\} \tag{6.16}$$

dont une solution est calculée explicitement avec une complexité polynomiale $O(n^2)$ au pire des cas Thiao et al.[140].

Sur l'algorithme 2 nous résumons l'algorithme DCA appliqué à (6.9).

Algorithm 2 DCA pour (6.9)

Entrées : $A \in S_+^n$, $1 \leq k < n$, $t > 0$, $(x^0, u^0) \in \mathbb{R}^n \times \mathbb{R}_+^n$ et la tolérance ϵ.
Sortie : (x, u)
Initialisation $l := 0$.
Répéter

1. $X^l := 2Ax^l$, $U^l := 2tu^l$.
2. Calculer x^{l+1} et par u^{l+1}

$$(x^{l+1}, u^{l+1}) \in \arg\min \left\{ -(X^l)^T x - (U^l)^T u : (x, u) \in C \right\}.$$

Jusqu'à $\left| f_t^1(x^{l+1}, u^{l+1}) - f_t^1(x^l, u^l) \right| \leq \epsilon$.
$x := x^l$ et $u := u^l$.

Algorithm 3 DCA pour (6.11)

Entrées : $A \in S_+^n$, $\rho > 0$, $t > 0$, $(x^0, u^0) \in \mathbb{R}^n \times \mathbb{R}_+^n$ et la tolérance ϵ.
Sortie : (x, u)
Initialisation $l := 0$.
Répéter

1. $X^l = 2Ax^l$, $U^l = -\rho e + t(2u^l - e)$.
2. Calculer x^{l+1} et par u^{l+1}

$$(x^{l+1}, u^{l+1}) \in \arg\min \left\{ -(X^l)^T x - (U^l)^T u : (x, u) \in D \right\}.$$

Jusqu'à $\left| \phi_t(x^{l+1}, u^{l+1}) - \phi_t(x^l, u^l) \right| \leq \epsilon$.
$x := x^l$ et $u := u^l$.

6.4.2 DCA pour la reformulation du problème ℓ_0-régularisé

Une formulation DC de (6.11) est donnée par

$$\min \left\{ G(x, u) - H(x, u) : (x, u) \in \mathbb{R}^n \times \mathbb{R}^n \right\}, \tag{6.17}$$

où

$$G(x, u) := \chi_D(x, u); \quad H(x, u) := x^T Ax - \rho e^T u + t(u^T u - e^T u);$$

$$D := \left\{ (x, u) \in \mathbb{R}^n \times \mathbb{R}^n : \ x^T x \leq 1, \ |x_j| \leq u_j, j = 1, ..., n, \ u \in [0, 1]^n \right\}.$$

$$(X^l, U^l) \in \partial H(x^l, u^l) \Leftrightarrow X^l = 2Ax^l, U^l = -\rho e + t(2u^l - e), \tag{6.18}$$

et

$$(x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l) \Leftrightarrow (x^{l+1}, u^{l+1}) \in \arg\min \left\{ -(X^l)^T x - (U^l)^T u : (x, u) \in C \right\}.$$

Sur l'algorithme 3 nous résumons l'algorithme DCA appliqué à (6.11).

6.5 Expérimentation

Nous présentons dans ce paragraphe des expérimentions numériques préliminaires dans le contexte de l'Analyse en Composantes Principales (ACP) parcimonieuse.

6.5.1 ACP

Etant donné une matrice de covariance, A_0 symétrique semi-définie postive, modélisant l'information contenue sur les données, le but de l'ACP est d'extraire les r vecteurs propres dominants correspondants aux r valeurs propres dominantes de la matrice A_0. Chacun de ces vecteurs propres donne une composante principale et les r premiers vecteurs correspondent aux r premières composantes principales. Généralement, des techniques de déflation sont utilisées pour extraire séquentiellement les différents vecteurs propres [159, 131].
A la l-ième itération, nous extrayons un vecteur propre dominant de A_{l-1} en résolvant

$$x_l \in \arg\max \left\{ x^T A_{l-1} x : x^T x = 1 \right\},$$

$$A_l = A_{l-1} - x_l x_l^T A_{l-1} x_l x_l^T$$

et le $(l+1)$-ième vecteur propre dominant de A_0 étant le vecteur propre dominant de A_l.

6.5.2 ACP parcimonieuse

Le but ici est d'extraire les r premières composantes principales parcimonieuses. A la l-ième itération, nous extrayons une composante principale parcimonieuse de A_{l-1} en résolvant

$$x_l \in \arg\max \left\{ x^T A_{l-1} x : x^T x = 1, \|x\|_0 \leq k_l \right\}, \tag{6.19}$$

où A_{l-1} est une matrice de déflation et k_l est nombre entier positif controlant la parcimonie désirée. Nous utilisons ici une technique de déflation orthogonalisée (on pourra consulter les travaux de Mackey [105] pour les techniques de déflation pour l'ACP parcimonieuse)

$$A_0 = A,\ v_l = (I - V_{l-1} V_{l-1}^T) x_l / \left\| (I - V_{l-1} V_{l-1}^T) x_l \right\|,$$

$$A_l = (I - v_l v_l^T) A_{l-1} (I - v_l v_l^T),$$

où $v_1 = x_1$ et V_{l-1} la matrice dont les vecteurs colonnes sont les v_1, ..., v_{l-1}.

6.5.3 Tests numériques

Nous proposons dans cette section des tests numériques dans le cadre de l'ACP parcimonieuse dans laquelle l'algorithme 2 est utilisé pour résoudre le problème ℓ_0

TABLE 6.2 – Composantes trouvées et les variances capturées par les deux premières composantes sur les données artificielles.

	PC	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	VARIANCE
	1	0	0	0	0	.5	.5	.5	.5	0	0	40.9%
DCA	2	.5	.5	.5	.5	0	0	0	0	0	0	39.5%

(6.19), avec $A := A_{l-1}$ et $k := k_l$, pour l'extraction des différentes composantes principales. L'évaluation numérique porte sur les variances capturées et la parcimonie des composantes principales trouvées.

Dans la suite de la section, nous appellerons *cardinalité*, le nombre $\|x\|_0$ pour une composante principale x donnée.

Les données considérées sont des données benchmark utilisées pour tester les méthodes pour l'ACP parcimonieuse. Nous commençons d'abord par des simulations sur les données artificielles proposées par Zou [168], ensuite nous considérons les données pit props Jeffers [71] et nous terminons par des simulations sur les données colon tumor Alon et al.[5].

6.5.3.1 Données artificielles

Considérons l'exemple proposé par Zou [168] qui définit 3 facteurs V_1, V_2 et V_3,

$$V_1 \sim \mathcal{N}(0, 290),\ V_2 \sim \mathcal{N}(0, 300),\ V_3 = -0.3V_1 + 0.925V_2 + \epsilon,\ \epsilon \sim \mathcal{N}(0, 1),$$

avec V_1, V_2 et ϵ indépendants et ensuite génère 10 variables comme suit :

$$X_i = V_j + \epsilon_i^j,\ \epsilon_i^j \sim \mathcal{N}(0, 1),$$

avec $j = 1$ pour $i = 1, ..., 4$, $j = 2$ pour $i = 5, ..., 8$ et $j = 3$ pour $i = 9, 10$ et ϵ_i^j indépendantes pour $j = 1, 2, 3$, $i = 1, ..., 10$.

Les variances des trois facteurs sont 290, 300 et 283.8, respectivement. Le nombre de variables associées aux trois facteurs sont 4, 4 et 2. Les deux premières composantes principales expriment plus de 99% de la variance totale et donc nous ne considérons ici que ces deux premières composantes pour le calcul des variables parcimonieuses dérivées. Idéalement, la première variable dérivée devrait retrouver le facteur V_2 en utilisant seulement (X_5, X_6, X_7, X_8) et la deuxième variable dérivée devrait retrouver le facteur V_1 avec seulement (X_1, X_2, X_3, X_4).

En prenant $k = 4$ pour les deux premières composantes principales, notre algorithme DCA retrouve exactement les deux facteurs V_2 et V_1 et les résutats sont réprésentés dans le tableau 6.2.

6.5.3.2 Données pit props

Les données pit props ont été proposées par Jeffers [71] et permettent de tester les algorithmes pour l'ACP parcimonieuse. Les comparaisons sont effectuées sur

Table 6.3 – Pit Props : Résultats sur les trois composantes principales obtenues avec SPCA [168], DSPCA [34] et DC-PCA [135] et les six composantes obtenues avec notre algorithme DCA.

	PC	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}
	C_1	-.477	-.476	0	0	.177	0	-.250	-.344	-.416	-.400	0	0	0
SPCA	C_2	0	0	.785	.620	0	0	0	-.021	0	0	0	.013	0
	C_3	0	0	0	0	.640	.589	.492	0	0	0	0	0	-.015
	C_1	-.560	-.583	0	0	0	0	-.263	-.099	-.371	-.362	0	0	0
DSPCA	C_2	0	0	.707	.707	0	0	0	0	0	0	0	0	0
	C_3	0	0	0	0	0	-.793	-.610	0	0	0	0	0	.012
	C_1	.449	.459	0	0	0	0	.374	.332	.403	.419	0	0	0
DC-PCA	C_2	0	0	.707	.707	0	0	0	0	0	0	0	0	0
	C_3	0	0	0	0	0	.816	.578	0	0	0	0	0	0
	C_1	-.444	-.453	0	0	0	0	-.379	-.341	-.403	-.419	0	0	0
DCA	C_2	0	0	.707	.707	0	0	0	0	0	0	0	0	0
	C_3	0	0	0	0	.694	.721	0	0	0	0	0	0	0
	C_4	0	0	0	0	0	0	0	0	0	0	0	0	1
	C_5	0	0	0	0	0	0	0	0	0	0	1	0	0
	C_6	0	0	0	0	0	0	0	0	0	0	0	-1	0

Table 6.4 – Pits Props : variation de la variance capturée suivant k pour la première composante principale.

k	1	2	3	4	5	6	7	8	9	10	11	12	13
Var.%	7.69	15.03	19.04	22.56	26.20	29.00	30.74	31.30	31.83	32.10	32.37	32.44	32.45

les six premières composantes principales qui capturent 87% de la variance totale. Comme il a été montré dans Sriperumbudur et al.[135] que DC-PCA donne de meilleurs résultats, sur ces données, que DSPCA et SPCA avec une cardinalité (cumulée) de 13 et une variance (cumulée) de 77.1% de la variance totale, nous allons l'utiliser pour les comparaisons. Le tableau 6.3 montre les trois premières composantes principales obtenues avec les algorithmes SPCA, DSPCA, DC-PCA et les six premières composantes principales données par l'algorithme 2. Avec $(k_1, k_2, k_3, k_4, k_5, k_6) := (6, 2, 2, 1, 1, 1)$, notre algorithme donne la même cardinalité cumulée (13) et capture presque la même variance (77.05%). Nous observons aussi que toutes les composantes principales $C_1, ..., C_6$ obtenues satisfont la propriété d'orthonormalité ($C_i^T C_i = 1$ et $C_i^T C_j = 0, \forall i \neq j$) ce qui n'est pas le cas de DC-PCA. Une autre avantage est que dans notre algorithme les cardinalités désirées sont explicitement mentionnées et que avec DC-PCA, il est difficile de choisir le paramètre de régularisation pour atteindre une cardinalité donnée. Le tableau 6.4 montre la variance capturée par la première composante en fonction de k.

6.5.3.3 Données colon cancer

Les données colon tumor sont des données micro-array proposées par Alon et al.[5]. Nous considérons les cinq premières composantes principales qui capturent 70% de la variance totale. La figure 6.1 montre que notre algorithme DCA donne de loin de meilleurs résultats que SPCA et DC-PCA. DC-PCA capture seulement 62% de variance cumulée avec plus de 6000 pour la cardinalité cumulée, tandis que notre algorithme DCA capture 65.94% de variance cumulée avec seulement 5000 de cardinalité cumulée avec $(k_1, k_2, k_3, k_4, k_5) := (1800, 800, 800, 800, 800)$. Dans le

TABLE 6.5 – Colon Cancer : variation de la variance capturée par la première composante principale suivant k.

k	200	400	600	800	1000	1200	1400	1600	1800	2000
VAR.%	7.70	14.35	20.22	25.51	30.25	34.41	38.11	41.30	43.76	44.96

tableau 6.5, nous représentons la variance capturée par la première composante principale en fonction de k.

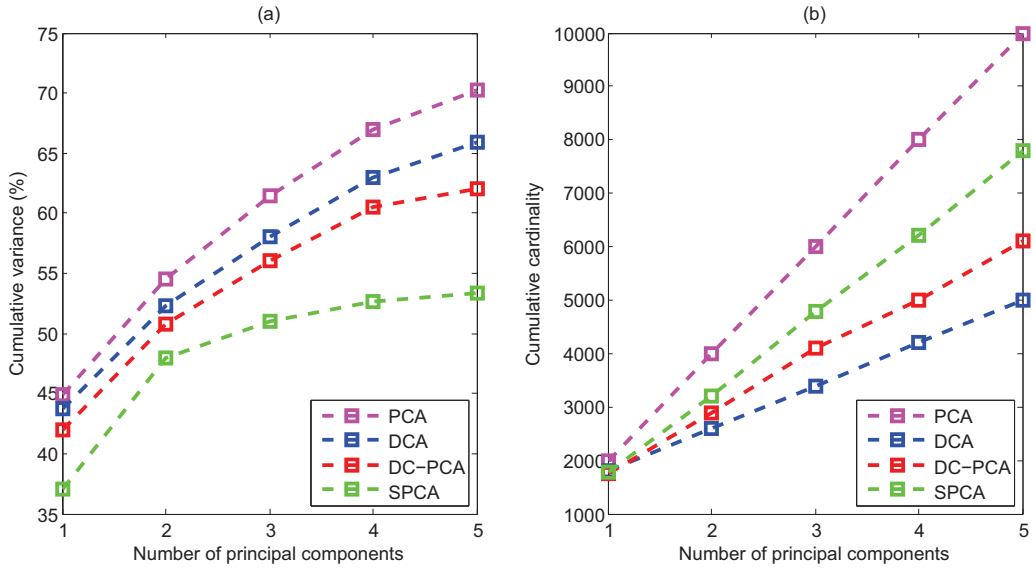

FIGURE 6.1 – Colon Cancer (a) variance cumulée (b) cardinalité cumulée pour les 5 premières composantes principales.

6.6 Conclusion

Ce présent chapitre nous a permis d'introduire une nouvelle approche concernant la modélisation parcimonieuse pour le problème de la valeur propre maximale. Nous avons établi des reformulations DC équivalentes de ces problèmes de parcimonie grâce à de nouvelles techniques de pénalité exacte en Programmation DC. Ces

reformulations sont d'autant plus originales sachant que les méthodes approchées utilisées afin de contourner la discontinuité du terme de parcimonie ne sont pas exactes et manquent de liens rigoureux avec les problèmes originaux la plupart du temps. Au point de vue numériques, les résultats préliminaires obtenus avec DCA illustrent bien le potentiel de cette nouvelle approche dont les résultats préliminaires dans le contexte de l'analyse en composantes sont très intéressants et très prometteurs. Ces résultats ont été présentés et publiés à la plus prestigieuse conférence internationale en Machine Learning ICML 2010.

Chapitre 7

Modélisation parcimonieuse dans les modéles de régression linéaire

Nous considérerons dans ce chapitre différents types de problèmes de parcimonie rencontrés dans les modèles de régression linéaire. Nous discuterons dans un premier temps de liens entre ces différents problèmes et ensuite nous proposerons différentes reformulations et montrerons l'équivalence avec ces problèmes de parcimonie. Nous présenterons aussi des extensions de ces reformulations. Nous détaillerons l'algorithme DCA appliqué à ces reformulations et présenterons un algorithme combiné DCA-SE (Séparation et Evaluation) pour l'optimisation globale. Enfin, nous terminerons par des simulations numériques préliminaires.

7.1 Introduction

Étant donné une matrice $A \in \mathbb{M}_{m,n}(\mathbb{R})$ représentant un ensemble de m données mesurées et leur réponse $b \in \mathbb{R}^m$, nous distinguons généralement, dans les modèles linéaires, les classes de problèmes de parcimonie suivantes

$$(P) \quad \begin{cases} \min_x & \|x\|_0 \\ s.t. & Ax = b, \\ & x \in \mathbb{R}^n, \end{cases}$$

$$(P^\lambda) \quad \begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n, \end{cases}$$

et

$$(Q^k) \quad \begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 \\ s.t. & \|x\|_0 \leq k \\ & x \in \mathbb{R}^n, \end{cases}$$

λ étant un réel strictement positif et k est un entier positif compris entre 1 et $n-1$.

Le premier problème détermine la solution la plus creuse du système linéaire $Ax = b$ tandis que dans le deuxième, le paramètre $\lambda > 0$, appelé paramètre de régularisation, permet de faire un compromis entre le critère d'erreur quadratique et la parcimonie des solutions et la dernière formulation permet de minimiser le critère

d'erreur quadratique tout en contrôlant la parcimonie des solutions par k.

Les problèmes ℓ_0 précédents sont discontinus et NP-difficiles pour des données générales A, b, λ, et k à cause du terme $\|.\|_0$ et alors difficiles à prendre en main directement. Généralement, des approximations continues plus douces du terme ℓ_0 sont utilisées pour contourner cette discontinuité. Outre les résultats sur la minimisation ℓ_1 ([39, 38, 56, 20, 21, 19]) qui montrent que, dans certains cas très spéciaux de données, la minimisation ℓ_1 permet de retrouver une solution du problème ℓ_0 original, les approximations proposées ne sont pas exactes et manquent très souvent de liens rigoureux entre le problème approché et le problème original. Dans ce travail nous proposons des reformulations DC exactes de ces différents problèmes ℓ_0 par de nouvelles techniques développées durant cette thèse. Ces techniques nous ont permis d'établir des propriétés très intéressantes et des reformulations quadratiques concernant les problèmes ℓ_0 précédents.

Dans le paragraphe 2, nous discuterons d'abord de liens entre ces différents problèmes ℓ_0. Des reformulations DC et quadratiques pour ces problèmes ℓ_0 ainsi que quelques extensions seront proposées dans les paragraphes 3, 4 et 5. Dans les paragraphes 6 et 7, nous détaillerons l'algorithme DCA appliqué aux différentes reformulations et nous présenterons un algorithme combiné DCA-SE pour l'optimisation globale. Et nous terminerons par quelques tests numériques préliminaires.

7.2 Liens entre (P), (P^λ) et (Q^k)

Existence de solution : Par un procédé énumératif fini, il est facile de voir que les problèmes (P^λ) et (Q^k) admettent toujours au moins une solution et que le problème (P) admet au moins une solution dès lors que le système $Ax = b$ est admissible.

Bien que ce procédé permet d'établir l'existence de solution et permet d'en trouver une, il n'est pas utilisable en pratique pour des problèmes avec des dimensions concrètes du fait de sa complexité exponentielle. D'où la nécessité de rechercher des méthodes de résolutions plus efficaces.

Liens : La structure du problème (P^λ), dont le critère est la somme pondérée des deux termes $\frac{1}{2}\|Ax-b\|_2^2$ et $\|x\|_0$, permet d'énoncer une propriété très importante sur le comportement de chaque terme en une solution lorsque λ varie. Intuitivement, si λ décroît, on attache plus d'importance à $\frac{1}{2}\|Ax-b\|_2^2$ et il semble normal que, si x^λ est une solution de (P^λ), $\frac{1}{2}\left\|Ax^\lambda-b\right\|_2^2$ décroisse et x^λ devient une bonne approximation d'une solution de (P). La proposition suivante énonce cela de façon rigoureuse, lorsque l'ensemble admissible M de (P), défini par

$$M = \{x \in \mathbb{R}^n : \quad Ax = b\},$$

est non vide. Elle donne aussi un lien entre (P) et (Q^k).

Proposition 7.2.1 *Si M est non vide, alors*

1. *il existe $\lambda_0 > 0$, tel que les problèmes (P) et (P^λ) admettent les mêmes solutions, pour tout λ, $0 < \lambda < \lambda_0$,*
2. *il existe k_0, tel que les problèmes (P) et (Q^{k_0}) admettent les mêmes solutions.*

Preuve Soit $\bar{x}$ une solution de (P).

1. Pour $\lambda > 0$, soit x^λ une solution de (P^λ).
   On a
   $$\frac{1}{2}\left\|Ax^{\lambda_1} - b\right\|_2^2 + \lambda_1 \left\|x^{\lambda_1}\right\|_0 \leq \frac{1}{2}\left\|Ax^{\lambda_2} - b\right\|_2^2 + \lambda_1 \left\|x^{\lambda_2}\right\|_0, \tag{7.1}$$
   et
   $$\frac{1}{2}\left\|Ax^{\lambda_2} - b\right\|_2^2 + \lambda_2 \left\|x^{\lambda_2}\right\|_0 \leq \frac{1}{2}\left\|Ax^{\lambda_1} - b\right\|_2^2 + \lambda_2 \left\|x^{\lambda_1}\right\|_0 \tag{7.2}$$
   pour tout $\lambda_1 > 0$, $\lambda_2 > 0$.
   En sommant ces deux inégalités nous obtenons
   $$(\lambda_1 - \lambda_2)\left(\left\|x^{\lambda_1}\right\|_0 - \left\|x^{\lambda_2}\right\|_0\right) \leq 0, \tag{7.3}$$
   pour tout $\lambda_1 > 0$, $\lambda_2 > 0$. Donc la suite $\left\{\left\|x^\lambda\right\|_0\right\}_{\lambda \downarrow 0}$ croît lorsque λ décroît. De plus elle prend un nombre fini de valeurs 0 ou 1,..., ou n, donc converge lorsque λ décroît vers 0 et donc stationnaire à partir d'un certain rang, i.e., il existe $\lambda_0 > 0$, $p \in \{0, 1, ..., n\}$, tel que
   $$\left\|x^\lambda\right\|_0 = p, \ \forall \lambda, \ 0 < \lambda < \lambda_0. \tag{7.4}$$
   Pour $0 < \lambda_1 < \lambda_0, 0 < \lambda_2 < \lambda_0$, les deux premières inégalités deviennent alors
   $$\frac{1}{2}\left\|Ax^{\lambda_1} - b\right\|_2^2 \leq \frac{1}{2}\left\|Ax^{\lambda_2} - b\right\|_2^2,$$
   $$\frac{1}{2}\left\|Ax^{\lambda_2} - b\right\|_2^2 \leq \frac{1}{2}\left\|Ax^{\lambda_1} - b\right\|_2^2,$$
   c'est à dire
   $$\left\|Ax^{\lambda_1} - b\right\|_2 = \left\|Ax^{\lambda_2} - b\right\|_2, \ \forall \lambda_1, \lambda_2, \ 0 < \lambda_1 < \lambda_0, 0 < \lambda_2 < \lambda_0.$$
   Donc il existe $\beta \geq 0$, tel que
   $$\left\|Ax^\lambda - b\right\|_2 = \beta, \ \forall \lambda, \ 0 < \lambda < \lambda_0. \tag{7.5}$$
   Comme $\bar{x}$ est admissible pour (P^λ), on a
   $$\frac{1}{2}\left\|Ax^\lambda - b\right\|_2^2 + \lambda \left\|x^\lambda\right\|_0 \leq \lambda \left\|\bar{x}\right\|_0, \forall \lambda > 0, \tag{7.6}$$
   et donc
   $$\frac{1}{2}\left\|Ax^\lambda - b\right\|_2^2 \leq \lambda\left(\left\|\bar{x}\right\|_0 - \left\|x^\lambda\right\|_0\right), \forall \lambda > 0.$$

Ainsi

$$\lim_{\lambda \downarrow 0} \left\| Ax^\lambda - b \right\|_2^2 = 0$$

et en considérant (7.5), nous obtenons

$$\left\| Ax^\lambda - b \right\|_2 = \beta = 0, \forall \lambda, 0 < \lambda < \lambda_0. \tag{7.7}$$

(7.6) et (7.7) donnent x^λ admissible pour (P) et $\left\|x^\lambda\right\|_0 \leq \|\bar{x}\|_0$, pour tout λ, $0 < \lambda < \lambda_0$. Donc x^λ solution de (P) et $\bar{x}$ solution de (P^λ), $\forall \lambda, 0 < \lambda < \lambda_0$.
Pour tout $\lambda, 0 < \lambda < \lambda_0$, nous venons de démontrer que
– $Ax^\lambda = b$ et x^λ est une solution de (P),
– toute solution de (P) est aussi solution de (P^λ).
Il reste donc à démontrer que toute solution de (P^λ) est aussi solution de (P).
Soit λ, $0 < \lambda < \lambda_0$ fixé et soit $\tilde{x}^\lambda$ une solution de (P^λ).
Comme $\tilde{x}^\lambda$ et x^λ sont des solutions de (P^λ), on a

$$\frac{1}{2} \left\| A\tilde{x}^\lambda - b \right\|_2^2 + \lambda \left\| \tilde{x}^\lambda \right\|_0 = \lambda \left\| x^\lambda \right\|_0$$

et

$$\left\| \tilde{x}^\lambda \right\|_0 \geq \left\| x^{\frac{\lambda+\lambda_0}{2}} \right\|_0 = \left\| x^\lambda \right\|_0 .$$

Ainsi $A\tilde{x}^\lambda = b$ et $\left\|\tilde{x}^\lambda\right\|_0 = \left\|x^\lambda\right\|_0$. Donc $\tilde{x}^\lambda$ est une solution de (P).

