

# Algorithme Branch-and-Bound pour le démixage spectral parcimonieux

Mehdi Latif<sup>a</sup>,  
Sébastien Bourguignon<sup>b</sup>, Ramzi Ben Mhenni<sup>c</sup>

<sup>a</sup>Université de Nantes - Master Optimisation en Recherche Opérationnelle

<sup>b</sup>Ecole Centrale Nantes - Equipe Signal Image et Son (LS2N)

<sup>c</sup>Ecole Centrale Nantes - Equipe Signal Image et Son (LS2N)

---

## Abstract

Le problème de démixage spectral vise à déterminer la composition, en termes de matériaux, d'une surface planétaire à partir de la mesure de la réflectance lumineuse sur plusieurs canaux spectraux. Un modèle classique considère que le spectre mesuré pour chaque un pixel peut être expliqué comme un mélange de spectres associés aux composants présents dans la matière et qui peut être exprimé comme une combinaison linéaire de signatures spectrales pures.

Une contrainte physique, appelée contrainte de parcimonie, stipule que qu'un faible nombre de composants dans le mélange suffit à décrire la composition de chaque spectre mesuré. Par ailleurs, les réflectances mesurées correspondant à une proportion de la lumière éclairant la scène, les coefficients du mélange, représentant des pourcentages, sont positifs et de somme unité.

Les travaux de recherche présentés dans ce rapport visent à proposer une méthode de résolution exacte pour le problème de démixage parcimonieux intégrant ces contraintes et reposent sur la formulation en nombres mixtes du problème de démixage parcimonieux et de sa résolution par un algorithme de branch-and-bound dédié.

*Keywords:* Démixage spectral, Parcimonie, Optimisation exacte, Branch-and-bound.

---

## Table des matières

<b>1</b>	<b>Introduction générale au problème de démixage spectral</b>	<b>5</b>
1.1	Imagerie hyperspectrale et parcimonie . . . . .	5
1.2	Formulation du problème de démixage spectral parcimonieux . . . . .	7
1.3	Etat de l'art et positionnement de ce travail de recherche . . . . .	7
<b>2</b>	<b>Algorithme de branch-and-bound pour le problème de démixage spectral</b>	<b>9</b>
2.1	Rappels généraux sur l'algorithme de branch-and-bound . . . . .	9
2.1.1	Principes généraux . . . . .	9
2.1.2	Détails sur l'algorithme pour un problème en minimisation . . . . .	9
2.2	Relaxation de la norme $\ell_0$ , Reformulation MIP et intégration des variables binaires. . . . .	10

---

*Email address:* mehdi.latif@etu.univ-nantes.fr (Mehdi Latif)

2.2.1	Relaxation convexe de la norme $\ell_0$ par la norme $\ell_1$ . . . . .	11
2.2.2	Reformulation MIP et ajout des variables binaires . . . . .	13
2.3	Etude de l'impact des branchements pour le problème $\hat{P}_{2/1}$ . . . . .	14
2.3.1	Notations . . . . .	14
2.3.2	Formulation des problèmes $\hat{P}_{2/0}$ et $\hat{P}_{2/1}$ . . . . .	15
2.3.3	Propriété de parcimonie inactive : . . . . .	15
2.3.4	Stratégies de branchements . . . . .	17
2.4	Synthèse de l'impact des branchements : . . . . .	20
2.5	Stratégies de recherche, branchement et de dominance . . . . .	20
2.6	Conclusion . . . . .	21
<b>3</b>	<b>Mise en œuvre de l'algorithme de branch-and-bound dédié</b> . . . . .	<b>22</b>
3.1	Spécification des type de problèmes en fonction des nœuds . . . . .	22
3.1.1	Notations des sous-ensembles d'indices . . . . .	22
3.1.2	Principes de branchements . . . . .	22
3.1.3	Précisions sur les problèmes résolus en fonction des nœuds . . . . .	22
3.1.4	Exemple d'exécution de l'algorithme sur un problème avec $K=3$ . . . . .	25
3.2	Implémentation de l'algorithme . . . . .	26
3.2.1	Notations . . . . .	27
3.2.2	Pseudo-code et principe de fonctionnement . . . . .	28
3.2.3	Application de l'algorithme de branch-and-bound dédié sur un exemple . . . . .	30
3.3	Conclusion . . . . .	31
<b>4</b>	<b>Stratégies développées pour l'évaluation des nœuds l'obtention de bornes inférieures</b> . . . . .	<b>34</b>
4.1	Préambule : stratégies de pré-calculs et de restriction des variables candidates au support . . . . .	34
4.1.1	Pré-calculs de produits matriciels . . . . .	34
4.1.2	Restriction des variables candidates au support . . . . .	34
4.2	Approches basées sur les solveurs en programmation quadratique . . . . .	35
4.2.1	Expression de $\hat{P}_{2/1}$ sous forme d'un programme quadratique . . . . .	35
4.2.2	Reformulation des problèmes dans les cas dégénérés. . . . .	36
4.2.3	Discrimination d'une famille de solveurs . . . . .	36
4.2.4	Choix de la meilleure stratégie pour le solveur sélectionné . . . . .	37
4.2.5	Conclusion de l'approche basée sur les solveurs QP . . . . .	37
4.3	Méthode de continuation homotopique pour l'obtention de bornes inférieures . . . . .	38
4.3.1	Principe général de la méthode . . . . .	38
4.3.2	Conclusion sur la méthode homotopique . . . . .	40
4.4	Méthode de résolution dédiée aux problèmes Fully Constrained Least Squares (FCLS) . . . . .	41
4.5	Conclusion sur les stratégies d'obtention de bornes inférieures . . . . .	41
<b>5</b>	<b>Expérimentations numériques et analyse des résultats</b> . . . . .	<b>42</b>
5.1	Source des données, des erreurs et génération des instances . . . . .	42
5.1.1	Dictionnaire USGS . . . . .	42
5.1.2	Le rapport signal/bruit . . . . .	42
5.1.3	Corrélation entre les atomes du dictionnaire $\mathbf{H}$ et erreur de support . . . . .	43
5.1.4	Génération des instances de test . . . . .	44
5.2	Résultats expérimentaux . . . . .	45
5.2.1	Profils de performance . . . . .	45
5.2.2	Analyse des résultats . . . . .	46

<b>6 Conclusion générale</b>	<b>51</b>
<b>7 Bibliographie</b>	<b>52</b>
<b>8 Annexes</b>	<b>52</b>

*Pluralitas non est ponenda sine necessitate*

**Le rasoir d'Ockham ou principe de parcimonie.**

## **Introduction et remerciements**

Durant ce stage de fin de Master, j'ai eu l'opportunité de travailler sur une problématique se situant à l'intersection des domaines de l'optimisation mathématique, de la recherche opérationnelle et du traitement du signal.

Ce rapport est le fruit de ces cinq mois de travail que j'ai réalisé sous la tutelle de Messieurs S.Bourguignon et R. Ben Mhenni au sein de l'équipe Signal IMage et Son du Laboratoire des Sciences du Numérique de Nantes.

Avant d'entamer la présentation de nos travaux, je souhaiterais tout d'abord remercier Sébastien et Ramzi pour leurs conseils, leur accompagnement, leur écoute, leur confiance, la patience dont ils ont fait preuve et pour m'avoir permis de découvrir le domaine du traitement du signal qui jusqu'alors m'était inconnu.

Je remercie également l'ensemble des doctorants de l'équipe pour m'avoir accueilli pendant les premiers mois de stage, pour m'avoir fait part de leurs projets et m'avoir permis d'apprécier au mieux le monde de la recherche.

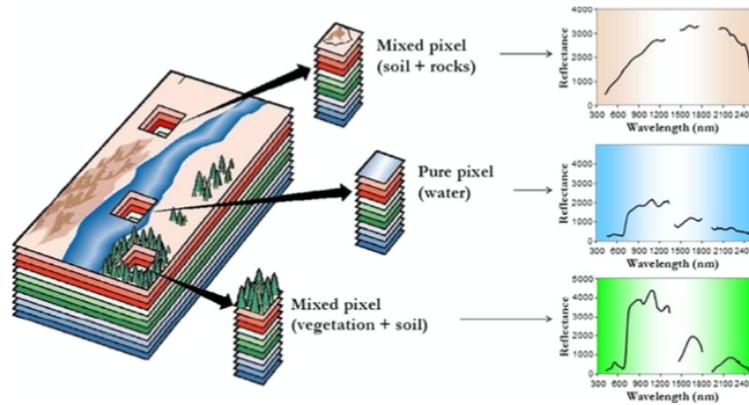


FIGURE 1: Principe de l'imagerie spectrale pour l'observation au sol - Source Iordache [1]

## 1. Introduction générale au problème de démixage spectral

### 1.1. Imagerie hyperspectrale et parcimonie

L'imagerie hyperspectrale correspond à la mesure, en plusieurs positions spatiales, de la réflectance lumineuse *i.e.* la proportion de lumière solaire réfléchiée par la surface observée et ceci sur plusieurs bandes spectrales. Un modèle classique en imagerie hyperspectrale considère que, en chaque pixel ainsi mesuré, ce spectre de réflectance correspond à la superposition de différentes composantes. Ainsi, le spectre obtenu pour chaque pixel observé peut être expliqué comme un mélange de spectres associés à des composants (*signatures spectrales pures, atomes ou endmembers*), ces derniers étant pondérés par leurs proportions (*abondances*).

La problématique consiste en l'étude du problème de démixage spectral parcimonieux. L'idée centrale de ce stage est qu'*un faible nombre de composants est suffisant pour expliquer certains phénomènes physiques*. Le caractère parcimonieux d'une solution, dans le domaine du traitement du signal, se traduit par le fait que seuls quelques coefficients sont non nuls. Dans ce contexte, nous avons à notre disposition un dictionnaire de signatures spectrales pures sur lequel nous pouvons nous baser pour déterminer la composition d'un mélange en un pixel donné.

Le dictionnaire de signatures spectrales pures contient un grand nombre de candidats à la décomposition, mais seulement quelques éléments vont être sélectionnés.

Singer & McCord[2] émettent l'hypothèse que le mélange pour un pixel peut être exprimé comme une combinaison linéaire de signatures spectrales pures pondérées par les abondances de chaque atome. La composition d'un pixel peut alors être exprimée par le modèle linéaire :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\varepsilon} \quad \Leftrightarrow \quad \mathbf{y} = \sum_{q \in Q} \mathbf{h}_q x_q + \boldsymbol{\varepsilon} \quad (1)$$

où :

$\mathbf{y} \in \mathbb{R}^{N \times 1}$  est un vecteur colonne de  $N$  observations nommé spectre de réflectance.

$\mathbf{H} \in \mathbb{R}^{N \times Q}$  est un dictionnaire contenant  $Q$  spectres de référence observés sur  $N$  bandes spectrales ;

$\mathbf{h}_q \in \mathbb{R}^{N \times 1}$  la  $q$ -ème colonne du dictionnaire  $\mathbf{H}$  correspondant à un atome ;

$\mathbf{x} \in \mathbb{R}^{Q \times 1}$  est un vecteur contenant les abondances des  $Q$  signatures spectrales composant le pixel ;

$\boldsymbol{\varepsilon} \in \mathbb{R}^{N \times 1}$  un vecteur de bruit ajouté au modèle.

Un tel modèle linéaire est généralement sous-déterminé *i.e.* le nombre d'observations  $N$  est inférieur au nombre d'atomes  $Q$ . La prise en compte du caractère parcimonieux (*sparsity* en anglais) du vecteur  $\mathbf{x} \in \mathbb{R}^Q$  répond à des contraintes physiques réelles observées dans la nature *e.g.* un nombre limité de composants de la matière - Iordache [1].

La mesure intuitive pour quantifier la parcimonie est une fonction de comptage, appelée *norme*  $\ell_0$ . Elle consiste à compter le nombre de composants non nulles dans un vecteur  $\mathbf{x} \in \mathbb{R}^Q$ . Ainsi, nous pouvons définir cette *norme* pour un vecteur  $\mathbf{x} \in \mathbb{R}^Q$  par :

$$\|\mathbf{x}\|_0 := \text{card}\left(\{q \in Q \mid x_q \neq 0\}\right).$$

L'utilisation du terme *norme* est un abus de langage ; en effet, cette fonction de comptage ne satisfait la propriété d'homogénéité caractéristique des normes *i.e.*  $\forall \lambda \in \mathbb{R}$  et  $\lambda \neq 1$ ,  $\|\lambda \mathbf{x}\| \neq |\lambda| \cdot \|\mathbf{x}\|_0$ . Dans la suite de ce document, nous conserverons le terme *normes* dans un souci de cohérence avec la littérature.

Nous appelons *support* d'un vecteur  $\mathbf{x} \in \mathbb{R}^Q$  l'ensemble des indices  $\mathbf{x}$  dont les composants sont non nulles :

$$\text{supp}(\mathbf{x}) := \{q \in Q : x_q \neq 0\}.$$

Dans la suite du rapport, nous noterons  $S_1$ , le support du vecteur  $\mathbf{x}$ .

Un vecteur  $\mathbf{x} \in \mathbb{R}^Q$  est dit  $K$ -sparse si il possède au plus  $K$  composants non nulles :

$$\|\mathbf{x}\|_0 := \text{card}\left(\text{supp}(\mathbf{x})\right) \leq K.$$

Ce projet de recherche est la continuité des travaux de R. Ben Mhenni[3] dans lesquels l'auteur a travaillé sur des formulations en nombres mixtes pour l'optimisation parcimonieuse ; l'auteur insiste notamment sur l'utilisation de deux contraintes présentées dans Iordache [1], Singer & McCord[2] et Heinz[4] :

**la contrainte de non négativité** (ou Abundance nonnegativity constraint) imposant une borne inférieure sur les composants du vecteur  $\mathbf{x} \in \mathbb{R}^Q$  :

$$\mathbf{x} \geq 0 \quad \Leftrightarrow \quad x_q \geq 0 \quad \forall q \in Q \quad (2)$$

**la contrainte de somme à 1** (ou Abundance sum-to-one constraint) contraignant la somme des abondances du vecteur  $\mathbf{x} \in \mathbb{R}^Q$  :

$$\mathbf{1}_Q^T \mathbf{x} \leq 1 \quad \Leftrightarrow \quad \sum_{q \in Q} x_q \leq 1 \quad (3)$$

Ces deux contraintes sont naturelles dès lors que l'on considère que les abondances des composants incluses dans le support de  $\mathbf{x} \in \mathbb{R}^Q$  s'expriment comme des pourcentages. Notons également que lorsque la contrainte de somme à 1 est saturée, elle définit l'enveloppe convexe de l'espace de recherche *i.e.* l'ensemble des combinaisons convexes d'éléments. Soit  $X$ , un l'ensemble admissible pour le problème de démelange spectral parcimonieux :

$$\text{conv}(X) = \left\{ \mathbf{x} \in \mathbb{R}^Q \mid \mathbf{x} = \sum_{q \in Q} \mathbf{h}_q x_q, \sum_{q \in Q} x_q = 1, x_q \geq 0 \right\}$$

L'objectif de ce stage est d'étudier l'impact des contraintes de non négativité et de somme à 1 sur les algorithmes de résolution en nombres mixtes et ceci, en se basant sur les travaux réalisés par R.Ben Mhenni[3]

qui, durant sa thèse de doctorat, a développé des méthodes en nombres mixtes pour le problème de démixtion parcimonieux sans contrainte.

### 1.2. Formulation du problème de démixtion spectral parcimonieux

Le problème de démixtion parcimonieux peut être formulé mathématiquement comme un problème de minimisation de l'erreur quadratique au sens des moindres carrés sous contrainte d'une borne sur le nombre de coefficients non nul de  $\mathbf{x} \in \mathbb{R}^Q$  i.e. la norme  $\ell_0$ , de somme à 1 et de non négativité. La formulation de ce problème, notée  $\mathcal{P}_{2/0}$ , est donnée par :

$$\min_{\mathbf{x} \in [0,1]^p} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.c.} \quad \|\mathbf{x}\|_0 \leq K, \quad \|\mathbf{x}\|_1 \leq 1 \quad (\mathcal{P}_{2/0})$$

où la valeur du coefficient de parcimonie  $K$  est fixée *a priori*.

Les problèmes d'approximation du support d'un vecteur  $\mathbf{x} \in \mathbb{R}^Q$  à partir des données observées  $\mathbf{y} \in \mathbb{R}^Q$  utilisant la norme  $\ell_0$  sont  $\mathcal{NP}$ -difficiles.

En effet, dès lors que le minimiseur de ce problème possède un coefficient de parcimonie d'au plus  $K \in \mathbb{N}^*$ , l'approche directe pour trouver  $\mathbf{x}$  consiste en la résolution de l'ensemble des systèmes rectangulaires  $\mathbf{H}_S \mathbf{u} = \mathbf{y}$  i.e. l'ensemble des systèmes carrés  $\mathbf{H}_S^T \mathbf{H}_S \mathbf{u} = \mathbf{H}_S^T \mathbf{y}$  avec  $\mathbf{u} \in \mathbb{R}^S$  et  $S$  représente les sous-ensembles de  $K$  composantes prises parmi  $\llbracket 1, Q \rrbracket$  i.e. l'ensembles combinaisons possibles de  $K$  éléments parmi  $Q$  et  $\mathbf{H}_S$  la sous-matrice composée des colonnes de  $S$ .

Assez naturellement, nous pouvons nous rendre compte que le nombre  $\binom{Q}{K}$  de sous ensembles est assez grand et cette approche devient peu réaliste. Par exemple, si nous considérons un dictionnaire à  $Q = 200$  colonnes et un coefficient de parcimonie  $K = 5$  :

$$\binom{Q}{K} = \frac{Q!}{K!(Q-K)!} \geq \left(\frac{Q}{K}\right)^K \quad \text{et} \quad \binom{200}{5} \geq \left(\frac{200}{5}\right)^5 = 102400000 \text{ combinaisons d'atomes possibles}$$

La résolution d'une telle quantité de systèmes carrés de taille  $K \times K$  n'est donc pas envisageable. Une preuve de la complexité des problèmes de minimisation en norme  $\ell_0$  a été initialement proposée par Natarajan[5] réalisant une réduction polynomiale avec le problème Exact Cover by 3-Sets (X3C). Cette même preuve de réduction a été adaptée aux problèmes de minimisation en norme  $\ell_0$  avec contrainte de non-négativité par T.T Nguyen[6]. Nous verrons dans ce document que la contrainte de somme à 1 n'aura que peu d'effet sur l'aspect combinatoire du problème 24.

Pour résoudre de tels problèmes, il existe deux catégories de méthodes : les méthodes approchées et les méthodes exactes. Les méthodes approchées ont pour but de trouver une solution approchée rapidement, mais pas nécessairement optimale. En revanche, les méthodes exactes nous permettent de trouver la solution optimale, mais elles sont souvent beaucoup plus coûteuses quand il s'agit de problèmes de grande taille. La communauté scientifique travaillant sur les problèmes de minimisation en norme  $\ell_0$  a recours aux méthodes de résolution approchées qui ont pour avantage un très faible temps de calcul.

Ce stage s'inscrit dans la philosophie du *small data* où nous préférons assurer la qualité de méthodes de résolution *exactes* sur un volume de données raisonnable.

### 1.3. Etat de l'art et positionnement de ce travail de recherche

Etant donné la classe de complexité des problèmes de minimisation en norme  $\ell_0$ , des méthodes glouttes ont été développées ; Iordache [1] et Tropp & Wright [7] présentent une série d'algorithmes permettant

d'obtenir des solutions approchées (Orthogonal Matching Pursuit, Basis Pursuit, Iterative Hard Thresholding, LASSO Methods, ...). Tropp & Wright [7] évoquent également une série de méthodes basées sur la relaxation convexe ou encore l'optimisation non convexe.

Parmi les tentatives notables visant à résoudre de manière exacte les problèmes de démixtion spectrale, Bourguignon & Ninin[8] abordent le problème sous sa forme exacte via une reformulation MIP et proposent une méthode de relaxation de la norme *norme*  $\ell_0$  basée sur l'introduction de variables binaires. Cette approche a été par la suite développée par R. Ben Mhenni[3] durant sa thèse et notamment dans R. Ben Mhenni[9].

Enfin, Bourguignon & Ninin[8] et R. Ben Mhenni & Bouguignon[10] ont montré que la méthode de résolution basée sur la reformulation MIP permettait d'obtenir de meilleures solutions comparées à celle retournées par des solveurs commerciaux.

Les dernières publications citées nous montrent l'intérêt de l'utilisation d'algorithmes dédiés pour la résolution exacte de problèmes de démixtion spectrale parcimonieux; cette approche prometteuse a motivé les travaux que nous allons maintenant présenter.

Comme nous l'avons évoqué dans ce chapitre, le problème de démixtion spectrale répond à une problématique réelle dans l'observation et la compréhension des images. L'ajout du coefficient parcimonie répond à des contraintes physiques réelles. La complexité de ce problème, que nous cherchons à résoudre grâce à des approches exactes, nous amène à trouver des solutions mais ne nous permet pas d'apporter la preuve de l'optimalité de ces dernières.

La suite de ce document va présenter l'ensemble des travaux qui ont été réalisés durant ce stage pour développer une méthode d'approximation parcimonieuse et ceci en intégrant les contraintes de non négativité des abondances et de somme à 1.

## 2. Algorithme de branch-and-bound pour le problème de démélange spectral

Dans ce chapitre, nous allons nous intéresser à la mise en place de l'algorithme de type branch-and-bound pour la résolution du problème  $\mathcal{P}_{2/0}$ . Tout d'abord, nous commencerons par rappeler les grands principes de l'algorithme puis, dans un second temps, nous étudierons la reformulation Mixed Integer Problem (MIP) et l'introduction de variables binaires proposée par Bourguignon & Ninin[8]. Enfin, nous étudierons l'impact des contraintes non négativité et de somme à 1 sur les branchements opérés pour un problème relâché.