2. Il suffit de prendre $k_0 = \|\bar{x}\|_0$. □

La proposition précédente montre que toute méthode permettant de résoudre efficacement (P^λ) ou (Q^k) devrait permettre aussi la résolution efficace de (P). Ces méthodes de résolution utilisent très souvent des outils d'optimisation continue d'où la nécessité de techniques de reformulations continues adaptées aux problèmes ℓ_0.

7.3 Reformulations DC

Dans ce paragraphe, nous allons proposer des reformulations DC au problème de parcimonie

$$(P^\lambda) \quad \left\{ \begin{array}{ll} \min_x & \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n, \end{array} \right.$$

ainsi qu'à sa version avec contrainte

$$(Q^k) \left\{ \begin{array}{ll} \min_x & \frac{1}{2} \|Ax - b\|_2^2 \\ s.t. & \|x\|_0 \leq k. \end{array} \right.$$

De façon à démontrer l'équivalence avec les reformulations DC proposées, nous passons par quelques notions en programmation DC. Pour cela, nous rappelons qu'un programme DC est un problème d'optimisation de la forme

$$(P_{dc}) \;\; \alpha = \inf \left\{ f(z) := g(z) - h(z) : z \in \mathbb{R}^p \right\},$$

avec $g, h \in \Gamma_0(\mathbb{R}^p)$ et que les conditions KKT généralisées sont données par

$$\partial h(z) \cap \partial g(z) \neq \varnothing.$$

7.3.1 Problème moindres carrés ℓ_0-régularisé (P^λ)

Rappelons que (P^λ) est donné par

$$(P^\lambda) \quad \begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n. \end{cases}$$

Pour $\tau > 0$, nous considérons le problème suivant

$$(P_\tau^\lambda) \begin{cases} \min & \phi_\tau(x,u) := \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau |x|^T (e-u) \\ s.t. & x \in \mathbb{R}^n, u \in [0,1]^n. \end{cases}$$

Il est clair que (P_τ^λ) est non convexe à cause du terme $|x|^T (e-u)$. Plus précisément, c'est un programme DC avec la reformulation DC suivante

$$\min \{G(x,u) - H(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{7.8}$$

où

$$G(x,u) := \frac{1}{2}(\tau+\beta)x^T x + \frac{1}{2}\tau u^T u + \tau e^T |x| + \lambda e^T u + \chi_{[0,1]^n}(u), \tag{7.9}$$

$$H(x,u) := \frac{1}{2}\tau \sum_{i=1}^n (|x_i| + |u_i|)^2 + \frac{1}{2}\beta x^T x - \frac{1}{2}\|Ax-b\|_2^2, \tag{7.10}$$

$\beta \geq \lambda_{\max}(A^T A)$.
Nous allons montrer dans ce qui suit que (P^λ) et (P_τ^λ) sont équivalents pour des valeurs de τ bien choisies. Nous aurons besoin en premier de calculer les sous-différentiels $\partial G(x,u)$ et $\partial H(x,u)$.

$$(X,U) \in \partial G(x,u) \Leftrightarrow X \in (\tau+\beta)x + \tau \partial(e^T|x|), U \in \tau u + \lambda e + \partial \chi_{[0,1]^n}(u), \tag{7.11}$$

$$\left[\partial(e^T|x|)\right]_i = \begin{cases} \{1\} & \text{if } x_i > 0, \\ \{-1\} & \text{if } x_i < 0, \\ [-1,1] & \text{if } x_i = 0, \end{cases} \quad i = 1,...,n.$$

$$\partial \chi_{[0,1]^n}(u) = \{\gamma - \delta : \gamma_i(u_i - 1) = 0, \delta_i u_i = 0, \gamma_i \geq 0, \delta_i \geq 0, i = 1,...,n\}.$$

$$(X,U) \in \partial H(x,u) \Leftrightarrow X = \beta x - A^T(Ax-b) + \tau r, U = \tau s, \tag{7.12}$$

$$\begin{pmatrix} r_i \\ s_i \end{pmatrix} \in \begin{cases} \left\{ \begin{pmatrix} x_i + u_i \\ x_i + u_i \end{pmatrix} \right\} & \text{if } x_i u_i > 0, \\ \left\{ \begin{pmatrix} x_i - u_i \\ -x_i + u_i \end{pmatrix} \right\} & \text{if } x_i u_i < 0, \quad i = 1,...,n. \\ \left[\begin{pmatrix} x_i + u_i \\ x_i + u_i \end{pmatrix}, \begin{pmatrix} x_i - u_i \\ -x_i + u_i \end{pmatrix} \right] & \text{if } x_i u_i = 0, \end{cases}$$

Nous aurons besoin aussi du lemme suivant qui donne quelques propriétés des solutions de (P_τ^λ).

Lemme 7.3.1.1 *Soit $\tau \geq \|A\|_1 \|b\|_2$. Alors toute solution $(\bar{x}, \bar{u})$ de (P_τ^λ) vérifie $\phi_\tau(\bar{x}, \bar{u}) = \frac{1}{2}\|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0$.*

Preuve Soit $\tau \geq \|A\|_1 \|b\|_2$ et soit $(\bar{x}, \bar{u})$ une solution de (P_τ^λ). Posons

$$F := \{i : \bar{x}_i \neq 0, \bar{u}_i = 0\}, \; G := \{i : \bar{x}_i = 0, \bar{u}_i > 0\} \text{ et } I := \{i : \bar{x}_i(1 - \bar{u}_i) \neq 0\}.$$

$(\bar{x}, \bar{u})$ vérifie nécessairement les conditions KKT généralisées appliquées à (7.8)

$$\partial H(x, u) \cap \partial G(x, u) \neq \varnothing. \tag{7.13}$$

Nous allons montrer d'abord que $F = \varnothing$ et $G = \varnothing$. Nous procédons par contradiction.
Supposons $F \neq \varnothing$ et soit $i_0 \in F$. On a $\bar{x}_{i_0} \neq 0$ et $\bar{u}_{i_0} = 0$. Donc des conditions KKT conditions (7.13), nous tirons

$$\left[A^T(A\bar{x} - b)\right]_{i_0} + \tau sign(\bar{x}_{i_0}) = 0. \tag{7.14}$$

Comme $(0, 0)$ est admissible pour (P_τ^λ), on a $\phi_\tau(\bar{x}, \bar{u}) \leq \frac{1}{2}\|b\|_2^2$ et donc

$$\|A\bar{x} - b\|_2 < \|b\|_2. \tag{7.15}$$

On a

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| \leq \left\|A^T(A\bar{x} - b)\right\|_\infty \leq \left\|A^T\right\|_\infty \|A\bar{x} - b\|_\infty \leq \|A\|_1 \|A\bar{x} - b\|_2,$$

donc en combinant avec (7.15), nous obtenons finalement

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| < \|A\|_1 \|b\|_2 \leq \tau.$$

Ce qui contredit (7.14) et donc $F = \varnothing$.
Supposons $G \neq \varnothing$ et soit $i_0 \in G$. On a $\bar{x}_{i_0} = 0$ et $\bar{u}_{i_0} > 0$. D'après les conditions KKT (7.13), il existe $\gamma_{i_0} \geq 0$ tel que

$$\lambda + \gamma_{i_0} = 0.$$

Ce qui contredit $\lambda > 0$, donc $G = \varnothing$.
Maintenant considérons les cas suivants : $I = \varnothing$ et $I \neq \varnothing$.

- $I = \varnothing$: comme $G = \varnothing$, on a $e^T\bar{u} = \|\bar{x}\|_0$ et $|\bar{x}|^T(e - \bar{u}) = 0$, donc

$$\phi_\tau(\bar{x}, \bar{u}) = \frac{1}{2}\|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0.$$

- $I \neq \varnothing$: soit $i_0 \in I$. Comme $F = \varnothing$, on a $0 < \bar{u}_{i_0} < 1$ et $\bar{x}_{i_0} \neq 0$. Des conditions KKT (7.13), nous tirons $|\bar{x}_{i_0}| = \frac{\lambda}{\tau}$ et donc

$$|\bar{x}_i| = \frac{\lambda}{\tau}, \forall i \in I.$$

$$\begin{aligned}\phi_\tau(\bar{x},\bar{u}) &= \tfrac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda e^T\bar{u} + \tau\,|\bar{x}|^T(e-\bar{u})\\ &= \tfrac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda e^T\bar{u} + \tau\textstyle\sum_{i\in I}|\bar{x}_i|\,(1-\bar{u}_i)\\ &= \tfrac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda e^T\bar{u} + \lambda\textstyle\sum_{i\in I}(1-\bar{u}_i)\\ &= \tfrac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda(\textstyle\sum_{i\notin I}\bar{u}_i + \sum_{i\in I}1).\end{aligned}$$

Comme $G=\varnothing$, nous obtenons finalement

$$\phi_\tau(\bar{x},\bar{u}) = \frac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda\|\bar{x}\|_0. \qquad \square$$

Nous sommes maintenant en mesure d'établir l'équivalence entre les problèmes (P^λ) et (P^λ_τ).

Proposition 7.3.1.2 *Soit $\tau \geq \tau_1(A,b,\lambda) := \|A\|_1\|b\|_2$. Alors (P^λ) et (P^λ_τ) sont équivalents dans le sens :*

- *si $\bar{x}$ est une solution de (P^λ), alors il existe $\bar{u}$ tel que $(\bar{x},\bar{u})$ est une solution de (P^λ_τ).*
- *si $(\bar{x},\bar{u})$ est une solution de (P^λ_τ), alors $\bar{x}$ est une solution de (P^λ).*

Preuve Soit $(\bar{x},\bar{u})$ une solution de (P^λ_τ) et soit $\tilde{x}$ une solution de (P^λ). Posons $\tilde{u}_i := 1$ si $\tilde{x}_i \neq 0$ et $\tilde{u}_i := 0$ sinon. D'après le lemme 7.3.1.1, on a

$$\phi_\tau(\bar{x},\bar{u}) = \frac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda\|\bar{x}\|_0$$

et donc

$$\phi_\tau(\tilde{x},\tilde{u}) = \frac{1}{2}\|A\tilde{x}-b\|_2^2 + \lambda\|\tilde{x}\|_0 \leq \frac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda\|\bar{x}\|_0 = \phi_\tau(\bar{x},\bar{u}) \leq \phi_\tau(\tilde{x},\tilde{u}).$$

Parsuite

$$\phi_\tau(\tilde{x},\tilde{u}) = \frac{1}{2}\|A\tilde{x}-b\|_2^2 + \lambda\|\tilde{x}\|_0 = \frac{1}{2}\|A\bar{x}-b\|_2^2 + \lambda\|\bar{x}\|_0 = \phi_\tau(\bar{x},\bar{u}),$$

et alors $\bar{x}$ solution de (P^λ) et $(\tilde{x},\tilde{u})$ solution de (P^λ_τ). $\square$

7.3.2 Problème moindres carrés avec contrainte ℓ_0 (Q^k)

Rappelons que (Q^k) est donné par

$$(Q^k)\begin{cases}\min_x & \frac{1}{2}\|Ax-b\|_2^2\\ s.t. & \|x\|_0 \leq k.\end{cases}$$

La démarche est analogue à la section précédente. Pour $t>0$, nous considérons dans ce cas le problème non convexe

$$(Q^k_t)\begin{cases}\min_{(x,u)} & \psi_t(x,u) := \frac{1}{2}\|Ax-b\|_2^2 + t\,|x|^T(e-u)\\ s.t. & e^Tu \leq k,\\ & x\in\mathbb{R}^n,\\ & u\in[0,1]^n.\end{cases}$$

C'est un programme DC avec la formulation DC suivante

$$\min\left\{G(x,u)-H(x,u):(x,u)\in\mathbb{R}^n\times\mathbb{R}^n\right\}, \tag{7.16}$$

où

$$G(x,u):=\frac{1}{2}(t+\beta)x^Tx+\frac{1}{2}tu^Tu+te^T|x|+\chi_C(u), \tag{7.17}$$

$$H(x,u):=\frac{1}{2}t\sum_{i=1}^{n}(|x_i|+|u_i|)^2+\frac{1}{2}\beta x^Tx-\frac{1}{2}\|Ax-b\|_2^2, \tag{7.18}$$

$$C:=\left\{u\in[0,1]^n:\ e^Tu\leq k\right\}\ \text{et}\ \beta\geq\lambda_{\max}(A^TA).$$

Comme précédemment, nous allons déterminer les valeurs de t pour lesquelles (Q^k) et (Q_t^k) sont équivalents. Nous commençons d'abord par le calcul des sous-différentiels ∂G et ∂H.

$$(X,U)\in\partial G(x,u)\Leftrightarrow X\in(t+\beta)x+t\partial(e^T|x|),U\in tu+\partial\chi_C(u), \tag{7.19}$$

$$\left[\partial(e^T|x|)\right]_i=\begin{cases}\{1\} & \text{if } x_i>0,\\ \{-1\} & \text{if } x_i<0,\\ [-1,1] & \text{if } x_i=0,\end{cases}\quad i=1,...,n.$$

$$\partial\chi_C(u)=\left\{\gamma-\delta+\eta e:\ \eta(e^Tu-k)=0,\gamma_i(u_i-1)=0,\delta_iu_i=0,\eta\geq0,\gamma_i\geq0,\delta_i\geq0,\forall i\right\}.$$

$$(X,U)\in\partial H(x,u)\Leftrightarrow X=\beta x-A^T(Ax-b)+tr,U=ts, \tag{7.20}$$

$$\begin{pmatrix}r_i\\ s_i\end{pmatrix}\in\begin{cases}\left\{\begin{pmatrix}x_i+u_i\\ x_i+u_i\end{pmatrix}\right\} & \text{if } x_iu_i>0,\\ \left\{\begin{pmatrix}x_i-u_i\\ -x_i+u_i\end{pmatrix}\right\} & \text{if } x_iu_i<0,\\ \left[\begin{pmatrix}x_i+u_i\\ x_i+u_i\end{pmatrix},\begin{pmatrix}x_i-u_i\\ -x_i+u_i\end{pmatrix}\right] & \text{if } x_iu_i=0,\end{cases}\quad i=1,...,n.$$

Le résultat préliminaire suivant donne quelques propriétés des solutions de (Q_t^k).

Lemme 7.3.2.1 *Soit $t\geq\|A\|_1\|b\|_2$. Si $(\bar{x},\bar{u})$ est une solution de (Q_t^k), alors $|\bar{x}|^T(e-\bar{u})=0$ et $\|\bar{x}\|_0\leq k$.*

Preuve Soit $t\geq\|A\|_1\|b\|_2$ et soit $(\bar{x},\bar{u})$ une solution de (Q_t^k). Posons

$$F:=\{i:\bar{x}_i\neq0,\bar{u}_i=0\},\ G:=\{i:\bar{x}_i=0,\bar{u}_i>0\}\ \text{et}\ I:=\{i:\bar{x}_i(1-\bar{u}_i)\neq0\}.$$

$(\bar{x},\bar{u})$ vérifie les conditions KKT généralisées appliquées à (7.16)

$$\partial H(x,u)\cap\partial G(x,u)\neq\varnothing. \tag{7.21}$$

Nous allons d'abord montrer que $F = \varnothing$. Nous procédons par contradiction.
Supposons $F \neq \varnothing$ et soit $i_0 \in F$. On a $\bar{x}_{i_0} \neq 0$ et $\bar{u}_{i_0} = 0$. Des conditions KKT généralisées (7.21), nous tirons

$$\left[A^T(A\bar{x} - b)\right]_{i_0} + tsign(\bar{x}_{i_0}) = 0. \tag{7.22}$$

Comme $(0, 0)$ est admissible pour (Q_τ^k), on a $\psi_\tau(\bar{x}, \bar{u}) \leq \frac{1}{2}\|b\|_2^2$ et donc

$$\|A\bar{x} - b\|_2 < \|b\|_2 . \tag{7.23}$$

On a

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| \leq \|A\|_1 \|A\bar{x} - b\|_2$$

donc en combinant avec (7.23), nous obtenons

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| < \|A\|_1 \|b\|_2 \leq t.$$

Ce qui contredit (7.22) et donc $F = \varnothing$.
Nous allons montrer maintenant que $I \neq \varnothing$. Nous procédons par contradiction. Supposons $I \neq \varnothing$.
Soit $i_0 \in I$, comme $F = \varnothing$ on a $\bar{x}_{i_0} \neq 0$ et $0 < \bar{u}_{i_0} < 1$. Les conditions KKT (7.21) donnent alors

$$|\bar{x}_{i_0}| = \frac{\eta}{t},\ \eta > 0,\ e^T\bar{u} = k,$$

et donc

$$|\bar{x}_i| = \frac{\eta}{t},\ \forall i \in I. \tag{7.24}$$

Considérons maintenant les cas $G \neq \varnothing$ et $G = \varnothing$.

- $G \neq \varnothing$. Soit $g \in G$ i.e. $\bar{x}_g = 0$ et $\bar{u}_g > 0$. Alors les conditions KKT (7.21) donnent

  $$\eta + \gamma_g = 0 \ \text{ et } \ \gamma_g \geq 0.$$

  Ce qui contredit $\eta > 0$.
- $G = \varnothing$. De $e^T\bar{u} = k$, nous tirons

  $$\sum_{i \in I} \bar{u}_i = k - \sum_{i \notin I} \bar{u}_i.$$

  Comme $G = \varnothing$, on a $\sum_{i \notin I} \bar{u}_i \in \mathbb{N}$ et donc $\sum_{i \in I} \bar{u}_i \in \mathbb{N}$. De plus, $\sum_{i \in I} \bar{u}_i > 0$ $(F = \varnothing)$, ainsi

  $$\sum_{i \in I} \bar{u}_i \in \{1, ..., k\} .$$

  On pose $c := \sum_{i \in I} \bar{u}_i$. Il est clair que $c \in \{1, ..., k\}$ et $c < Card(I)$.
  Soit $\{i_1, i_2, ..., i_c\} \subset I$ et soit $\tilde{u}$ le point définit par

  $$\tilde{u}_i := \begin{cases} \bar{u}_i & \forall i \notin I, \\ 1 & \forall i \in \{i_1, i_2, ..., i_c\}, \\ 0 & \text{ailleurs.} \end{cases}$$

On a $\tilde{u} \in [0,1]^n$, $e^T\tilde{u} = k$ et

$$t\,|\bar{x}|^T(e-\bar{u}) = t\sum_{i\in I}|\bar{x}_i|\,(1-\bar{u}_i) = \eta\sum_{i\in I}(1-\bar{u}_i) = \eta\sum_{i\in I}(1-\tilde{u}_i) = t\,|\bar{x}|^T(e-\tilde{u}). \tag{7.25}$$

Donc $(\bar{x},\tilde{u})$ est une solution de (Q^k_τ) et alors vérifie les conditions KKT (7.21). Soit $r \in I \setminus \{i_1,i_2,...,i_c\}$, on a $\bar{x}_r \neq 0$ et $\tilde{u}_r = 0$. Ce qui donne avec les conditions KKT (7.21)

$$\left[A^T(A\bar{x}-b)\right]_r + tsign(\bar{x}_r) = 0.$$

Cette dernière condition est similaire à la condition (7.22) et donc contredit $t \geq \|A\|_1\,\|b\|_2$.

Parsuite $I = \varnothing$ et $\|\bar{x}\|_0 \leq k$. Ce qui termine la preuve. □

Nous pouvons maintenant établir le lien entre (Q^k) et (Q^k_t).

Proposition 7.3.2.2 *Soit $t \geq t_1(A,b,k) := \|A\|_1\,\|b\|_2$. Alors (Q^k_t) est équivalent à (Q^k) dans le sens :*

- *si $\bar{x}$ est une solution de (Q^k), alors il existe $\bar{u}$ tel que $(\bar{x},\bar{u})$ est une solution de (Q^k_t).*
- *si $(\bar{x},\bar{u})$ est une solution de (Q^k_t), alors $\bar{x}$ est une solution de (Q^k).*

Preuve Soit $(\bar{x},\bar{u})$ une solution de (Q^k_t) et soit $\tilde{x}$ une solution de (Q^k). Posons $\tilde{u}_i := 1$ si $\tilde{x}_i \neq 0$ et $\tilde{u}_i := 0$ sinon. On a $e^T\tilde{u} \leq k$ et d'après le lemme 7.3.2.1 $|\bar{x}|^T(e-\bar{u}) = 0$ et $\|\bar{x}\|_0 \leq k$. Ainsi

$$\psi_t(\tilde{x},\tilde{u}) = \frac{1}{2}\|A\tilde{x}-b\|_2^2 \leq \frac{1}{2}\|A\bar{x}-b\|_2^2 = \psi_t(\bar{x},\bar{u}) \leq \psi_t(\tilde{x},\tilde{u}),$$

et donc

$$\psi_t(\tilde{x},\tilde{u}) = \frac{1}{2}\|A\tilde{x}-b\|_2^2 = \frac{1}{2}\|A\bar{x}-b\|_2^2 = \psi_t(\bar{x},\bar{u}).$$

$\bar{x}$ solution de (Q^k) et $(\tilde{x},\tilde{u})$ solution de (Q^k_t). □

Les reformulations données par les propositions 7.3.1.2 et 7.3.2.2 précédentes dépendent des valeurs $\tau_1(A,b,\lambda)$ et $t_1(A,b,k)$. Nous discutons du choix de ces deux valeurs dans la section suivante.

7.3.3 Commentaires sur les paramètres de pénalité

La valeur $\tau_1(A,b,\lambda)$ (resp. $t_1(A,b,k)$) donnée dans la proposition 7.3.1.2 (resp. 7.3.2.2) pour obtenir l'équivalence pour tout $\tau \geq \tau_1(A,b,\lambda)$ (resp. $t \geq t_1(A,b,k)$) représente une valeur majorante de la valeur limite $\tau_0(A,b,\lambda)$ (resp. $t_0(A,b,k)$) définie par

$$\tau_0(A,b,\lambda) := \inf\left\{\tau > 0 :\ (P^\lambda) \ \Leftrightarrow\ (P^\lambda_\tau)\right\}, \tag{7.26}$$

respectivement

$$t_0(A,b,k) := \inf\left\{t>0:\ (Q^k)\ \Leftrightarrow\ (Q^k_t)\right\}. \tag{7.27}$$

Les propositions précédentes restent donc valides dès que $\tau > \tau_0(A,b,\lambda)$ (resp. $t > t_0(A,b,k)$).

Il est difficile de calculer la valeur exacte $\tau_0(A,b,\lambda)$ (resp. $t_0(A,b,k)$). Cependant, il est important de noter que plus le majorant $\tau_1(A,b,\lambda)$ (resp. $t_1(A,b,k)$) est proche de la valeur limite mieux c'est, en particulier pour des calculs numériques.

Nous proposons, dans ce qui suit, une technique pour améliorer ces valeurs majorantes dès que l'on a en main un point $\alpha \in \mathbb{R}^n$ qui fait mieux que le point 0 par exemple, i.e.,

1. $\frac{1}{2}\|A\alpha-b\|_2^2+\lambda\|\alpha\|_0 < \frac{1}{2}\|A0-b\|_2^2+\lambda\|0\|_0 = \frac{1}{2}\|b\|_2^2$, dans le cas de $\tau_1(A,b,\lambda)$.
2. $\frac{1}{2}\|A\alpha-b\|_2^2 < \frac{1}{2}\|A0-b\|_2^2 = \frac{1}{2}\|b\|_2^2$ et $\|\alpha\|_0 \le k$, dans le cas de $t_1(A,b,k)$.

Pour $\tau_1(A,b,\lambda)$. Soit $s = \min\left\{\frac{1}{2}\|A\alpha-b\|_2^2 : x\in\mathbb{R}^n\right\}$. Comme l'on sait qu'il existe une solution α de ce problème tel que $\|\alpha\|_0 \le m$, il suffit de poser $u_j := 1$ si $\alpha_j \neq 0$ et $u_j := 0$ sinon, $j = 1,...,n$, pour obtenir un point (α,u) admissible de (P^λ_τ). En prenant ce point (α,u) à la place du point $(0,0)$ dans la preuve du lemme 7.3.1.1, nous obtenons $\phi_\tau(\bar{x},\bar{u}) \le \phi_\tau(\alpha,u) = \frac{1}{2}\|A\alpha-b\|_2^2+\lambda\|\alpha\|_0 \le \frac{1}{2}\|A\alpha-b\|_2^2+\lambda m$ et alors

$$\|A\bar{x}-b\|_2 < \sqrt{\|A\alpha-b\|_2^2+2\lambda\|\alpha\|_0} \le \sqrt{\|A\alpha-b\|_2^2+2\lambda m}. \tag{7.28}$$

De $\left|\left[A^T(A\bar{x}-b)\right]_{i_0}\right| \le \|A\|_1\|A\bar{x}-b\|_2$ et (7.28), nous tirons

$$\left[A^T(A\bar{x}-b)\right]_{i_0} + \|A\|_1\sqrt{\|A\alpha-b\|_2^2+2\lambda\|\alpha\|_0} > 0.$$

Ainsi on pourra prendre

$$\tau_1(A,b,\lambda) := \|A\|_1 \min(\|b\|_2, \sqrt{2s+2\lambda\|\alpha\|_0}), \tag{7.29}$$

si le calcul de α, $\|\alpha\|_0 < m$ n'est pas trop coûteux, et

$$\tau_1(A,b,\lambda) := \|A\|_1 \min(\|b\|_2, \sqrt{2s+2\lambda m}), \tag{7.30}$$

sinon.
On peut noter qu'avec ces choix, $\tau_1(A,b,\lambda)$ décroît vers $\|A\|_1\min(\|b\|_2,\|A\alpha-b\|_2)$, lorsque λ décroît vers zéro. Et que (7.30) devient

$$\tau_1(A,b,\lambda) = \|A\|_1 \min(\|b\|_2, \sqrt{2\lambda m}), \tag{7.31}$$

si le système $Ax=b$ est admissible.

Pour $t_1(A,b,\lambda)$. Si l'on dispose d'un point $\alpha\in\mathbb{R}^n$ vérifiant 2. alors pour des raisons analogues au cas précédent on pourra prendre

$$t_1(A,b,\lambda) := \|A\|_1\|A\alpha-b\|_2 < \|A\|_1\|b\|_2. \tag{7.32}$$

7.3.4 Problème de minimisation ℓ_0 (P)

La proposition qui suit donne une reformulation DC du problème (P) tirée de celle de (P^λ). Rappellons que

$$(P) \quad \begin{cases} \min_x & \|x\|_0 \\ s.t. & Ax = b, \\ & x \in \mathbb{R}^n, \end{cases}$$

$$(P^\lambda) \begin{cases} \min & \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n, \end{cases}$$

$$(P_\tau^\lambda) \begin{cases} \min & \phi_\tau(x,u) := \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau |x|^T (e-u) \\ s.t. & x \in \mathbb{R}^n, u \in [0,1]^n, \end{cases}$$

et $M = \{x \in \mathbb{R}^n : \ Ax = b\}$.

Proposition 7.3.4.1 *Si $M \neq \varnothing$, alors il existe $\lambda_0 > 0$, tel que pour tout λ, $0 < \lambda < \lambda_0$ et pour tout τ, $\tau > \tau_0(A,b,\lambda)$, (P) et (P_τ^λ) sont équivalents dans le sens :*

- *si $\bar{x}$ est une solution de (P), alors il existe $\bar{u}$ tel que $(\bar{x},\bar{u})$ est une solution de (P_τ^λ).*
- *si $(\bar{x},\bar{u})$ est une solution de (P_τ^λ), alors $\bar{x}$ est une solution de (P).*

Preuve Utiliser les propositions 7.2.1 et 7.3.1.2. □

La proposition suivante donne une borne inférieure concernant la valeur absolue des composantes non nulles de toute solution du problème (P^λ).

Proposition 7.3.4.2 *Soient $\lambda > 0$ et $\tau > \tau_0(A,b,\lambda)$. Soit x^λ une solution de (P^λ). Alors*

$$\left|x_i^\lambda\right| \geq \frac{\lambda}{\tau}, \ \ \forall i \in \mathrm{supp}(x^\lambda).$$

Plus particulièrement

$$\left|x_i^\lambda\right| \geq \frac{\lambda}{\tau_1(A,b,\lambda)} \geq \frac{\lambda}{\|A\|_1 \|b\|_2}, \ \ \forall i \in \mathrm{supp}(x^\lambda).$$

Preuve Soit x^λ une solution de (P^λ). On définit u^λ par

$$u_i^\lambda := \begin{cases} 1 & \text{si } x_i^\lambda \neq 0, \\ 0 & \text{sinon,} \end{cases} \quad i = 1, ...n.$$

(x^λ, u^λ) est donc une solution de (P_τ^λ) $(\tau > \tau_0(A,b,\lambda))$ et vérifie donc les conditions KKT généralisées

$$\partial H(x^\lambda, u^\lambda) \cap \partial G(x^\lambda, u^\lambda) \neq \varnothing.$$

Soit $i \in \mathrm{supp}(x^\lambda)$. D'après ces dernières conditions, il existe $\gamma_i \geq 0$ tel que

$$\lambda + \gamma_i = \tau \left|x_i^\lambda\right|.$$

Ainsi

$$\left|x_i^\lambda\right| \geq \frac{\lambda}{\tau},\ \forall \tau > \tau_0(A,b,\lambda).$$

et donc

$$\left|x_i^\lambda\right| \geq \frac{\lambda}{\tau},\ \forall \tau > \tau_0(A,b,\lambda),$$

à fortiori pour $\tau = \tau_1(A,b,\lambda) \leq \|A\|_1 \|b\|_2$. □

7.3.5 Quand est-ce que $(0,0)$ est un point critique ?

Ici nous examinons dans quelle situation $(0,0)$ est un point critique de (P_τ^λ) ou de (Q_t^k).

Proposition 7.3.5.1
- $(0,0)$ *est un point critique de* (P_τ^λ) *si et seulement si* $\tau \geq \left\|A^T b\right\|_\infty$.
- $(0,0)$ *est un point critique de* (Q_t^k) *si et seulement si* $t \geq \left\|A^T b\right\|_\infty$.

Preuve Nous détaillons juste le premier cas (P_τ^λ). Le second s'obtient par un raisonnement similaire. Rappelons qu'un point (x,u) est un point critique de $G-H$ si et seulement si $\partial H(x,u) \cap \partial G(x,u) \neq \varnothing$. Alors, $(0,0)$ l'est si et seulement si

$$\partial H(0,0) \cap \partial G(0,0) \neq \varnothing. \tag{7.33}$$

Comme $\partial H(0,0) = \left\{(A^T b, 0)\right\}$, (7.33) devient

$$(A^T b, 0) \ \in \ \partial G(0,0).$$

En explicitant cette dernière condition, nous obtenons

$$-\tau \ \leq \ \left[A^T b\right]_i \ \leq \ \tau \ \ \forall i = 1, ..., n,$$

i.e.,

$$\tau \geq \left\|A^T b\right\|_\infty. \tag{7.34}$$

□

7.3.6 Versions quadratiques

Dans les reformulations précédentes, on constate que le terme $|x|$ non différentiable rend les problèmes (P_τ^λ) et (Q_t^k) non différentiables donc non quadratiques. Dans ce qui suit, nous allons présenter des versions quadratiques des problèmes (P_τ^λ) et (Q_t^k). Rappelons que

$$(P_\tau^\lambda) \quad \begin{cases} \min_{(x,u)} & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau \left|x\right|^T (e-u) \\ s.t. & x \in \mathbb{R}^n, \\ & u \in [0,1]^n, \end{cases}$$

et

$$(Q_t^k) \quad \begin{cases} \min_{(x,u)} & \frac{1}{2}\|Ax-b\|_2^2 + t\,|x|^T(e-u) \\ s.t. & e^T u \leq k, \\ & x \in \mathbb{R}^n, \\ & u \in [0,1]^n. \end{cases}$$

Sachant que tout point $x \in \mathbb{R}^n$ peut s'écrire $x = x^+ - x^-$, avec $x_i^+ \geq 0$, $x_i^- \geq 0$ et $x_i^+ x_i^- = 0$, $i = 1, ..., n$, nous dérivons une reformulation quadratique avec contraintes boites de (P_τ^λ)

$$(QuadP_\tau^\lambda) \quad \begin{cases} \min_{(\alpha,\beta,u)} & \frac{1}{2}\|A(\alpha-\beta)-b\|_2^2 + \lambda e^T u + \tau(\alpha+\beta)^T(e-u) \\ s.t. & \alpha \in \mathbb{R}^n_+, \beta \in \mathbb{R}^n_+ \\ & u \in [0,1]^n, \end{cases}$$

et une reformulation quadratique de (Q_t^k)

$$(QuadQ_t^k) \quad \begin{cases} \min_{(\alpha,\beta,u)} & \frac{1}{2}\|A(\alpha-\beta)-b\|_2^2 + t(\alpha+\beta)^T(e-u) \\ s.t. & e^T u \leq k, \\ & \alpha \in \mathbb{R}^n_+, \beta \in \mathbb{R}^n_+ \\ & u \in [0,1]^n. \end{cases}$$

Les versions quadratiques $(QuadP_\tau^\lambda)$ et $(QuadQ_t^k)$ permettent ainsi de résoudre les problèmes P^λ et (Q^k) par des techniques de la programmation quadratique non convexe ou tout autre méthode d'optimisation différentiable non convexe adaptée.

Dans le cas où une contrainte de positivité sur la variable x ($x \geq 0$) est ajoutée sur le problème de parcimonie, nous obtenons directement une reformulation quadratique équivalente.

7.4 Formulations dans le cas avec contrainte de positivité

Considérons les problèmes de parcimonie précédents avec des contraintes de positivité

$$(PosP) \quad \begin{cases} \min_x & \|x\|_0 \\ s.t. & Ax = b, \\ & x \geq 0, \end{cases}$$

$$(PosP^\lambda) \quad \begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_0 \\ s.t. & x \geq 0, \end{cases}$$

et

$$(PosQ^k) \begin{cases} \min_x & \frac{1}{2}\|Ax-b\|_2^2 \\ s.t. & \|x\|_0 \leq k, \\ & x \geq 0. \end{cases}$$

On pose $PosM = \{x \in \mathbb{R}^n : \ Ax = b, x \geq 0\}$. Comme dans le cas sans contraintes de positivité, le problème $(PosP^\lambda)$ est équivalent à $(PosP)$ pour $\lambda > 0$ suffisamment

petit, de même il existe $k_0 \in \{1, ..., n\}$ tel que $(PosQ^{k_0})$ soit équivalent à $(PosP)$. Nous aurons besoin de rappeler quelques résultats préliminaires très bien connus en optimisation linéaire.

Lemme 7.4.1 *Soit $x \in PosM$. Alors les deux propriétés suivantes sont équivalentes :*

1. *x est un sommet de $PosM$.*
2. *Les vecteurs colonnes de A, correspondant aux indices $i \in supp(x)$, sont linéairement indépendants.*

Preuve Supposons (2.) faux. Il existe alors un élément $z \in \mathbb{R}^n$, $z \neq 0$, tel que $z_i = 0$ pour $i \notin \text{supp}(x)$, vérifiant $Az = 0$. Pour ϵ réel de module $|\epsilon|$ assez petit, on a

$$x \pm \epsilon z \geq 0 \ \text{ et } \ A(x \pm \epsilon z) = b,$$

donc $x \pm \epsilon z \in PosM$. Comme

$$x = \frac{1}{2}\left[(x + \epsilon z) + (x - \epsilon z)\right],$$

on a alors (1.) qui est faux.

Réciproquement, supposons que (1.) est faux. Il existe alors y et $z \in PosM$, $y \neq z$, et $\alpha \in]0, 1[$, tels que

$$x = \alpha y + (1 - \alpha)z.$$

Soit $i \in \{1, ..., n\}$. On a $x_i = \alpha y_i + (1-\alpha)z_i$, avec $y_i \geq 0$, $z_i \geq 0$, $x_i \geq 0$, et $0 < \alpha < 1$. Par suite, x_i est nul si et seulement si y_i et z_i sont tous deux nuls. Le vecteur $y - z$, non identiquement nul, vérifie $y_i - z_i = 0$ pour tout $i \notin \text{supp}(x)$, et il est dans le noyau de A puisque $A(y - z) = 0$, ce qui prouve que (2.) est faux. □

Proposition 7.4.2 *$PosM$ a un nombre fini de sommets inférieur ou égal à C_n^m (rappelons que C_n^m est le nombre de façons de choisir m colonnes parmi les n colonnes de A).*

Définition 7.4.3 (rayon infini) *On dit qu'un vecteur $y \in \mathbb{R}^n$, $y \geq 0$ est un rayon infini du polyèdre $PosM$ si pour tout $x \in PosM$ et tout $\alpha \geq 0$, $x + \alpha y \in PosM$.*

Remarquons que y est un rayon infini si et seulement si

$$Ay = 0, \ \ y \geq 0.$$

Définition 7.4.4 (rayons extrémaux) *Considérons l'ensemble*

$$R = \left\{y \in \mathbb{R}^n : \ Ay = 0, \ \ e^T y = 1, \ \ y \geq 0\right\}.$$

On appelle rayons extrémaux de $PosM$ les sommets du polyèdre convexe R.

Proposition 7.4.5 *Tout point du polyèdre convexe $PosM$ est somme d'une combinaison convexe de ses points sommets et d'une combinaison linéaire à coefficients positifs ou nuls de ses rayons extrémaux.*

Proposition 7.4.6 *Si $PosM$ est non vide, alors toute solution de $(PosP)$ est un sommet de $PosM$.*

Preuve Nous procédons par contradiction. Soit $\bar{x}$ une solution de $(PosP)$. Supposons que $\bar{x}$ n'est pas un sommet de $PosM$. Alors d'après le lemme précédent 7.4.1, il existe $w \neq 0$, tel que $Aw = 0$ et $supp(w) \subset supp(\bar{x})$.
Pour $t \in \mathbb{R}$, posons $v(t) := \bar{x} + tw$, on a $Av(t) = b$ et l'ensemble des réels t tel que $v(t) \geq 0$ est un intervalle fermé de $\mathbb{R}$ contenant un intervalle ouvert contenant 0, nécessairement borné au moins sur un coté. Soit t_m une des extrémités finies de cet intervalle. Alors $v(t_m) \in PosM$ et $\|v(t_m)\|_0 < \|\bar{x}\|_0$. Contradiction $\square$.

7.4.1 Cas de la minimisation ℓ_0 $(PosP)$

Dans ce paragraphe, nous allons présenter une reformulation quadratique de $(PosP)$ en utilisant les outils d'optimisation linéaire précédents. Rappelons que $(PosP)$ est donné par

$$(PosP) \quad \begin{cases} \min_x & \|x\|_0 \\ s.t. & Ax = b, \\ & x \geq 0. \end{cases}$$

Pour $\tau > 0$, considérons le problème quadratique

$$(PosP_\tau) \begin{cases} \min & \psi_\tau(x,u) := e^T u + \tau x^T (e-u) \\ s.t. & Ax = b, \\ & x \geq 0, \\ & u \in [0,1]^n . \end{cases}$$

Soit V l'ensemble des points sommets de $PosM$. Soit

$$\tau_0 := \frac{1}{\min\{x_j : x \in V, j \in supp(x)\}}.$$

Nous allons déterminer les valeurs de τ pour lesquelles les deux problèmes sont équivalents. Nous aurons besoin d'abord du lemme suivant qui nous donne quelques propriétés des solutions de $(PosP_\tau)$.

Lemme 7.4.1.1 *Soit $\tau > \tau_0$. On suppose $PosM$ non vide. Alors toute solution $(\bar{x}, \bar{u})$ de $(PosP_\tau)$ vérifie $\bar{x} \in V$ et $\psi_\tau(\bar{x}, \bar{u}) = \|\bar{x}\|_0$.*

Preuve Soit $(\bar{x}, \bar{u})$ solution de $(PosP_\tau)$. Nous allons d'abord montrer que $\bar{x} \in V$ et ensuite $\psi_\tau(\bar{x}, \bar{u}) = \|\bar{x}\|_0$. Nous procédons par contradiction.

1. Supposons $\bar{x} \notin V$. D'après la proposition 7.4.5 on a

$$\bar{x} = \sum_{i \in I} \alpha_i v^i + \sum_{j \in J} \beta_j E^j, \tag{7.35}$$

avec

$$0 < \alpha_i \leq 1, \forall i \in I, \sum_{i \in I} \alpha_i = 1,\ v^i \in V, \forall i \in I,\ \beta_j \geq 0, \forall j \in J,$$

et $E^j \geq 0, j \in J$ des rayons extrémaux de $PosM$.
De (7.35), nous tirons

$$\bar{x}^T(e - \bar{u}) = \sum_{i \in I} \alpha_i v^{iT}(e - \bar{u}) + \sum_{j \in J} \beta_j E^{jT}(e - \bar{u}),$$

et alors

$$\bar{x}^T(e - \bar{u}) \geq \sum_{i \in I} \alpha_i v^{iT}(e - \bar{u}),\ \ 0 < \alpha_i \leq 1, \forall i \in I, \sum_{i \in I} \alpha_i = 1. \tag{7.36}$$

Comme $(\bar{x}, \bar{u})$ est solution de $(PosP_\tau)$ et les points $(v^i, \bar{u}), i \in I$ sont admissibles, on a

$$\psi_\tau(\bar{x}, \bar{u}) = e^T\bar{u} + \tau\bar{x}^T(e - \bar{u}) \leq \psi_\tau(v^i, \bar{u}) = e^T\bar{u} + \tau v^{iT}(e - \bar{u}),\ \forall i \in I.$$

Ainsi

$$\bar{x}^T(e - \bar{u}) \leq v^{iT}(e - \bar{u}),\ \forall i \in I. \tag{7.37}$$

En combinant cette dernière inégalité avec l'inégalité (7.36), nous obtenons

$$\bar{x}^T(e - \bar{u}) = v^{iT}(e - \bar{u}),\ \forall i \in I. \tag{7.38}$$

Parsuite

$$\psi_\tau(\bar{x}, \bar{u}) = \psi_\tau(v^i, \bar{u}),\ \forall i \in I \tag{7.39}$$

et comme $\beta_j \geq 0,\ E^j \geq 0,\ e - \bar{u} \geq 0$, on a de plus

$$\beta_j E^j_l(1 - \bar{u}_l) = 0,\ \forall l = 1, ..., n,\ \ \forall j \in J. \tag{7.40}$$

$$\begin{aligned} \psi_\tau(v^i, u) - \psi_\tau(\bar{x}, \bar{u}) &= \psi_\tau(v^i, u) - \psi_\tau(v^i, \bar{u}) \\ &= e^T u + \tau v^{iT}(e - u) - e^T\bar{u} - \tau v^{iT}(e - \bar{u}) \\ &= e^T(u - \bar{u}) - \tau v^{iT}(u - \bar{u}) \\ &= (e - \tau v^i)^T(u - \bar{u}),\ \ \forall i \in I, \forall u \in [0, 1]^n\,. \end{aligned} \tag{7.41}$$

Nous allons maintenant construire un point admissible (x, u) tel que $\psi_\tau(x, u) < \psi_\tau(\bar{x}, \bar{u})$ ce qui contredirait le fait que $(\bar{x}, \bar{u})$ est solution de $(PosP_\tau)$. Pour cela nous considérons les cas suivants :

– $\bar{x}_j(1-\bar{u}_j)=0, \forall j$: soit $v(t_m)$ comme dans la preuve de la proposition 7.4.6 et posons $x := v(t_m)$,

$$u_i := \begin{cases} 1 \text{ si } v(t_m)_i > 0, \\ 0 \text{ sinon,} \end{cases} \quad i = 1, ..., n.$$

On a $\psi_\tau(x,u) = \|x\|_0 = \|x\|_0 < \|\bar{x}\|_0 \leq e^T\bar{u} \leq \psi_\tau(\bar{x},\bar{u})$ et donc

$$\psi_\tau(x,u) - \psi_\tau(\bar{x},\bar{u}) < 0. \text{ Contradiction}$$

– il existe l tel que $\bar{x}_l(1-\bar{u}_l) > 0$: nous distinguons les deux sous cas
  – $0 < \bar{u}_l < 1$: posons $x = v^i$, $u_j := \bar{u}_j, \forall j \neq l$ et

$$u_l := \begin{cases} 1 \text{ si } v_l^i > 0, \\ 0 \text{ sinon.} \end{cases}$$

$$\psi_\tau(x,u) - \psi_\tau(\bar{x},\bar{u}) = (1-\tau v_l^i)(u_l - \bar{u}_l) < 0. \text{ Contradiction}$$

  – $\bar{u}_l = 0$: d'après 7.35 et 7.40, on a

$$\beta_j E_l^j = 0, \ \ \forall j \in J \text{ et } \bar{x}_l = \sum_{i\in I} \alpha_i v_l^i.$$

Comme $\bar{x}_l > 0$, il existe alors $i_0 \in I$ tel que $v_l^{i_0} > 0$. Posons alors $x = v^{i_0}$, $u_j := \bar{u}_j, \forall j \neq l$ et $u_l = 1$.

$$\psi_\tau(x,u) - \psi_\tau(\bar{x},\bar{u}) = (1-\tau v_l^{i_0}) < 0. \text{ Contradiction}$$

Nous venons de montrer, dans tous les cas, qu'il existe un point (x,u) tel que $\psi_\tau(x,u) - \psi_\tau(\bar{x},\bar{u}) < 0$ ce qui contredit le fait que $(\bar{x},\bar{u})$ est solution de $(PosP_\tau)$. Parsuite, $\bar{x}$ est un nécessairement un sommet de $PosM$, c'est à dire $\bar{x} \in V$.