### 2.1. Rappels généraux sur l'algorithme de branch-and-bound

#### 2.1.1. Principes généraux

Le branch-and-bound est une classe d'algorithme permettant de produire des solutions exactes pour des problèmes d'optimisation  $\mathcal{NP}$ -difficiles. Initialement proposé par Land & Doig[11], cette procédure de résolution permet de réaliser une énumération implicite de l'ensemble des solutions réalisables en résolvant une série de sous-problèmes relâchés *i.e.* dans lesquelles nous avons retiré au moins une contrainte et ceci en stockant les valeurs et compositions des solutions dans une structure arborescente. Le principe de cette approche *divide & conquer* consiste, pour un nœud non encore exploré par l'algorithme, en un partitionnement de l'espace de recherche associé à ce nœud en deux sous-nœuds correspondant à des sous-espaces restreints ; il s'agit de la procédure de branchement (Branch).

Considérons un problème de minimisation, durant son exécution, l'algorithme conserve en mémoire la valeur de la borne primale<sup>1</sup> *i.e.* une borne supérieure, et calcule une série de bornes duales *i.e.* des bornes inférieures, pour tenter de mettre à jour la valeur de la meilleure borne primale trouvée jusqu'alors. Pour un nœud donné pour lequel nous résolvons le problème d'optimisation correspondant, l'algorithme possède une série de règles permettant de prouver que la valeur de borne duale obtenue est sous-optimale et que le sous-espace de recherche associé au nœud peut être omis ; il s'agit de la procédure de dominance (Bound). Dès lors que l'arbre de recherche a été exploré dans sa totalité, l'algorithme s'arrête en ayant trouvé la solution au problème et en ayant prouvé son optimalité. Dans les cas où l'algorithme n'a pas pu réaliser d'opération de dominance, la résolution consiste en une énumération exhaustive de l'ensemble des solutions réalisable au problème étudié.

Au-delà de la structure arborescente, un algorithme de branch-and-bound peut être caractérisé par trois règles :

1. la stratégie de recherche *i.e.* l'ordre dans lequel les sous-problèmes vont être explorés ;
2. la stratégie de branchement *i.e.* la méthode de partitionnement de l'espace de recherche en un nœud permettant d'obtenir deux sous-espaces de recherche restreints ;
3. la stratégie de dominance *i.e.* un ensemble de règles permettant de sonder les nœuds qui engendrent des solutions sous-optimales.

#### 2.1.2. Détails sur l'algorithme pour un problème en minimisation

Une présentation mathématique de l'algorithme de branch-and-bound est proposé par Wolsey[12] pour la résolution de problème en maximisation. Nous présentons dans le paragraphe suivant la méthode adaptée à la résolution de problèmes en minimisation.

Soit  $\mathcal{P} = (X, f)$  un problème d'optimisation où  $X$  représente l'espace de recherche *i.e.* un ensemble de solutions réalisables pour  $\mathcal{P}$  et  $f : X \rightarrow \mathbb{R}$  est la fonction objectif. L'objectif de l'algorithme de branch-and-bound est de trouver la solution optimale  $x^*$  telle que  $x^* \in \min_{x \in X} f(x)$ . Pour résoudre le problème  $\mathcal{P}$ , l'algorithme construit itérativement un arbre de recherche  $T$  où chaque nœud correspond à un sous-problème *i.e.* à un

---

1. Selon les notations proposées par Wolsey [12]

sous-ensemble de l'espace  $X$ . Comme nous l'avons mentionné dans le paragraphe précédent, une solution courante  $\hat{x} \in X$  est stockée correspondant à la meilleure borne primale trouvée jusqu'alors. Nous notons  $L$  la liste dans laquelle sont stockés les nœuds non encore explorés; pour chaque itération, l'algorithme sélectionne un sous-problème  $S \subseteq X$  stocké dans  $L$  et résout  $S$  afin d'obtenir la solution  $\hat{x}'$ . Si cette solution  $\hat{x}' \in S$  permet d'obtenir une valeur telle que  $f(\hat{x}') < f(\hat{x})$ , alors la valeur de la borne primale est mise à jour. Si il est possible de montrer que l'ensemble des solutions  $x \in S$  ne permet pas d'obtenir une meilleure borne primale *i.e.*  $\forall x \in S f(x) \geq f(\hat{x})$ , alors le problème est sondé et l'exploration de cette branche de l'arbre de recherche est stoppée. Si aucun des deux cas n'est rencontré dans l'évaluation de la solution  $\hat{x}' \in S$ , alors des sous-nœuds sont obtenus à partir du nœud courant en partitionnant l'ensemble  $S = \bigcup_{i=1}^r S_i$ ; les  $r$  sous-problèmes sont alors stockés dans la liste  $L$  et ajoutés à l'arborescence  $T$ . Dès lors que la liste  $L$  est vide *i.e.* il ne reste plus aucun sous-problème à explorer, alors la solution courante  $\hat{x}$  est retournée avec la preuve de son optimalité.

---

**Algorithme 1** : Structure générale d'un algorithme de branch-and-bound

---

**Entrées** :  $X$  l'espace de recherche,  $f$  la fonction objectif

**Sorties** :  $\hat{x}$  la solution courante

**Initialisation** :

$L \leftarrow \{X\}$ ,  $\hat{x} \leftarrow \emptyset$ ,  $f(\hat{x}) = +\infty$

**tant que**  $L \neq \emptyset$  **faire**

    Sélectionner  $S$  un sous-problème dans  $L$

$L \leftarrow L \setminus S$

**si**  $\exists \hat{x}' \in \{x \in S | f(x) < f(\hat{x})\}$  **alors**

$\hat{x} \leftarrow \hat{x}'$

**fin**

**si**  $S$  ne peut être sondé **alors**

        Partitionner  $S$  tel que  $S = \bigcup_{i=1}^r S_i$

$L \leftarrow L \cup \{\bigcup_{i=1}^r S_i\}$

**fin**

**fin**

---

La complexité d'un algorithme de type branch-and-bound est en  $O(Mb^d)^2$  où :

$b$  : le branching factor *i.e.* le nombre de sous-nœuds générés à chaque partition de l'ensemble  $S$ .

$d$  : la profondeur de recherche *i.e.* la longueur du plus long chemin entre la racine et une feuille.

$M$  : une borne sur le temps nécessaire pour explorer un nœud *i.e.* un sous-problème.

## 2.2. Relaxation de la norme $\ell_0$ , Reformulation MIP et intégration des variables binaires.

Nous commençons par rappeler la formulation de notre problème  $\mathcal{P}_{2/0}$  intégrant les contraintes de positivité et de somme à 1 :

---

2. En étude des algorithmes et théorie de la complexité, la notation Landau présentée correspond à une majoration asymptotique en pire cas définie telle que :

$$O(f) := \{g : g \in \mathbb{R}^{\mathbb{R}} : \exists c \in \mathbb{R}_+^*, \exists x_0 \in \mathbb{R}, \forall x \geq x_0 : 0 \leq g(x) \leq cf(x)\}$$

$$\min_{\mathbf{x} \in [0,1]^n} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.c.} \quad \|\mathbf{x}\|_0 \leq K, \quad \|\mathbf{x}\|_1 \leq 1 \quad (\mathcal{P}_{2/0})$$

Comme nous l'avons évoqué précédemment, les problèmes de minimisation en *norme*  $\ell_0$  sont  $\mathcal{NP}$ -difficiles et ne peuvent être résolus en temps polynomial à l'aide d'algorithmes déterministes, source de motivation pour la mise en place d'algorithmes de résolution exacts.

Dans cette partie, nous allons montrer qu'il est possible de réaliser une relaxation de la norme  $\ell_0$  tout en conservant le caractère parcimonieux du problème  $\mathcal{P}_{2/0}$ . Dans un second temps, nous décrivons l'introduction des variables binaires proposée par Bourguignon & Ninin[8] afin de résoudre le problème en utilisant des algorithmes de type branch-and-bound.

### 2.2.1. Relaxation convexe de la norme $\ell_0$ par la norme $\ell_1$

Un vecteur parcimonieux au sens le plus strict du terme signifie que seuls quelques-uns de ses coefficients ont des valeurs non nulles. Étant donnée cette définition, la norme  $\ell_0$ , qui peut être également vue comme la limite de la *norme*<sup>3</sup>  $\ell_p$  quand  $p$  tend vers zéro :

$$\|\mathbf{x}\|_0 = \lim_{p \rightarrow 0} \|\mathbf{x}\|_p^p = \lim_{p \rightarrow 0} \sum_{i=1}^n |x_i|^p,$$

est la fonction la plus adaptée pour mesurer la parcimonie.

Par définition, l'ensemble des normes  $p \geq 1$  sont convexes. En effet, en utilisant l'inégalité triangulaire et la propriété d'homogénéité, nous pouvons vérifier les combinaisons convexes de deux vecteurs  $\mathbf{x}, \mathbf{y}$  avec  $\forall \lambda \in [0, 1]$  :

$$\|\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}\|_p \leq \lambda \|\mathbf{x}\|_p + (1 - \lambda) \|\mathbf{y}\|_p = \lambda \|\mathbf{x}\|_p + (1 - \lambda) \|\mathbf{y}\|_p$$

La propriété de convexité n'est donc pas vérifiée par la norme  $\ell_0$ ; en effet, pour  $\mathbf{x} = (0, 1)^\top$ ,  $\mathbf{y} = (1, 0)^\top$  et  $\lambda = \frac{1}{2}$

$$\|\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}\|_0 = 2 > \lambda \|\mathbf{x}\|_0 + (1 - \lambda) \|\mathbf{y}\|_0 = 1$$

Notons que la propriété d'homogénéité n'est également pas vérifiée  $\forall \lambda \in \mathbb{R}$ ; en effet, pour  $\mathbf{x} = (0, 1)^\top$  et en posant  $\lambda = 2$ , nous avons :

$$\|\lambda \mathbf{x}\|_0 = \|\mathbf{x}\|_0$$

Il a été montré que le remplacement de la norme  $\ell_0$  par une fonction de la forme  $\sum_{i=1}^n \varphi(|x_i|)$  produisait des solutions parcimonieuses, *i.e.* comprenant des valeurs nulles, si et seulement si  $\varphi$  est strictement croissante en 0 (Voir Moulin & Liu[13]). C'est le cas de la norme  $\ell_p$  avec  $0 < p \leq 1$ . Si le cas  $p < 1$ , générant des problèmes d'optimisation difficiles car non convexes, a été étudié dans plusieurs travaux Lai & Wang[14]. La norme  $\ell_1$  ( $\|\mathbf{x}\|_1 := \sum_i |x_i|$ ) reste la mesure de parcimonie la plus fréquemment utilisée et la plus facile à prendre en compte en raison de sa nature convexe (voir Figure 2). Du fait de la non-convexité de la *norme*  $\ell_0$ , une méthode souvent utilisée pour l'approximation parcimonieuse en traitement du signal consiste en la modification de la fonction objectif, pour le moment non calculable, par une fonction objectif convexe rendant le problème plus simple à résoudre. De telles transformations peuvent être réalisées à l'aide des normes  $\ell_q$  avec  $q > 1$  qui sont, par définition convexes et strictement convexes dans le cas de la norme  $\ell_2$  ce qui permet d'assurer l'unicité du minimiseur et son calcul par des algorithmes de descente classiques. Nous

3. Le terme de *norme*  $\ell_p$  pour  $0 < p < 1$  est également un abus de langage : cette fonction, ne vérifiant pas l'inégalité triangulaire, est seulement une quasi-norme *i.e.* une norme satisfaisant les propriétés de séparation et d'homogénéité mais pas l'inégalité triangulaire, cette dernière étant remplacée par  $\|\mathbf{x} + \mathbf{y}\|_p \leq c(\|\mathbf{x}\| + \|\mathbf{y}\|)$ ,  $\forall c > 0$ .

devons ainsi trouver, parmi les normes  $\ell_q, q \geq 1$ , celle qui offrira le meilleur compromis entre convexité et parcimonie.

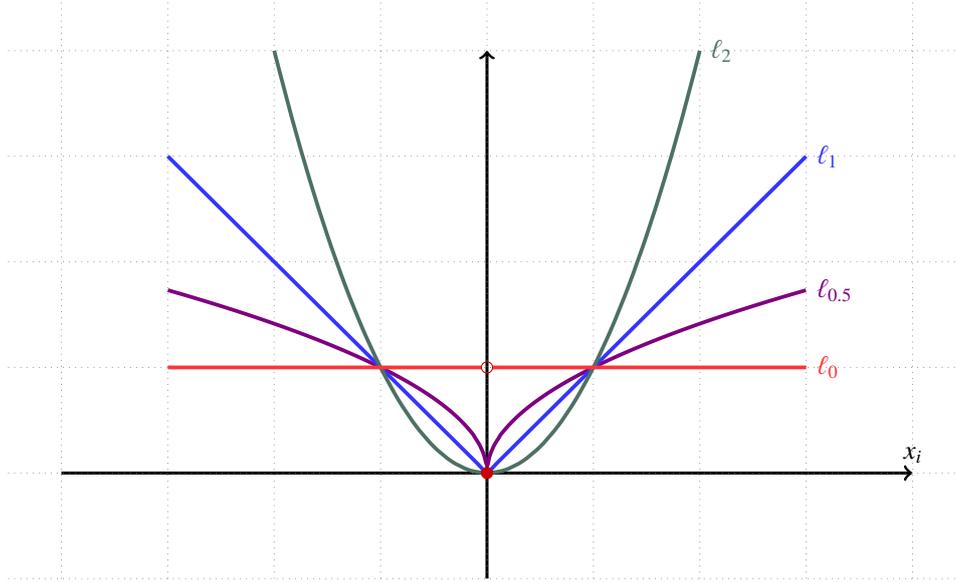


FIGURE 2: Fonctions de la variable scalaire  $\varphi(x_i)$  intervenant dans la définition de différentes normes  $\ell_p$ , avec  $\|\mathbf{x}\|_p^p = \sum_i \varphi(|x_i|)$ , pour  $p = 0, 0.5, 1$  et  $2$  - Source : R.Ben Mhenni[3]

Définissons temporairement le problème d'optimisation  $\mathcal{P}_q$  tel que

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_q^q \quad \text{s.c.} \quad \mathbf{y} = \mathbf{H}\mathbf{x} \quad \text{avec } q \geq 1$$

Une méthode intuitive pour représenter le problème d'optimisation  $\mathcal{P}_q$  est de considérer une boule  $\ell_q$  centrée à l'origine du repère et dont nous allons progressivement augmenter le rayon  $r$  jusqu'à atteindre un point d'intersection avec l'hyperplan définissant les contraintes  $\mathbf{y} = \mathbf{H}\mathbf{x}$ . Le point obtenu sera alors le vecteur minimum réalisable pour la norme  $\ell_q$  mais également, la solution optimale du problème  $\mathcal{P}_q$ .

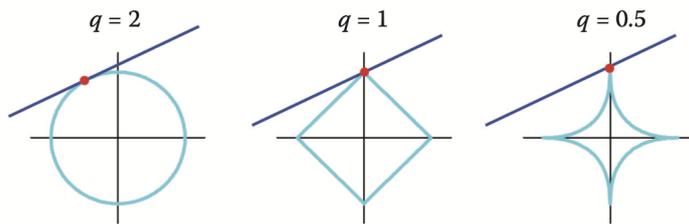


FIGURE 3: Représentation de l'optimisation de  $\mathcal{P}_q$  réalisée en augmentant le rayon de la boule  $\ell_q$  centrée à l'origine jusqu'à l'intersection avec l'ensemble des points réalisables  $\mathbf{y} = \mathbf{H}\mathbf{x}$ . Rish & Grabarnik[15]

Considérant le problème, nous pouvons constater que la boule  $\ell_q$  présente également des "angles" si  $q \leq 1$  sur les axes du repère, correspondant à des solutions parcimonieuses pour lesquelles certaines coordonnées sont nulles. Ainsi, une boule  $\ell_q$  a de plus fortes probabilités d'atteindre l'hyperplan  $\mathbf{y} = \mathbf{H}\mathbf{x}$  sur les angles de l'enveloppe convexe avec  $0 < q \leq 1$  que si l'on considère une relaxation avec la norme  $\ell_2$  pour laquelle l'intersection n'apparaît jamais en pratique ce qui entraîne l'obtention de solutions non parcimonieuses.

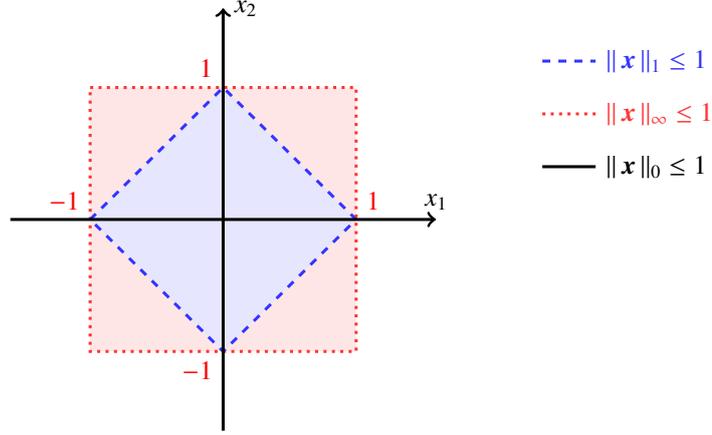


FIGURE 4: Représentation de la norme  $\ell_1$  comme l'enveloppe convexe de l'intersection des boules  $\ell_0$  et  $\ell_\infty$ .

Il est également possible d'apporter une justification supplémentaire quant à l'utilisation de la norme  $\ell_1$  pour assurer une structure parcimonieuse au problème relâché en remarquant que cette dernière est en réalité l'enveloppe convexe de l'intersection des boules  $\ell_0$  et  $\ell_\infty$ . La preuve de cette propriété se fait par inclusion mutuelle  $\mathcal{B}_{\ell_1} \subseteq \text{conv}(\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty})$  et  $\text{conv}(\mathcal{B}_{\ell_0} \cap \mathcal{B}_{\ell_\infty}) \subseteq \mathcal{B}_{\ell_1}$ .

En guise de conclusion, nous pouvons en déduire que la relaxation convexe avec la norme  $\ell_1$  permet de rendre le problème calculable tout en conservant la nature parcimonieuse des solutions obtenues avec cette approximation.

### 2.2.2. Reformulation MIP et ajout des variables binaires

Dans leur article, Bourguignon & Ninin[8] proposent une méthode pour transformer le problème 24 en MIP et ceci en intégrant des variables binaires  $b_q \in \{0, 1\} \forall q \in Q$  pour lesquelles chaque variable binaire traduit la contrainte logique  $x_q = 0 \Leftrightarrow b_q = 0 \forall q \in Q$ . Cette méthode permet alors de relâcher la contrainte de norme en formulation la *norme*  $\ell_0$  comme la somme des variables binaires  $b_q \forall q \in Q$  ainsi qu'en définissant une valeur de  $M \in \mathbb{R}^*$ ; sous contrainte de somme à 1, nous avons  $M = 1$  nous pouvons alors écrire :

$$\|\mathbf{x}\|_0 = \sum_{q \in Q} b_q \text{ et } \|\mathbf{x}\|_\infty \leq 1$$

Dès lors, le problème 24 peut être réécrit de la manière suivante :

$$\min_{\substack{\mathbf{x} \in \{0,1\}^Q \\ \mathbf{b} \in \{0,1\}^Q}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.c.} \quad \sum_{q \in Q} b_q \leq K, \quad \|\mathbf{x}\|_1 \leq 1, \quad \mathbf{x} \leq \mathbf{b} \quad (\hat{\mathcal{P}}_{2/0})$$

Notons de plus que l'ajout de cette variable binaire et la relaxation ne modifient pas la solution obtenue pour 24; en effet, posons  $S \subseteq Q$  un ensemble d'atomes tels que  $\text{card}(S) = n$ , par la reformulation avec variables binaires et la définition de la norme  $\ell_0$ , nous avons :

$$\|\mathbf{x}\|_0 = \text{card}\{j \in S | x_j \neq 0\} = \sum_{\substack{q \in S \\ x_q \neq 0}} b_q = \sum_{\substack{q \in S \\ x_q \neq 0}} 1 \quad (4)$$

L'égalité précédente est possible car  $x_q \neq 0 \Rightarrow b_q = 1 \forall q \in Q$ . Notons de plus que la contrainte de somme à

1 nous permet d'écrire :

$$\|\mathbf{x}\|_\infty \leq 1 \Leftrightarrow \sup_{i \in \llbracket 1, n \rrbracket} \{|x_i|\} \leq 1 \Rightarrow |x_i| \leq 1 \quad \forall i \in \llbracket 1, n \rrbracket \quad (5)$$

En intégrant l'inégalité (5) dans l'expression de la norme  $\ell_0$  présentée en (4), nous obtenons :

$$\|\mathbf{x}\|_0 = \sum_{\substack{q \in S \\ x_q \neq 0}} 1 \geq \sum_{\substack{q \in S \\ x_q \neq 0}} |x_q| \Rightarrow \|\mathbf{x}\|_0 \geq \|\mathbf{x}\|_1 \quad (6)$$

Considérant cette majoration, nous pouvons alors réaliser une relaxation continue des variables  $\mathbf{b}$  pour  $\hat{\mathcal{P}}_{2/0}$  et obtenons ainsi le nouveau problème :

$$\min_{\substack{\mathbf{x} \in [0,1]^Q \\ \mathbf{b} \in [0,1]^Q}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.c.} \quad \sum_{q \in Q} b_q \leq K, \quad \|\mathbf{x}\|_1 \leq 1, \quad \mathbf{x} \leq \mathbf{b} \quad (\hat{\mathcal{P}}_{2/1})$$

A l'optimum du problème  $\hat{\mathcal{P}}_{2/1}$ , nous avons  $|x_q^*| = b_q^* \quad \forall q \in Q$ ; en effet, si il existait une composante  $q'$  telle que  $|x_{q'}^*| < b_{q'}^*$ , alors cela impliquerait qu'il existerait une séquence de variables binaires  $\mathbf{b}'$  telle que  $b'_{q'} = |x_{q'}^*| < b_{q'}^*$  qui serait aussi réalisable avec une fonction de coût inférieure.