2. On va maintenant montrer que $\psi_\tau(\bar{x},\bar{u}) = \|\bar{x}\|_0$. Pour cela on pose

$$G := \{i : \ \bar{x}_i = 0, \ \bar{u}_i > 0\} \ \text{ et } \ F := \{i : \ \bar{x}_i(1-\bar{u}_i) > 0\}.$$

Il suffit alors de montrer que $G = \varnothing$ et $F = \varnothing$. Nous procédons par contradiction.

Supposons $G \neq \varnothing$. Soit $i_0 \in G$. Posons $x := \bar{x}$, $u_i := \bar{u}_i, \forall i \neq i_0$ et $u_{i_0} = 0$. On a

$$\psi_\tau(x,u) - \psi_\tau(\bar{x},\bar{u}) = -\bar{u}_{i_0} < 0.$$

Contradiction et donc $G = \varnothing$.

Supposons $F \neq \varnothing$. Soit $i_0 \in F$. Comme $G = \varnothing$, on a $0 < \bar{u}_{i_0} < 1$ et $\bar{x}_{i_0} > 0$. Posons alors $x := \bar{x}$, $u_i := \bar{u}_i, \forall i \neq i_0$ et $u_{i_0} = 1$. Ainsi

$$\psi_\tau(x,u) - \psi_\tau(\bar{x},\bar{u}) = (1-\tau\bar{x}_{i_0}).$$

D'après 1., on a $\bar{x} \in V$ et donc $(1-\tau\bar{x}_{i_0}) < 0$. Contradiction. Donc $F = \varnothing$. Ce qui termine la preuve. □

Nous sommes maintenant en mesure d'établir l'équivalence entre $(PosP)$ et $(PosP_\tau)$.

Proposition 7.4.1.2 *Soit $\tau > \tau_0$. Alors $(PosP)$ et $(PosP_\tau)$ sont équivalents dans le sens :*

- *si $\bar{x}$ est une solution de $(PosP)$, alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{u})$ est une solution de $(PosP_\tau)$.*
- *si $(\bar{x}, \bar{u})$ est une solution de $(PosP_\tau)$, alors $\bar{x}$ est une solution de $(PosP)$.*

Preuve Soit $\tau > \tau_0$. Soit $(\bar{x}, \bar{u})$ une solution de $(PosP_\tau)$ et soit $\tilde{x}$ une solution de $(PosP)$. On pose

$$\tilde{u}_i := \begin{cases} 1 \text{ si } \tilde{x}_i > 0, \\ 0 \text{ sinon,} \end{cases} \quad i = 1, ..., n.$$

D'après le lemme 7.4.1.1 précédent $\psi_\tau(\bar{x}, \bar{u}) = \|\bar{x}\|_0$.

$$\psi_\tau(\tilde{x}, \tilde{u}) = \|\tilde{x}\|_0 \leq \|\bar{x}\|_0 = \psi_\tau(\bar{x}, \bar{u}) \leq \psi_\tau(\tilde{x}, \tilde{u})$$

et donc

$$\psi_\tau(\tilde{x}, \tilde{u}) = \|\tilde{x}\|_0 = \|\bar{x}\|_0 = \psi_\tau(\bar{x}, \bar{u}).$$

Parsuite $\bar{x}$ solution de $(PosP)$ et $(\tilde{x}, \tilde{u})$ solution de $(PosP_\tau)$. □

7.4.2 Cas des moindres carrés ℓ_0-régularisés $(PosP^\lambda)$

Rappelons que $(PosP^\lambda)$ est donné par

$$(PosP^\lambda) \quad \begin{cases} \min_x & \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \geq 0. \end{cases}$$

Pour $\tau > 0$, considérons le problème quadratique avec contraintes boites

$$(PosP_\tau^\lambda) \begin{cases} \min & \phi_\tau(x, u) := \frac{1}{2}\|Ax - b\|_2^2 + \lambda e^T u + \tau x^T (e - u) \\ s.t. & x \geq 0, \\ & u \in [0, 1]^n. \end{cases}$$

Dans ce qui suit, nous allons montrer que $(PosP^\lambda)$ et $(PosP_\tau^\lambda)$ sont équivalents pour des valeurs de τ que nous préciserons. La démarche est analogue à celle du lemme 7.3.1.1. La différence est qu'ici nous allons utiliser les conditions KKT classiques au lieu des conditions KKT généralisées en programmation DC.

Proposition 7.4.2.1 *Soit $\tau \geq \|A\|_1 \|b\|_2$. Alors $(PosP^\lambda)$ et $(PosP_\tau^\lambda)$ sont équivalents dans le sens :*

- *si $\bar{x}$ est une solution de $(PosP^\lambda)$, alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{u})$ est une solution de $(PosP_\tau^\lambda)$.*

– *si* $(\bar{x}, \bar{u})$ *est une solution de* $(PosP_\tau^\lambda)$*, alors* $\bar{x}$ *est une solution de* $(PosP^\lambda)$.

Preuve Soit $\tau \geq \|A\|_1 \|b\|_2$. Soit $(\bar{x}, \bar{u})$ une solution de $(PosP_\tau^\lambda)$ et soit $\tilde{x}$ une solution de $(PosP^\lambda)$. On pose $\tilde{u}_i := 1$ si $\tilde{x}_i > 0$ et $\tilde{u}_i := 0$ sinon.

Nous allons d'abord montrer que $\phi_\tau(\bar{x}, \bar{u}) = \frac{1}{2}\|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0$ et nous montrons ensuite que $\bar{x}$ est une solution de $(PosP^\lambda)$ et que $(\tilde{x}, \tilde{u})$ est une solution de $(PosP_\tau^\lambda)$.Posons

$$F := \{i : \bar{x}_i > 0, \bar{u}_i = 0\}, \; G := \{i : \bar{x}_i = 0, \bar{u}_i > 0\} \text{ et } I := \{i : \bar{x}_i(1 - \bar{u}_i) > 0\}.$$

$(\bar{x}, \bar{u})$ étant solution de $(PosP^\lambda)$ satisfait nécessairement les conditions KKT classiques appliquées à $(PosP_\tau^\lambda)$

$$\begin{cases} A^T(A\bar{x} - b) + \tau(e - \bar{u}) - \beta = 0, \\ \lambda e - \tau\bar{x} + \gamma - \delta = 0, \\ \beta_i\bar{x}_i = 0, \gamma_i(\bar{u}_i - 1) = 0, \delta_i\bar{u}_i = 0, i = 1, ..., n, \\ x \geq 0, u \in [0, 1]^n, \beta \geq 0, \gamma \geq 0, \delta \geq 0. \end{cases} \tag{7.42}$$

Montrons que $F = \varnothing$ et $G = \varnothing$. Nous allons procéder par contradiction.
Suposons $F \neq \varnothing$ et soit let $i_0 \in F$. On a $\bar{x}_{i_0} > 0$ et $\bar{u}_{i_0} = 0$, en utilisant les conditions KKT (7.42), nous obtenons

$$\left[A^T(A\bar{x} - b)\right]_{i_0} + \tau = 0. \tag{7.43}$$

Comme 0 est un point admissible pour (P_τ^λ), on a $\phi_\tau(\bar{x}, \bar{u}) \leq \frac{1}{2}\|b\|_2^2$ et alors

$$\|A\bar{x} - b\|_2 < \|b\|_2. \tag{7.44}$$

De plus

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| \leq \|A\|_1 \|A\bar{x} - b\|_2$$

et alors en combinant avec (7.44), nous obtenons $\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| < \|A\|_1 \|b\|_2$ et alors

$$\left[A^T(A\bar{x} - b)\right]_{i_0} + \tau \geq \left[A^T(A\bar{x} - b)\right]_{i_0} + \|A\|_1 \|b\|_2 > 0.$$

Ce qui contredit (7.43) et donc $F = \varnothing$.
Supposons $G \neq \varnothing$ et soit $i_0 \in G$. Comme $\bar{x}_{i_0} = 0$ et $\bar{u}_{i_0} > 0$, des conditions KKT (7.42) nous tirons $\lambda + \gamma_{i_0} = 0$ et $\gamma_{i_0} \geq 0$ et alors $\lambda = 0$. Contradiction, donc $G = \varnothing$.
Maintenant nous allons considérer les cas : $I = \varnothing$ et $I \neq \varnothing$.

– Cas $I = \varnothing$: Comme $G = \varnothing$, on a $e^T\bar{u} = \|\bar{x}\|_0$ et $\bar{x}^T(e - \bar{u}) = 0$, donc

$$\phi_\tau(\bar{x}, \bar{u}) = \frac{1}{2}\|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0.$$

– Cas $I \neq \varnothing$: Soit $i_0 \in I$. Comme $F = \varnothing$, on a $0 < \bar{u}_{i_0} < 1$ et $\bar{x}_{i_0} > 0$, en combinant avec les conditions KKT (7.42) nous obtenons

$$\lambda - \tau\bar{x}_{i_0} = 0,$$

donc $\bar{x}_{i_0} = \frac{\lambda}{\tau}$ et alors

$$\bar{x}_i = \frac{\lambda}{\tau}, \forall i \in I.$$

$$\begin{aligned} \phi_\tau(\bar{x}, \bar{u}) &= \tfrac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda e^T \bar{u} + \tau \textstyle\sum_{i \in I} \bar{x}_i (1 - \bar{u}_i) \\ &= \tfrac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda e^T \bar{u} + \lambda \textstyle\sum_{i \in I} (1 - \bar{u}_i) \\ &= \tfrac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda (\textstyle\sum_{i \notin I} \bar{u}_i + \sum_{i \in I} 1). \end{aligned}$$

De plus, comme $G = \varnothing$, nous obtenons

$$\phi_\tau(\bar{x}, \bar{u}) = \frac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0 .$$

Ce qui prouve que $\phi_\tau(\bar{x}, \bar{u}) = \frac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0$.
Maintenant nous allons montrer que $\bar{x}$ est une solution de $(PosP^\lambda)$ et que $(\tilde{x}, \tilde{u})$ est une solution de $(PosP_\tau^\lambda)$.
On a

$$\phi_\tau(\tilde{x}, \tilde{u}) = \frac{1}{2} \|A\tilde{x} - b\|_2^2 + \lambda \|\tilde{x}\|_0 \leq \frac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0 = \phi_\tau(\bar{x}, \bar{u}) \leq \phi_\tau(\tilde{x}, \tilde{u}),$$

et alors

$$\phi_\tau(\tilde{x}, \tilde{u}) = \frac{1}{2} \|A\tilde{x} - b\|_2^2 + \lambda \|\tilde{x}\|_0 = \frac{1}{2} \|A\bar{x} - b\|_2^2 + \lambda \|\bar{x}\|_0 = \phi_\tau(\bar{x}, \bar{u}).$$

Ce qui signifie $\bar{x}$ est une solution de $(PosP^\lambda)$ et que $(\tilde{x}, \tilde{u})$ est une solution de $(PosP_\tau^\lambda)$. □

7.4.3 Cas des moindres carrés avec contrainte ℓ_0 $(PosQ^k)$

Rappelons que $(PosQ^k)$ est donné par

$$(PosQ^k) \begin{cases} \min_x & \frac{1}{2} \|Ax - b\|_2^2 \\ s.t. & \|x\|_0 \leq k, \\ & x \geq 0. \end{cases}$$

Pour $t > 0$, considérons le problème quadratique suivant

$$(PosQ_t^k) \begin{cases} \min & \psi_\tau(x, u) := \frac{1}{2} \|Ax - b\|_2^2 + t x^T (e - u) \\ s.t. & e^T u \leq k, \\ & x \geq 0, \\ & u \in [0, 1]^n . \end{cases}$$

Comme précédemment, nous allons montrer que $(PosQ^k)$ et $(PosQ_t^k)$ sont équivalents pour des valeurs de t que nous préciserons.

Proposition 7.4.3.1 *Soit $\tau \geq \|A\|_1 \|b\|_2$. Alors $(PosQ^k)$ et $(PosQ_t^k)$ sont équivalents dans le sens :*

- *si $\bar{x}$ est une solution de $(PosQ^k)$, alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{u})$ est une solution de $(PosQ_t^k)$.*
- *si $(\bar{x}, \bar{u})$ est une solution de $(PosQ_t^k)$, alors $\bar{x}$ est une solution de $(PosQ^k)$.*

Preuve Soit $\tau \geq \|A\|_1 \|b\|_2$. Soit $(\bar{x}, \bar{u})$ une solution de $(PosQ_t^k)$ et soit $\tilde{x}$ une solution de $(PosQ^k)$. On pose $\tilde{u}_i := 1$ si $\tilde{x}_i > 0$ et $\tilde{u}_i := 0$ sinon.
Nous allons d'àbord montrer que $|\bar{x}|^T (e - \bar{u}) = 0$ et $\|\bar{x}\|_0 \leq k$ et nous montrons ensuite que $\bar{x}$ est une solution de $(PosQ_t^k)$ et que $(\tilde{x}, \tilde{u})$ est une solution de $(PosQ_t^k)$. Posons

$$F := \{i : \bar{x}_i > 0, \bar{u}_i = 0\}, \; G := \{i : \bar{x}_i = 0, \bar{u}_i > 0\} \text{ et } I := \{i : \bar{x}_i(1 - \bar{u}_i) > 0\}.$$

$(\bar{x}, \bar{u})$ vérifie les conditions KKT classiques appliquées à $(PosQ_t^k)$

$$\begin{cases} A^T(A\bar{x} - b) + \tau(e - \bar{u}) - \beta = 0, \\ \eta e - \tau\bar{x} + \gamma - \delta = 0, \\ \eta(e^T\bar{u} - k) = 0, \; \beta_i\bar{x}_i = 0, \; \gamma_i(\bar{u}_i - 1) = 0, \; \delta_i\bar{u}_i = 0, \; i = 1, ..., n, \\ e^T\bar{u} \leq k, \; x \geq 0, \; u \in [0, 1]^n, \; \eta \geq 0, \; \beta \geq 0, \; \gamma \geq 0, \; \delta \geq 0. \end{cases} \tag{7.45}$$

Nous allons d'abord montrer que $F = \varnothing$. Nous procédons par contradiction.
Supposons $F \neq \varnothing$ et soit $i_0 \in F$. On a $\bar{x}_{i_0} > 0$ et $\bar{u}_{i_0} = 0$. Des conditions KKT (7.45), nous tirons

$$\left[A^T(A\bar{x} - b)\right]_{i_0} + t = 0. \tag{7.46}$$

Comme $(0, 0)$ est admissible pour $(PosQ_\tau^k)$, on a $\psi_\tau(\bar{x}, \bar{u}) \leq \frac{1}{2}\|b\|_2^2$ et donc

$$\|A\bar{x} - b\|_2 < \|b\|_2. \tag{7.47}$$

On a

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| \leq \|A\|_1 \|A\bar{x} - b\|_2$$

donc en combinant avec (7.47), nous obtenons

$$\left|\left[A^T(A\bar{x} - b)\right]_{i_0}\right| < \|A\|_1 \|b\|_2 \leq t.$$

Ce qui contredit (7.46) et donc $F = \varnothing$.
Montrons que $I \neq \varnothing$. Pour cela supposons $I \neq \varnothing$.
Soit $i_0 \in I$, comme $F = \varnothing$ on a $\bar{x}_{i_0} > 0$ et $0 < \bar{u}_{i_0} < 1$. Les conditions KKT (7.45) donnent alors

$$\bar{x}_{i_0} = \frac{\eta}{t}, \; \eta > 0, \; e^T\bar{u} = k,$$

et donc

$$\bar{x}_i = \frac{\eta}{t}, \; \forall i \in I. \tag{7.48}$$

Considérons maintenant les cas $G \neq \varnothing$ et $G = \varnothing$.

- $G \neq \varnothing$. Soit $g \in G$ i.e. $\bar{x}_g = 0$, et $\bar{u}_g > 0$. Alors les conditions KKT (7.45) donnent $\eta + \gamma_g = 0$, $\gamma_g(\bar{u}_g - 1) = 0$ et $\gamma_g \geq 0$. Ce qui contredit $\eta > 0$.

– $G = \varnothing$. A partir de $e^T\bar{u} = k$, nous tirons

$$\sum_{i\in I} \bar{u}_i = k - \sum_{i\notin I} \bar{u}_i.$$

Comme $G = \varnothing$, on a $\sum_{i\notin I} \bar{u}_i \in \mathbb{N}$ et donc $\sum_{i\in I} \bar{u}_i \in \mathbb{N}$. De plus, $\sum_{i\in I} \bar{u}_i > 0$ ($F = \varnothing$), ainsi

$$\sum_{i\in I} \bar{u}_i \in \{1, ..., k\}.$$

On pose $c := \sum_{i\in I} \bar{u}_i$. Il est clair que $c \in \{1, ..., k\}$ et $c < Card(I)$.
Soit $\{i_1, i_2, ..., i_c\} \subset I$ et soit $\tilde{u}$ le point définit par

$$\tilde{u}_i := \begin{cases} \bar{u}_i & \forall i \notin I, \\ 1 & \forall i \in \{i_1, i_2, ..., i_c\}, \\ 0 & \text{ailleurs.} \end{cases}$$

On a

$$\tilde{u} \in [0,1]^n,\ e^T\tilde{u} = k$$

et

$$t\bar{x}^T(e-\bar{u}) = t\sum_{i\in I} \bar{x}_i(1-\bar{u}_i) = \eta\sum_{i\in I}(1-\bar{u}_i) = \eta\sum_{i\in I}(1-\tilde{u}_i) = t\bar{x}^T(e-\tilde{u}). \quad (7.49)$$

Donc $(\bar{x}, \tilde{u})$ est une solution de $(PosQ^k_\tau)$ et alors vérifie les conditions KKT (7.45).
Soit $r \in I \setminus \{i_1, i_2, ..., i_c\}$, on a $\bar{x}_r \neq 0$ et $\tilde{u}_r = 0$. Ce qui donne avec les conditions KKT (7.45)

$$\left[A^T(A\bar{x} - b)\right]_r + t = 0.$$

Cette dernière condition est identique à la condition (7.46) et donc contredit $t \geq \|A\|_1 \|b\|_2$.

Parsuite $I = \varnothing$. Ce qui signifie $\bar{x}^T(e - \bar{u}) = 0$ et $\|\bar{x}\|_0 \leq k$.
$\bar{x}$ étant admissible pour $(PosQ^k)$, on a

$$\psi_\tau(\tilde{x}, \tilde{u}) = \frac{1}{2}\|A\tilde{x} - b\|_2^2 \leq \frac{1}{2}\|A\bar{x} - b\|_2^2 = \psi_\tau(\bar{x}, \bar{u}) \leq \psi_\tau(\tilde{x}, \tilde{u}),$$

et donc

$$\psi_\tau(\tilde{x}, \tilde{u}) = \frac{1}{2}\|A\tilde{x} - b\|_2^2 = \frac{1}{2}\|A\bar{x} - b\|_2^2 = \psi_\tau(\bar{x}, \bar{u}).$$

Ainsi $\bar{x}$ est une solution de $(PosQ^k_t)$ et $(\tilde{x}, \tilde{u})$ est une solution de $(PosQ^k_t)$. □

Sur le tableau 7.1 nous résumons les différentes reformulations obtenues.

Table 7.1 – Tableau récapitulatif des différentes reformulations.

Problèmes ℓ_0	Reformulations DC
$\min_x \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_0$ $s.t. \quad x \in \mathbb{R}^n.$	$\min_{(x,u)} \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau \|x\|^T (e-u)$ $s.t. \quad x \in \mathbb{R}^n$ $u \in [0,1]^n.$ Pour $\tau \geq \|A\|_1 \|b\|_2$.
$\min_x \ \frac{1}{2}\|Ax-b\|_2^2$ $s.t. \quad \|x\|_0 \leq k$ $x \in \mathbb{R}^n.$	$\min_{(x,u)} \ \frac{1}{2}\|Ax-b\|_2^2 + t \|x\|^T (e-u)$ $s.t. \quad e^T u \leq k,$ $x \in \mathbb{R}^n,$ $u \in [0,1]^n.$ Pour $t \geq \|A\|_1 \|b\|_2$.
$\min_x \ \|x\|_0$ $s.t. \quad Ax = b,$ $x \in \mathbb{R}^n.$	$\min_{(x,u)} \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau \|x\|^T (e-u)$ $s.t. \quad x \in \mathbb{R}^n,$ $u \in [0,1]^n.$ Pour $0 < \lambda < \lambda_0$, $\tau \geq \|A\|_1 \|b\|_2$.
$\min_x \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda \|x\|_0$ $s.t. \quad x \geq 0.$	$\min_{(x,u)} \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau x^T (e-u)$ $s.t. \quad x \geq 0,$ $u \in [0,1]^n.$ Pour $\tau \geq \|A\|_1 \|b\|_2$.
$\min_x \ \frac{1}{2}\|Ax-b\|_2^2$ $s.t. \quad \|x\|_0 \leq k,$ $x \geq 0.$	$\min_{(x,u)} \ \frac{1}{2}\|Ax-b\|_2^2 + t x^T (e-u)$ $s.t. \quad e^T u \leq k,$ $x \geq 0,$ $u \in [0,1]^n.$ Pour $t \geq \|A\|_1 \|b\|_2$.
$\min_x \ \|x\|_0$ $s.t. \quad Ax = b,$ $x \geq 0.$	$\min_{(x,u)} \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + t x^T (e-u)$ $s.t. \quad x \geq 0,$ $u \in [0,1]^n.$ Pour $0 < \lambda < \lambda_0$, $t \geq \|A\|_1 \|b\|_2$.
$\min_x \ \|x\|_0$ $s.t. \quad Ax = b,$ $x \geq 0.$	$\min_{(x,u)} \ e^T u + \tau x^T (e-u)$ $s.t. \quad Ax = b,$ $x \geq 0,$ $u \in [0,1]^n.$ Pour $\tau > \tau_0$.

Nous venons de présenter différentes techniques de reformulation de problèmes de parcimonie dans les modèles de régression linéaire. Ces techniques peuvent être étendues à d'autres problèmes de parcimonie dans d'autres modèles comme la régression logistique dont la fonction perte est convexe lipschitzienne par exemple.

7.5 Extensions

Nous nous proposons dans cette section d'étendre ces techniques de reformulation aux problèmes ℓ_0 de la forme

$$(\bar{P}^\lambda) \quad \min \{f(x,y) + \lambda \|x\|_0 : x \in \mathbb{R}^n, y \in \mathbb{R}^m\},$$

$$(\bar{Q}^k) \quad \min \{f(x,y) : \|x\|_0 \leq k,\ x \in \mathbb{R}^n, y \in \mathbb{R}^m\},$$

où $\lambda > 0$, $k = 1, ..., n-1$ et f est une fonction convexe et lipschitzienne telle que

$$\min \{f(x,y) : (x,y) \in \mathbb{R}^n \times \mathbb{R}^m\}$$

admet au moins une solution.

L'hypothèse f admet au moins une solution nous assure l'existence de solution pour les problèmes $(\bar{P}^\lambda)$ et $(\bar{Q}^k)$. En effet, par un procédé énumératif fini on peut trouver une solution. Cependant, ce procédé n'est pas utilisable en pratique du fait de sa complexité exponentielle.

L'hypothèse f convexe et lipschitzienne nous permet ici de déterminer le paramètre de pénalité pour obtenir des reformulations équivalentes. Cependant, cette hypothèse ne constitue pas forcément l'hypothèse la plus faible pour obtenir des reformulations équivalentes.

Nous aurons besoin du résultat préliminaire suivant.

Lemme 7.5.1 *Soit $\theta : \mathbb{R}^p \to \mathbb{R}$ une fonction convexe K-lipschitzienne partout finie. Alors*

$$\|r\|_\infty \leq K,\ \forall\ r \in \partial\theta(z),\ \forall\ z \in \mathbb{R}^p.$$

Preuve Soit $z \in \mathbb{R}^p$. Soit $r \in \partial\theta(z)$. Comme θ est une fonction convexe $K-$lipschitzienne partout finie, on a

$$(x-z)^T r \leq \theta(x) - \theta(z), \forall x \in \mathbb{R}^p \text{ (propriété de convexité)}$$

et

$$|\theta(x) - \theta(z)| \leq K \|x - z\|, \forall x \in \mathbb{R}^p \text{ (propriété de lipschitz)}.$$

Alors avec $x^i = z + r_i e^i$ (e^i, $i = 1, ..., n$ représentent les éléments de la base canonique de $\mathbb{R}^n$), nous obtenons

$$r_i^2 \leq \theta(x^i) - \theta(z)$$

et

$$\left|\theta(x^i) - \theta(z)\right| \leq K |r_i|, \quad i = 1, ...n.$$

En combinant ces deux dernières inégalités nous obtenons finalement

$$r_i^2 \leq \theta(x^i) - \theta(z) = \left|\theta(x^i) - \theta(z)\right| \leq K\left|r_i\right|,\ i = 1, ..., n,$$

donc

$$|r_i| \leq K,\ i = 1, ..., n.$$

d'où $\|r\|_\infty \leq K$. □

Nous allons maintenant appliquer les techniques de reformulation aux problèmes $(\bar{P}^\lambda)$ et $(\bar{Q}^k)$.

7.5.1 Cas ℓ_0-régularisé $(\bar{P}^\lambda)$

Rappelons que $(\bar{P}^\lambda)$ est donné par

$$(\bar{P}^\lambda) \quad \min\left\{f(x,y) + \lambda\|x\|_0 : x \in \mathbb{R}^n, y \in \mathbb{R}^m\right\}.$$

Pour $\tau > 0$, considérons le problème non convexe

$$(\bar{P}_\tau^\lambda)\begin{cases} \min & \psi_\tau(x,y,u) := f(x,y) + \lambda e^T u + \tau\left|x\right|^T(e-u) \\ s.t. & x \in \mathbb{R}^n, y \in \mathbb{R}^m, u \in [0,1]^n. \end{cases}$$

Une formulation DC de $(\bar{P}_\tau^\lambda)$ est donnée par

$$\min\left\{G(x,y,u) - H(x,y,u) : (x,y,u) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n\right\}, \tag{7.50}$$

où

$$G(x,y,u) := f(x,y) + \frac{1}{2}\tau x^T x + \frac{1}{2}\tau u^T u + \tau e^T\left|x\right| + \lambda e^T u + \chi_{[0,1]^n}(u), \tag{7.51}$$

et

$$H(x,y,u) := \frac{1}{2}\tau\sum_{i=1}^n(|x_i| + |u_i|)^2. \tag{7.52}$$

Comme le minimum de f est atteint, les problèmes $(\bar{P}^\lambda)$ et $(\bar{P}_\tau^\lambda)$ admettent au moins une solution. Et alors toute solution de $(\bar{P}_\tau^\lambda)$ vérifie nécessairement les conditions KKT généralisées

$$\partial H(x,y,u) \cap \partial G(x,y,u) \neq \varnothing.$$

On a

$$(X,Y,U) \in \partial G(x,y,u) \ \Leftrightarrow\ X = \tau x + v + \tau z,\ \ Y = w,\ \ U = \tau u + \lambda e + p, \tag{7.53}$$

$$(v,w) \in \partial f(x,y),\ \ z \in \partial e^T\left|x\right|,\ \ p \in \partial\chi_{[0,1]^n}(u),$$

et

$$(X,Y,U) \in \partial H(x,y,u) \ \Leftrightarrow\ X = \tau r,\ \ Y = 0,\ \ U = \tau s, \tag{7.54}$$

$$\begin{pmatrix} r_i \\ s_i \end{pmatrix} \in \begin{cases} \left\{ \begin{pmatrix} x_i + u_i \\ x_i + u_i \end{pmatrix} \right\} & \text{if } x_i u_i > 0, \\ \left\{ \begin{pmatrix} x_i - u_i \\ -x_i + u_i \end{pmatrix} \right\} & \text{if } x_i u_i < 0, \quad i = 1, ..., n. \\ \left[\begin{pmatrix} x_i + u_i \\ x_i + u_i \end{pmatrix}, \begin{pmatrix} x_i - u_i \\ -x_i + u_i \end{pmatrix} \right] & \text{if } x_i u_i = 0, \end{cases}$$

La proposition suivante établit le lien entre $(\bar{P}^\lambda)$ et $(\bar{P}^\lambda_\tau)$.

Proposition 7.5.1.1 *Soit $\tau > K$. Alors $(\bar{P}^\lambda)$ et $(\bar{P}^\lambda_\tau)$ sont équivalents dans le sens :*
- *si $(\bar{x}, \bar{y})$ est une solution de $(\bar{P}^\lambda)$, alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{y}, \bar{u})$ est une solution de $(\bar{P}^\lambda_\tau)$.*
- *si $(\bar{x}, \bar{y}, \bar{u})$ est une solution de $(\bar{P}^\lambda_\tau)$, alors $(\bar{x}, \bar{y})$ est une solution de $(\bar{P}^\lambda)$.*

Preuve Soit $\tau > K$. Soit $(\bar{x}, \bar{y}, \bar{u})$ une solution de $(\bar{P}^\lambda_\tau)$ et $(\tilde{x}, \tilde{y})$ une solution de $(\bar{P}^\lambda)$. Posons $\tilde{u}_i := 1$ si $\tilde{x}_i \neq 0$ et $\tilde{u}_i := 0$ sinon.
Nous allons d'abord montrer que $\psi_\tau(\bar{x}, \bar{y}, \bar{u}) = f(\bar{x}, \bar{y}) + \lambda \|\bar{x}\|_0$ et ensuite que $(\bar{x}, \bar{y})$ et $(\tilde{x}, \tilde{y}, \tilde{u})$ sont solutions de $(\bar{P}^\lambda)$ et $(\bar{P}^\lambda_\tau)$ respectivement. On pose

$$F := \{i : \bar{x}_i \neq 0, \bar{u}_i = 0\}, \; G := \{i : \bar{x}_i = 0, \bar{u}_i > 0\} \text{ et } I := \{i : \bar{x}_i(1 - \bar{u}_i) \neq 0\}.$$

$(\bar{x}, \bar{y}, \bar{u})$ vérifie les conditions KKT généralisées appliquées à (7.50)

$$\partial H(x, y, u) \cap \partial G(x, y, u) \neq \varnothing. \tag{7.55}$$

Montrons que $F = \varnothing$ et $G = \varnothing$. Nous procédons par contradiction.
Supposons $F \neq \varnothing$. Soit $i_0 \in F$. On a $\bar{x}_{i_0} \neq 0$ et $\bar{u}_{i_0} = 0$. D'après les conditions KKT généralisées (7.55), il existe $(\bar{v}, \bar{w}) \in \partial f(\bar{x}, \bar{y})$ tel que

$$\bar{v}_{i_0} + \tau sign(\bar{x}_{i_0}) = 0. \tag{7.56}$$

Comme $|\bar{v}_{i_0}| \leq K$ (lemme 7.5.1), (7.56) contredit $\tau > K$. Donc $F = \varnothing$.
Supposons $G \neq \varnothing$ et soit $i_0 \in G$. Comme $\bar{x}_{i_0} = 0$ et $\bar{u}_{i_0} > 0$, les conditions (7.55) impliquent qu'il existe $\gamma_{i_0} \geq 0$ tel que

$$\lambda + \gamma_{i_0} = 0.$$

Ce qui contredit $\lambda > 0$, donc $G = \varnothing$.
Maintenant considérons les cas suivants : $I = \varnothing$ et $I \neq \varnothing$.

- Cas $I = \varnothing$: comme $G = \varnothing$, on a $e^T \bar{u} = \|\bar{x}\|_0$ et $|\bar{x}|^T (e - \bar{u}) = 0$, donc

$$\phi_\tau(\bar{x}, \bar{y}, \bar{u}) = f(\bar{x}, \bar{y}) + \lambda \|\bar{x}\|_0 .$$

- Cas $I \neq \varnothing$: soit $i_0 \in I$. Comme $F = \varnothing$, on a $0 < \bar{u}_{i_0} < 1$ et $\bar{x}_{i_0} \neq 0$. De (7.55) nous tirons $|\bar{x}_{i_0}| = \frac{\lambda}{\tau}$ et alors

$$|\bar{x}_i| = \frac{\lambda}{\tau}, \forall i \in I.$$

$$\begin{aligned}\phi_\tau(\bar{x},\bar{y},\bar{u}) &= f(\bar{x},\bar{y}) + \lambda e^T\bar{u} + \tau\left|\bar{x}\right|^T(e-\bar{u})\\ &= f(\bar{x},\bar{y}) + \lambda e^T\bar{u} + \tau\textstyle\sum_{i\in I}\left|\bar{x}_i\right|(1-\bar{u}_i)\\ &= f(\bar{x},\bar{y}) + \lambda e^T\bar{u} + \lambda\textstyle\sum_{i\in I}(1-\bar{u}_i)\\ &= f(\bar{x},\bar{y}) + \lambda(\textstyle\sum_{i\notin I}\bar{u}_i + \sum_{i\in I}1).\end{aligned}$$

Comme $G=\varnothing$, nous obtenons finalement

$$\phi_\tau(\bar{x},\bar{y},\bar{u}) = f(\bar{x},\bar{y}) + \lambda\left\|\bar{x}\right\|_0.$$

On a alors

$$\phi_\tau(\tilde{x},\tilde{y},\tilde{u}) = f(\tilde{x},\tilde{y}) + \lambda\left\|\tilde{x}\right\|_0 \le f(\bar{x},\bar{y}) + \lambda\left\|\bar{x}\right\|_0 = \phi_\tau(\bar{x},\bar{y},\bar{u}) \le \phi_\tau(\tilde{x},\tilde{y},\tilde{u}),$$

et donc

$$\phi_\tau(\tilde{x},\tilde{y},\tilde{u}) = f(\tilde{x},\tilde{y}) + \lambda\left\|\tilde{x}\right\|_0 = f(\bar{x},\bar{y}) + \lambda\left\|\bar{x}\right\|_0 = \phi_\tau(\bar{x},\bar{y},\bar{u}).$$

Ce qui termine la preuve. □

7.5.2 Cas avec contrainte ℓ_0 $(\bar{Q}^k)$

Rappelons que $(\bar{Q}^k)$ est donné par

$$(\bar{Q}^k)\quad \min\left\{f(x,y):\ \left\|x\right\|_0\le k,\ x\in\mathbb{R}^n, y\in\mathbb{R}^m\right\},$$

Pour $t>0$, considérons le problème non convexe

$$(\bar{Q}^k_t)\ \begin{cases}\min_{(x,u)} & \psi_t(x,y,u) := f(x,y) + t\left|x\right|^T(e-u)\\ s.t. & e^Tu\le k,\\ & x\in\mathbb{R}^n,\\ & y\in\mathbb{R}^m,\\ & u\in[0,1]^n.\end{cases}$$

C'est un programe DC avec la formulation DC suivante

$$\min\left\{G(x,y,u)-H(x,y,u):(x,y,u)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^n\right\}, \tag{7.57}$$

où

$$G(x,y,u) := f(x,y) + \frac{1}{2}tx^Tx + \frac{1}{2}tu^Tu + te^T\left|x\right| + \chi_C(u), \tag{7.58}$$

$$H(x,y,u) := \frac{1}{2}t\sum_{i=1}^n(\left|x_i\right|+\left|u_i\right|)^2, \tag{7.59}$$

$C := \left\{u\in[0,1]^n:\ e^Tu\le k\right\}$.

$$(X,Y,U)\in\partial G(x,y,u)\ \Leftrightarrow\ X=tx+v+tz,\ \ Y=w,\ \ U=tu+p, \tag{7.60}$$

$$(v,w)\in\partial f(x,y),\ \ z\in\partial e^T\left|x\right|,\ \ p\in\partial\chi_C(u),$$

et

$$(X,Y,U) \in \partial H(x,y,u) \;\Leftrightarrow\; X = tr,\;\; Y = 0,\;\; U = ts, \tag{7.61}$$

$$\begin{pmatrix} r_i \\ s_i \end{pmatrix} \in \begin{cases} \left\{ \begin{pmatrix} x_i + u_i \\ x_i + u_i \end{pmatrix} \right\} & \text{if } x_i u_i > 0, \\ \left\{ \begin{pmatrix} x_i - u_i \\ -x_i + u_i \end{pmatrix} \right\} & \text{if } x_i u_i < 0, \quad i = 1, ..., n. \\ \left[\begin{pmatrix} x_i + u_i \\ x_i + u_i \end{pmatrix}, \begin{pmatrix} x_i - u_i \\ -x_i + u_i \end{pmatrix} \right] & \text{if } x_i u_i = 0, \end{cases}$$

La proposition suivante énonce le lien entre $(\bar{Q}_t^k)$ et $(\bar{Q}_t^k)$.

Proposition 7.5.2.1 *Soit $t > K$. Alors $(\bar{Q}^k)$ et $(\bar{Q}_t^k)$ sont équivalents dans le sens :*
- *si $(\bar{x}, \bar{y})$ est une solution de $(\bar{Q}^k)$, alors il existe $\bar{u}$ tel que $(\bar{x}, \bar{y}, \bar{u})$ est une solution de $(\bar{Q}_t^k)$.*
- *si $(\bar{x}, \bar{y}, \bar{u})$ est une solution de $(\bar{Q}_t^k)$, alors $(\bar{x}, \bar{y})$ est une solution de $(\bar{Q}^k)$.*

Preuve Soit $t > K$. Soit $(\bar{x}, \bar{y}, \bar{u})$ une solution de $(\bar{Q}_t^k)$ et soit $(\tilde{x}, \tilde{y})$ une solution de $(\bar{Q}^k)$. Posons $\tilde{u}_i := 1$ si $\tilde{x}_i \neq 0$ et $\tilde{u}_i := 0$ sinon.
Nous allons d'abord montrer que $|\bar{x}|^T (e - \bar{u}) = 0$ et $\|\bar{x}\|_0 \leq k$. Pour cela posons

$$F := \{i : \bar{x}_i \neq 0, \bar{u}_i = 0\},\; G := \{i : \bar{x}_i = 0, \bar{u}_i > 0\} \text{ et } I := \{i : \bar{x}_i(1 - \bar{u}_i) \neq 0\}.$$

$(\bar{x}, \bar{u})$ vérifie les conditions KKT généralisées appliquées à (7.57)

$$\partial H(x,y,u) \cap \partial G(x,y,u) \neq \varnothing. \tag{7.62}$$

Nous allons d'abord montrer que $F = \varnothing$. Nous procédons par contradiction. Supposons $F \neq \varnothing$ et soit $i_0 \in F$. On a $\bar{x}_{i_0} \neq 0$ et $\bar{u}_{i_0} = 0$. D'après les conditions KKT généralisées (7.62), il existe $(\bar{v}, \bar{w}) \in \partial f(\bar{x}, \bar{y})$ tel que

$$\bar{v}_{i_0} + t sign(\bar{x}_{i_0}) = 0. \tag{7.63}$$

Comme $|\bar{v}_{i_0}| \leq K$ (lemma 7.5.1), (7.63) contredit $\tau > K$. Donc $F = \varnothing$.
Montrons que $I = \varnothing$. Nous procédons par contradiction. On suppose $I \neq \varnothing$.
Soit $i_0 \in I$, comme $F = \varnothing$ on a $\bar{x}_{i_0} \neq 0$ et $0 < \bar{u}_{i_0} < 1$. Les conditions KKT (7.62) donnent alors

$$|\bar{x}_{i_0}| = \frac{\eta}{t},\; \eta > 0,\; e^T \bar{u} = k,$$

et donc

$$|\bar{x}_i| = \frac{\eta}{t},\; \forall i \in I. \tag{7.64}$$

Considérons maintenant les cas $G \neq \varnothing$ et $G = \varnothing$.
- $G \neq \varnothing$. Soit $g \in G$ i.e. $\bar{x}_g = 0$, et $\bar{u}_g > 0$. Alors les conditions KKT (7.62) donnent

  $$\eta + \gamma_g = 0 \;\text{ et }\; \gamma_g \geq 0.$$

  Ce qui contredit $\eta > 0$.

– $G = \varnothing$. A partir de $e^T\bar{u} = k$, nous tirons $\sum_{i\in I}\bar{u}_i = k - \sum_{i\notin I}\bar{u}_i$.
Comme $G = \varnothing$, on a $\sum_{i\notin I}\bar{u}_i \in \mathbb{N}$ et donc $\sum_{i\in I}\bar{u}_i \in \mathbb{N}$. De plus, $\sum_{i\in I}\bar{u}_i > 0$ $(F = \varnothing)$, ainsi

$$\sum_{i\in I}\bar{u}_i \in \{1, ..., k\}.$$

On pose $c := \sum_{i\in I}\bar{u}_i$. Il est clair que $c \in \{1, ..., k\}$ et $c < Card(I)$.
Soit $\{i_1, i_2, ..., i_c\} \subset I$ et soit $\hat{u}$ le point défini par

$$\hat{u}_i := \begin{cases} \bar{u}_i & \forall i \notin I, \\ 1 & \forall i \in \{i_1, i_2, ..., i_c\}, \\ 0 & \text{ailleurs.} \end{cases}$$

On a $\hat{u} \in [0, 1]^n$, $e^T\hat{u} = k$ et

$$t\left|\bar{x}\right|^T(e-\bar{u}) = t\sum_{i\in I}\left|\bar{x}_i\right|(1-\bar{u}_i) = \eta\sum_{i\in I}(1-\bar{u}_i) = \eta\sum_{i\in I}(1-\hat{u}_i) = t\left|\bar{x}\right|^T(e-\hat{u}). \tag{7.65}$$

Donc $(\bar{x}, \hat{u})$ est une solution de (Q^k_τ) et alors vérifie les conditions KKT (7.62).
Soit $r \in I \setminus \{i_1, i_2, ..., i_c\}$, on a $\bar{x}_r \neq 0$ et $\hat{u}_r = 0$. D'après les conditions KKT (7.62), il existe $(\bar{v}, \bar{w}) \in \partial f(\bar{x}, \bar{y})$ tel que

$$\bar{v}_r + tsign(\bar{x}_r) = 0.$$

Cette dernière condition est identique à la condition (7.63) et donc contredit $t > K$.