Nous pouvons alors conclure que  $\|\mathbf{x}\|_0 \geq \|\mathbf{x}\|_1$  et qu'à l'optimalité, nous avons  $\|\mathbf{x}^*\|_0 = \|\mathbf{x}^*\|_1$  comme unique minimiseur du problème ( $\hat{\mathcal{P}}_{2/1}$ ) et par reformulation du problème (24). Nous avons ainsi prouvé que l'introduction des variables binaires et la relaxation des ces dernières assurent l'unicité de la solution optimale.

### 2.3. Etude de l'impact des branchements pour le problème $\hat{\mathcal{P}}_{2/1}$

L'objectif de cette section est d'étudier l'impact des stratégies de branchement pour le problème  $\mathcal{P}_{2/0}$  pour lequel nous avons ajouté les contraintes de non négativité et de somme à 1.

On s'intéresse notamment aux relations entre les minimiseurs obtenus pour le problème à un nœud donné avec ceux obtenus pour les deux sous-problèmes générés par l'opération de branchement. Dans ce branch-and-bound, les sous-problèmes sont créés en ajoutant une contrainte assignant une variable binaire  $b_{i_0}$  à une valeur 0 ou 1,  $i_0$  est ici l'indice d'une variable qui n'a pas encore été inspectée durant les branchements réalisés auparavant.

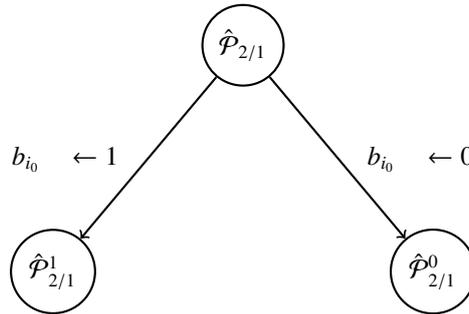


FIGURE 5: Exemple de branchement sur la variable  $b_{i_0}$  et création des nœuds fils : fils gauche  $\hat{\mathcal{P}}_{2/1}^1$  et fils droit  $\hat{\mathcal{P}}_{2/1}^0$ .

#### 2.3.1. Notations

Nous détaillons ci-dessous les notations utilisées dans la suite de cette étude :

$i_0 \in \llbracket 1, n \rrbracket$  : l'indice d'une variable à un nœud de l'arbre sur lequel on souhaite opérer une opération de branchement.

$\hat{\mathcal{P}}$  : un problème reformulé à l'aide des variables binaires  $\mathbf{b}$ .

$\hat{\mathcal{P}}^0$  : le problème obtenu à partir de  $\mathcal{P}$  en fixant la variable binaire  $b_{i_0}$  à 0.

$\hat{\mathcal{P}}^1$  : le problème obtenu à partir de  $\mathcal{P}$  en fixant la variable binaire  $b_{i_0}$  à 1.

$S = \llbracket 1, n \rrbracket$  : l'ensemble des indices des variables associées aux colonnes du dictionnaire

$S_1 \subseteq S$  : l'ensemble des indices des variables assignées à 1 défini tel que  $S_1 = \{i | b_i = 1\}$  avec  $n_1 = \text{card}(S_1)$

$S_0 \subseteq S$  : l'ensemble des indices des variables assignées à 0 défini tel que  $S_0 = \{i | b_i = 0\}$

$\bar{S} \subseteq S$  : l'ensemble des indices des variables non assignées défini tel que  $\bar{S} = S \setminus \{S_1 \cup S_0\}$  avec  $\bar{n} = \text{card}(\bar{S})$

$\text{Argmin}(\mathcal{P})$  : l'ensemble des minimiseurs du problème  $\mathcal{P}$

### 2.3.2. Formulation des problèmes $\hat{\mathcal{P}}_{2/0}$ et $\hat{\mathcal{P}}_{2/1}$

En un nœud donné de l'arborescence et utilisant les notations présentées dans la section précédente, nous présentons ici le problème initial  $\mathcal{P}_{2/0}$  étudié :

$$(\hat{\mathcal{P}}_{2/0}) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in \{0,1\}^{\bar{n}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{n_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in \bar{S}} b_i \leq K - \text{card}(S_1) \quad (7.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \quad (7.2)$$

$$\sum_{i \in \bar{S}} x_i \leq 1 \quad (7.3)$$

où la fonction objectif correspond au calcul de l'erreur en norme  $\ell_2$  excluant les variables fixées à 0 et la contrainte de parcimonie se voit retirer le nombre de composantes ayant été ajouté au support *i.e.* dans l'ensemble  $S_1$ .

Dans la contrainte (7.2) du problème présenté ci-dessus, nous pouvons observer que fixer la variable binaire  $b_i$  à la valeur 1 n'entraîne pas nécessairement  $x_i \neq 0$ ; en effet, dans cette contrainte, la valeur de  $b_i$  définit une borne supérieure pour la variable  $x_i$ , cette dernière restant libre de prendre une valeur dans l'intervalle  $[0, 1]$ .

Nous présentons ci-dessous le problème relâché  $\hat{\mathcal{P}}_{2/1}$  où les variables binaires  $b_i$ ,  $i \in \bar{S}$ , prennent leurs valeurs dans l'intervalle  $[0, 1]$  et que nous résolvons à l'optimalité pour isoler l'indice d'une variable  $i_0 \in \bar{S}$  sur laquelle nous allons réaliser une opération de branchement.

$$(\hat{\mathcal{P}}_{2/1}) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in [0,1]^{\bar{n}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{n_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in \bar{S}} b_i \leq K - \text{card}(S_1) \quad (8.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \quad (8.2)$$

$$\sum_{i \in \bar{S}} x_i \leq 1 \quad (8.3)$$

### 2.3.3. Propriété de parcimonie inactive :

A l'optimum de tout problème relâché de type  $\hat{\mathcal{P}}_{2/1}$ , la contrainte de parcimonie (8.1) est non active et peut alors être retirée de l'expression du problème.

En effet, cette propriété est due au fait que la contrainte de somme à 1 (8.3) rend la contrainte (8.1) redondante dès lors que  $K > \text{card}(S_1)$ . Nous pouvons donc reformuler le problème initial en retirant la contrainte de parcimonie :

$$(\hat{\mathcal{P}}'_{2/1}) \quad \min_{\substack{\mathbf{b}_S \in [0,1]^n, \mathbf{x}_S \in [0,1]^n \\ \mathbf{x}_{S_1} \in [0,1]^n}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_S \mathbf{x}_S\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (9.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \quad (9.2)$$

En considérant cette observation, nous souhaitons démontrer que la contrainte de parcimonie (8.1) du problème  $\hat{\mathcal{P}}_{2/1}$  ne modifie pas la valeur optimale qui sera également la solution optimale au problème  $\hat{\mathcal{P}}'_{2/1}$ . En d'autres termes, nous souhaitons montrer qu'il est possible, en fixant  $b_i = x_i, i \in \bar{S}$ , de trouver une solution réalisable  $(\mathbf{b}^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0)$  au problème  $\hat{\mathcal{P}}_{2/1}$  qui reste réalisable pour  $\hat{\mathcal{P}}'_{2/1}$  dans lequel la contrainte de parcimonie redondante a été supprimée.

Pour ce faire, nous allons montrer qu'il existe un minimiseur de  $\hat{\mathcal{P}}_{2/1}$  pour lequel la contrainte de parcimonie est insaturée sous les contraintes de somme à 1 et de non négativité *i.e.* est également un minimiseur de  $\hat{\mathcal{P}}'_{2/1}$  :

$$\exists (\tilde{\mathbf{b}}_S^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0) \in \text{Argmin}(\hat{\mathcal{P}}_{2/1}) \text{ tel que } \sum_{i \in \bar{S}} \tilde{b}_i^0 \leq 1 \Rightarrow (\tilde{\mathbf{b}}_S^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0) \in \text{Argmin}(\hat{\mathcal{P}}'_{2/1})$$

Preuve :

Soient  $(\mathbf{b}^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0)$  une solution réalisable pour le problème  $\hat{\mathcal{P}}_{2/1}$  et  $(\tilde{\mathbf{b}}_S^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0)$  un minimiseur de  $\hat{\mathcal{P}}_{2/1}$  tel que  $\tilde{\mathbf{b}}_S^0 = \mathbf{x}_S^0$ . Nous pouvons déduire de l'égalité précédente :

$$\tilde{\mathbf{b}}_S^0 := \mathbf{x}_S^0 \Rightarrow \sum_{i \in \bar{S}} \tilde{b}_i^0 = \sum_{i \in \bar{S}} x_i^0 \quad (10)$$

Nous remarquons qu'il nous est possible de reformuler la contrainte (8.3) de  $\mathcal{P}^R$  telle que :

$$\sum_{i \in \bar{S}} x_i \leq 1 \Leftrightarrow \sum_{i \in S_1} x_i + \sum_{i \in \bar{S}} x_i \leq 1 \Leftrightarrow \sum_{i \in \bar{S}} x_i \leq 1 - \sum_{i \in S_1} x_i \leq 1 \quad (11)$$

Nous pouvons alors utiliser la majoration (11) avec l'égalité (10) afin obtenir la relation suivante :

$$\sum_{i \in \bar{S}} \tilde{b}_i^0 \leq 1 \text{ et } 1 \leq K - \text{card}(S_1).$$

Nous observons que cette dernière inégalité nous permet de montrer que la contrainte (8.3), lorsqu'elle est saturée, rend la contrainte (8.1) redondante car nécessairement insaturée.

Nous avons ainsi prouvé que  $(\tilde{\mathbf{b}}_S^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0)$  est un minimiseur réalisable pour  $\hat{\mathcal{P}}_{2/1}$ , vérifiant la propriété  $\sum_{i \in \bar{S}} \tilde{b}_i^0 \leq 1$ , pour lequel la contrainte de parcimonie est inactive et la contrainte de somme à 1 est respectée *i.e.*  $(\tilde{\mathbf{b}}_S^0, \mathbf{x}_S^0, \mathbf{x}_{S_1}^0)$  est un minimiseur de  $\hat{\mathcal{P}}'_{2/1}$ .

Dans la suite de ce document, nous appliquerons la propriété de parcimonie inactive et nous considérerons le problème initial sous sa forme relâchée :

$$(\hat{\mathcal{P}}_{2/1}) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in [0,1]^{\bar{n}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{n_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (12.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \quad (12.2)$$

### 2.3.4. Stratégies de branchements

Soit  $\mathbf{x}^* \in [0, 1]^n$  une solution optimale obtenue par résolution du problème relâché  $\hat{\mathcal{P}}_{2/1}$  et notons  $i_0 \in \bar{S}$  la coordonnée de  $\mathbf{x}^*$  sur laquelle nous souhaitons réaliser une opération de branchement. En définissant  $b_{i_0} \in \{0, 1\}$  la variable binaire associée à la composante  $x_{i_0} \in [0, 1]$ , nous allons présenter les quatre cas de branchements possibles qui peuvent influencer la résolution des sous-problèmes de  $\hat{\mathcal{P}}_{2/1}$  à savoir :

**Cas 1 :**  $x_{i_0}^* \neq 0$  et  $b_{i_0} \leftarrow 1$

**Cas 3 :**  $x_{i_0}^* = 0$  et  $b_{i_0} \leftarrow 1$

**Cas 2 :**  $x_{i_0}^* \neq 0$  et  $b_{i_0} \leftarrow 0$

**Cas 4 :**  $x_{i_0}^* = 0$  et  $b_{i_0} \leftarrow 0$

#### 2.3.4.1 Cas 1 : $x_{i_0}^* \neq 0$ et $b_{i_0} \leftarrow 1$

Le problème obtenu par application de cette stratégie de branchement est le suivant :

$$(\hat{\mathcal{P}}_{2/1}^1) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in [0,1]^{\bar{n}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{n_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S_1 \cup \{i_0\}} \mathbf{x}_{S_1 \cup \{i_0\}} - \mathbf{A}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (13.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (13.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \cup \{i_0\} \quad (13.3)$$

Nous pouvons constater que cette stratégie de branchement  $b_{i_0} \leftarrow 1$  ne définit qu'une borne supérieure pour la valeur de  $x_{i_0}^*$  mais laisse cette dernière *libre* dans l'intervalle  $[0, 1]$ . Notons que puisque la contrainte de parcimonie a été retirée du modèle par application de la propriété, la cardinalité du support, qui aurait été mise à jour après ce branchement, n'est plus contrainte par la valeur  $K$ .

Enfin, remarquons que la variable  $\mathbf{b} \in \{0, 1\}^n$  n'impacte pas la fonction de coût de  $\hat{\mathcal{P}}_{2/1}^1$  et que  $\forall b_{i_0} \in [b_{i_0}^0, 1]$  la valeur à l'optimum de  $\hat{\mathcal{P}}_{2/1}^1$  sera identique (avec  $b_{i_0}^0$  la valeur de la variable binaire à l'optimum relâché de  $\hat{\mathcal{P}}_{2/0}^1$  ; Il est donc possible de réécrire le problème  $\hat{\mathcal{P}}_{2/1}^1$  en observant que la contrainte de somme à 1 entraîne une majoration implicite des  $x_i \in [0, 1]$  par 1 ;  $\hat{\mathcal{P}}_{2/1}^1$  devient alors :

$$(\hat{\mathcal{P}}_{2/1}^{1'}) \quad \min_{\substack{\mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{n_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1 \cup \{i_0\}} \mathbf{x}_{S_1 \cup \{i_0\}} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (14.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \cup \{i_0\} \quad (14.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (14.3)$$

La figure ci-dessus illustre la propriété permettant, pour une solution optimale au problème relâché  $\hat{\mathcal{P}}_{2/1}$  et en observant que le vecteur de variables binaires  $\mathbf{b}$  n'intervient pas dans le calcul de la fonction objectif, de trouver une autre solution réalisable  $(\tilde{\mathbf{b}}_{\bar{S}}^0, \mathbf{x}_{\bar{S}}, \mathbf{x}_{S_1})$  tel que  $\sum_{i \in \bar{S}} \tilde{b}_i^0 \leq 1$ . L'ensemble de ces solutions réalisables équivalentes est modélisé sur cette figure par le segment joignant les deux points dans l'espace des solutions du problème relâché.

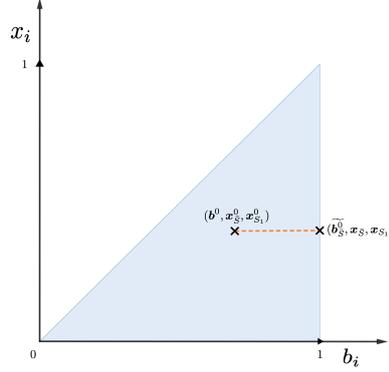


FIGURE 6: Représentation graphique de l'admissibilité d'une solution  $(\bar{\mathbf{b}}_{\bar{S}}^0, \mathbf{x}_{\bar{S}}, \mathbf{x}_{S_1})$  au problème relâché  $\hat{\mathcal{P}}_{2/1}$

### 2.3.4.2 Cas 2 : $x_{i_0}^* \neq 0$ et $b_{i_0} \leftarrow 0$

Le problème obtenu par application de cette stratégie de branchement est le suivant :

$$(\hat{\mathcal{P}}_{2/1}^0) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in [0,1]^{\bar{p}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{p}} \\ \mathbf{x}_{S_1} \in [0,1]^{\bar{p}_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in \bar{S}} x_i \leq 1 \quad (15.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (15.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \quad (15.3)$$

Dans le cas présent, le branchement  $b_{i_0} \leftarrow 0$  force la composante  $x_{i_0}^* \neq 0$  pour  $\hat{\mathcal{P}}_{2/1}$  à être nulle dans la solution de  $\hat{\mathcal{P}}_{2/1}^0$ . Ce branchement correspond à une stratégie *exploratoire* de l'espace des solutions ; en effet, l'exclusion de la variable  $x_{i_0}^*$  dans le problème  $\hat{\mathcal{P}}_{2/1}^0$  nécessite une ré-optimisation pour obtenir une nouvelle solution réalisable ne contenant pas la variable  $x_{i_0}^*$  et proposant une nouvelle borne inférieure pour le problème  $\hat{\mathcal{P}}_{2/1}$  pouvant potentiellement permettre l'élagage de futures branches dans l'arbre.

De la même manière que dans le cas précédent, nous pouvons réécrire le problème  $\hat{\mathcal{P}}_{2/1}^0$  en considérant que les variables  $b_i, \forall i \in \bar{S} \setminus \{i_0\}$  définissent une majoration implicite des variables  $\mathbf{x}$  :

$$(\hat{\mathcal{P}}_{2/1}^{0'}) \quad \min_{\substack{\mathbf{x}_{\bar{S}} \in [0,1]^{\bar{p}} \\ \mathbf{x}_{S_1} \in [0,1]^{\bar{p}_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in \bar{S}} x_i \leq 1 \quad (16.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (16.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \quad (16.3)$$

2.3.4.3 Cas 3 :  $x_{i_0}^* = 0$  et  $b_{i_0} \leftarrow 1$

Le problème obtenu par application de cette stratégie de branchement est le suivant :

$$(\hat{\mathcal{P}}_{2/1}^1) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in [0,1]^{\bar{n}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{\eta_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1 \cup \{i_0\}} \mathbf{x}_{S_1 \cup \{i_0\}} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (17.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (17.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \cup \{i_0\} \quad (17.3)$$

Encore une fois, on observe que l'affectation  $b_{i_0} \leftarrow 1$  dans  $\hat{\mathcal{P}}_{2/1}^1$  ne contraint pas la valeur de  $x_{i_0}^*$  qui reste *libre* dans l'intervalle  $[0, 1]$ . Ainsi, étant donné que ce branchement n'interdit pas  $x_{i_0}^* = 0$  à l'optimum de  $\hat{\mathcal{P}}_{2/1}^1$  qui est l'hypothèse de ce cas, nous pouvons en déduire :

$$\min(\hat{\mathcal{P}}_{2/1}) = \min(\hat{\mathcal{P}}_{2/1}^1).$$

La majoration des variables binaires  $b_i, \forall i \in \bar{S} \setminus \{i_0\}$  nous permet de réécrire le problème relâché tel que :

$$(\hat{\mathcal{P}}_{2/1}^{1'}) \quad \min_{\substack{\mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{\eta_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1 \cup \{i_0\}} \mathbf{x}_{S_1 \cup \{i_0\}} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (18.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (18.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \cup \{i_0\} \quad (18.3)$$

2.3.4.4 Cas 4 :  $x_{i_0}^* = 0$  et  $b_{i_0} \leftarrow 0$

Le problème obtenu par application de cette stratégie de branchement est le suivant :

$$(\hat{\mathcal{P}}_{2/1}^0) \quad \min_{\substack{\mathbf{b}_{\bar{S}} \in [0,1]^{\bar{n}}, \mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{\eta_1}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (19.1)$$

$$0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (19.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \quad (19.3)$$

Par hypothèse,  $x_{i_0}^* = 0$  pour  $\hat{\mathcal{P}}_{2/1}$ . Ainsi, fixer  $b_{i_0} \leftarrow 0$  dans  $\hat{\mathcal{P}}_{2/1}^0$  n'entraîne pas de modification supplémentaire sur la valeur de la solution à l'optimum de  $\hat{\mathcal{P}}_{2/1}^0$ . Nous pouvons donc en déduire :

$$\min(\hat{\mathcal{P}}_{2/1}) = \min(\hat{\mathcal{P}}_{2/1}^0).$$

La relaxation continue  $\hat{\mathcal{P}}_{2/1}^0$  étant identique à  $\hat{\mathcal{P}}_{2/1}$ , nous pouvons émettre l'hypothèse qu'aucune réoptimisation ne sera nécessaire et comme les variables binaires sont majorées par 1, nous pouvons dès lors réécrire le problème  $\hat{\mathcal{P}}_{2/1}^0$  tel que :

Valeur de $x_{i_0}$ à l'optimum de $(P^R)$	Branchement sur $b_{i_0}$		Impact sur la solution
$x_{i_0}^* \neq 0$	et	$b_{i_0} \leftarrow 1$	La solution à $\hat{\mathcal{P}}_{2/1}^1$ est identique à celle de $\hat{\mathcal{P}}_{2/1}$
$x_{i_0}^* \neq 0$	et	$b_{i_0} \leftarrow 0$	La solution à $\hat{\mathcal{P}}_{2/1}^0$ est modifiée par exclusion de $x_{i_0}$
$x_{i_0}^* = 0$	et	$b_{i_0} \leftarrow 1$	La solution à $\hat{\mathcal{P}}_{2/1}^1$ est identique à celle de $\hat{\mathcal{P}}_{2/1}$
$x_{i_0}^* = 0$	et	$b_{i_0} \leftarrow 0$	La solution à $\hat{\mathcal{P}}_{2/1}^0$ est identique à celle de $\hat{\mathcal{P}}_{2/1}$

TABLE 1: Tableau récapitulatif de l'impact des choix de branchement en fonction de la valeur de la variable  $x_{i_0}$  à l'optimum de  $\hat{\mathcal{P}}_{2/1}$

$$(\hat{\mathcal{P}}_{2/1}^{0'}) \quad \min_{\substack{\mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{|S_1|}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S} \setminus \{i_0\}} \mathbf{x}_{\bar{S} \setminus \{i_0\}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (20.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in \bar{S} \setminus \{i_0\} \quad (20.2)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \quad (20.3)$$

#### 2.4. Synthèse de l'impact des branchements :

L'analyse des stratégies de branchements sur le problème  $\hat{\mathcal{P}}_{2/0}$  nous permet de montrer que l'ajout des contraintes de somme à 1 et non négativité permet de réécrire les problèmes indépendamment des variables binaires et a pour effet de rendre caduque la contrainte de parcimonie. Ces problèmes ont été introduits par Heinz[4] sous le nom Fully Constrained Least Squares (FCLS).

Nous présentons dans le tableau (1) la synthèse des quatre cas de branchements possibles pour notre problème  $\hat{\mathcal{P}}_{2/1}$ .

*Notes :* Les variables  $b_{i_0}$  sont laissées dans ce tableau pour rappeler les stratégies de branchement ; nous avons démontré l'inutilité des variables binaires dans la formulation du problème  $\hat{\mathcal{P}}_{2/1}$ .

#### 2.5. Stratégies de recherche, branchement et de dominance

Comme nous l'avons évoqué au début de cette section, un algorithme de branch-and-bound peut être caractérisé par un ensemble de stratégies permettant de définir le comportement global de l'algorithme durant son exécution. Notons également que lors des différents paragraphes de cette section, nous avons conclu à l'inactivité de la contrainte de parcimonie et l'inutilité des variables binaires. L'implémentation de notre algorithme de branch-and-bound dédié devra donc, en tout nœud, garder une trace du nombre de variables fixées à 1 et ceci pour activer ou non la résolution des sous-problèmes et garder en mémoire la contrainte de parcimonie.

Nous détaillons ci-dessous nos choix de stratégies :

**Stratégie de recherche :** Parmi les stratégies proposées dans l'ouvrage de Wolsey[12], nous avons décidé de mettre en place un parcours de type profondeur d'abord (DFS, Depth-First Search) qui explore en priorité les nœuds les plus éloignés de la racine. Cette stratégie de recherche exploite la structure non symétrique de l'arbre afin de trouver rapidement des solutions réalisables *i.e.* des solutions avec  $K$  composantes non nulles (Voir R.Ben Mhenni[3]).

**Stratégie de branchement :** La stratégie mise en place dans le choix du partitionnement consiste, comme nous l'avons évoqué précédemment, à fixer la présence ou l'absence d'une variable  $x_{i_0}$  dans le support de la solution. Le choix de la variable à fixer est réalisé en prenant celle dont l'amplitude est maximum dans la solution obtenue après résolution du problème relâché  $\hat{\mathcal{P}}_{2/1}$ . Cette stratégie est utilisée par R.Ben Mhenni[3] dans ses travaux sur le démelange spectral et l'approximation parcimonieuse.

**Stratégie de dominance :** Pour sonder les nœuds durant l’exploration de notre arbre de recherche, nous mettons en place 3 règles classiques de dominance permettant d’élaguer des branches :

**Règle 1 :** par non réalisabilité qui dans notre problème, se caractérise par la consommation du budget de calcul alloué pour la résolution des sous problèmes aux nœud.

**Règle 2 :** par dominance, la valeur de la borne inférieure obtenue en un nœud est supérieure à la meilleure borne supérieure trouvée jusqu’alors.

**Règle 3 :** par optimalité, une solution proposant une meilleure borne primale a été trouvée. Nous définissons l’optimalité d’une solution en fonction du seuil d’intégrité fixé à  $\tau = 1e^{-8}$  ;  $\tau$  spécifie la quantité pour laquelle une variable entière peut être différente d’un nombre entier et être toujours considérée comme réalisable.

$$\lfloor x \rfloor - x < \tau$$

## 2.6. Conclusion

Dans ce chapitre, nous avons montré que la formulation des problèmes relâchés incluant les contraintes de somme à 1 et de non négativité nous ramenait à résoudre des problèmes de moindres carrés avec somme à 1 et positivité (FCLS). Nous avons également montré que cette relaxation continue avec la méthode d’ajout de variables binaires proposée par Bourguignon & Ninin[8]. nous assure d’obtenir une solution relâchée identique à celle obtenue à la norme  $\ell_0$  et ceci en nous permettant d’introduire des variables sur lesquelles nous pouvons effectuer des opérations de branchement. Enfin, nous avons montré l’impact des contraintes de somme à 1 et non négativité qui rendent inactive la parcimonie, contrainte fondamentale dans notre problème d’approximation.

Dans la section suivante, nous allons présenter la mise en œuvre des résultats démontrés précédemment à l’aide d’un algorithme de type branch-and-bound nous permettant alors de déterminer la composition du support en limitant les recherches exhaustives.

### 3. Mise en œuvre de l’algorithme de branch-and-bound dédié

Dans le chapitre précédent, nous avons montré que le problème étudié  $\mathcal{P}_{2/0}$  après reformulation MIP voyait les variables binaires et la contrainte de parcimonie disparaître. Nous avons également montré que l’évaluation du problème relâché en fonction de certains branchements n’amenait pas à obtenir des solutions différentes (voir tableau 1). Nous avons donc construit un algorithme de branch-and-bound intégrant ces différentes observations, déterminant le type de problème à résoudre et activant ou non la contrainte de parcimonie. Dans la suite de cette section, nous présenterons en détail l’implémentation de notre algorithme de branch-and-bound.

#### 3.1. Spécification des type de problèmes en fonction des nœuds

##### 3.1.1. Notations des sous-ensembles d’indices

Pour présenter les différentes résolutions réalisées dans l’arbre de recherche, nous proposons, pour un dictionnaire  $\mathbf{H} \in \mathbb{R}^{N \times Q}$  et un support  $S$ , les notations ensemblistes suivantes

- $S \subseteq Q$  et tel que  $\text{card}(S) = n$ , l’ensemble des  $n$  indices correspondants aux atomes de  $\mathbf{H}$
- $S_1 = \{i \in S | b_i \neq 0\}$  et tel que  $\text{card}(S_1) = n_1$ , l’ensemble des indices des variables incluses dans le support.
- $S_0 = \{i \in S | b_i = 0\}$  et tel que  $\text{card}(S_0) = n_0$ , l’ensemble des indices des variables exclues du support.
- $\bar{S} = Q \setminus \{S_1 \cup S_0\}$  et tel que  $\text{card}(\bar{S}) = \bar{n}$ , l’ensemble des indices des variables indéterminées.

*Remarque :* Bien que nous ayons montré que les variables binaires  $\mathbf{b} \in \{0, 1\}^Q$  disparaissent dans les reformulations avec les contraintes additionnelles inhérentes à notre problèmes, nous conserverons ces notations par souci de simplification des explications lorsque nous présenterons l’implémentation de l’algorithme.

##### 3.1.2. Principes de branchements

Comme nous l’avons évoqué dans la section précédente, la stratégie de branchement correspond à un partitionnement de l’espace de recherche en ”pariant” sur la présence ou l’absence d’une variable dans le support de la solution. Pour présenter le principe de branchement de notre algorithme, nous définissons  $\hat{\mathcal{P}}_{2/1}$  le problème associé à un nœud quelconque de notre arborescence de recherche. Comme nous l’avons évoqué dans la chapitre précédent, la règle de branchement que nous avons mis en place consiste en le choix de l’atome qui, dans la solution courante, possède l’amplitude maximale. Nous notons  $q \in \bar{S}$  l’indice de cette variable et sur laquelle nous réalisons une opération de branchement :

$$q = \arg \max_{i \in \bar{S}} \{x_i\}.$$

où  $\mathbf{x} \in [0, 1]^Q$  est une solution du nœud parent que nous allons subdiviser.

Nous utilisons les variables binaires  $\mathbf{b} \in \{0, 1\}^Q$  comme indicatrices de la présence (ou de l’absence) d’un atome dans le support de la solution. Le branchement sur la variable  $q \in \bar{S}$  engendre deux sous-problèmes :

- $\hat{\mathcal{P}}_{2/1}^{(g)}$  obtenu en fixant la valeur  $b_q \leftarrow 1$  et ayant pour effet d’imposer  $x_q \neq 0$  ;
- $\hat{\mathcal{P}}_{2/1}^{(d)}$  obtenu en fixant la valeur  $b_q \leftarrow 0$  et ayant pour effet d’imposer  $x_q = 0$  ;

Les opérations ensemblistes associées à cette stratégie de branchement sont présentées dans la figure (7).

##### 3.1.3. Précisions sur les problèmes résolus en fonction des nœuds

Comme nous l’avons mentionné la section 2.4, nous avons démontré que certaines propriétés de notre problème étaient supprimées par l’ajout des contraintes de somme à 1 et non négativité, notamment la contrainte de parcimonie, et que certaines stratégies de branchement ne permettaient pas d’obtenir de nouvelles solutions. Nous avons alors implémenté un algorithme de branch-and-bound permettant d’intégrer ces différentes observations, de détecter si une résolution doit être réalisée en un nœud et, dans ce cas, de déterminer le type de problème et de le résoudre.

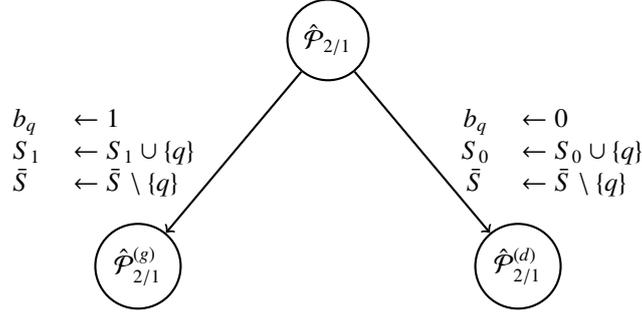


FIGURE 7: Exemple de branchement sur la variable  $b_q$  et création des nœuds fils : fils gauche  $\hat{\mathcal{P}}_{2/1}^{(g)}$  et fils droit  $\hat{\mathcal{P}}_{2/1}^{(d)}$ .

### 3.1.3.1 Sur la numérotation des nœuds

La numérotation des nœuds que nous avons mise en place est la suivante. Soit  $n$ , le numéro d'un nœud dans l'arbre,

- si le nœud est la racine de l'arbre, alors  $n = 0$
- si une opération de branchement peut être réalisée sur le nœud  $n$ , alors son fils
  - gauche aura pour indice  $n' = 2n + 1$  ;
  - droit aura pour indice  $n' = 2n + 2$  ;

Le principal avantage de cette numérotation des nœuds est qu'elle nous permet de connaître à tout moment le chemin qui est emprunté par l'algorithme<sup>4</sup>.

### 3.1.3.2 Sur la formulation des problèmes

Pour notre algorithme de branch-and-bound dédié à la résolution de problème parcimonieux sous contrainte de somme à 1 et non négativité, nous avons constaté que la formulation du problème dépendait de la position dans l'arbre et notamment du type de nœud rencontré. Dans la section suivante, nous allons décrire les trois types de nœuds entraînant une résolution ainsi que les formulations des problèmes associées à ces mêmes nœuds.

#### Problème de type racine $\hat{\mathcal{P}}_0^r$

Au nœud racine ( $n = 0$ ), aucune variable n'est fixée à 0 ou 1 ; nous avons donc  $\bar{S} = S = \llbracket 1, n \rrbracket$  et  $S_1 = S_0 = \emptyset$ . En ce nœud précis, le problème à résoudre est donc le suivant :

$$(\hat{\mathcal{P}}_0^r) \quad \min_{x_{\bar{S}} \in [0,1]^{\bar{n}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \text{ s.c. } \sum_{i \in \bar{S}} x_i \leq 1 \quad (21.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in \bar{S} \quad (21.2)$$

La résolution au nœud racine nous permet d'obtenir une première solution, non parcimonieuse, où les valeurs des variables  $x_i \forall i \in \bar{S}$  vont nous permettre de réaliser les  $K$  premiers branchements.

4. Par propriété de connexité des arbres binaires de recherche, il existe un unique chemin entre la racine et tout nœud du graphe

### Problème de type feuille $\hat{\mathcal{P}}_n^f$

Les nœuds feuilles d'indice  $n$  impair sont obtenus après avoir branché  $K$  fois sur le fils gauche *i.e.* après avoir fixé  $K$  variables  $b_i$   $i \in \mathcal{Q}$  à 1. Dans ce cas particulier, nous devons activer la contrainte de parcimonie rendue inactive par la contrainte de somme à 1. Pour ce faire, nous devons forcer  $x_{q_0} = 0$   $\forall q_0 \in \mathcal{S}_0 \cup \bar{\mathcal{S}}$  afin d'estimer les abondances des  $K$  variables sélectionnées  $x_{q_1}$   $\forall q_1 \in \mathcal{S}_1$ . Le problème *feuille* peut être formulé de la manière suivante :

$$(\hat{\mathcal{P}}_n^f) \quad \min_{x_{S_1} \in [0,1]^{|S_1|}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in S_1} x_i \leq 1 \quad (22.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \quad (22.2)$$

$$x_i = 0 \quad \forall i \in \bar{\mathcal{S}} \cup \mathcal{S}_0 \quad (22.3)$$

La dénomination de ce nœud provient du fait que nous sommes sûrs de nous arrêter après avoir fixé  $K$  variables à 1. Ces nœuds, s'ils ne sont pas élagués dans la suite de l'exécution de l'algorithme, ont pour objectif de proposer *les meilleures solutions réalisables* à  $K$  composantes *i.e.* des bornes supérieures sur la valeur de la borne primale en saturant la contrainte de parcimonie  $\|\cdot\|_0 \leq K$ .

La première résolution d'un nœud feuille va également permettre d'obtenir une première borne supérieure *i.e.* une première valeur de l'erreur en norme  $\ell_2$  pour une solution satisfaisant les contraintes ; le reste de l'exécution de l'algorithme va permettre de raffiner cette erreur et au final, de prouver l'optimalité de cette borne primale.

### Problème de type exploration $\hat{\mathcal{P}}_n^e$

Les nœuds d'exploration d'indice  $n$  pair sont obtenus après avoir résolu un nœud feuille par application de la stratégie d'exploration DFS. Suivant le principe de parcours de l'arbre de recherche par profondeur d'abord, un nœud d'exploration correspond à l'estimation des abondances en omettant l'indice de la dernière composante ajoutée au support ; la composante exclue du support est celle qui, considérant la dernière solution obtenue, possède la plus faible amplitude et qui est la dernière composante sur laquelle nous avons réalisé une opération de branchement. La formulation d'un problème de type exploration est la suivante :

$$(\hat{\mathcal{P}}_n^e) \quad \min_{\substack{x_{\bar{\mathcal{S}}} \in [0,1]^{\bar{\mathcal{S}}} \\ x_{S_1} \in [0,1]^{|S_1|}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{\mathcal{S}}} \mathbf{x}_{\bar{\mathcal{S}}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in S_1 \cup \bar{\mathcal{S}}} x_i \leq 1 \quad (23.1)$$

$$0 \leq x_i \leq 1 \quad \forall i \in S_1 \cup \bar{\mathcal{S}} \quad (23.2)$$

Les nœuds d'explorations sont obtenus après avoir réalisé le branchement  $x_i \neq 0$  et  $b_i \leftarrow 0$  ; cette stratégie permet d'intégrer un nouvel atome à la *meilleure solution réalisable* à  $K$  composantes pour laquelle nous avons mis à jour la borne supérieure globale. La résolution de ces problèmes aux nœuds feuilles nous permet d'obtenir des bornes inférieures sur la valeur optimale de l'erreur en norme  $\ell_2$ .

Les nœuds d'explorations visent à répondre aux questions suivantes : *Pouvons nous trouver une solution minimisant l'erreur en norme  $\ell_2$  avec une composante différente ?* ou encore *La dernière composante que nous avons incluse dans le support est-elle réellement indispensable ?*

*Remarque :* nous noterons  $\hat{\mathcal{P}}_n^b$  le problème associé au nœud  $n$  pour lequel l'algorithme n'a pas exécuté de résolution mais seulement un branchement.

### 3.1.4. Exemple d'exécution de l'algorithme sur un problème avec $K=3$

Utilisant les notations que nous avons défini précédemment et les problèmes que nous avons formulé, nous présentons dans ce paragraphe l'exécution de l'algorithme de branch-and-bound dédié pour une instance possédant un coefficient de parcimonie  $K = 3$ . L'exécution de l'algorithme est présenté sur la figure 8.

A partir des observations réalisées dans les paragraphes 2.3.4 et 2.4, nous avons montré que le branchement  $x_q \neq 0$  et  $b_q \leftarrow 1$  avec  $q = \arg \max_{i \in \bar{S}} (|x_i|)$  et  $x_i$  est la solution du problème calculé au noeud parent :

1. ne nécessite pas de ré-optimisation lors de la création de ses nœuds fils et ceci tant que  $\text{card}(S_1) < K$
2. nécessite de lancer une résolution dès lors que  $\text{card}(S_1) = K$  ; la contrainte de parcimonie devient active car *plus forte* que la contrainte de somme à 1

C'est sur ce principe que vont s'exécuter les explorations de nœuds dans notre arbre de recherche afin de construire une première solution réalisable à  $K$  composantes.

Tout d'abord, plaçons nous dans le nœud racine de notre arbre et résolvons le problème associé  $\hat{\mathcal{P}}_0^r$ , nous obtenons alors une première solution non parcimonieuse  $\mathbf{x}^*$ . Nous calculons ensuite l'indice  $q_0$  de la variable ayant l'amplitude maximale *i.e.*  $q_0 = \arg \max_{i \in \bar{S}} (x_i^*)$  et nous réalisons une première opération de branchement sur cette variable en mettant à jour l'ensemble des valeurs fixées à 1 pour le fils gauche ( $S_1 \leftarrow S_1 \cup \{q_0\}$ ), l'ensemble des valeurs fixées à 0 pour le fils droit ( $S_0 \leftarrow S_0 \cup \{q_0\}$ ) et retirons  $q_0$  de l'ensemble des indices non encore déterminés ( $\bar{S} \leftarrow \bar{S} \setminus \{q_0\}$ ). L'opération réalisée sur le fils gauche entraîne la soustraction d'une unité à la valeur de  $K$  ; nous avons alors  $\sum_{i \in \bar{S}} b_i \leq K - \text{card}(S_1)$  avec  $\text{card}(S_1) = 1$ . En appliquant notre règle de parcours DFS, nous allons répéter cette opération de branchement sur le fils gauche de chaque nœud  $\hat{\mathcal{P}}_n^b$  jusqu'à ce que  $\text{card}(S_1) = K$  *i.e.* nous allons à partir de  $q_0$  et de  $\mathbf{x}^*$  afin de détecter l'indice  $q_1$  tel que :

$$q_1 = \arg \max_{i \in \bar{S} \setminus \{q_0\}} \{x_i^*\}$$

Dès lors que la profondeur  $K$  est atteinte, nous allons pouvoir résoudre un problème de type feuille  $\hat{\mathcal{P}}_n^f$  dans lequel les variables dont l'indice est dans  $\bar{S}$  sont forcées à être nulles. La contrainte de parcimonie ayant été activée, nous avons obtenu une solution réalisable à  $K$  composantes qui nous permet de calculer une borne supérieure sur la solution optimale. En pratique, les  $K$  premiers branchements sur fils gauches nous permettent de mettre à jour la valeur de la meilleure solution courante *i.e.* la borne supérieure.

Après avoir réalisé les  $K$  premier branchement sur fils gauches, la règle de parcours DFS nous amène à réaliser un branchement de type  $x_q \neq 0$  et  $b_q \leftarrow 0$  ; ce branchement entraîne la résolution d'un noeud de type exploration. Cette même résolution nous permet d'obtenir une solution alternative où l'indice du dernier atome inclus dans le support  $S_1$  est retiré.

L'obtention de cette nouvelle solution se fait par résolution du problème de type exploration noté  $\hat{\mathcal{P}}_n^e$ . Pour notre exemple où  $K = 3$ , nous isolons alors un atome  $q_3 \in \bar{S}$  pouvant remplacer à l'atome d'indice  $q_2$  dans le support  $S_1$ .

Cette résolution nous permet de réaliser une relaxation et d'obtenir une borne inférieure sur la valeur de la solution optimale ; la branche correspondante peut être élaguée si la borne inférieure est supérieure à la borne primale courante signifiant alors que l'atome  $q_2$  est indispensable pour minimiser l'erreur en norme  $\ell_2$ .