Parsuite $I = \varnothing$. Ainsi $\left|\bar{x}\right|^T(e - \bar{u}) = 0$ et $\left\|\bar{x}\right\|_0 \leq k$.
On a

$$\psi_t(\tilde{x}, \tilde{y}, \tilde{u}) = f(\tilde{x}, \tilde{y}) \leq f(\bar{x}, \bar{y}) = \psi_\tau(\bar{x}, \bar{y}, \bar{u}) \leq \psi_\tau(\tilde{x}, \tilde{y}, \tilde{u}),$$

donc

$$\psi_t(\tilde{x}, \tilde{y}, \tilde{u}) = f(\tilde{x}, \tilde{y}) = f(\bar{x}, \bar{y}) = \psi_\tau(\bar{x}, \bar{y}, \bar{u}).$$

□

Les techniques de reformulation précédentes nous ont permis d'obtenir des problèmes équivalents aux problèmes de parcimonie

$$(P) \quad \begin{cases} \min_x & \left\|x\right\|_0 \\ s.t. & Ax = b, \\ & x \in \mathbb{R}^n, \end{cases}$$

$$(P^\lambda) \quad \begin{cases} \min_x & \frac{1}{2}\left\|Ax - b\right\|_2^2 + \lambda\left\|x\right\|_0 \\ s.t. & x \in \mathbb{R}^n, \end{cases}$$

et

$$(Q^k) \quad \begin{cases} \min_x & \frac{1}{2}\left\|Ax - b\right\|_2^2 \\ s.t. & \left\|x\right\|_0 \leq k \\ & x \in \mathbb{R}^n. \end{cases}$$

Les problèmes non convexes obtenus possèdent des structures DC voire quadratiques et donc peuvent être attaqués par des méthodes de résolution en optimisation continue notamment la programmation quadratique et la programmation DC. La structure DC se prête bien à l'algorithme DCA et aux algorithmes combinés DCA, SE (branch and Bound) et relaxation DC.

7.6 DCA pour les différentes reformulations DC

Nous présentons l'algorithme DCA appliqué aux reformulations DC (P_τ^λ) et (Q_t^k). Une fois que le problème est mis sous forme DC

$$\min \{G(x,u) - H(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{7.66}$$

où G et H sont des éléments de $\Gamma_0(\mathbb{R}^n \times \mathbb{R}^n)$, DCA consiste à calculer les deux suites primale et duale $\left\{(x^l, u^l)\right\}$ et $\left\{(X^l, U^l)\right\}$ définies par

$$(X^l, U^l) \in \partial H(x^l, u^l), (x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l). \tag{7.67}$$

De façon schématique

$$\begin{array}{lcl} (x^l, u^l) & \longrightarrow & (X^l, U^l) \in \partial H(x^l, u^l) \\ & \swarrow & \\ (x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l) & \longrightarrow & (X^{l+1}, U^{l+1}) \in \partial H(x^{l+1}, u^{l+1}). \end{array}$$

Etant donné que toute fonction DC possède une infinité de décompositions DC, tout programme DC possède alors une infinité de versions de DCA. Dans notre étude nous explicitons l'algorithme DCA obtenu avec les décompositions précédentes pour chacune des reformulations DC (P_τ^λ) et (Q_t^k).

La question du choix d'une décomposition optimale, c'est à dire celle qui pourrait donner une solution globale du problème, reste une question ouverte. Ce choix dépendra sans doute de la structure spécifique du problème considéré. Ainsi, on se contente très souvent d'une décomposition dont le calcul des suites générées par DCA est simple et pas trop cher en temps de calcul.

A coté de cette question sur le choix de la décomposition se pose une autre question concernant le choix d'un bon point initial. Cette dernière question est aussi ouverte. On se contente aussi d'utiliser des techniques d'approximation et de relaxation afin de trouver un bon point initial pour DCA qui utilise ses propriétés de décroissance et de convergence pour trouver rapidement une bonne solution. On peut aussi utiliser des techniques de redémarrage (restarting techniques) de DCA afin d'améliorer la meilleure solution obtenue jusqu'à maintenant.

Une autre technique consiste à combiner DCA avec des méthodes d'optimisation globale telles que la méthode Séparation et Evaluation (SE) pour assurer la globalité de la solution. Dans ce cas, DCA intervient pour accélérer la convergence en permettant de trouver très vite une solution globale.

7.6.1 DCA pour (P_τ^λ)

Rappelons que (P_τ^λ) est défini par

$$(P_\tau^\lambda) \quad \begin{cases} \min_{(x,u)} & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u\tau\, |x|^T (e-u) \\ s.t. & x \in \mathbb{R}^n, \\ & y \in \mathbb{R}^m, \\ & u \in [0,1]^n, \end{cases}$$

et qu'une formulation DC est donnée par

$$\min\{G(x,u) - H(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{7.68}$$

où

$$G(x,u) := \frac{1}{2}(\tau+\beta)x^T x + \frac{1}{2}\tau u^T u + \tau e^T |x| + \chi_C(u), \tag{7.69}$$

$$H(x,u) := \frac{1}{2}\tau \sum_{i=1}^{n}(|x_i| + |u_i|)^2 + \frac{1}{2}\beta x^T x - \frac{1}{2}\|Ax-b\|_2^2, \tag{7.70}$$

$C := [0,1]^n$, et $\beta \geq \lambda_{\max}(A^T A)$.
Le calcul de (X^l, U^l) est donné par (7.12). Ici nous donnons seulement les détails pour le calcul de (x^{l+1}, u^{l+1}).
$(x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l)$ si et seulement si

$$(x^{l+1}, u^{l+1}) \in \arg\min\left\{G(x,u) - X^{lT}x - U^{lT}u : x \in \mathbb{R}^n, u \in \mathbb{R}^n\right\}$$

c'est à dire

$$x^{l+1} = \arg\min\left\{\frac{1}{2}(\tau+\beta)x^T x + \tau e^T |x| - X^{lT}x : x \in \mathbb{R}^n\right\}, \tag{7.71}$$

$$u^{l+1} = \arg\min\left\{\frac{1}{2}\tau u^T u + \lambda e^T u - U^{lT}u : u \in [0,1]^n\right\}. \tag{7.72}$$

u^{l+1} est donc la projection de $\frac{1}{\tau}(U^l - \lambda e)$ sur $[0,1]^n$, i.e.,

$$u^{l+1} = P_{[0,1]^n}(\frac{1}{\tau}(U^l - \lambda e)), \tag{7.73}$$

et $0 \in \partial\left[\frac{1}{2}(\tau+\beta)x^T x + \tau e^T |x| - X^{lT}x\right](x^{l+1})$, i.e.,

$$x_i^{l+1} = \begin{cases} \frac{X_i^l - \tau}{\tau+\beta} & \text{if } X_i^l > \tau, \\ \frac{X_i^l + \tau}{\tau+\beta} & \text{if } X_i^l < -\tau, \\ 0 & \text{if } -\tau \leq X_i^l \leq \tau, \quad i = 1, ..., n. \end{cases}$$

Un résumé de DCA appliqué au problème (P_τ^λ) est donné par l'algorithme 4.

Algorithm 4 DCA pour (P_τ^λ)

Entrées : $A \in M_{m,n}(\mathbb{R})$, $b \in \mathbb{R}^m$, $\lambda > 0$, $\tau > 0$, $\beta \geq \lambda_{\max}(A^TA)$, $(x^0, u^0) \in C$ et la tolérance $\epsilon > 0$.
Sortie : (x, u)
Initialisation $l := 0$.
Répéter

1. $(X^l, U^l) \in \partial H(x^l, u^l)$ (7.12).
2. Calculer x^{l+1} par

$$x_i^{l+1} := \begin{cases} \frac{X_i^l - \tau}{\tau+\beta} & \text{si } X_i^l > \tau, \\ \frac{X_i^l + \tau}{\tau+\beta} & \text{if } X_i^l < -\tau, \\ 0 & \text{if } -\tau \leq X_i^l \leq \tau, \end{cases} \qquad i = 1, ..., n.$$

et u^{l+1} par

$$u_i^{l+1} := \begin{cases} 0 \text{ if } \frac{1}{\tau}(U_i^l - \lambda) \leq 0, \\ 1 \text{ if } \frac{1}{\tau}(U_i^l - \lambda) \geq 1, \\ \frac{1}{\tau}(U_i^l - \lambda) \text{ otherwise,} \end{cases} \qquad i = 1, ..., n.$$

Jusqu'à $\left|\phi_\tau(x^{l+1}, u^{l+1}) - \phi_\tau(x^l, u^l)\right| / \left|\phi_\tau(x^l, u^l)\right| \leq \epsilon$.
$x := x^l$ et $u := u^l$.

7.6.2 DCA pour (Q_t^k)

Rappelons que (Q_t^k) est défini par

$$(Q_t^k) \quad \begin{cases} \min_{(x,u)} & \frac{1}{2}\|Ax - b\|_2^2 + t\,|x|^T (e - u) \\ s.t. & e^T u \leq k, \\ & x \in \mathbb{R}^n, \\ & y \in \mathbb{R}^m, \\ & u \in [0,1]^n, \end{cases}$$

et qu'une reformulation DC est donnée par

$$\min \{G(x,u) - H(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{7.74}$$

où

$$G(x,u) := \frac{1}{2}(\tau+\beta)x^Tx + \frac{1}{2}\tau u^Tu + \tau e^T|x| + \chi_C(u), \tag{7.75}$$

$$H(x,u) := \frac{1}{2}\tau \sum_{i=1}^n (|x_i| + |u_i|)^2 + \frac{1}{2}\beta x^Tx - \frac{1}{2}\|Ax - b\|_2^2, \tag{7.76}$$

avec $C := \left\{u \in [0,1]^n : \ e^Tu \leq k\right\}$, et $\beta \geq \lambda_{\max}(A^TA)$.
Le calcul de (X^l, U^l) est donné par (7.20). Ici nous donnons seulement les détails

pour le calcul de (x^{l+1}, u^{l+1}).
$(x^{l+1}, u^{l+1}) \in \partial G^*(X^l, U^l)$ si et seulement si

$$(x^{l+1}, u^{l+1}) \in \arg\min \left\{ G(x,u) - X^{lT}x - U^{lT}u : x \in \mathbb{R}^n, u \in \mathbb{R}^n \right\}$$

c'est à dire

$$x^{l+1} = \arg\min \left\{ \frac{1}{2}(\tau+\beta)x^T x + \tau e^T |x| - X^{lT}x : x \in \mathbb{R}^n \right\}, \tag{7.77}$$

$$u^{l+1} = \arg\min \left\{ \frac{1}{2} t u^T u - U^{lT}u : \ e^T u \leq k, \ u \in [0,1]^n \right\}. \tag{7.78}$$

$0 \in \partial \left[\frac{1}{2}(t+\beta)x^T x + te^T |x| - X^{lT}x \right] (x^{l+1})$, i.e.,

$$x_i^{l+1} = \begin{cases} \frac{X_i^l - t}{t+\beta} & \text{if } X_i^l > t, \\ \frac{X_i^l + t}{t+\beta} & \text{if } X_i^l < -t, \\ 0 & \text{if } -t \leq X_i^l \leq t, \ \ i = 1, ..., n, \end{cases}$$

et u^{l+1} est donc la projection de $\frac{1}{t}U^l$ sur $C = \left\{ u \in [0,1]^n : \ e^T u \leq k \right\}$. Cette projection peut être explicitée en utilisant les conditions KKT classiques appliquées au problème convexe (7.78). Elles sont données par

$$tu_i - U_i^l + \alpha + \gamma_i - \delta_i = 0, \ i = 1, ..., n,$$

$$\gamma_i(u_i - 1) = 0, \ \delta_i u_i = 0, \ u_i \in [0,1], \ \gamma_i \geq 0, \ \delta_i \geq 0, \ i = 1, ..., n,$$
$$\alpha(e^T u - k) = 0, \ e^T u \leq k, \ \alpha \geq 0.$$

Ces conditions étant équivalentes aux conditions suivantes

$$u(\alpha)_i = \min[\frac{1}{t}\max(U_i^l - \alpha, 0), 1], \ i = 1, ..., n, \tag{7.79}$$

$$\gamma(\alpha)_i = \max[-t + \max(U_i^l - \alpha, 0), 0], \ i = 1, ..., n, \tag{7.80}$$

$$\delta(\alpha)_i = \max(\alpha - U_i^l, 0), \ i = 1, ..., n, \tag{7.81}$$

$$\alpha(e^T u(\alpha) - k) = 0, \ e^T u(\alpha) \leq k, \ \alpha \geq 0. \tag{7.82}$$

Il reste donc à déterminer α vérifiant (7.82).
Noter que la formule (7.79) s'exprime aussi sous la forme

$$u(\alpha)_i = \begin{cases} 1 & \text{si } \alpha \leq U_i^l - t, \\ \frac{U_i^l - \alpha}{t} & \text{si } U_i^l - t < \alpha \leq U_i^l, \quad i = 1, ..., n. \\ 0 & \text{si } U_i^l < \alpha, \end{cases} \tag{7.83}$$

Un algorithme proposé par Helgason et al. [61] permet de déterminer un α vérifiant $e^T u(\alpha) = k$. Ici, on va utiliser une version modifiée de cet algorithme pour déterminer α vérifiant (7.82). Pour cela il nous faudra définir le terme suivant

$$z(\alpha) = e^T u(\alpha). \tag{7.84}$$

Il est clair que les fonctions $\alpha \mapsto u(\alpha)_i$, $i = 1, ..., n$, sont continues linéaires par morceaux et décroissantes. Comme la somme de functions préserve cette propriété, on a $\alpha \mapsto z(\alpha)$ qui est aussi continue linéaire par morceaux et décroissante.

Nous sommes en mesure, maintenant, d'énoncer une procédure de calcul de u^{k+1}, qu'on appellera procédure de Helgason et al. [61] :

- *si* $z(0) \leq k$, *alors prendre* $u^{k+1} = u(0)$,
- *sinon déterminer* α *par l'algorithme de Helgason et al. [61], et prendre* $u^{k+1} = u(\alpha)$.

Un résumé de DCA appliqué au problème (Q_τ^k) est donné par l'algorithme 5.

Algorithm 5 DCA pour (Q_t^k)

Entrées : $A \in M_{m,n}(\mathbb{R})$, $b \in \mathbb{R}^m$, $k \in \{1, ..., n\}$, $t > 0$, $\beta \geq \lambda_{\max}(A^T A)$, $(x^0, u^0) \in C$ et $\epsilon > 0$ la tolérence.
Sortie : (x, u)
Initialisation $l := 0$.
Répéter

1. $(X^l, U^l) \in \partial H(x^l, u^l)$ (7.20).
2. Calculer x^{l+1} par

$$x_i^{l+1} := \begin{cases} \frac{X_i^l - t}{t+\beta} & \text{si } X_i^l > t, \\ \frac{X_i^l + t}{t+\beta} & \text{if } X_i^l < -t, \\ 0 & \text{if } -t \leq X_i^l \leq t, \end{cases}$$

et u^{l+1} par la procédure de Helgason et al. [61].

Jusqu'à $\left|\psi_t(x^{l+1}, u^{l+1}) - \psi_t(x^l, u^l)\right| / \left|\psi_t(x^l, u^l)\right| \leq \epsilon$.
$x := x^l$ et $u := u^l$.

7.7 Une technique combinée DCA et S.E. via la relaxation DC pour (P_τ^λ)

Dans ce paragraphe, on s'intéresse à la résolution globale du problème (P^λ)

$$(P^\lambda) \quad \begin{cases} \min_x & f(x) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n, \end{cases}$$

à partir de sa reformulation

$$(P_\tau^\lambda) \begin{cases} \min & \phi_\tau(x, u) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda e^T u + \tau |x|^T (e - u) \\ s.t. & x \in \mathbb{R}^n, \\ & u \in [0, 1]^n, \end{cases}$$

$\tau > \tau_0(A, b, \lambda)$.
Pour cela nous allons utiliser une méthode combinée DCA, S.E et relaxation DC

appliquée au problème suivant

$$(\hat{P}_\tau^\lambda)\begin{cases}\min & \phi_\tau(x,u)=\frac{1}{2}\|Ax-b\|_2^2+\lambda e^T u+\tau\left|x\right|^T(e-u)\\ s.t. & x\in[-\Delta,\Delta]^n,\\ & u\in[0,1]^n.\end{cases}$$

Noter que pour Δ suffisamment grand, (P_τ^λ) et $(\hat{P}_\tau^\lambda)$ possèdent le même ensemble de solutions. Δ nous permet de borner l'ensemble admissible afin de pouvoir définir des relaxations convexes.

7.7.1 Relaxation DC de $(\hat{P}_\tau^\lambda)$

Une formulation DC de $(\hat{P}_\tau^\lambda)$ est donnée par

$$\min\left\{G(x,u)-H(x,u):(x,u)\in\mathbb{R}^n\times\mathbb{R}^n\right\}, \tag{7.85}$$

$$G(x,u):=\frac{1}{2}\|Ax-b\|_2^2+\lambda e^T u+\frac{1}{2}\tau x^T x+\frac{1}{2}\tau u^T u+\tau e^T\left|x\right|+\chi_C(x,u), \tag{7.86}$$

$$H(x,u):=\frac{1}{2}\tau\sum_{i=1}^n H_i(x,u)=\frac{1}{2}\tau\sum_{i=1}^n(|x_i|+|u_i|)^2, \tag{7.87}$$

$C:=[-\Delta,\Delta]^n\times[0,1]^n$. D'après le paragraphe 5.3.4, une minorante convexe $\hat{\psi}_\tau$ de ϕ_τ sur C est obtenue en additionnant G et $\frac{1}{2}\tau\overline{co}_C(-H)$. Du paragraphe 4.3, il suit

$$\overline{co}_{[-\Delta,\Delta]\times[0,1]}(-H_i)=-\left[(2\Delta+1)u_i+\Delta^2\right],\ i=1,...,n \tag{7.88}$$

et

$$\overline{co}_{[-\Delta,\Delta]^n\times[0,1]^n}(-H)=\sum_{i=1}^n\overline{co}_{[-\Delta,\Delta]\times[0,1]}(-H_i)=-\sum_{i=1}^n\left[(2\Delta+1)u_i+\Delta^2\right]. \tag{7.89}$$

Nous obtenons finalement le problème convexe relaxé suivant

$$\begin{cases}\min & \frac{1}{2}\|Ax-b\|_2^2+\lambda e^T u+\frac{1}{2}\tau x^T x+\frac{1}{2}\tau u^T u+\tau e^T\left|x\right|+\overline{co}_{[-\Delta,\Delta]^n\times[0,1]^n}(-H)\\ s.t. & x\in[-\Delta,\Delta]^n,\\ & u\in[0,1]^n.\end{cases}$$

7.7.2 Processus de séparation

Supposons que, sur un noeud dans notre procédure SE, nous avons à résoudre le problème

$$\begin{cases}\min & \phi_\tau(x,u)=\frac{1}{2}\|Ax-b\|_2^2+\lambda e^T u+\tau\left|x\right|^T(e-u)\\ s.t. & (x,u)\in M.\end{cases} \tag{7.90}$$

Le but de la séparation est de diviser l'ensemble convexe M en deux sous-ensembles convexes et de considérer le problème sur chacun de ces sous-ensembles afin de localiser un sous-ensemble contenant une solution globale sur M. En utilisant le

caractère binaire de la partie $\bar{u}$ d'une solution optimale globale $(\bar{x}, \bar{u})$, une solution globale peut être localisée avec les deux sous-ensembles suivants

$$M^1 = \{(x,u) \in M : \ u_i = 0\} \text{ et } M^2 = \{(x,u) \in M : \ u_i = 1\}, \tag{7.91}$$

où $i \in \{1, ..., n\}$ est fixé.
Plus explicitement, supposons que nous avons à résoudre le problème

$$(SP) \begin{cases} \min & \phi_\tau(x,u) = \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau |x|^T (e-u) \\ s.t. & x \in [-\Delta, \Delta]^n, \\ & u_i = 0 \text{ si } i \in I, \\ & u_i = 1 \text{ si } i \in J, \\ & 0 \le u_i \le 1 \text{ si } i \in K := \{1, ..., n\} \setminus I \cup J, \end{cases}$$

avec M représentant son ensemble admissible.

Soit (x^R, u^R) une solution optimale de la relaxation DC de (SP) et soit $\beta^R(M)$ la borne inférieure de (SP) donnée par la relaxation.

Si $\phi_\tau(x^R, u^R) = \beta^R(M)$, alors (x^R, u^R) est aussi une solution de (SP). Sinon, choisir un indice $i^* \in K$ et remplacer le problème (SP) par deux sous problèmes en posant $u_{i^*} = 0$ et $u_{i^*} = 1$, c'est à dire

$$(SP1) \begin{cases} \min & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau |x|^T (e-u) \\ s.t. & x \in [-\Delta, \Delta]^n, \\ & u_i = 0 \text{ si } i \in I \bigcup \{i^*\}, \\ & u_i = 1 \text{ si } i \in J, \\ & 0 \le u_i \le 1 \text{ si } i \in K \setminus \{i^*\} \end{cases}$$

et

$$(SP2) \begin{cases} \min & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau |x|^T (e-u) \\ s.t. & x \in [-\Delta, \Delta]^n, \\ & u_i = 0 \text{ si } i \in I, \\ & u_i = 1 \text{ si } i \in J \bigcup \{i^*\}, \\ & 0 \le u_i \le 1 \text{ si } i \in K \setminus \{i^*\}, \end{cases}$$

dont les ensembles admissibles sont respectivement M^1 et M^2.

L'indice i^* est choisi de sorte que le gap entre

$$-(\left|x_i^R\right| + \left|u_i^R\right|)^2$$

et son enveloppe convexe

$$\overline{co}_{[-\Delta,\Delta]\times[0,1]} \left[-(\left|x_i^R\right| + \left|u_i^R\right|)^2\right] = -\left[(2\Delta+1)u_i^R + \Delta^2\right],$$

soit maximal, c'est à dire,

$$i^* \in \arg\max \left\{-(\left|x_i^R\right| + \left|u_i^R\right|)^2 + \left[(2\Delta+1)u_i^R + \Delta^2\right] : \ i \in K\right\}. \tag{7.92}$$

7.7.3 Processus d'évaluation

Il consiste à estimer les bornes inférieures et les bornes supérieures durant la procédure SE.

7.7.3.1 Bornes inférieures

Considérons les deux sous problèmes précédents $(SP1)$ et $(SP2)$ avec leur ensemble admissible M^1 et M^2 respectivement. Dans le cas où l'ensemble d'indices $K \setminus \{i^*\}$ est vide, ces sous problèmes sont convexes et peuvent être résolus directement, dans le cas contraintre il nous faut déterminer des bornes inférieures $\beta(M^1)$ et $\beta(M^2)$ pour les valeurs optimales des sous problèmes $(SP1)$ et $(SP2)$. Dans ce cas nous utilisons une relaxation DC pour déterminer ces bornes inférieures. Noter que les sous problèmes $(SP1)$ et $(SP2)$ sont de la même forme que le problème (SP). Dans la suite nous allons détailler comment construire une relaxation DC de (SP).

Considérons le problème (SP),

$$(SP)\begin{cases} \min & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau |x|^T (e-u) \\ s.t. & x \in [-\Delta, \Delta]^n, \\ & u_i = 0 \text{ si } i \in I, \\ & u_i = 1 \text{ si } i \in J, \\ & 0 \le u_i \le 1 \text{ si } i \in K. \end{cases}$$

(SP) peut s'écrire aussi

$$\begin{cases} \min & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau \sum_{i\in I} |x_i| + \tau \sum_{i\in K} |x_i|(1-u_i) \\ s.t. & x \in [-\Delta, \Delta]^n, \\ & u_i = 0 \text{ si } i \in I, \\ & u_i = 1 \text{ si } i \in J, \\ & 0 \le u_i \le 1 \text{ si } i \in K. \end{cases} \tag{7.93}$$

Une formulation DC de (7.93) est donnée par

$$\min\{G_{SP}(x,u) - H_{SP}(x,u) : (x,u) \in \mathbb{R}^n \times \mathbb{R}^n\}, \tag{7.94}$$

$$G_{SP}(x,u) := \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau \sum_{i\in I\cup K} |x_i| + \frac{1}{2}\tau \sum_{i\in K} x_i^2 + \frac{1}{2}\tau \sum_{i\in K} u_i^2 + \chi_M(x,u), \tag{7.95}$$

$$H_{SP}(x,u) := \frac{1}{2}\tau \sum_{i\in K} (|x_i| + |u_i|)^2, \tag{7.96}$$

où M représente l'ensemble admissible de (SP). Pour chaque $i \in K$, on a

$$\overline{co}_{[-\Delta,\Delta]\times[0,1]}(-(|x_i| + |u_i|)^2) = -\left[(2\Delta+1)u_i + \Delta^2\right], \tag{7.97}$$

et donc nous obtenons le problème convexe relaxé suivant

$$\begin{cases} \min & \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T u + \tau \sum_{i\in I\cup K} |x_i| + \frac{1}{2}\tau \sum_{i\in K} x_i^2 + \frac{1}{2}\tau \sum_{i\in K} u_i^2 \\ & -\frac{1}{2}\tau \sum_{i\in K} \left[(2\Delta+1)u_i + \Delta^2\right] \\ s.t. & x \in [-\Delta, \Delta]^n, \\ & u_i = 0 \text{ si } i \in I, \\ & u_i = 1 \text{ si } i \in J, \\ & 0 \le u_i \le 1 \text{ si } i \in K. \end{cases}$$

7.7.3.2 Amélioration de la borne supérieure

Pour $x \in \mathbb{R}^n$, on définit le vecteur $u(x) \in \{0,1\}^n$ par

$$u(x)_i := \begin{cases} 1 & \text{si } x_i \neq 0, \\ 0 & \text{sinon.} \end{cases} \quad i = 1, ..., n. \tag{7.98}$$

Soit (x^*, u^*) le point donnant la meilleure borne supérieure jusqu'à maintenant. Soit (x, u) un point obtenu lors du calcul d'une borne inférieure d'un sous probème durant la procédure. Si la valeur de $\phi_\tau(x, u)$ est plus petite que la borne supérieure actuelle, alors nous redémarrons DCA, avec comme point initial (x, u), pour obtenir un nouveau point $(\tilde{x}, \tilde{u})$. Si $(\tilde{x}, u(\tilde{x}))$ réalise une meilleure borne supérieure que (x^*, u^*), alors nous mettons à jour la borne supérieure.

7.7.3.3 Algorithme combiné DCA-SE

Nous présentons ici, un algorithme combiné DCA-SE et relaxation DC. Le but de l'algorithme est de résoudre globalement le problème (P^λ) en utilisant sa reformulation (P^λ_τ). L'intérêt de cette combinaison repose sur plusieurs points

- la procédure SE permet d'assurer l'optimalité de la solution trouvée mais elle est généralement trop lente,
- DCA grâce à son efficacité et sa rapidité, permet de trouver très rapidement une solution globale en proposant de bonnes bornes supérieures et ainsi accélère la convergence de SE,
- la relaxation DC quand à elle permet de calculer des bornes inférieures très rapidement.

L'algorithme permet aussi de tester l'optimalité de la solution proposée par DCA.

Sur l'algorithme 6, nous résumons l'algorithme combiné obtenu.

7.8 Tests numériques préliminaires

Dans ce paragraphe, nous présentons quelques tests numériques préliminaires concernant le problème

$$(P^\lambda) \quad \begin{cases} \min_x & f(x) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_0 \\ s.t. & x \in \mathbb{R}^n. \end{cases}$$

Nous utilisons DCA appliqué à la reformulation

$$(P^\lambda_\tau) \begin{cases} \min & \phi_\tau(x, u) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda e^T u + \tau |x|^T (e - u) \\ s.t. & x \in \mathbb{R}^n, \\ & u \in [0,1]^n, \end{cases}$$

$\tau > \tau_0(A, b, \lambda)$.

Nous présentons aussi quelques comparaisons avec l'approximation ℓ_1 et le modèle DC utilisant l'approximation ℓ_p ([52])

$$(\ell_1) \begin{cases} \min_x & \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \\ s.t. & x \in \mathbb{R}^n, \end{cases}$$

Algorithm 6 Algorithme combiné DCA-SE

Résoudre la relaxation DC pour obtenir (x^{M_0}, u^{M_0}) et une borne inférieure $\beta(M_0)$.
Lancer DCA avec le point initial (x^{M_0}, u^{M_0}) pour obtenir $(\tilde{x}, \tilde{u})$.
Poser $x \leftarrow \tilde{x}$ et $u \leftarrow u(\tilde{x})$ et $UB \leftarrow \phi_\tau(\tilde{x}, u(\tilde{x}))$.
Si $\beta(M_0) = UB$**, alors** (x, u) **est une solution optimal**.
sinon
Tant que VRAI faire
Choisir un sous-ensemble $M_k \in \mathcal{S}$ tel que $\beta(M_k) = \min\{\beta(M) : M \in \mathcal{S}\}$.
Diviser le sous-ensemble M_k en deux sous-ensembles M_k^1 et M_k^2 via (7.91) et (7.92).
Résoudre les relaxations DC de (SP_i) pour obtenir $\beta(M_k^i)$ et $(x^{M_k^i}, u^{M_k^i})$, $i = 1, 2$.
Poser $\mathcal{S} \leftarrow \mathcal{S} \cup \{M_k^i : \beta(M_k^i) < UB, i = 1, 2\} \setminus \{M_k\}$
Si $\phi_\tau(x^{M_k^i}, u(x^{M_k^i})) < UB$, **alors**
Mettre à jour $UB \leftarrow \phi_\tau(x^{M_k^i}, u(x^{M_k^i}))$ et $x \leftarrow x^{M_k^i}$, $u \leftarrow u(x^{M_k^i})$.
Fin Si
Si $\phi_\tau(x^{M_k^i}, u^{M_k^i}) < UB$, **alors**
Lancer DCA avec le point initial $(x^{M_k^i}, u^{M_k^i})$ pour obtenir $(\tilde{x}, \tilde{u})$.
Si $\phi_\tau(\tilde{x}, u(\tilde{x})) < UB$, **alors**
Mettre à jour $UB \leftarrow \phi_\tau(\tilde{x}, u(\tilde{x}))$ et $x \leftarrow \tilde{x}$, $u \leftarrow u(\tilde{x})$.
Fin Si
Fin Si
Mettre à jour $\mathcal{S} \leftarrow \mathcal{S} \setminus \{M_j : \beta(M_j) > UB\}$
Si $(\mathcal{S} = \varnothing)$ ou $(UB = \min\{\beta(M) : M \in \mathcal{S}\})$, **alors**
STOP, (x, u) **est une solution optimale**.
sinon $k \leftarrow k + 1$.
Fin Si
Fin Tant que
Fin Si

et

$$(\ell_p) \begin{cases} \min_x & \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_p^p \\ s.t. & x \in \mathbb{R}^n. \end{cases}$$

Nous utilisons les données normalisées suivantes : bupa liver ($m = 345$, $n = 6$), breast cancer wisconsin (BCW) ($m = 683$, $n = 9$), ionosphere ($m = 351$, $n = 34$) et sonar ($m = 208$, $n = 60$) prises à partir du site UCI machine learning repository.

Sur les tableaux 7.2 et 7.3 sont représentés la valeur optimale globale obtenue par DCA-SE et la valeur de la fonction objectif donnée par DCA (seul) sur les données bupa et BCW. Tandis que sur les tableaux 7.4 et 7.6, nous présentons les résultats numériques obtenus par ℓ_1, ℓ_p et DCA concernant les valeurs suivantes

- $obj := \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_0$,
- $\|x\|_0$,

pour différentes valeurs de λ. Les temps de calcul correspondant sont présentés sur les tableaux 7.5 et 7.7.

Nous observons sur le tableau 7.2 que DCA (seul) peut trouver une solution

TABLE 7.2 – Bupa : valeur fonction objectif et parcimonie. Nombre de DCA représente le nombre de redémarrage de DCA pour obtenir une borne supérieure égale à la valeur optimale.

	DCA-SE		DCA seulement
λ	Valeur Optimale	Nombre de DCA	Obj.
1.00	155.6790	3	**155.6790**
2.00	160.6790	3	**160.6790**
3.00	165.6790	3	173.4018
4.00	167.8282	3	172.3373
5.00	169.8282	2	173.3373
6.00	171.8282	2	172.5000
7.00	172.5000	1	**172.5000**
8.00	172.5000	1	**172.5000**
9.00	172.5000	1	**172.5000**
10.00	172.5000	1	**172.5000**

optimale globale (deux premières et quatres dernières lignes du tableau) et que pour les autres lignes les solutions sont seulement locales. La même situation est observée sur le tableau 7.3 où toutes les solutions trouvées sont locales (non globales). Nous constatons aussi sur ces deux tableaux que le nombre de redémarrage de DCA pour obtenir une borne supérieure égale à la valeur optimale varie de 1 à 3 pour le premier et est de 4 pour le deuxième. DCA arrive à trouver rapidement une solution globale après quelques redémarrages. Cependant, il faut noter que la borne inférieure ne monte pas très vite et que la confirmation de la globalité de la solution est obtenue après un temps correspondant à la complexité au pire des cas (exponentielle).

Les résultats de DCA sur le tableau 7.4 sont significativement meilleurs que ℓ_1 et ℓ_p en terme de la valeur de la fonction objectif et de la parcimonie des solutions trouvées. Nous constatons aussi la même chose sur le tableau 7.6 en ce qui concerne la parcimonie des solutions. Cependant, les résultats sur les valeurs de la fonction objectif sont beaucoup plus nuancés. ℓ_p donne de meilleures valeurs pour la fonction objectif pour les valeurs $\lambda = 0.2, 0.3, 0.6, 0.7, 0.8$.

En terme de temps de calcul (tableaux 7.5 et 7.7), on constate que ℓ_1 est toujours plus rapide que les deux autres et que DCA est plus rapide que ℓ_p.

Au vu de ces différents résultats préliminaires, nous pouvons dire que DCA appliqué à nos reformulations permet de trouver de bon résultats dans certains cas. Des améliorations pourraient permettre d'obtenir de bien meilleurs résultats. En ce qui concerne l'algorithme combiné DCA-SE pour trouver une solution globale, des améliorations seront sans doute nécessaires concernant le calcul de bornes inférieures afin d'obtenir plus vite la confirmation de la globalité des solutions.

Table 7.3 – BCW : valeurs fonction objectif et parcimonie. Nombre de DCA représente le nombre de redémarrage de DCA pour obtenir une borne supérieure égale à la valeur optimale.

	DCA-SE		DCA seulement
λ	Valeur Optimale	Nombre de DCA	Obj.
1.00	85.6419	4	87.4526
2.00	90.0585	4	93.7425
3.00	94.0585	4	98.1059
4.00	97.9178	4	104.1058
5.00	100.9178	4	110.1058
6.00	103.9178	4	116.1058
7.00	106.9178	4	122.1057
8.00	109.4291	4	128.1058
9.00	111.4291	4	134.1058
10.00	113.4291	4	140.1058

7.9 Conclusion

Ce chapitre décrit quelques travaux effectués durant la thèse concernant les problèmes de parcimonie dans les modèles linéaires. D'une part, nous avons introduit une nouvelle manière de voir ces problèmes de parcimonie grâce à de nouvelles techniques de reformulations exactes. Nous avons pu dériver des propriétés intéressantes, des reformulations DC et aussi quadratiques grâce à la programmation DC. D'autre part, les résultats numériques préliminaires obtenus avec l'algorithme DCA sont prometteurs. Des améliorations sur l'algorithme DCA et l'algorithme combiné DCA-SE pourraient sans doute permettre d'obtenir de meilleurs résultats.

Nous envisageons dans le futur, d'investir les points suivants

- approfondissement des propriétés des reformulations obtenues,
- amélioration des algorithmes DCA et DCA-SE pour l'optimisation globale,
- développement de méthodes d'optimisation globales et de la programmation quadratique non convexe.

TABLE 7.4 – Ionosphere : valeurs fonction objectif et parcimonie. DCA est itulisé ici avec $t = 100\lambda$ avec 1000 itérations.

	ℓ_1		ℓ_p, $p = 0.5$		DCA	
λ	Obj.	$\|x\|_0$	Obj.	$\|x\|_0$	Obj.	$\|x\|_0$
0.10	78.6894	32	78.4856	31	**78.3320**	**28**
0.20	82.3117	32	81.4986	30	**80.8521**	**24**
0.30	85.5952	30	83.6536	26	**82.9820**	**20**
0.40	87.3144	25	85.8581	24	**84.8943**	**19**
0.50	89.4835	23	87.4089	21	**87.0322**	**18**
0.60	92.3808	23	89.6117	21	**88.8322**	**18**
0.70	96.0892	24	91.3585	20	**90.2217**	**16**
0.80	98.4222	23	92.7471	18	**91.9119**	**14**
0.90	99.5948	21	94.6457	18	**93.0385**	**13**
1.00	101.3451	20	96.5639	18	**95.7269**	**11**

TABLE 7.5 – Ionosphere : Temps de calcul(s).

λ	ℓ_1	ℓ_p, $p = 0.5$	DCA
0.10	0.2744	5.1584	0.4869
0.20	0.2865	36.3320	0.5843
0.30	0.3291	27.7302	0.6638
0.40	0.2720	25.1007	0.7099
0.50	0.2742	21.4650	0.6245
0.60	0.3733	22.6362	0.8074
0.70	0.3082	10.0085	0.8083
0.80	0.2837	2.9266	0.7688
0.90	0.2737	3.1117	0.8226
1.00	0.2538	3.3393	0.6412

Table 7.6 – sonar : valeurs fonction objectif et parcimonie. DCA est itulisé ici avec $t = 100\lambda$ avec 10000 itérations.

	ℓ_1		ℓ_p, $p = 0.5$		DCA	
λ	Obj.	$\|x\|_0$	Obj.	$\|x\|_0$	Obj.	$\|x\|_0$
0.10	46.1209	49	44.9413	46	**44.9363**	**38**
0.20	52.6615	47	**48.4554**	36	48.8715	**34**
0.30	58.1528	41	**53.0581**	31	53.9876	**29**
0.40	62.7578	39	56.7316	29	**56.7157**	**26**
0.50	66.9760	37	59.6172	28	**58.7554**	**24**
0.60	71.4199	36	**62.0920**	26	62.8408	**19**
0.70	73.6413	32	**63.8860**	23	64.7544	**17**
0.80	75.7419	29	**65.2779**	21	65.3350	**15**
0.90	78.1050	27	70.2545	21	**67.0218**	**14**
1.00	80.6230	26	71.6871	20	**67.7577**	**13**

Table 7.7 – Sonar : Temps de calcul(s).

λ	ℓ_1	ℓ_p, $p = 0.5$	DCA
0.10	0.6997	40.8855	1.9774
0.20	0.6925	31.7391	2.2013
0.30	0.5762	18.5078	1.6396
0.40	0.5363	15.7222	1.8380
0.50	0.4727	14.0655	2.0163
0.60	0.4470	11.8747	1.5497
0.70	0.4424	10.7696	1.5659
0.80	0.3971	4.6557	1.3155
0.90	0.3972	3.6118	1.4364
1.00	0.3927	8.7945	1.5078

Conclusion et Perspectives

Cette thèse s'inscrit dans le cadre de l'étude théorique et algorithmique des problèmes de parcimonie rencontrés en modélisation parcimonieuse de données. L'approche utilisée est la programmation DC et DCA qui est une approche déterministe pour résoudre des problèmes non convexes. Cette approche utilise des outils d'analyse convexe et exploite la structure DC de ces problèmes.

Pour chacun des problèmes de parcimonie rencontrés, nous avons pu les reformuler sous forme de problèmes ayant des strutures DC, voire sous forme quadratique, grâce à de nouvelles techniques de pénalité exacte en Programmation DC développées durant la thèse. Au point de vue théorique, ces techniques nous ont permis d'établir une nouvelle façon de voir ces problèmes et aussi de dégager de nouvelles propriétés et caractéristiques pour une meilleure compréhension et prise en main de ces problèmes. Ces propriétés seront sans doute utiles pour la résolution de ces problèmes. Il faut noter que ces problèmes de parcimonie sont difficiles à attaquer directement à cause du terme de parcimonie et qu'une grande difficulté concernant ces problèmes de parcimonie est la reformulation. Très souvent, les méthodes utilisées dans la littérature passent par des approximations continues plus douces du terme de parcimonie afin de contourner cette difficulté. Cependant, il manque de liens rigoureux entre le problème original et l'approximation dans la plupart des cas. Au point de vue numérique, grâce à ces reformulations et propriétés, nous avons pu appliquer l'algorithme DCA qui s'est montré très efficace et très performant dans la pratique. Les résultats numériques préliminaires obtenus illustrent bien le potentiel de cette nouvelle approche et montrent qu'elle est très prometteuse pour résoudre des problèmes de parcimonie en modélisation parcimonieuse de données. Durant cette thèse, nous avons travaillé sur les problèmes de parcimonie suivant :

1. La modélisation parcimonieuse dans le problème de la valeur propre maximale. Ce problème est très utile en Analyse en Composantes Principales. Nous avons considéré les problèmes originaux avec le terme de parcimonie et nous avons pu les reformuler sous forme DC et quadratique grâce à de nouvelles techniques de pénalité. La structure simple des reformulations obtenues est très adaptée à l'algorithme DCA. Les excellents résultats numériques obtenus avec DCA illustrent bien le potentiel de cette nouvelle approche.
2. La modélisation parcimonieuse dans les modèles de régression linéaire. Ces problèmes sont très utiles, voire indispensables, lorsqu'on est en face à de grandes quantités de données, en apprentissage (ils permettent d'éviter le sur-apprentissage), en traitement de signaux et d'images (ils permettent une bonne

reconstruction des signaux et images). Nous avons pu reformuler ces problèmes sous forme DC et quadratique grâce à de nouvelles techniques de pénalité (différentes de celles présentées précédemment). Ce qui nous a permis de dégager certaines caractéristiques et propriétés importantes concernant les solutions de ces problèmes. Nous avons aussi étendu ces techniques à une classe plus grande de problèmes de parcimonie. Une étude numérique préliminaire, avec DCA et un méthode combinée DCA, SE (Branch and Bound) et relaxation DC pour l'optimisation globale, a été présentée et les résultats obtenus sont très prometteurs.

Il nous reste encore beaucoup de développement théorique et algorithmique à faire concernant ces problèmes par exemple :

- Développement de méthode d'optimisation globale par exemple en combinant DCA, SE, relaxation DC et relaxation SDP.
- Amélioration et raffinement des fonctions de pénalité et paramètres de pénalité.
- Recherche de nouvelles propriétés concernant les solutions.
- Extension de ces nouvelles techniques à d'autre classes de problèmes de parcimonie.

Nous pensons aussi à appliquer cette approche à des applications réelles notamment en

- fouille de données (séléction de caractéristiques).
- traitement de signaux et d'images (reconstruction de signaux parcimonieux).

Bibliographie

[1] DC Programming and DCA : Theory, Algorithms and Applications-Local and Global Approaches : http ://lita.sciences.univ-metz.fr/∼lethi/.

[2] F. Akoa. Combining DC Algorithms (DCAs) and Decomposition Techniques for the Training of Nonpositive–Semidefinite Kernels. *IEEE Transactions on Neural Networks*, 19 :1854–1872, 2008.

[3] A.D. Aleksandrov. On surfaces represented as the difference of convex functions. *Izvestiya Akad, Nauk Kazah. SSR. 60, Ser. Mat. Meh.*, 3, 1949.

[4] F. Alizadeh. Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization. *SIAM Journal on Optimization*, 5 :13–51, 1993.

[5] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues. *Cell Biology*, 96 :6745–6750, 1999.

[6] A. Argyriou, R. Hauser, C. A. Micchelli, and M. Pontil. A DC-programming algorithm for kernel selection. *In Proceedings of the 23rd International Conference on Machine Learning, ICML*, pages 41–48, 2006.

[7] A. Astorino, A. Fuduli, and M. Gaudioso. DC models for spherical separation. *Jounal of Global Optimization*, 48 :657–669, 2010.

[8] M. Atteia and A. Elqortobi. Quasi-convex duality. *In A. Auslender et al. (ed.), Optimization and Optimal Control, Proc. Conference Oberwolfach March 1980, Lecture notes in Control and Inform. Sci. 30, 3-8, Springer-Verlag, Berlin*, 1981.

[9] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28 :253–263, 2008.

[10] M. T. Belghiti, H. A. Le Thi, and T. Pham Dinh. Clustering via DC programming & DCA. *Modelling, Computation and Optimization in Information Systems and Management Sciences. Hermes Science Publishing*, pages 499–507, 2004.

[11] K. P. Bennet and A. Demiriz. Semi-Supervised Support Vector Machines. *In Proceedings of the Conference Neural Information Processing Systems (NIPS)*, 1998.

[12] P. Bloomfield and W. Steiger. Least Absolute Deviations : Theory, Applications, and Algorithms. *Birkhauser*, 1983.

[13] S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, 2004.

[14] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, pages 82–90, 1998.

[15] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51 :34–81, 2009.

[16] C. J. Burges. A tutorial on Support Vector Machines for patterns recognition. *Data Mining and Knowledge Discovery*, 2 :121–167, 1998.

[17] J. Cadima and I. T. Jolliffe. Loadings and correlations in the interpretation of principal components. *Applied Statistics*, pages 203–214, 1995.

[18] E. Candès and J. Romberg. l1-magic. *http ://www.l1-magic.org/*, 2007.

[19] E. Candès, J. Romberg, and T. Tao. Near-optimal signal recovery from random projections : Universal encoding strategies ? *IEEE Trans. Inform. Theory*, 52 :5406–5425, 2006.

[20] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles : exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52 :489–509, 2006.

[21] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59 :1207–1223, 2006.

[22] E. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51 :4203–4215, 2005.

[23] E. Candès and T. Tao. The Dantzig selector : statistical estimation when P is much larger than N. *The Annals of Statistics*, 35 :2392–2404, 2007.

[24] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted l1 minimization. *J. Fourier Analysis and Applications*, 2008.

[25] C. Chapelle, V. Sindhwani, S. S. Keerthi, and N. Cristianini. Optimization Techniques for Semi-Supervised Support Vector Machines. *Journal of Machine Learning Research*, 9 :203–233, 2008.