Ce processus de recherche et de résolution, représenté dans la figure 8, est répété tant que la preuve de l'optimalité de la solution primale n'est pas obtenue et que le temps total de résolution n'est pas atteint. La figure 9 nous permet de visualiser la composition de la solution en différents nœuds et d'illustrer l'impact des ensembles  $S_1$ ,  $S_0$  et  $\bar{S}$  dans les différentes résolutions des trois types de problèmes. La figure :

- 9a représente la composition du mélange au noeud racine pour lequel aucun atome n'a été sélectionné ( $S_1 = S_0 = \emptyset$ ,  $\bar{S} = \mathcal{Q}$ ). Les composantes sont non nulles ; certaines d'entre elles sont numériquement

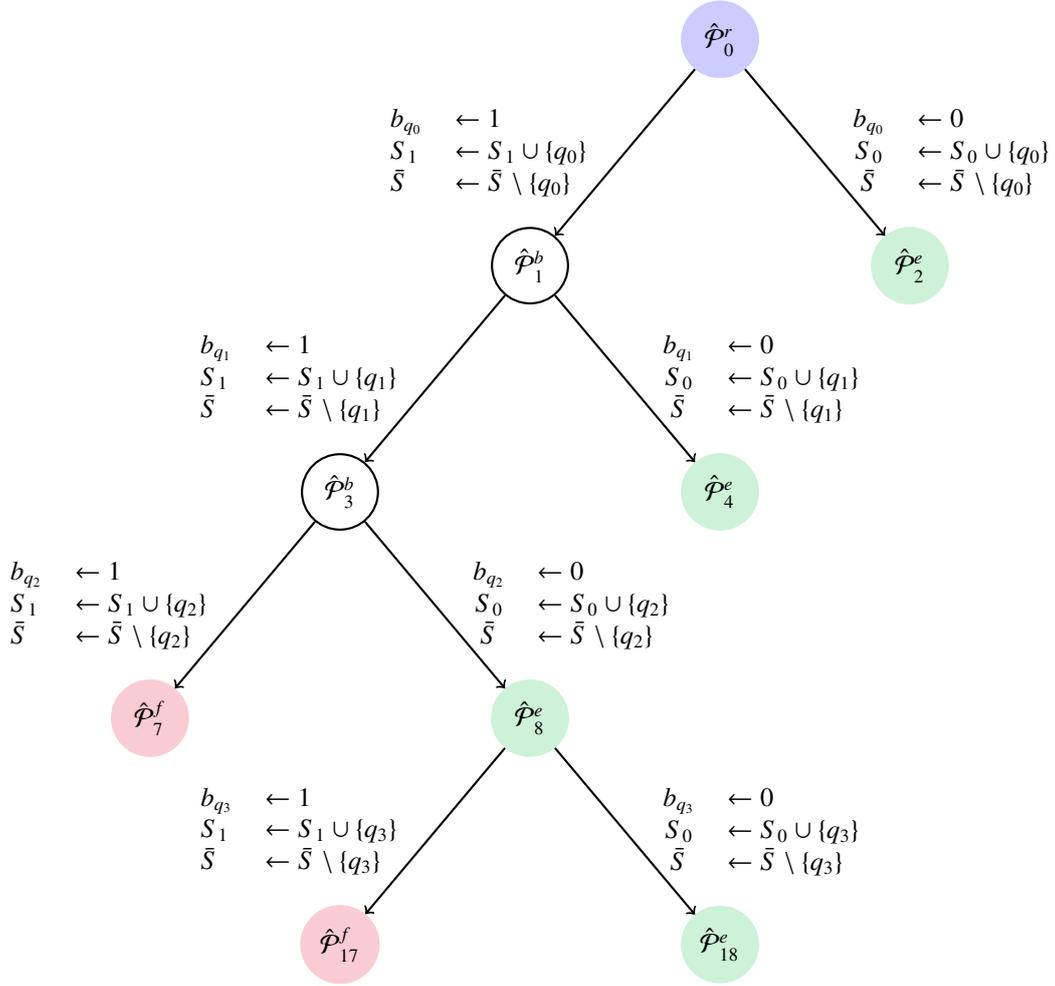


FIGURE 8: Arbre de branch-and-bound - exemple d'exécution sur un problème avec  $K=3$  - nœuds de types **racine**, **feuille**, **exploration**

faibles (de l'ordre de  $1e^{-7}$ );

- 9b correspond au support obtenu après avoir branché  $K = 3$  fois sur les fils gauches des nœuds et pour lequel nous avons  $S_1 = \{q_0, q_1, q_2\}$  avec  $q_0 = 16$ ,  $q_1 = 6$ ,  $q_2 = 3$ ; nous avons ainsi  $x_q \neq 0 \forall q \in S_1$  avec  $\text{card}(S_1) = K$  et  $x_{\bar{q}} = 0 \forall \bar{q} \in \bar{S} \cup S_0$  avec  $\text{card}(\bar{S}) = \text{card}(Q \setminus K)$ ;
- 9c montre le support  $S_1$  obtenu après avoir forcé l'atome  $q_2$  à être nul dans le mélange ( $S_0 \leftarrow S_0 \cup \{q_2\}$ ) et pour lequel nous avons recalculé les abondances; après résolution, un nouvel atome  $q_3 = 2$  est ajouté au support  $S_1$ . Nous obtenons ainsi  $x_q \neq 0 \forall q \in S_1 \cup (\bar{S} \setminus \{q_2\})$  et  $x_{\bar{q}} = 0 \forall \bar{q} \in S_0$ .

### 3.2. Implémentation de l'algorithme

Maintenant que nous avons étudié les méthodes de branchement, de mise à jour des ensembles d'indices d'atomes et les différents types de problèmes résolus en fonction du nœud exploré, nous allons pouvoir présenter l'algorithme de branch-and-bound que nous avons développé pour répondre au problème de minimisation de l'erreur  $\ell_2$  sous contrainte de parcimonie, de non négativité et de somme à 1.

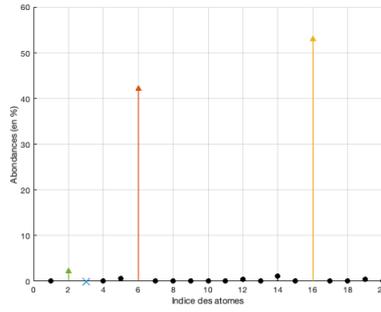
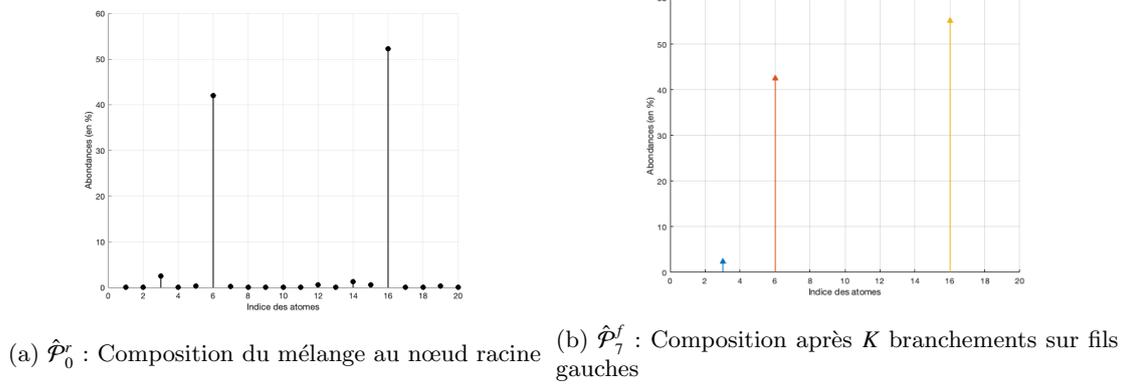


FIGURE 9: Solutions obtenues en fonction des résolutions de l'arbre de branch-and-bound. Les atomes mentionnés dans les commentaires sont  $q_0$ ,  $q_1$ ,  $q_2$  et  $q_3$ ; les atomes  $\bar{q} \in \bar{\mathcal{S}}$  sont colorés en noir.

Dans un premier temps, nous commencerons par présenter le pseudo-code et les notations associées et dans un second temps, nous illustrerons le fonctionnement de cet algorithme sur une instance de  $\mathcal{P}_{2/0}$ .

### 3.2.1. Notations

Nous présentons l'ensemble des notations que nous utilisons dans la description du pseudo-code de notre algorithme de branch-and-bound :

- $K$  : le coefficient de parcimonie du problème  $\mathcal{P}_{2/0}$ .
- $Q$  : le nombre d'atomes présents dans le dictionnaire  $\mathbf{H}$ .
- $n$  : le numéro du nœud courant.
- $L$  : l'ensemble des sous-problèmes actifs. La structure associée est une pile qui nous permet d'accéder au dernier nœud stocké dans  $L^5$ .
- $\hat{\mathbf{x}}$  : la meilleure solution réalisable connue.
- $z_U$  : la borne supérieure courante.
- $y \in \{r, f, e\}$  : type de problème rencontré au nœud courant.
- $\hat{\mathcal{P}}_n^y$  : le sous-problème relâché au nœud  $n$  de type  $y$ .

5. La stratégie de gestion de la pile des problèmes actifs est donc de type LIFO. Pour la suite de l'algorithme, on l'opération  $L \leftarrow L \setminus X$  est réalisée lors de l'appel à la fonction `sommet(L)`

---

**Algorithme 2** : Structure générale de notre algorithme de branch-and-bound

---

**Entrées** :  $K$  le coefficient de parcimonie,  $\mathbf{H}$  le dictionnaire,  $\mathbf{y}$  le vecteur d'observations

**Sortie** :  $z_U$  l'optimum global du problème et  $\hat{\mathbf{x}}$  la solution associée

**Initialisation** :

$n \leftarrow 0$

$z_U \leftarrow +\infty, \hat{\mathbf{x}} \leftarrow \emptyset, k^n = 0$

$\mathbf{lb} \leftarrow \mathbf{0}_Q, \mathbf{ub} \leftarrow \mathbf{1}_Q$

$L \leftarrow \{(\hat{\mathcal{P}}_n^r, k^n, \mathbf{lb}, \mathbf{ub})\}$

$r \leftarrow 1, f \leftarrow 0, e \leftarrow 0$

**Exécution de l'algorithme** :

**tant que** ( $L \neq \emptyset$ ) **faire**

$\{(\hat{\mathcal{P}}_n^y), k^n, \mathbf{lb}, \mathbf{ub}\} \leftarrow \text{sommet}(L)$

$S_1 \leftarrow \{i \in Q \mid \text{lb}(i) = 1 \wedge \text{ub}(i) = 1\}$

$S_0 \leftarrow \{i \in Q \mid \text{lb}(i) = 0 \wedge \text{ub}(i) = 0\}$

$\bar{S} \leftarrow Q \setminus \{S_1 \cup S_0\}$

    Aller à l'étape *Détermination du type de nœud* (Algorithme 3)

    Aller à l'étape *Résolution du nœud courant* (Algorithme 4)

    Aller à l'étape *Élagage* (Algorithme 5)

    Aller à l'étape *Branchement* (Algorithme 6)

**fin**

---

- $z_L^n$  : la borne inférieure obtenue au nœud  $n$  par résolution de  $\hat{\mathcal{P}}_n^y$ .
- $\mathbf{x}_L^n$  : la solution associée à la borne inférieure  $z_L^n$ .
- $k^n$  : le nombre de valeurs fixées à 1 au nœud  $n$  - compteur de parcimonie.
- $\text{sep} \in \{0, 1\}$  : booléen permettant de d'indiquer si une opération de séparation peut être réalisée.
- $(r, f, e) \in \{0, 1\}^3$  : booléens permettant de d'indiquer le type de problème rencontré
- $\mathbf{ub} \in \{0, 1\}^Q$  (resp.  $\mathbf{lb} \in \{0, 1\}^Q$ ) : un vecteur de bornes supérieures (resp. inférieures) sur les variables  $\mathbf{x} \in [0, 1]^Q$
- $\tau$  : seuil d'intégrité.

### 3.2.2. Pseudo-code et principe de fonctionnement

Nous présentons maintenant le pseudo-code de l'algorithme. Afin d'assurer une meilleure compréhension, nous avons séparé l'algorithme en plusieurs sous-séquences d'instructions appelées par une boucle présentée ci-contre.

L'algorithme 2 présente le pseudo-code global ainsi que l'initialisation des vecteurs de bornes supérieures (resp. inférieures) notés  $\mathbf{lb}$  (resp.  $\mathbf{ub}$ ) et de la valeur de la borne supérieure globale  $z_U$  ainsi que la solution courante associée  $\hat{\mathbf{x}}$ . Le compteur de parcimonie courant  $k^n$  est également fixé à 0. Pour simplifier la lecture de ce code, nous avons choisi de stocker dans  $L$  des 4-uplets composés de la structure de problème courant à résoudre (ici  $\hat{\mathcal{P}}_n^r$  où  $r$  correspond au nœud racine), le coefficient de parcimonie courant  $k^n$  ainsi que les vecteurs de bornes. Le booléen  $r$ , indicateur du nœud racine est fixé à 1 ; le booléen  $f$  (resp.  $e$ ) indicateur du type feuille (resp. exploration) est fixé à 0 lors de l'initialisation.

---

**Algorithme 3** : Détermination du type de nœud

---

```
si (mod (n, 2) = 0 ∧ r = 0) alors
| r ← 0, f ← 0, e ← 1
sinon
| si k^n = K alors
| | r ← 0, f ← 1, e ← 0
| fin
fin
```

---

---

**Algorithme 4** : Résolution du nœud courant

---

```
si r = 1 alors
| Résoudre  $\hat{\mathcal{P}}_0^r$  avec  $\mathbf{lb} = \mathbf{0}_Q$ ,  $\mathbf{ub} = \mathbf{1}_Q$ 
| sep ← 1
sinon
| si f = 1 alors
| | Résoudre  $\hat{\mathcal{P}}_n^f$  avec  $\mathbf{lb} = \mathbf{0}_Q$ ,  $\mathbf{ub} = \mathbf{1}^{S_1}$ 
| | sep ← 0
| sinon
| | si e = 1 alors
| | | Résoudre  $\hat{\mathcal{P}}_n^e$  avec  $\mathbf{lb} = \mathbf{0}_Q$ ,  $\mathbf{ub} = \mathbf{1}^{S_1 \cup \bar{S}}$ 
| | | sep ← 1
| | fin
| fin
fin
```

---

L'exécution de l'algorithme peut alors commencer. La première étape consiste en la récupération des informations relatives au problème courant à résoudre *i.e.* le 4-uplet ; à partir de ces informations, les différents ensembles  $S_1$ ,  $S_0$  et  $\bar{S}$  sont calculés pour construire le problème courant conformément aux formulations présentées en début de section.

La suite de l'algorithme consiste en une série d'appels à des séquences d'opérations que nous allons maintenant décrire. Comme nous l'avons montré le paragraphe 3.1.3.2, les résolutions de problèmes ne sont amorcées que pour certains type de nœuds. Nous retrouvons alors dans l'algorithme 3 l'intérêt de la numérotation des nœuds présentée précédemment pour la détection des nœuds où nous devons utiliser la formulation de type exploration. Rappelons également que certains nœuds ne servent qu'aux opérations de branchement et pour lesquels aucune ré-optimisation n'est nécessaire.

A partir de la détermination du type de problème à résoudre obtenue avec l'algorithme 3, nous pouvons alors appliqué la formulation adaptée dans l'algorithme 4. Le nœud racine sera résolu une seule et unique fois pour obtenir une solution que nous allons utiliser pour lancer les premiers branchements (sep ← 1). De par la relative simplicité des problèmes résolus, notre implémentation ne conserve en mémoire que les vecteurs de bornes ( $\mathbf{lb}$ ,  $\mathbf{ub}$ ) associés aux problèmes à résoudre. La notation indicatrice  $\mathbf{1}^S$  utilisée dans ce pseudo-code signifie :

$$\mathbf{1}^S \in \{0, 1\}^Q \quad \text{avec} \quad \mathbf{1}^S(i) = \begin{cases} 1 & \text{si } i \in S \\ 0 & \text{sinon} \end{cases} \quad \forall i \in Q$$

Rappelons que dans le cas d'un problème de type feuille, les opérations de branchement sont arrêtées car la contrainte de parcimonie est active ( $\|\mathbf{x}\|_0 \leq K$ ) ce qui explique alors la valeur sep ← 0. Enfin, le dernier

---

**Algorithme 5** : Procédure d'élagage

---

```
si  $(r = 0 \wedge f = 1) \vee (r = 0 \wedge e = 1)$  alors
|
| si  $\mathbf{x}_L^n$  est non réalisable alors
| | sep  $\leftarrow 0$  /* Elagage par non réalisabilité */
| sinon
| | si  $z_L^n \geq z_U$  alors
| | | sep  $\leftarrow 0$  /* Elagage par dominance */
| | sinon
| | | si  $|\lfloor \mathbf{x}_L^n(i) \rfloor - \mathbf{x}_L^n(i)| < \tau \forall i \in Q$  alors
| | | | sep  $\leftarrow 0$  /* Elagage par optimalité */
| | | |  $\hat{\mathbf{x}} \leftarrow \mathbf{x}_L^n$ 
| | | |  $z_U \leftarrow z_L^n$ 
| | | fin
| | fin
| fin
fin
```

---

cas concerne les nœuds de type exploration avec lesquels nous tentons d'obtenir des bornes inférieures sur la valeur optimale en résolvant des sous-problèmes relâchés avec les atomes d'indices  $i \in S_1 \cup \bar{S}$ .

L'algorithme 5 présente les trois règles classiques d'élagage communes à bon nombre d'algorithmes de branch-and-bound. Nous pouvons cependant noter la spécificité de notre problème qui est de n'activer ces opérations de test de dominance seulement pour les nœuds de types feuille ou exploration. La séparation est alors rendue impossible par la mise à jour du booléen  $\text{sep} \leftarrow 0$ .

Nous retrouvons dans l'algorithme 5 les trois règles nous permettant d'élaguer les branches de notre arbre de recherche et présentées dans le paragraphe 2.5. Notons plus particulièrement l'expression de la règle nous permettant de sonder des nœuds par optimalité; cette dernière faisant intervenir le seuil d'intégrité  $\tau$  nous permettant de spécifier la quantité pour laquelle laquelle une variable entière peut être différente d'un nombre entier et être toujours considérée comme réalisable. Si cette condition est vérifiée, alors la séparation est rendue inactive, la valeur de la borne primale  $z_U$  et la solution associée  $\hat{\mathbf{x}}$  sont mises à jour.

Enfin, l'algorithme 6 présente la procédure de branchement que nous avons illustré dans la figure 8. Le vecteur  $\bar{S}$  nous permet de détecter les indices des variables pour lesquelles aucun branchement n'a encore été réalisé. Si lors de la résolution et la procédure d'élagage, la valeur de  $\text{sep}$  est encore égale à 1, alors la procédure de branchement est lancée; cette dernière consiste en la mise à jour des vecteurs de bornes  $(\mathbf{lb}, \mathbf{ub})$  en ajoutant l'indice de la variable qui possède, localement, l'amplitude maximale dans les différents ensembles  $S_1$  ou  $S_0$  en la retirant de  $\bar{S}$ . Pour assurer le comportement de recherche Profondeur d'abord, nous ajoutons d'abord le problème correspondant au branchement  $b_q \leftarrow 0$  dans la pile  $L$  puis celui associé au branchement  $b_q \leftarrow 1$ . La numérotation des nœuds se fait conformément à celle présentée précédemment dans cette même section et le compteur de parcimonie n'est incrémenté que dans le cas du branchement  $b_q \leftarrow 1$  pour lequel nous forçons une variable à être présente dans le support de la solution.

### 3.2.3. Application de l'algorithme de branch-and-bound dédié sur un exemple

Nous présentons un exemple d'exécution de notre algorithme de branch-and-bound dédié sur un problème simulé avec :

- $Q = 20$  le nombre d'atomes dans le dictionnaire  $\mathbf{H}$
- $N = 224$  le nombre d'observations

---

**Algorithme 6** : Procédure de branchement

---

```
 $\bar{S} \leftarrow \{i \in Q \mid lb(i) \neq ub(i)\}$  /* L'ensemble des indices valides pour le branchement */  
si ( $sep = 1 \wedge card(\bar{S}) \neq \emptyset \wedge f = 0$ ) alors  
    /* Indice sélectionné pour l'opération de branchement */  
     $i_{sep} \leftarrow \arg \max_{i \in \bar{S}} \{x_L^n\}$   
    /* Branchement sur fils droit  $b_{i_{sep}} \leftarrow 0$  */  
     $lb(i) \leftarrow \begin{cases} 1 & \text{si } i \in S_1 \\ 0 & \text{si } i \in \bar{S} \cup S_0 \cup \{i_{sep}\} \end{cases} \quad \forall i \in Q$   
     $ub(i) \leftarrow \begin{cases} 1 & \text{si } i \in S_1 \\ 0 & \text{si } i \in \bar{S} \cup S_0 \cup \{i_{sep}\} \end{cases} \quad \forall i \in Q$   
     $k^{2n+2} = k^n$   
     $L \leftarrow L \cup \{(\hat{\mathcal{P}}_{2/1}^{2n+2}), k^{2n+2}, \mathbf{lb}, \mathbf{ub}\}$   
    /* Branchement sur fils gauche  $b_{i_{sep}} \leftarrow 1$  */  
     $lb(i) \leftarrow \begin{cases} 1 & \text{si } i \in S_1 \cup \{i_{sep}\} \\ 0 & \text{si } i \in \bar{S} \cup S_0 \end{cases} \quad \forall i \in Q$   
     $ub(i) \leftarrow \begin{cases} 1 & \text{si } i \in S_1 \cup \{i_{sep}\} \\ 0 & \text{si } i \in \bar{S} \cup S_0 \end{cases} \quad \forall i \in Q$   
     $k^{2n+1} = k^n + 1$   
     $L \leftarrow L \cup \{(\hat{\mathcal{P}}_{2/1}^{2n+1}), k^{2n+1}, \mathbf{lb}, \mathbf{ub}\}$   
fin
```

---

- $K = 5$  le coefficient de parcimonie
- SNR= 45 le bruit appliqué au données<sup>6</sup>

L'arbre de recherche est présenté par la figure 11 et l'évolution de la borne supérieure globale est représentée sur la figure 10. Nous pouvons observer sur ces mêmes figures que l'algorithme a réussi à prouver l'optimalité du problème et ceci :

- en explorant 17 nœuds ;
- en réalisant 2 élagages par optimalité aux nœuds feuilles 33 et 65 ;
- en réalisant 7 élagages par dominance aux nœuds feuilles 133 et 269 ainsi qu'aux nœuds d'explorations 270, 16, 8, 4 et 2 ;

*Remarque* : l'erreur en norme  $\ell_2$  ne peut être négative. La fonction objectif utilisée dans l'implémentation de notre branch-and-bound est donnée par la partie quadratique de la norme  $\ell_2$  soit  $\|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 - \|\mathbf{y}\|_2^2$ . Ainsi, nous pouvons observer que la valeur de la borne primale à la fin de l'exécution de l'algorithme vaut  $-21,67561$ .

La figure 10 permet d'observer la diminution de la borne supérieure globale et ceci à deux reprises aux itérations 6 et 8 de l'algorithme correspondant aux nœuds feuilles 33 et 65. Notons que l'absence de trait pour la courbe de la borne primale signifie que cette dernière est encore fixée à  $+\infty$ .

### 3.3. Conclusion

Ce chapitre nous a permis de présenter l'implémentation de l'algorithme de branch-and-bound dédié au problème de démixage sous contrainte de parcimonie, de somme à 1 et de non négativité. Nous avons pu

---

6. La notion de SNR sera définie dans le paragraphe 5.1.2 du rapport.

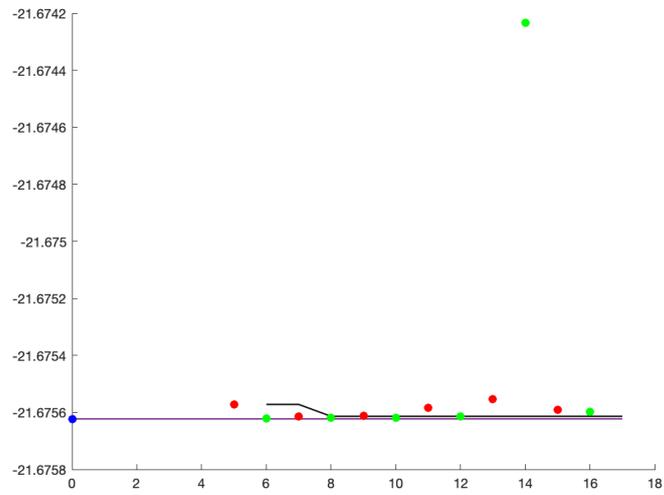


FIGURE 10: Evolution des bornes primale et duale - exécution sur une instance ( $Q = 20$ ,  $K = 5$ ). Types des nœuds : *racine*, *feuille*, *exploration*, la borne primale est représentée en noire et la borne duale en violet.

observer la mise en place des différentes formulations du problème en fonction des types de nœuds rencontrés mais également l'impact des branchements et de l'ajout des contraintes qui rendent inactive la contrainte de parcimonie, cette dernière devant être activée en certains nœuds de l'arbre.

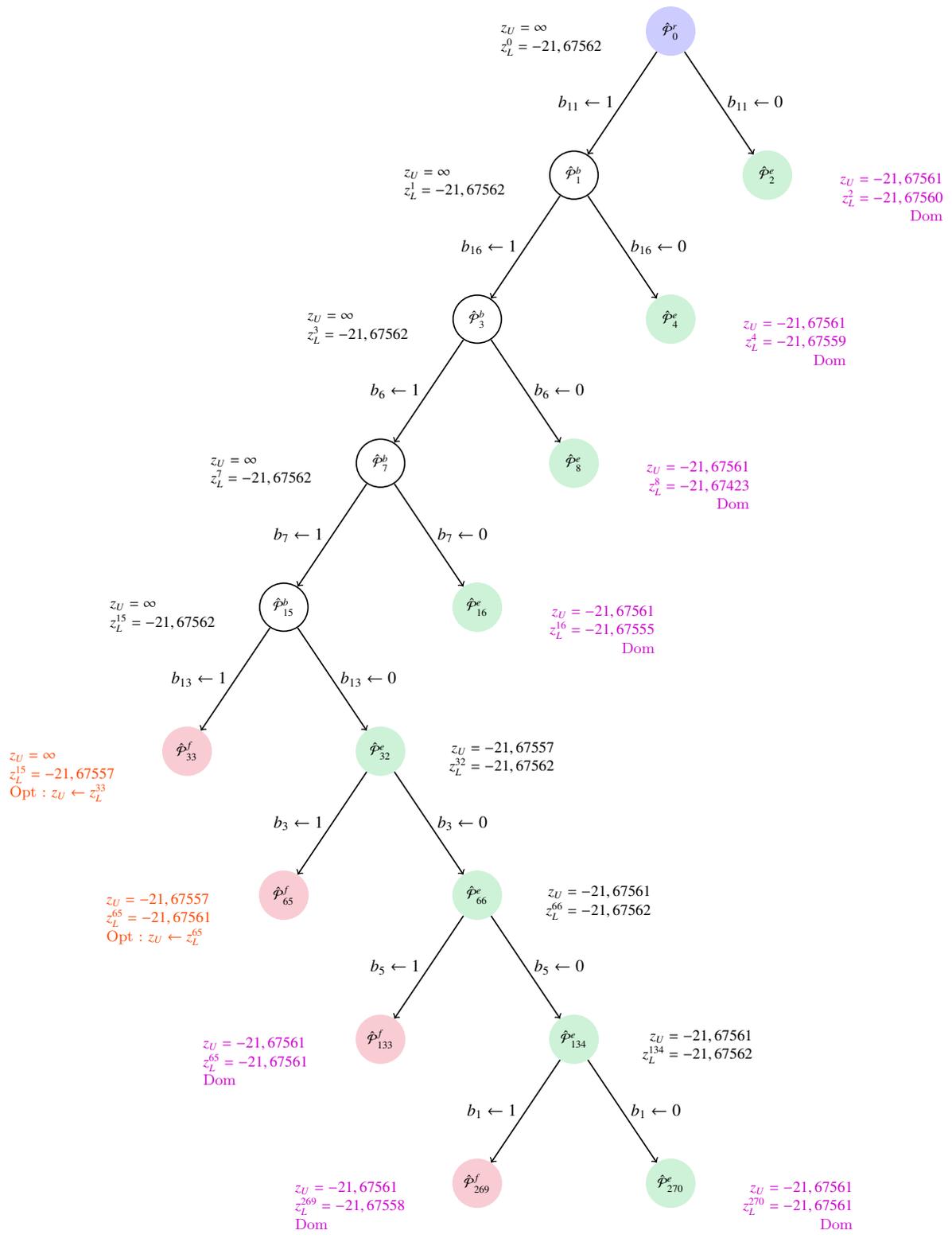


FIGURE 11: Arbre de branch-and-bound - exécution sur une instance (Q = 20, K = 5)

#### 4. Stratégies développées pour l'évaluation des nœuds l'obtention de bornes inférieures

Tout d'abord, nous rappelons les formulations des problèmes  $\mathcal{P}_{2/0}$  et  $\hat{\mathcal{P}}_{2/1}$  :

$$(\mathcal{P}_{2/0}) \quad \min_{\mathbf{x} \in [0,1]^n} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.c.} \quad \|\mathbf{x}\|_0 \leq K, \quad \|\mathbf{x}\|_1 \leq 1 \quad (24)$$

et la formulation relâchée en un nœud  $n$  de notre arborescence de recherche :

$$(\hat{\mathcal{P}}_{2/1,n}) \quad \min_{\substack{b_{\bar{S}} \in [0,1]^{\bar{p}}, x_{\bar{S}} \in [0,1]^{\bar{p}} \\ x_{S_1} \in [0,1]^{|S_1|}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1}\mathbf{x}_{S_1} - \mathbf{H}_{\bar{S}}\mathbf{x}_{\bar{S}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in \bar{S}} x_i \leq 1, \quad 0 \leq x_i \leq b_i \quad \forall i \in \bar{S} \quad (25)$$

Dans le chapitre précédent, nous avons présenté la structure générale de l'algorithme de branch-and-bound dédié au problème  $\mathcal{P}_{2/0}$  dans lequel nous intégrons les différentes observations réalisées sur la structure du problème relâché  $\hat{\mathcal{P}}_{2/1}$ .

Notre objectif dans cette section est de présenter cette fois ci les méthodes de résolutions exactes et dédiées que nous avons mises en œuvre pour résoudre, quand cela est nécessaire, les sous-problèmes en chaque nœud de notre arborescence et ainsi tenter d'obtenir rapidement de bonnes bornes inférieures.

##### 4.1. Préambule : stratégies de pré-calculs et de restriction des variables candidates au support

L'objectif principal de ce stage est de mettre en avant la compétitivité de méthodes de résolution dédiées par rapport aux solveurs génériques tels que Cplex et sa formulation MIQP. Pour ce faire, nous devons donc réfléchir aux stratégies nous permettant d'être plus efficaces dans l'évaluation des bornes inférieures.

En plus du développement d'algorithmes de résolution dédiés qui constitue déjà un stratégie permettant d'accélérer l'évaluation des nœuds de notre arbre de branch-and-bound, deux leviers supplémentaires peuvent être mis en place pour augmenter la rapidité d'exécution de nos méthodes pour obtenir des bornes duales ; nous allons présenter succinctement les méthodes de pré-calcul et de restriction des variables candidates au support. Notons également qu'il sera possible, dans la suite de cette étude, de réfléchir à d'autres leviers permettant de réduire les temps d'évaluation des sous-problèmes de notre arbre de recherche.

###### 4.1.1. Pré-calculs de produits matriciels

Comme nous l'avons mentionné dans la chapitre précédent, de par leur relative simplicité, nous faisons le choix de reconstruire les problèmes  $\hat{\mathcal{P}}_{2/1,n}$  dès lors que ce dernier doit être résolu. Cependant, la formulation des attributs de certains solveurs utilisés nécessite le passage en paramètre des quantités  $\mathbf{H}^T\mathbf{H}$  et  $\mathbf{y}^T\mathbf{H}$ . Afin de gagner en rapidité lors des appels aux solveurs, nous avons proposé une méthode permettant de réaliser une seule et unique fois l'évaluation de ces deux quantités et de les passer directement en paramètres de la fonction objectif dont le développement est présenté dans l'équation 28 en page 35. Les résultats présentés dans le chapitre suivant montreront l'importance du gain de temps en termes de nombre de nœuds explorés dans le cas d'instances compliquées.

###### 4.1.2. Restriction des variables candidates au support

Un autre axe dans la recherche de performance de nos solveurs dédiés peut être la réduction de la taille des problèmes que nous souhaitons résoudre en chaque nœud. En considérant notre stratégie de branchement qui consiste en la mise à jour des ensembles  $S_1$ ,  $S_0$  et  $\bar{S}$  (Cf. paragraphe 3.1.1 en page 22), nous pouvons utiliser ces ensembles pour exclure des variables qui ont été placées dans l'ensemble  $S_0$  par opérations de branchement sur des fils droits (Cf. figure 8) ; il nous est donc possible de réduire les tailles des matrices

calculées en sélectionnant les indices utiles en un nœud et telle que  $\mathbf{H}_{S_1}^\top \mathbf{H}_{S_1}$  et  $\mathbf{y}^\top \mathbf{H}_{S_1}$  dans le cas des nœuds feuilles et  $\mathbf{H}_{S_1 \cup \bar{S}}^\top \mathbf{H}_{S_1 \cup \bar{S}}$  et  $\mathbf{y}^\top \mathbf{H}_{S_1 \cup \bar{S}}$  dans le cas des nœuds d'explorations. Notons de plus que cette stratégie s'avère des plus efficaces dès lors que l'algorithme descend profondément dans une branche de l'arbre et pour laquelle nous réalisons systématiquement des branchements de type  $b_q \leftarrow 0$  *i.e.* nous excluons l'atome d'indice  $q$  des atomes candidats au support. Ce levier nous permet donc d'évaluer plus rapidement des nœuds qui sont "en bas" de notre arborescence.

## 4.2. Approches basées sur les solveurs en programmation quadratique

### 4.2.1. Expression de $\hat{\mathcal{P}}_{2/1}$ sous forme d'un programme quadratique

Notre première approche pour évaluer les bornes inférieures obtenues aux nœuds de notre arbre de branch-and-bound fut d'utiliser différents solveurs de programmation quadratique (QP). La structure générale de ces problèmes est donnée par la formulation suivante :

$$(\mathcal{P}_Q) \quad \min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^\top \mathbf{H}^\top \mathbf{H} \mathbf{x} - \mathbf{y}^\top \mathbf{H} \mathbf{x} \quad \text{s.c.} \quad \mathbf{A}_{\text{ineq}} \mathbf{x} \leq \mathbf{b}_{\text{ineq}}, \quad \mathbf{A}_{\text{eq}} \mathbf{x} = \mathbf{b}_{\text{eq}}, \quad \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub} \quad (26)$$

où  $\mathbf{A}_{\text{ineq}}$  (resp.  $\mathbf{A}_{\text{eq}}$ ) est une matrice correspondant au membre gauche des contraintes d'inégalité (resp. égalités) et  $\mathbf{b}_{\text{ineq}}$  (resp.  $\mathbf{b}_{\text{eq}}$ ) est le vecteur du membre droit des contraintes d'inégalité (resp. d'égalité). Les vecteurs  $\mathbf{lb}$  et  $\mathbf{ub}$  sont les vecteurs de bornes sur la variable  $\mathbf{x}$ . Considérons par exemple l'expression du problème  $\hat{\mathcal{P}}_{2/1}$  au nœud racine de notre arborescence :

$$(\hat{\mathcal{P}}_{2/1,0}^r) \quad \min_{\mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \quad \text{s.c.} \quad \sum_{i \in \bar{S}} x_i \leq 1, \quad 0 \leq x_i \leq 1 \quad \forall i \in \bar{S} \quad (27)$$

En adaptant la formulation (26), il nous est, par exemple, alors possible de réécrire, le problème au nœud racine (27) sous forme matricielle avec  $\mathbf{H} \in \mathbb{R}^{\bar{n} \times Q}$ ,  $\mathbf{y} \in \mathbb{R}^{1 \times \bar{n}}$  et  $\mathbf{x} \in [0,1]^Q$  tel que :

$$\mathbf{A}_{\text{ineq}} = (\mathbf{1}_{1 \times Q}) \quad \mathbf{b}_{\text{ineq}} = (1) \quad \mathbf{lb} = (\mathbf{0}_{1 \times Q})^\top \quad \mathbf{ub} = (\mathbf{1}_{1 \times Q})^\top$$

L'expression de la fonction objectif de notre problème découle du développement de l'erreur en norme  $\ell_2$  au coefficient multiplicateur près :

$$\|\mathbf{y} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 = \|\mathbf{y}\|_2^2 + \|\mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 - 2\mathbf{y} \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}} = \mathbf{y}^\top \mathbf{y} + \mathbf{x}_{\bar{S}}^\top \mathbf{H}_{\bar{S}}^\top \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}} - 2\mathbf{y} \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}} = 2 \left( \frac{1}{2} \mathbf{x}_{\bar{S}}^\top \mathbf{H}_{\bar{S}}^\top \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}} - \mathbf{y} \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}} \right) + \mathbf{y}^\top \mathbf{y} \quad (28)$$

Il convient de supprimer le terme  $\mathbf{y}^\top \mathbf{y}$  de la fonction objectif, ce dernier étant constant comme produit de deux vecteurs et de ne conserver que la partie quadratique de l'expression de l'erreur en norme  $\ell_2$ .

Comme nous l'avons évoqué précédemment, le problème  $\hat{\mathcal{P}}_{2/1}$  sont connus dans la littérature sous le nom FCLS (Cf. paragraphe 2.4 en page 20) ; ces derniers possédant des solveurs dédiés *e.g.* Cplex lsqlin, Matlab lsqlin... Cependant, lors de expérimentations numériques, il s'est avéré que ces derniers étaient moins efficaces pour résoudre nos problèmes<sup>7</sup>. Nous avons donc décidé d'écarter les solveurs dédiés aux problèmes de types moindres carrés et de nous concentrer sur les résultats obtenus à l'aide de solveurs de programmation quadratique.

7. Nous cherchons encore l'explication. Un argument pour expliquer ce résultat surprenant serait que cette classe de solveur proposerait de moindres performances pour les systèmes sous-déterminés *i.e.* des systèmes où il y a plus de variables que de contraintes.

#### 4.2.2. Reformulation des problèmes dans les cas dégénérés.

Dans le cas de systèmes sous-déterminés ( $\mathbf{H} \in \mathbb{R}^{N \times Q}$  et  $N \leq Q$ ), nous avons rencontrés des erreurs avec le solveur Cplex précisant que le problème n'était pas strictement convexe *i.e.* la matrice associée à la forme quadratique  $\mathbf{x}^\top \mathbf{H}^\top \mathbf{H} \mathbf{x}$  serait semie définie positive mais non forcément définie positive; le solveur en question considère que le problème est dégénéré et retourne l'erreur de non stricte convexité de la forme quadratique. Cependant, pour le problème  $\mathcal{P}_{2/0}$ , la contrainte de parcimonie nous assure qu'il est peu probable d'obtenir des modèles parcimonieux non inversibles *i.e.* la formulation passée au solveur minimise la fonction objectif  $\|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2$  avec un vecteur  $\mathbf{x}$  de taille supérieure à celle de  $\mathbf{y}$  ce qui est le cas puisque, par hypothèse, nous avons  $\text{card}(\mathbf{x}) = Q$  et  $\text{card}(\mathbf{y}) = N$  avec  $N \leq Q$ .

Dans ce cas dégénéré, il existe une infinité de solution de type  $\mathbf{y} = \mathbf{H}\mathbf{x}$  avec la forme quadratique  $\mathbf{x}^\top \mathbf{H}^\top \mathbf{H} \mathbf{x}$  non nécessairement définie positive (les valeurs de propres de cette forme quadratique sont nulles ou numériquement faibles); nous pouvons alors seulement affirmer que  $\mathbf{x}^\top \mathbf{H}^\top \mathbf{H} \mathbf{x} \geq 0$  mais ne pouvons prouver que la solution trouvée  $\mathbf{x}^*$  vérifie l'implication  $\mathbf{x}^{*\top} \mathbf{H}^\top \mathbf{H} \mathbf{x}^* = 0 \Rightarrow \mathbf{x}^* = 0$ .

Dans le cadre des problèmes parcimonieux, nous supposons que le vecteur  $\mathbf{x}$  recherché possède au plus  $K$  composantes non nulles avec  $K \ll N$ . Les sous-problèmes que nous résolvons dans les différents nœuds de l'arbre s'avèrent être de la forme  $\|\mathbf{H}_K \mathbf{x}_K - \mathbf{y}\|_2^2$  où  $\mathbf{H}_K$  (resp.  $\mathbf{x}_K$ ) sont des sous-matrices (resp. vecteurs) réduites à un sous-ensemble de  $K$  atomes valides; la matrice  $\mathbf{H}_K^\top \mathbf{H}_K$  est donc inversible car de rang plein  $\text{rg}(\mathbf{H}_K) = K$  et la forme quadratique associée  $\mathbf{x}_K^\top \mathbf{H}_K^\top \mathbf{H}_K \mathbf{x}_K$  est strictement convexe *i.e.*  $K$  étant *petit*, alors nous pouvons prouver que  $\mathbf{x}_K^\top \mathbf{H}_K^\top \mathbf{H}_K \mathbf{x}_K = 0 \Rightarrow \mathbf{x}_K = 0$  avec  $\text{rg}(\mathbf{H}_K) = K$ .

Pour parer à cette erreur du solveur, une reformulation du problème peut être mise en place en substituant dans la fonction objectif la matrice  $\mathbf{H}\mathbf{x}$  par un vecteur  $\mathbf{z}$  et en définissant  $N$  contraintes d'égalités. Le problème résolu dans les cas où  $N \leq Q$  devient alors :

$$\min_{\mathbf{x} \in [0,1]^p} \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 \text{ s.c. } \sum_{i \in \mathcal{S}} x_i \leq 1, 0 \leq x_i \leq 1 \quad \forall i \in \mathcal{S}, \quad \mathbf{z} = \mathbf{H}_{\mathcal{S}} \mathbf{x}_{\mathcal{S}} \quad (29)$$

La formulation matricielle du problème (29) est alors donnée par :

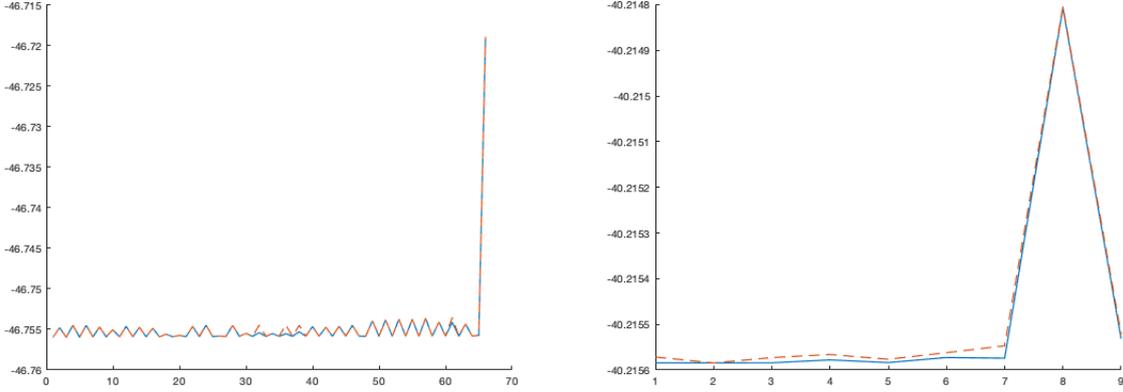
$$\begin{aligned} \mathbf{A}_{\text{ineq}} &= \begin{pmatrix} \mathbf{1}_{1 \times Q} & \mathbf{0}_{1 \times \bar{n}} \end{pmatrix} & \mathbf{b}_{\text{ineq}} &= (1) & \mathbf{l}\mathbf{b} &= \begin{pmatrix} \mathbf{0}_{1 \times Q} & -\infty_{1 \times \bar{n}} \end{pmatrix}^\top & \mathbf{u}\mathbf{b} &= \begin{pmatrix} \mathbf{1}_{1 \times Q} & \infty_{1 \times \bar{n}} \end{pmatrix}^\top \\ \mathbf{A}_{\text{eq}} &= \begin{pmatrix} \mathbf{H}_{\mathcal{S}} & -\mathbf{I}_{\bar{n}} \end{pmatrix} & \mathbf{b}_{\text{eq}} &= \begin{pmatrix} \mathbf{0}_{\bar{n} \times 1} \end{pmatrix} \end{aligned}$$

#### 4.2.3. Discrimination d'une famille de solveurs

Ayant auparavant retiré de l'analyse les solveurs de type moindres carrés pour favoriser les solveurs de programmation quadratique, nous avons cherché à savoir s'il n'était pas encore une fois possible de n'utiliser qu'une seule classe de méthode pour résoudre l'ensemble de nos problèmes. L'analyse des bornes inférieures obtenue à l'aide des solveurs QP proposés par Cplex et Matlab nous a permis de conclure. Nous présentons sur la figure 12 les valeurs de bornes inférieures obtenues sur des instances de notre problème en comparant les performances des solveurs Cplex et Matlab.