[26] R. Chartrand and V. Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24 :1–14, 2008.

[27] S. S. Chen, D. L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20 :33–61, 1998.

[28] X. Chen, F. M. Xu, and Y. Ye. Lower Bound Theory of Nonzero Entries in Solutions of L2-Lp Minimization. *SIAM J. Scientific Computing*, 32 :5 :2832–2852, 2010.

[29] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large Scale Transductive SVMs. *Journal of Machine Learning Research*, 7 :1687–1712, 2006.

[30] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading Convexity for Scalability. *In Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, 2006.

[31] C. Cortes and V. Vapnik. Support Vector Networks. *Machine Learning*, 20 :273–297, 1995.

[32] A. d'Aspremont, F. Bach, and L. El Ghaoui. Full regularization path for sparse principal component analysis. *In Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, 2007.

[33] A. d'Aspremont, F. Bach, and L. El Ghaoui. Optimal Solutions for Sparse Principal Component Analysis. *Journal of Machine Learning Research*, 9 :1269–1294, 2008.

[34] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3) :434–448, 2007.

[35] I. Daubechies, M. Defrise, and C. de Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57 :1413–1457, 2004.

[36] R. A. DeVore and V. N. Temlyakov. Some remarks on greedy algorithms. *Adv. Comput. Math.*, 5 :173–187, 1996.

[37] D. L. Donoho. For Most Large Underdetermined Systems of Linear Equations the Minimal l1-norm Solution is also the Sparsest Solution. *Comm. Pure Appl. Math*, 59 :797–829, 2004.

[38] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proc. Natl. Acad. Sci. USA*, 100 :2197–2202, 2003.

[39] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47 :2845–2862, 2001.

[40] D. L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proc. Natl. Acad. Sci. USA*, 102 :9446–9451, 2005.

[41] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann Stat*, 32 :407–499, 2004.

[42] L. El Ghaoui. On the quality of a semidefinite programming bound for sparse principal component analysis. *arXive.org*, 2006.

[43] R. ELLAIA. Contribution à l'analyse et l'optimisation de différences de fonctions convexes. *Thèse de Doctorat 3ieme cycle, Université Paul Sabatier, Toulouse*, 1984.

[44] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456) :1348–1360, 2001.

[45] M.A. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient projection for sparse reconstruction : Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 2007.

[46] I. Frank and J. Friedman. A statistical view of som chemometrics regression tools (with discussion). *Technometrics*, 35 :109–148, 1993.

[47] J. Friedman, R. Tibshirani, and T. Hastie. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33 :2010, 2010.

[48] W.J. Fu. Penalized regression : the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7 :397–416, 1998.

[49] G. M. Fung and O. L. Mangasarian. Semi-Supervised Support Vector Machines for Unlabeled Data Classificatio. *Optimization Methods and Software*, 15 :29–44, 2001.

[50] G.M. Fung and O. L. Mangasarian. Equivalence of Minimal 0-Norm and p-Norm Solutions of Linear Equalities, Inequalities and Linear Programs for Sufficiently Small p. *Journal of Optimization Theory and Applications. To appear*, 151, 2011.

[51] A. Gammerman, V. Vapnik, and V. Vovk. Learning by Transduction. *Uncertainty in Artificial Intelligence*, 1998.

[52] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and DC Programming. *IEEE Trans. Signal Processing*, 57 :4686–4698, 2009.

[53] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer : class discovery and class prediction by gene expression monitoring. *Science*, 286 :531–537, 1999.

[54] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using focuss : a re-weighted minimum norm algorithm. *IEEE Trans. Signal Processing*, 45 :600–616, 1997.

[55] R. Gribonval. Sur quelques problèmes mathématiques de modélisation parcimonieuse. *HDR, Université de Rennes 1*, 2007.

[56] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49 :3320–3325, 2003.

[57] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3 :1157–1182, 2003.

[58] E. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization : Methodology and convergence. *SIAM J. Optim.*, 19 :1107–1130, 2008.

[59] P. Hartman. On functions representable as a difference of two convex functions. *Pacific J. Math.*, 9 :707–713, 1959.

[60] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning. *Springer, Heidelberg*, 2001.

[61] R. Helgason, J. Kennington, and H. Lall. A polynomially bounded algorithm for a singly constrained quadratic program. *Mathematical Programming, North-Holland Publishing Company*, 18 :338–343, 1980.

[62] J. B. Hiriart-Urruty. Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. *Lecture Notes in Economics and Mathematical System 256, Springer-Verlag*, pages 37–70, 1986.

[63] J. B. Hiriart-Urruty and C. Lemarechal. Convex Analysis and Minimization algorithms. *Springer, Berlin*, 1993.

[64] R. Horst. Deterministic global optimization with partition sets whose feasiblity is not known. Application to concave minimization, reverse convex constraints, DC programming and Lipschitzian optimization. *J. of Theory and Application*, 58 :11–37, 1988.

[65] R. Horst. A general class of branch-and-bound methods in global optimization with some new approaches for concave minimization. *J. of Optimization Theory and Application*, 51 :271–291, 1988.

[66] R. Horst, P. M. Pardalos, and N. V. Thoai. Introduction to global optimization. *Second edition. Kluwer Academic Publishers, Netherlands*, 2000.

[67] R. Horst and N. V. Thoai. DC Programming : Overview. *Journal of Optimization Theory and Applications*, 103 :1–43, 1999.

[68] R. Horst and H. Tuy. On the convergence of global methods in multiextremal optimization. *J. of Optimization Theory and Application*, 54 :253–271, 1987.

[69] R. Horst and H. Tuy. Global Optimization : Deterministic Approaches. *2nd revised edition. Springer, Berlin*, 1993.

[70] J. Huang, J. Horowitz, and S. Ma. Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *Annals of Statistics*, 36 :587–613, 2008.

[71] J. Jeffers. Two case studies in the application of principal components. *Applied Statistics*, 16 :225–236, 1967.

[72] T. Joachims. Transductive inference for text classification using support vector machines. *In Proceedings of the International Conference on Machine Learning, ICML*, 1999.

[73] I. T. Jolliffe, N. V. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of computational and Graphical Statistics*, 12 :531–547, 2003.

[74] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An Interior-Point Method for Large-Scale l1-Regularized Least Squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4) :606–617, 2007.

[75] K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28 :1356–1378, 2000.

[76] R. Kohavi and G. H. John. Wrappers for Feature Subset Selection. *Artificial Intelligence*, 97 :273–324, 1997.

[77] N. Krause and Y. Singer. Leveraging the margin more carefully. *In Proceedings of the 21st International Conference on Machine Learning (ICML 2004)*, 2004.

[78] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11 :796–817, 2001.

[79] M. Le Hoai, H. A. Le Thi, and T. Pham Dinh. Hierarchical Clustering Based on Mathematical Optimization. *Advances on Knowledge Discovery and Data Mining, Lecture Notes in Artificial Intelligence*, 3918 :160–173, 2006.

[80] H. A. Le Thi. Analyse numérique des algorithmes de l'optimisation DC approches locales et globales. Codes et simulations numériques en grande dimension. Applications. *Thèse de doctorat de l'Université de Rouen, Décembre*, 1994.

[81] H. A. Le Thi. Contribution à l'optimisation non convexe et l'optimisation globale : Théorie, Algorithmes et Applications. *Habilitation à Diriger des Recherches, (HDR) (Juin 1997) Université de Rouen*, 1997.

[82] H. A. Le Thi. An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints. *Mathematical programming, Ser. A*, 87 :401–426, 2000.

[83] H. A. Le Thi. Solving large scale molecular distance geometry problems by a smoothing technique via the gaussian transform and DC programming. *Jounal of Global Optimization*, 27 :375–397, 2003.

[84] H. A. Le Thi, M. T. Belghiti, and T. Pham Dinh. A new efficient algorithm based on DC programming and DCA for clustering. *J. Glob. Opt.*, 37 :593–608, 2007.

[85] H. A. Le Thi, N. Huynh Van, and T. Pham Dinh. Convergence analysis of DC Algorithm for DC programming with subanalytic data. *Technical Report National Institute for Applied Sciences-Rouen, France*, 2009.

[86] H. A. Le Thi, M. Le Hoai, P. Nguyen Trong, and T. Pham Dinh. Noisy Image Segmentation by a Robust Clustering Algorithm Based on DC Programming and DCA. *In Proceedings of the 8th industrial conference on Advances in Data Mining : Medical Applications, E-Commerce, Marketing, and Theoretical Aspects*, 2008.

[87] H. A. Le Thi, M. Le Hoai, V. Nguyen Van, and T. Pham Dinh. A DC programming approach for feature selection in support vector machines learning. *Advances in Data Analysis and Classification*, 2 :259–278, 2008.

[88] H. A. Le Thi, M. Le Hoai, and T. Pham Dinh. Fuzzy clustering based on nonconvex optimisation approaches using difference of convex (DC) functions algorithms. *Advances in Data Analysis and Classification*, 1 :85–104, 2007.

[89] H. A. Le Thi, M. Le Hoai, and T. Pham Dinh. Optimization based DC programming and DCA for Hierarchical Clustering. *European Journal of Operational Research*, 183 :1067–1085, 2007.

[90] H. A. Le Thi, M. Le Hoai, T. Pham Dinh, and N. Huynh Van. Binary classification via spherical separator by DC programming and DCA. *To appear in Journal of Global Optimzation*, 2011.

[91] H. A. Le Thi, M. Le Hoai, T. Pham Dinh, and N. Huynh Van. Block Clustering based on DC programming and DCA. *Submitted*, 2011.

[92] H. A. Le Thi, V. V. Nguyen, and S. Ouchani. Gene Selection for Cancer Classification Using DCA. *Advanced Data Mining And Appications*, pages 62–72, 2008.

[93] H. A. Le Thi and T. Pham Dinh. Solving a class of linearly contrained indefinite quadratic problems by DC Algorithms. *Journal of Global Optimization*, 11 :253–285, 1997.

[94] H. A. Le Thi and T. Pham Dinh. A Branch and Bound Method via DC Optimization Algorithms and Ellipsoidal Technique for Box Constrained Nonconvex Quadratic Problems. *Journal of Global Optimization*, 13 :171–206, 1998.

[95] H. A. Le Thi and T. Pham Dinh. A continuous approach for globally solving linearly constrained quadratic zero-one programming problems. *Optimization*, 50(1-2) :93–120, 2001.

[96] H. A. Le Thi and T. Pham Dinh. A continuous approach for large-scale constrained quadratic zero-one programming. *(In Honor of Professor ELSTER, Founder of the Journal Optimization) Optimization*, 45 :1–28, 2001.

[97] H. A. Le Thi and T. Pham Dinh. Large Scale Molecular Optimization From Distance Matrices by a DC Optimization Approach. *SIAM Journal on Optimization*, 4(1) :77–116, 2003.

[98] H. A. Le Thi and T. Pham Dinh. The DC (difference of convex functions) Programming and DCA revisited with DC models of real world non convex optimization problems. *Annals of Operations Research*, 133 :23–46, 2005.

[99] H. A. Le Thi and T. Pham Dinh. Minimum Sum-of-Squares Clustering by DC Programming and DCA. *In Proceedings of the Intelligent computing 5th international conference on Emerging intelligent computing technology and applications (ICIC)*, 5755 :327–340, 2009.

[100] H. A. Le Thi, T. Pham Dinh, and N. Huynh Van. Exact Penalty and Error Bound in DC programming. *To appear in Journal of Global Optimization.*, 2011.

[101] H. A. Le Thi, T. Pham Dinh, and M. Le Dung. Exact Penalty in DC Programming. *Vietnam Journal of Mathematics*, 27(2) :169–179, 1999.

[102] M. Le Thi, H. A.and Le Hoai, V. V. Nguyen, and T. Pham Dinh. A DC programming approach for Feature Selection in Support Vector Machines learning. *Journal of Advances in Data Analysis and Classification*, 2 :259–278, 2008.

[103] Y. Liu and X. Shen. Multicategory Psi-learning. *Journal of the American Statistical Association*, 101 :500–509, 2006.

[104] Y. Liu, X. Shen, and H. Doss. Multicategory psi-learning and support vector machine, Computational tools. *Journal of Computational and Graphical Statistics*, 14 :219–236, 2005.

[105] L. Mackey. Deflation methods for Sparse PCA. *In Proceedings of the Conference Neural Information Processing Systems (NIPS 2008)*, 2008.

[106] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12) :3397–1993, 1993.

[107] O. L. Mangasarian. Machine learning via polyhedral concave minimization. *In H. Fischer, B. Riedmueller, and S. Schaeffler, editors, Applied Mathematics and Parallel Computing. Physica-Verlag A Springer-Verlag Company, Heidelberg*, pages 175–188, 1996.

[108] S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Uniform uncertainty principle for Bernoulli and subgaussian ensembles. *arXiv :math/0608665*, 2007.

[109] B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse LDA. *In Proceedings of the 27th International Conference on Machine Learning (ICML 2006)*, 2006.

[110] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA : Exact and greedy algorithms. *Advances in Neural Information Processing Systems*, 18, 2006.

[111] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24 :227–234, 1995.

[112] J. Neumann, C. Schnörr, and G. Steidl. SVM-based Feature Selection by Direct Objective Minimisation, Pattern Recognition. *Proc. of 26th DAGM Symposium, LNCS, Springer, August 2004*, 2004.

[113] J. Neumann, C. Schnörr, and G. Steidl. Combined SVM-Based Feature Selection and Classification. *Machine Learning*, 61 :129–150, 2005.

[114] Y. S. Niu. Programmation DC et DCA en Optimisation Combinatoire et Optimisation Polynomiale via les techniques de SDP. *Thèse de Doctorat, Institut National des Sciences Appliquées de Rouen*, 2010.

[115] M. Osborne, B. Presnell, and B. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20 :389–403, 2000.

[116] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition. *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1 :40–44, 1993.

[117] J.P. Penot. Duality for anticonvex programs. *Journal of Global Optimization archive*, 19 :163–182, 2001.

[118] T. Pham Dinh. Algorithmes de calcul du maximum des formes quadratiques sur la boule unité de la norme maximum. *Séminaire d'analyse numérique, Grenoble, numéro 247*, 1976.

[119] T. Pham Dinh. Algorithmes de calcul d'une forme quadratique sur la boule unité de la norme maximum. *Num. Math.*, 45 :377–440, 1985.

[120] T. Pham Dinh and S. El Bernoussi. Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients. *Fermat Days 85. Mathematics for optimization, J.-B. Hiriart-Urruty (ed.), Elseiver Science Publishers, B. V., North-Holland*, 1986.

[121] T. Pham Dinh and S. El Bernoussi. Duality in DC (difference of convex functions) Optimization. Subgradient Methods. *In K.H. Hoffmann et al. (ed.), Trends in Mathematical Optimization, Volume 84 of International series of Numerisches Mathematics, Birkhauser, Basel*, pages 277–293, 1988.

[122] T. Pham Dinh and H. A. Le Thi. Stabilité de la dualité lagrangienne en optimisation DC (différence de deux fonctions convexes). *C.R. Acad. Paris. Série 1*, 1994.

[123] T. Pham Dinh and H. A. Le Thi. Convex Analysis Approaches to DC Programming : Theory, Algorithms and Applications. *Acta Mathematica Vietnamica*, 22(1) :289–355, 1997.

[124] T. Pham Dinh and H. A. Le Thi. A DC optimization algorithm for solving the trust region subproblem. *SIAM J. Optim.*, pages 476–507, 1998.

[125] T. Pham Dinh and H. A. Le Thi. DC Programming. Theory, Algorithms, Applications : The state of the art. *First International Workshop on Global Constrained Optimization and Constraint Satisfaction, Nice, October 2-4,*, 2002.

[126] T. Pham Dinh, H. A. Le Thi, and F. Akoa. Combining DCA and interior point techniques for large-scale nonconvex quadratic programming. *Optimization, Methods and Softwares*, 23(4) :609–629, 2008.

[127] T. Pham Dinh, H. A. Le Thi, and F. Akoa. Combining DCA (DC Algorithms) and Interior Point Techniques for large-scale Nonconvex Quadratic Programming. *Optimization Methods and Software*, 23(4) :609–629, 2008.

[128] T. Pham Dinh, N. Nguyen Canh, and H. A. Le Thi. An efficient combined DCA and B&B using DC/SDP relaxation for globally solving binary quadratic programs. *J. Glob. Opt.*, 48(4) :595–632, 2010.

[129] N. Quadrianto, J. Petterson, and A. J. Smola. Distribution Matching for Transduction. *In Proceedings of the Conference Neural Information Processing Systems (NIPS 2009)*, 2009.

[130] R.T. Rockafellar. Convex Analysis. *Princeton University Press, N.J.*, 1970.

[131] Y. Saad. Projection and deflation methods for partial pole assignment in linear state feedback. *IEEE Trans. Automat. Contr.*, 33 :290–297, 1998.

[132] T. Schüle, C. Schnörr, S. Weber, and J. Hornegger. Discrete tomography by convex - concave regularization and DC Programming. *Discr. Appl. Math.*, 151 :229–243, 2005.

[133] X. Shen, G.C. Tseng, X. Zhang, and W. H. Wong. On psi-Learning. *Journal of American Statistical Association*, 98 :724–734, 2003.

[134] A. J. Smola and B. Schölkopf. A tutorial on Support Vector Regression. *NeuroCOLT2 Technical Report Series, NC2-TR-1998-030*, 1998.

[135] B. K. Sriperumbudur, D. A. Torres, and G. R. G. Lanckriet. Sparse eigen methods by DC Programming. *In Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, pages 831–838, 2007.

[136] P. Thai Quynh, H. A. Le Thi, and T. Pham Dinh. On the global solution of linearly constrained indefinite quadratic minimization problems by decomposition branch and bound method. *RAIRO, Rech. Opér.*, 30 :31Ű–49, 1996.

[137] P. Thai Quynh, H.A. Le Thi, and T. Pham Dinh. Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems. *Oper. Res. Lett.*, 17 :215–222, 1996.

[138] M. Thiao, T. Pham Dinh, and H. A. Le Thi. DC programming approach for a class of nonconvex programs involving zero-norm. *Communications in Computer and Information Science (CCIS) , Springer*, 14 :358–367, 2008.

[139] M. Thiao, T. Pham Dinh, and H. A. Le Thi. A DC programming approach for sparse eigenvalue problem. *In Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 1063–1070, 2010.

[140] M. Thiao, T. Pham Dinh, and H.A. Le Thi. Solutions of a linear program with an additional Euclidean unit ball constraint by a customized polynomial algorithm. Technical report, Laboratory of Mathematics, LMI, Insa of Rouen, Saint-Etienne-du-Rouvray cedex, France, 2009.

[141] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 46 :431–439, 1996.

[142] J.F. Toland. Duality in Nonconvex Optimisation. *J. Mathematical Analysis and Applications*, 58 :415–428, 1978.

[143] J.F. Toland. On Subdifferential Calculus and Duality in Nonconvex Optimization. *Bull. Soc. Math. France, Mémoire 60*, pages 177–183, 1979.

[144] J. A. Tropp. Greed is good : Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50 :2231–2242, 2004.

[145] J. A. Tropp. Topics in Sparse Approximation. PhD thesis, University of Texas at Austin. *PhD thesis, University of Texas at Austin*, 2004.

[146] J. A. Tropp. Just relax : Convex programming methods for identifying sparse signals. *IEEE Trans. Info. Theory*, 51 :1030–1051, 2006.

[147] H. Tuy. Global minimization of a difference of two convex functions. *Selected Topics in Oper. Res. and Math. Economics, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag*, 256 :98–118, 1984.

[148] H. Tuy, T. V. Thieu, and N. Q. Thai. A conical algorithm for globally minimizing a concave function over a convex set. *Math. Operations Research*, 10 :498–514, 1985.

[149] E. van den Berg and M.P. Friedlander. In pursuit of a root. *Tech. Rep. TR-2007-19, Department of Computer Science, University of British Columbia*, 2007.

[150] V. Vapnik. The Nature of Statistical Learning Theory. *Springer-Verlag, New York*, 1995.

[151] V. Vapnik. Statistical Learning Theory. *New York. Wiley*, 1998.

[152] R.M.F. Ventura. Sparse Image Approximation with Application To Flexible Image Coding. PhD thesis, EPFL. *PhD thesis, EPFL*, 2005.

[153] J. Wang, X. Shen, and W. Pan. On Transductive Support Vector Machines. *Contemp. Math.*, 43 :7–19, 2007.

[154] J. Wang, X. Shen, and W. Pan. On efficient large margin semisupervised learning : method and theory. *Journal of Machine Learning Research*, 10 :719–742, 2009.

[155] S. Weber, A. Nagy, T. Schüle, C. Schnörr, and A. Kuba. A Benchmark Evaluation of Large-Scale optimization Approaches to Binary Tomography. *In Proceedings of the Conference on Discrete Geometry on Computer Imagery (DGCI 2006)*, 4245, 2006.

[156] S. Weber, T. Schüle, and C. Schnörr. Prior Learning and Convex-Concave Regularization of Binary Tomography. *Electr. Notes in Discr. Math.*, 20 :313–327, 2005.

[157] S. Weber, T. Shüle, J. Hornegger, and C. Schnörr. Binary Tomography by Iterating Linear Programs from Noisy Projections. *Proceedings of International Workshop on Combinatorial Image Analysis (IWCIA)*, 2004.

[158] J. Weston, A. Elisseeff, B. Schölkopf, and Tipping M. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3 :1439–1461, 2003.

[159] P. White. The Computation of Eigenvalues and Eigenvectors of a Matrix. *Journal of the Society for Industrial and Applied Mathematics*, 6(4) :393–437, 1958.

[160] H. Wolkowicz, R. Saigal, and L. Vandenberghe. Handbook of Semidefinite Programming. *Kluwer Academic Publishers*, 2000.

[161] Y. Wu and Y. Liu. Variable selection in quantile regression. *Statistica Sinica*, 19 :801–817, 2009.

[162] A. Y. Yang. Compressed sensing meets machine learning- classification of mixture subspace models via sparse representation. *Mini Lectures in Image Processing, TRUST Center Seminar, University of California, Berkeley, USA*, 2008.

[163] Y. Ying, K. Huang, and C. Campbell. Enhanced Protein Fold Recognition through a Novel Data Integration Approach. *BMC Bioinformatics*, 10, 2009.

[164] A. Yuille and A. Rangarajan. The Concave-Convex Procedure (CCCP). *Neural Computation*, 15 :915–936, 2003.

[165] T. Zhang. Some sharp performance bounds for least squares regression with l1 regularization. *Technical Report, Dept. of Statistics, Rutgers University*, 2007.

[166] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101 :1418–1429, 2006.

[167] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J R Stat Soc Ser B.*, 67 :301–320, 2005.

[168] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15 :265–286, 2006.