L'analyse résultats obtenus à l'aide des solveurs de programmation quadratique montre que le solveur proposé par Matlab fournit des bornes inférieures de moins bonne qualité comparées à celles obtenues avec le solveur Cplex<sup>8</sup>. A partir de ces observations, nous avons décidé de ne conserver qu'un seul solveur QP pour nos expérimentations numériques à savoir le solveur de programmation quadratique Cplex .

8. Sur la figure (12a), regarder les itérations 32, 36, 39 et 61 représentées sur l'axe des abscisses. Sur la figure (12b), regarder les itérations 1, 3, 4, 5, 6, 7, 8 et 9 représentées sur l'axe des abscisses



(a) Instance  $Q = 300$ ,  $K = 3$ , et fortement bruitée      (b) Instance  $Q = 300$ ,  $K = 7$  et très faiblement bruitée

FIGURE 12: Représentation comparée des bornes inférieures obtenues avec les solveurs QP de Cplex(-) et Matlab(- -)

#### 4.2.4. Choix de la meilleure stratégie pour le solveur sélectionné

Pour l'obtention de bornes inférieures dans notre arbre de branch-and-bound avec solveurs QP, nous avons montré qu'il était possible de ne conserver que la méthode Cplex, notée BB\_R\_Cplx\_QP. De plus, comme nous l'avons évoqué au début de ce chapitre, nous avons mis en place des stratégies visant à rendre notre méthode dédiée plus efficace en termes d'évaluation des nœuds; nous noterons par le suffixe `_RC` la restriction des variables candidates au support et par `_PC` le pré-calcul des matrices et vecteurs  $\mathbf{H}^T\mathbf{H}$  et  $\mathbf{y}^T\mathbf{H}$ . Nous allons donc tenter de déterminer la *meilleure* combinaison de stratégies combinée en étudiant les performances obtenues pour des niveaux de bruit variants *i.e.* des vecteurs  $\boldsymbol{\varepsilon} \in \mathbb{R}^N$  plus ou moins importants dans le modèle linéaire 1 présenté en page 5.

Dans le cas des instances où  $N \geq Q$  présentées sur la figure 13, nous pouvons observer que la combinaison BB\_R\_Cplx\_QP\_RC\_PC est préférée dans la majorité des cas et ceci pour tous niveaux de bruits. Pour les instances de type  $N \leq Q$  se trouvant sur la figure 14, nous pouvons observer que pour les problèmes sans bruit ou à un faible niveau, c'est la combinaison BB\_R\_Cplx\_QP\_PC qui semble être préférée. Cependant, la reformulation incluant les  $N$  contraintes d'égalité réduit l'impact de la stratégie de pré-calcul à l'estimation du vecteur  $\mathbf{y}^T\mathbf{H}$ . Une hypothèse formulée pour tenter d'expliquer ces observations serait que la méthode de recherche utilisée par la stratégie `_RC` n'est pas suffisamment efficace en termes de gain temporel pour favoriser la stratégie de restriction des variables candidates au support. Notons que ce n'est plus le cas lorsque les instances considérées ont un fort niveau de bruit et pour lesquelles la combinaison BB\_R\_Cplx\_QP\_RC\_PC est préférée.

#### 4.2.5. Conclusion de l'approche basée sur les solveurs QP

Dans cette section, nous avons présenté les méthodes basées sur les solveurs QP mais également montré que parmi l'ensemble des combinaisons possibles de stratégies développées, nous pouvions nous limiter à l'utilisation du solveur Cplex QP avec restriction des variables candidates au support et pré-calculs des termes  $\mathbf{H}^T\mathbf{H}$  et  $\mathbf{y}^T\mathbf{H}$ .

Notons que cette méthode fait appel à des fonctions externes que nous ne pouvons maîtriser; notre objectif étant de proposer des algorithmes dédiés à l'obtention de bornes inférieures, nous étudierons par la suite d'autres algorithmes permettant de résoudre les problèmes relâchés aux différents nœuds de l'arbre.

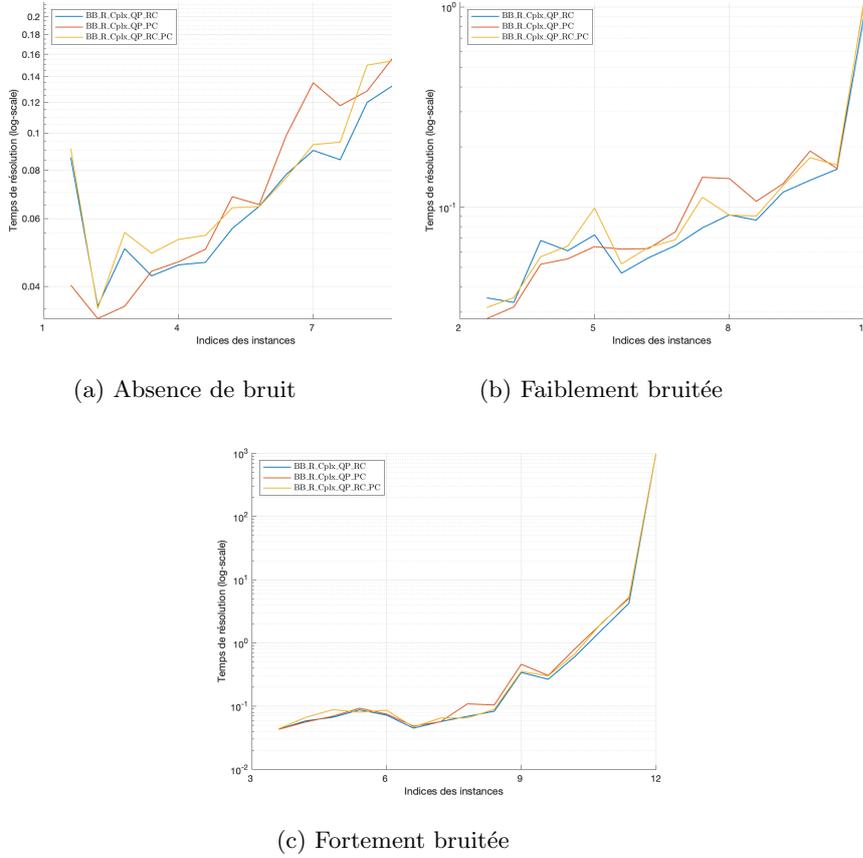


FIGURE 13: Instances  $N \geq Q$  - Représentation du temps de résolution fonction du bruit -  $BB\_R\_Cplx\_QP\_RC$ ,  $BB\_R\_Cplx\_QP\_PC$ ,  $BB\_R\_Cplx\_QP\_RC\_PC$

#### 4.3. Méthode de continuation homotopique pour l'obtention de bornes inférieures

Une méthode alternative dédiée appelée méthode de continuation Homotopique a été proposée dans R.Ben Mhenni[3] pour obtenir des bornes inférieures pour le problème d'approximation parcimonieuse sans contraintes non négativité et de somme à 1. Dans cette partie, nous allons étudier cette méthode appliquée à notre problème relâché  $\hat{\mathcal{P}}_{2/1}$ .

##### 4.3.1. Principe général de la méthode

La méthode homotopique pour l'optimisation en norme  $\ell_1$  proposée par Tibshirani, Osborne, Efron et Donoho[16, 17, 18, 19] repose sur l'intégration de la contrainte de somme à 1 dans la fonction objectif, pénalisée par le paramètre  $\lambda \in \mathbb{R}^+$ .

Nous initialisons le terme  $\lambda^{(0)} = \|\mathbf{y}^T \mathbf{H}\|_\infty$ ; le principe de cette méthode est de construire un *chemin de solution* i.e. une séquence d'intervalles  $[\lambda^{(t)}, \lambda^{(t-1)}]$  avec  $\lambda^{(t)} > \lambda^{(t-1)}, \forall t \in \llbracket 1, t_{max} \rrbracket$  où  $\lambda^{(t)}$  représente la prochaine valeur de pénalité pour laquelle une modification du support de la solution  $\mathbf{x}$  interviendra. Une modification du support peut être de deux types :

- Un atome d'indice  $i \in \bar{S}$  entre dans le support de la solution  $\mathbf{x}$  ( $i \in \bar{S} \rightarrow i \in S_1$ )
- Un atome d'indice  $i \in S_1$  est retiré du support de la solution  $\mathbf{x}$  ( $i \in S_1 \rightarrow i \in \bar{S}$ )

La méthode exploite le fait que le chemin de solution est linéaire par morceaux en fonction de  $\lambda$  et, partant de la valeur  $\lambda^{(0)}$ , calcule itérativement toutes les solutions en diminuant de façon continue le paramètre  $\lambda$

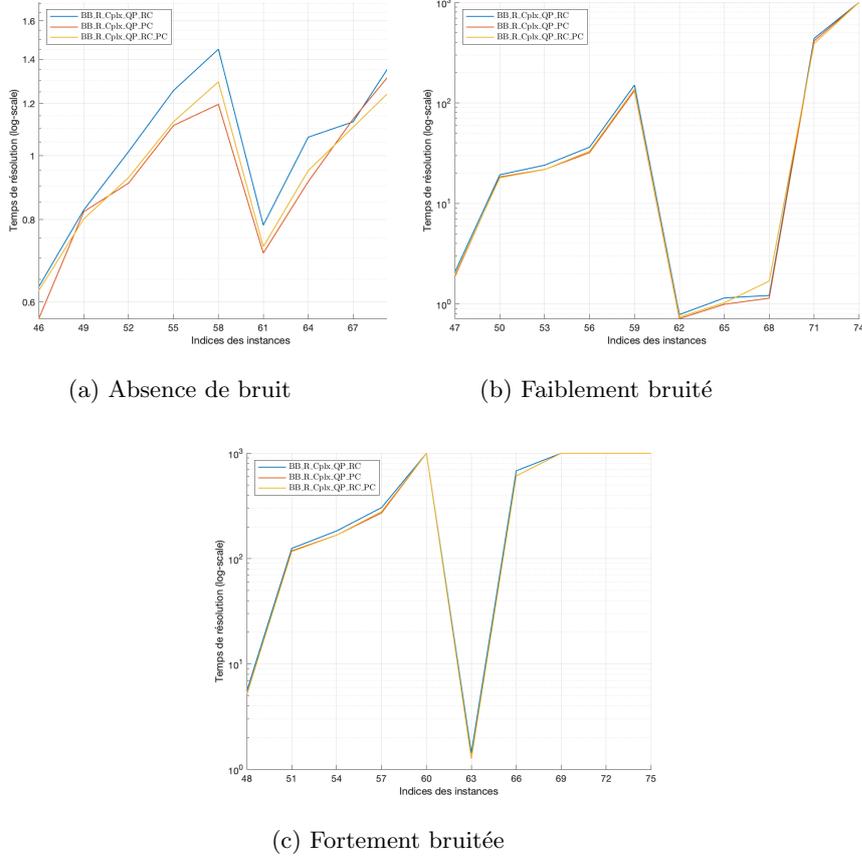


FIGURE 14: Instances  $N \leq Q$  - Représentation du temps de résolution fonction du bruit- `BB_R_Cplx_QP_RC`, `BB_R_Cplx_QP_PC`, `BB_R_Cplx_QP_RC_PC`

jusqu'à atteindre la valeur cible  $\lambda_c$  ; pour notre problème  $\hat{\mathcal{P}}_{2/1}$ , la valeur de  $\lambda_c$  correspond à la saturation de la contrainte de somme à 1. La méthode de continuation homotopique et converge en un nombre fini d'itération.

Cette approche a initialement proposée pour des problèmes de type moindres carrés et pénalisés par un terme en norme  $\ell_1$ . Cette méthode a été développé dans ce stage car elle permet d'intégrer les contraintes de non négativité et de somme à 1. Le problème  $\hat{\mathcal{P}}_{2/1}$  reformulé pour l'application de la méthode homotopique est donné ci-dessous :

$$\begin{aligned}
 \min_{\substack{\mathbf{x}_{\bar{S}} \in [0,1]^{\bar{n}} \\ \mathbf{x}_{S_1} \in [0,1]^{|S_1|}}} & F(\mathbf{x}, \lambda) = J(\mathbf{x}) + \lambda p(\mathbf{x}) \\
 \text{avec} & J(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{S_1} \mathbf{x}_{S_1} - \mathbf{H}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \\
 & p(\mathbf{x}) = \|\mathbf{x}\|_1 - 1
 \end{aligned} \tag{30}$$

L'homotopie positive exploite la condition d'optimalité du problème pénalisé 30 pour mettre à jour le support sur un intervalle  $[\lambda^{(t)}, \lambda^{(t-1)}]$  :

$$\mathbf{0} \in \nabla F(\mathbf{x}, \lambda) \iff \mathbf{0} \in \nabla J(\mathbf{x}^*) + \lambda \partial p(\mathbf{x}^*) \tag{31}$$

$$\iff \begin{cases} \mathbf{H}^\top(\mathbf{y} - \mathbf{H}\mathbf{x}) = \lambda \text{sign}(\mathbf{x}^*) & \forall x_q \neq 0 \iff q \in S_1 \\ |\mathbf{H}^\top(\mathbf{y} - \mathbf{H}\mathbf{x})| \leq \lambda & \forall x_q = 0 \iff q \in \bar{S} \end{cases} \tag{32}$$

Le principe générale de la méthode repose sur l'hypothèse que  $\forall \lambda \in [\lambda^{(t)}, \lambda^{(t-1)}] \forall t \in \llbracket 1, t_{max} \rrbracket$ , les ensembles  $S_1$  et  $\bar{S}$  restent constant *i.e.* la solution est identique sur ce même intervalle.

Lorsque nous faisons varier le paramètre de régularisation  $\lambda$ , dès lors que l'inégalité de (32) reste stricte, alors nous restons sur le même intervalle  $[\lambda^{(t)}, \lambda^{(t-1)}] \forall t \in \llbracket 1, t_{max} \rrbracket$ . Si pour une valeur de  $\lambda$ , la seconde ligne de (32) correspondant à l'inégalité devient stricte, alors la contrainte devient saturée et le support est mis à jour par ajout ou retrait d'un atome dans le support.

Le processus de modification du support courant est réalisé jusqu'à ce qu'aucune variable ne puisse être ajoutée ou retirée du support ou alors, lorsque la contrainte de parcimonie relâchée soit saturée *i.e.*  $\|\mathbf{x}\|_1 \geq \lambda_c$  avec  $\lambda_c = 1$ .

La figure 15 montre un chemin de solution typique pour un exemple avec 5 variables :  $\bar{S} = \{1, 2, 3\}$  et  $S_1 = \{4, 5\}$ .

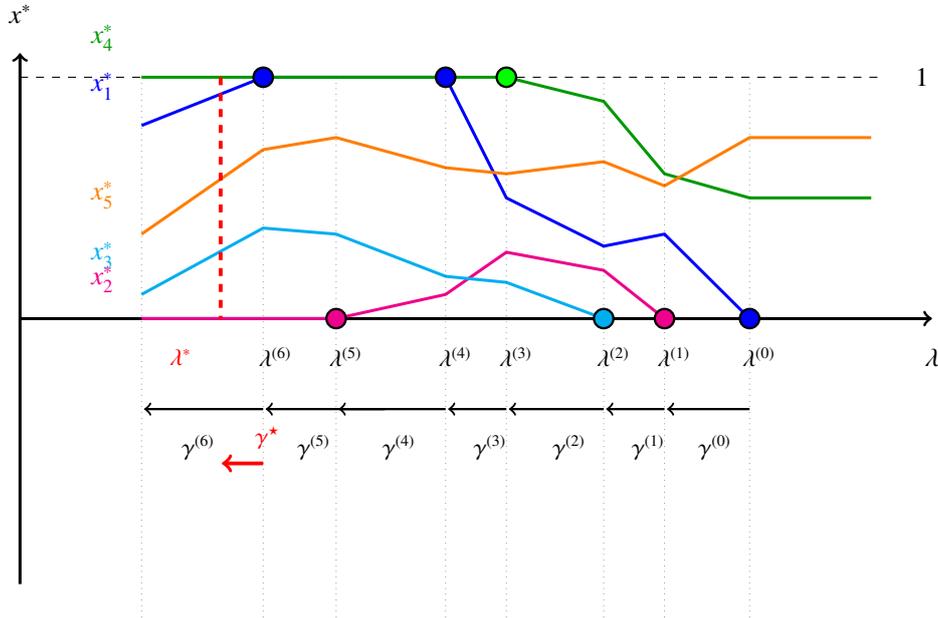


FIGURE 15: Exemple de chemin de solution donné par la solution  $\mathbf{x}^*$  du problème (30) en fonction de  $\lambda$ , avec 5 variables :  $\bar{S} = \{1, 2, 3\}$  et  $S_1 = \{4, 5\}$ . Chaque plan correspond à une nouvelle partie du chemin  $\lambda \in [\lambda^{(k)}, \lambda^{(k-1)}]$ . Les cercles représentent les événements provoquant un changement dans la configuration du support. Les lignes pointillées verticales représentent les points de rupture. La ligne rouge en pointillé représente l'itération à laquelle  $\lambda_c$  est atteint. Source : R.Ben Mhenni[3]

#### 4.3.2. Conclusion sur la méthode homotopique

Du fait de sa propriété de convergence en un nombre fini d'itération, cette méthode exacte nous assure de bonnes performances de résolution dans les cas où la relaxation continue est parcimonieuse ; le nombre d'itérations est alors de l'ordre du nombre de composantes non nulles. Son utilisation s'avère très intéressante pour résoudre les problèmes de type explorations de par le fait que durant la construction d'un chemin de solution, le nombre de variable entrant dans le support (variable non nulle) est toujours supérieur au nombre de variables sortant du support ; cette méthode nous permet alors de mieux saturer la contrainte de somme à 1.

Cependant, nous avons pu observer que, dans la plupart des cas, les résolutions des problèmes dans les nœuds de type feuilles ne saturaient pas la contrainte de somme à 1. La méthode de continuation homotopique n'est donc pas conseillée pour la résolution des problèmes dans les feuilles de notre arbre de recherche.

Pour exploiter au mieux les capacités de l'algorithme homotopique, nous avons décidé d'implémenter une stratégie de résolution *hybride* basée sur les performances des solveurs QP pour résoudre les nœuds de type feuilles afin d'obtenir des solutions parcimonieuses, l'homotopie est quant à elle utilisée pour résoudre les problèmes de type exploration en considérant que cet algorithme permet d'obtenir rapidement des solutions saturant la contrainte de somme à 1 et de bonnes qualités pour réaliser les opérations de branchements. Ajoutons également que les nœuds de type exploration sont ceux qui sont le plus visités lors du parcours de notre arbre de recherche ; cette observation légitime l'utilisation de la méthode de continuation homotopique.

#### 4.4. Méthode de résolution dédiée aux problèmes Fully Constrained Least Squares (FCLS)

Comme nous l'avons évoqué dans le paragraphe 2.4, les problèmes que nous résolvons dans les nœuds de notre arbre de branch-and-bound sont appelés Fully Constrained Least Squares et correspondent à des problèmes de type moindres carrés sous contraintes de non négativité et de somme à 1. La troisième stratégie pour l'obtention de bornes inférieures que nous avons mise en place pour la résolution des sous problèmes a été proposée par Heinz[4] ; l'auteur propose une méthode de résolution dédiée aux problèmes FCLS et appliquée au démixage spectral.

Le temps manquant, nous avons utilisé cet algorithme comme une *boîte noire* et l'avons appliqué à tout type de nœuds de notre arbre de branch-and-bound. Nous remercions S.Moussaoui pour nous avoir transmis le code de cet algorithme développé dans le cadre de ses travaux sur le démixage spectral.

Bien que cette méthode se soit avérée très compétitive par rapport au solveur Cplex QP, l'algorithme dédié n'est performant que pour les systèmes sous-déterminés *i.e.* avec  $N \leq Q$ . En effet, la méthode FCLS a été développée pour ce type de système *i.e.* l'estimation du vecteur d'abondances  $\mathbf{x}$  avec un *gros dictionnaire*  $\mathbf{H}$ . Notons de plus que cette méthode n'a pas été construite dans l'objectif d'inclure la contrainte de parcimonie et ainsi produire des vecteurs  $\mathbf{x}$  tels que  $\|\mathbf{x}\|_0 \leq K$ .

#### 4.5. Conclusion sur les stratégies d'obtention de bornes inférieures

Dans ce chapitre, nous avons étudié trois stratégies mises en place durant ce stage pour l'obtention de bornes inférieures pour notre algorithme de branch-and-bound. Comparées aux stratégies de résolution utilisant des solveurs quadratiques, les méthodes dédiées pour la résolution du problème relâché  $\hat{\mathcal{P}}_{2/1}$  montrent des performances accrues en termes de temps de calcul qui nous motivent à poursuivre l'étude des stratégies dédiées pour l'obtention de bornes inférieures. Dans le chapitre suivant, nous allons présenter les résultats des expérimentations numériques dans lesquelles nous avons mis en compétition les trois méthodes présentées pour l'obtention de bornes inférieures.

## 5. Expérimentations numériques et analyse des résultats

Nous allons à présent nous intéresser aux résultats obtenus à l'aide de nos différentes stratégies de résolutions pour le problème de démixage spectral. Dans ce chapitre, nous présenterons tout d'abord le dictionnaire de spectres réels que nous avons utilisé pour réaliser nos expérimentations ; dans un second temps, nous étudierons les sources d'erreurs pouvant impacter la qualité de nos estimations obtenues grâce à notre algorithme de branch-and-bound et nous finirons par la présentation de nos résultats expérimentaux.

### 5.1. Source des données, des erreurs et génération des instances

#### 5.1.1. Dictionnaire USGS

Le dictionnaire  $\mathbf{H} \in \mathbb{R}^{N \times Q}$  que nous utilisons pour réaliser nos simulations provient de l'*United States Geological Survey* proposé par Clark & Swayze[20]. Ce dernier contient  $Q = 498$  signatures spectrales pures de minéraux obtenues en laboratoire sur un ensemble de  $N = 224$  longueurs d'onde allant de 0,3 à 2,7  $\mu\text{m}$ . La figure 16 représente quatre spectres obtenus pour la calcite, un minéral présent dans les roches carbonatées et les météorites.

Dans notre problème, nous construisons un modèle linéaire correspondant à la composition d'un pixel à partir des endmembers présents dans ce dictionnaire :

$$\mathbf{y} = \sum_{i \in S_1} \mathbf{h}_i x_i + \boldsymbol{\varepsilon} \quad (33)$$

où  $\mathbf{h}_i$  est un atome de  $\mathbf{H}$  et  $\boldsymbol{\varepsilon}$  le terme de bruit supposé gaussien. Notre modèle est généré en sélectionnant de manière aléatoire  $K$  spectres et pour lesquelles nous générons des amplitudes positives dont la somme est égale à 1.

Dans les paragraphes suivants, nous présentons les paramètres pouvant engendrer des erreurs dans la détection des composantes présentes dans le pixel étudié.

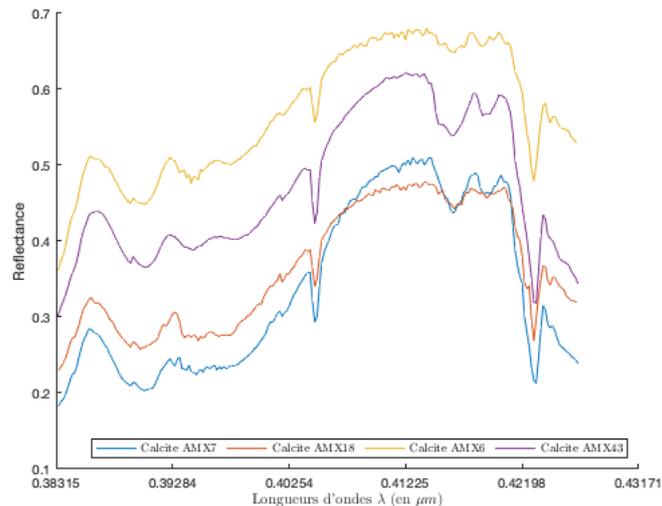


FIGURE 16: Représentation des spectres associés à la Calcite extraits de la base de données USGS - Clark & Swayze[20]

#### 5.1.2. Le rapport signal/bruit

Pour quantifier le niveau de bruit dans le modèle (33), nous utilisons le rapport signal/bruit (SNR) défini comme le rapport entre la puissance du signal (information significative) et la puissance du bruit (signal

indésirable) exprimé en utilisant l'échelle logarithmique des décibel :

$$\text{SNR} = \frac{P_{\text{signal}}}{P_{\text{bruit}}} = \frac{10 \log_{10}(P_{\text{signal}})}{10 \log_{10}(P_{\text{bruit}})} \quad (34)$$

Il est donc possible d'exprimer le SNR en décibel dans lequel nous substituons la puissance des signaux par les termes d'erreurs en norme  $\ell_2$  ; nous obtenons ainsi :

$$\text{SNR}^{\text{dB}} = 10 \log_{10}(\text{SNR}) = 10 \log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{bruit}}}\right) = 10 \log_{10}\left(\frac{\|\mathbf{H}\mathbf{x}\|_2^2}{\|\boldsymbol{\varepsilon}\|_2^2}\right) \quad (35)$$

Intéressons-nous plus particulièrement au dénominateur de (35) correspondant à la quantité d'erreur en norme  $\ell_2$  du bruit dans notre modèle :

$$\|\boldsymbol{\varepsilon}\|_2^2 = \sum_{i=1}^N |\varepsilon_i|^2 \quad (36)$$

Notons de plus que nous avons émis l'hypothèse que les échantillons  $\varepsilon_i$  sont des réalisations aléatoires indépendantes et distribuées selon la même loi gaussienne centrée  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2) \forall i \in \llbracket 1, N \rrbracket$ . Nous obtenons alors

$$\|\boldsymbol{\varepsilon}\|_2^2 = \sum_{i=1}^N |\varepsilon_i|^2 \simeq N\sigma^2 \quad (37)$$

En reprenant l'équation (35), nous pouvons à présent remplacer la valeur du dénominateur par une fonction de la variance de l'échantillon puis isoler cette même variance dans l'équation (35) :

$$\text{SNR} = 10 \log_{10}\left(\frac{\|\mathbf{H}\mathbf{x}\|_2^2}{N\sigma^2}\right) \iff \frac{N\sigma^2}{\|\mathbf{H}\mathbf{x}\|_2^2} = 10^{-\text{SNR}/10} \iff \sigma = \frac{\|\mathbf{H}\mathbf{x}\|_2}{N} \times 10^{-\text{SNR}/20} \quad (38)$$

Dès lors, nous pouvons réécrire notre modèle présenté en (33) en intégrant le ratio signal/bruit reformulé en (38) tel que :

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \sigma \mathbf{r} \quad (39)$$

où  $\mathbf{r} \in \mathbb{R}^N$  est un vecteur généré aléatoirement distribué suivant loi normale centrée réduite.

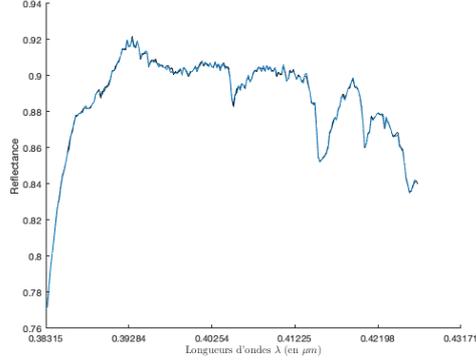
Pour illustrer l'impact du bruit sur les données observées, nous considérons quatre valeurs saillantes de  $\text{SNR} \in \{60, 50, 40, 30\}$  classé par ordre croissant d'impact sur le modèle ; ainsi, nous pouvons affirmer qu'un modèle possédant un coefficient de  $\text{SNR} = 30$  dB sera fortement impacté par le bruit comparé à un modèle avec un  $\text{SNR}$  à  $\infty$ . En effet, en reprenant l'expression de  $\sigma$  dans (38), nous avons :

- si le  $\text{SNR} = 0$ , alors cela signifie qu'il y a *autant de bruit que de signal* dans le modèle.
- si le  $\text{SNR} = \infty$ , alors cela signifie que le modèle est non bruité.

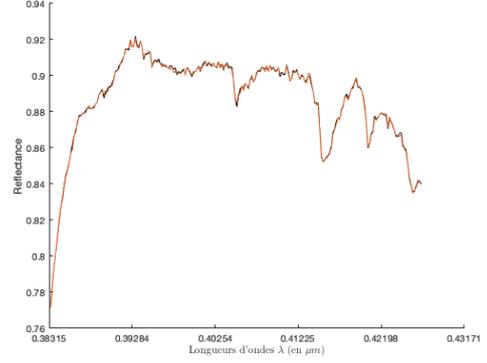
Nous pouvons alors observer l'impacte de la valeur du  $\text{SNR}$  sur un signal grâce à la figure 17 sur laquelle le signal sans bruit ( $\text{SNR} = \infty$ ) est représenté en noir.

### 5.1.3. Corrélation entre les atomes du dictionnaire $\mathbf{H}$ et erreur de support

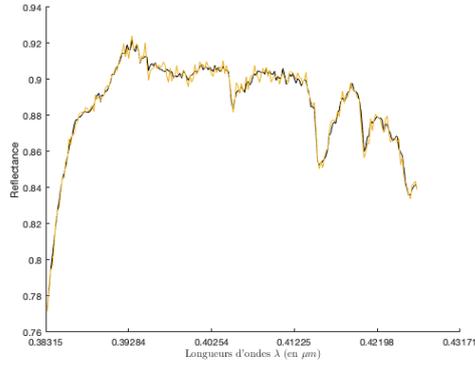
Comme nous l'avons présenté précédemment, le dictionnaire USGS contient une multitude de spectres obtenus en laboratoire. Cependant, les échantillons sur lesquels sont réalisés les études en laboratoire proviennent de différents sites ; ces multiples provenances peuvent entraîner des variations dans la compositions spectrales des atomes présents dans  $\mathbf{H}$  et créer des fortes ressemblances entre deux colonnes  $\mathbf{h}_i$  et  $\mathbf{h}_j$ ,  $i \neq j$  complexifiant le problème d'estimation. Ajoutons à cela le terme additionnel de bruit dans les observations, pouvant grandement favoriser les erreurs de détections.



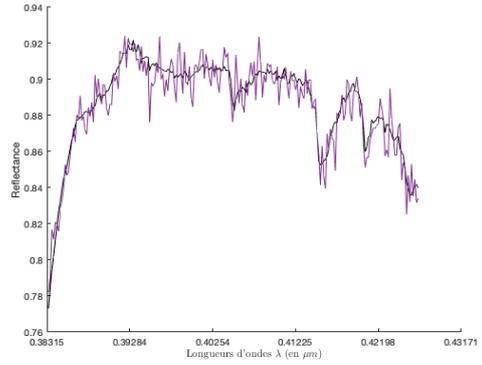
(a) SNR = 60 dB



(b) SNR = 50 dB



(c) SNR = 40 dB



(d) SNR = 30 dB

FIGURE 17: Représentation de l'atome  $\mathbf{h}_{10}$  observé avec différents niveaux de bruit extrait de la base de données USGS - Clark & Swayze[20]

Afin de mesurer la qualité de la solution obtenue et connaissant la composition exacte des échantillons que nous avons générés dans nos instances de test, nous sommes en mesure d'étudier l'erreur de prédiction entre les supports réels et ceux obtenus par application de nos différentes méthodes de résolution. Une telle mesure est appelée *erreur du support* et est calculée comme la somme des non-détections *i.e.* composantes présentes dans le mélange mais non trouvées par l'algorithme, et les fausses détections *i.e.* des atomes présents dans la solution retournée  $\mathbf{x}_{\text{exp}}$  et qui ne sont pas inclus dans le mélange réel  $\mathbf{x}_{\text{truth}}$ . Considérons l'exemple suivant pour lequel nous avons une générée une solution  $\mathbf{x}_{\text{truth}}$  de support  $S_1^t = \{1, 2, 3, 4\}$  et une solution obtenue par application de notre branch-and-bound  $\mathbf{x}_{\text{exp}}$  de support  $S_1^e = \{1, 2, 3, 5\}$ ; nous observons que

- la composante 4 de  $\mathbf{x}_{\text{truth}}$  n'est pas détectée par  $\mathbf{x}_{\text{exp}}$ , il s'agit d'une non détection.
- la composante 5 de  $\mathbf{x}_{\text{exp}}$  n'est pas présente dans le mélange réel  $\mathbf{x}_{\text{truth}}$ , il s'agit d'une fausse détection.

Nous pouvons donc en déduire que l'erreur du support dans l'exemple est égale à 2.

#### 5.1.4. Génération des instances de test

Nous présentons maintenant le protocole de génération d'instance respectant les contraintes de non négativité, de somme à 1, de parcimonie et permettant d'intégrer les différents niveaux de bruits. Les instances

que nous utilisons pour ces expériences sont générées à partir de sous matrices extraites du dictionnaire USGS susmentionné (noté  $\mathbf{H} \in \mathbb{R}^{N \times Q}$ ). Nous sélectionnons aléatoirement  $K$  indices dans l'ensemble  $\llbracket 1, Q \rrbracket$  auxquels sont allors associés des amplitudes non-nulles, tirées positives uniformément et normalisées de sorte que la somme fasse 1 ; Le processus de génération des  $K$  amplitudes est réalisé suivant une loi de Dirichlet *i.e.* le vecteur des abondances  $\{x_k\}_{k=1}^{k=K}$  vérifie les contraintes  $\sum_{i=1}^K x_i = 1$  et  $x_i \geq 0 \forall i \in \llbracket 1, K \rrbracket$ .

Pour réaliser nos instances de test, nous avons fixés les valeurs de  $Q \in \{20, 100, 160, 300, 400\}$ , de  $K \in \llbracket 3, 7 \rrbracket$  et de valeurs de SNR  $\in \{\infty, 55, 45\}$ . Le nombre d'observations  $N$  est fixé à 224 qui correspond au nombre de lignes présentes dans le dictionnaire USGS. Ces instances de problème nous permettent de tester les performances de notre algorithme de branch-and-bound sur une multitude de situations allant d'instances simples ( $K = 3$  et/ou SNR =  $\infty$ ) à des problèmes compliqués ( $K = 7$  et/ou SNR = 45 dB).

## 5.2. Résultats expérimentaux

Nous allons maintenant présenter les résultats numériques obtenus grâce à nos différentes stratégies de résolutions des problèmes aux noeuds de notre arborescence de recherche. Tout d'abord, nous présentons un outil permettant de comparer les performances d'algorithmes d'optimisation : le profil de performance.

### 5.2.1. Profils de performance

Pour réaliser l'étude des performances des différentes stratégies que nous avons mis en place, nous utilisons une technique proposée par Dolan & Moré[21] et appelée *profils de performance* ; cette dernière nous permet de définir une métrique permettant de comparer les comportements des solveurs d'obtenir une visualisation des capacités d'un algorithme à résoudre un ensemble d'instances de problèmes et ceci pour un critère donné (*e.g.* le temps de résolution ou le nombre de noeuds explorés dans l'arbre de recherche. Soient

$\mathcal{P}$  : un ensemble de problèmes pour lesquels nous recueillons les informations liées au critère choisi sous forme d'une matrice et tel que  $n_p = \text{card}(\mathcal{P})$

$\mathcal{S}$  : un ensemble de solveurs que nous utilisons pour résoudre les problèmes  $\mathcal{P}$  et tel que  $n_s = \text{card}(\mathcal{S})$

Nous définissons  $\forall p \in \mathcal{P}, \forall s \in \mathcal{S}$  le coût  $t_{p,s}$  associé au critère que l'on souhaite étudier (*e.g.* le temps de résolution mis par le solveur  $s$  pour le problème  $p$ ) et par hypothèse, nous posons  $t_{p,s} = +\infty$  si la méthode  $s \in \mathcal{S}$  ne sait pas résoudre  $p \in \mathcal{P}$ .

Le profil de performance prend en compte le nombre de problèmes en un temps donné (dans notre cas 1000s) ainsi que le *coût* de résolution qui sera évalué par rapport au meilleur algorithme proposant une solution. Cette valeur est appelée *rapport de performance* et est définie par :

$$\alpha_{p,s} = \frac{t_{p,s}}{\min_{s \in \mathcal{S}} \{t_{p,s} : p \in \mathcal{P}\}}, \quad \alpha_{p,s} \geq 1 \quad \forall p \in \mathcal{P}, \forall s \in \mathcal{S}$$

Notons que si  $\alpha_{p,s} = 1$ , alors cela signifie que  $s$  est le meilleur algorithme pour résoudre l'instance  $p$ .

La fonction de performance d'un algorithme  $s$  est alors définie comme la fonction de distribution cumulative de  $\alpha_{p,s}$  :

$$\begin{aligned} \rho_s : \mathbb{R} &\rightarrow [0, 1] \\ \tau &\mapsto \frac{1}{n_p} \text{card} \{p \in \mathcal{P} : \alpha_{p,s} \leq \tau\} \\ &\forall s \in \mathcal{S} \end{aligned}$$

Notons que les deux valeurs extrêmes de la fonction de performance  $\rho_s$  sont particulièrement intéressantes :  $\rho_s(1)$  : la proportion d'instances sur lesquelles le solveur  $s$  a été le plus performant. Nous parlons alors de *l'efficacité du solveur  $s \in \mathcal{S}$*  ;

$\rho_s(\alpha_{\max})$  : la fraction de problème résolu par l'algorithme  $s \in \mathcal{S}$  résolu en le temps maximum autorisé et avec  $\alpha_{\max} = \max_{p \in \mathcal{P}, s \in \mathcal{S}} \{\alpha_{p,s}\}$ . Nous parlons alors de la *robustesse du solveur*  $s \in \mathcal{S}$ .

### 5.2.2. Analyse des résultats

Ce stage avait pour objectif principal de développer un algorithme spécifique au problème de démixage, afin de s'affranchir de l'utilisation d'un solveur générique et de réduire le temps de calcul. Pour ce faire, nous allons présenter les résultats obtenus en résolvant le problème  $\mathcal{P}_{2/0}$  à l'aide du solveur MIP Cplex et de notre branch-and-bound dédié.

Nous présentons séparément les instances de type  $N \geq Q$  et  $N \leq Q$ . En effet, la méthode FCLS ne peut être utilisée que pour les problèmes où  $N \geq Q$  *i.e.* les systèmes sur-déterminés (Cf. paragraphe 4.4 en page 41). Le paramètre que nous ferons varier sera la coefficient de parcimonie  $K$  qui impacte directement la complexité des problèmes de démixage.

#### 5.2.2.1 Précision sur les notations

Les dénominations des méthodes de résolution que nous allons présenter sont les suivantes :

**MIP\_Cplex\_DT\_x** : correspond à la résolution du problème  $\mathcal{P}_{2/0}$  en utilisant le solveur MIP Cplex où DT correspond au saut de dualité  $\delta$  que nous faisons varier tel que  $\delta = 10^{-x}$  avec  $x \in \llbracket 4, 8 \rrbracket$  ;

**BB\_R\_Cplex\_QP** correspond à la résolution des problèmes relâchés  $\hat{\mathcal{P}}_{2/1}$  en utilisant le solveur Cplex QP ;

**BB\_R\_Hom** correspond à la résolution des problèmes relâchés  $\hat{\mathcal{P}}_{2/1}$  en utilisant la méthode Homotopique sauf sur les nœuds de types feuilles pour les raisons mentionnées dans le paragraphe 4.3.2 et pour lesquels nous utilisons le solveur Cplex QP :

**BB\_R\_FCLS** correspond à la résolution des problèmes relâchés  $\hat{\mathcal{P}}_{2/1}$  en utilisant l'algorithme FCLS.

Les trois méthodes de résolution dédiées intègrent les stratégies de restrictions des variables candidates au support et de pré-calcul des produits matriciels présentées dans le paragraphe 4.1. Les stratégies de branchement et d'explorations présentées dans le paragraphe 2.5 sont identiques pour les trois méthodes.

#### 5.2.2.2 Analyse de l'impact du saut de dualité $\delta$ sur la solution

L'objectif de cette étude est de vérifier si le choix d'une faible valeur de  $\delta$  nous permet d'obtenir une meilleure approximation de la composition du mélange.

Nous ne présentons dans la figure 18 la valeur de la fonction de coût sur les instances de problème de type sous-déterminés ( $N \leq Q$ ) et les profils de performances des méthodes MIP\_Cplex\_DT\_x pour  $x \in \llbracket 4, 8 \rrbracket$  sur la figure 19.

Le résultat est sans appel, le solveur MIP\_Cplex\_DT\_4 est certes plus rapide mais au détriment de l'approximation de la partie quadratique de la norme  $\ell_2$  ; en effet, nous pouvons constater des variations de l'erreur pour des instances possédant un faible coefficient de SNR *i.e.* très bruitées.

Ce constat s'explique simplement en considérant que plus le saut de dualité est élevé (*e.g.*  $\delta = 10^{-4}$ ), plus l'algorithme est rapide puisque nous relâchons les exigences sur les tolérances numériques. En revanche, la qualité de la solution fournie se dégrade lorsque ce saut augmente par application du même argument.

Par cette observation, nous montrons que l'utilisation des solveurs Cplex MIQP nous impose de réaliser un compromis entre la précision et le temps de résolution.

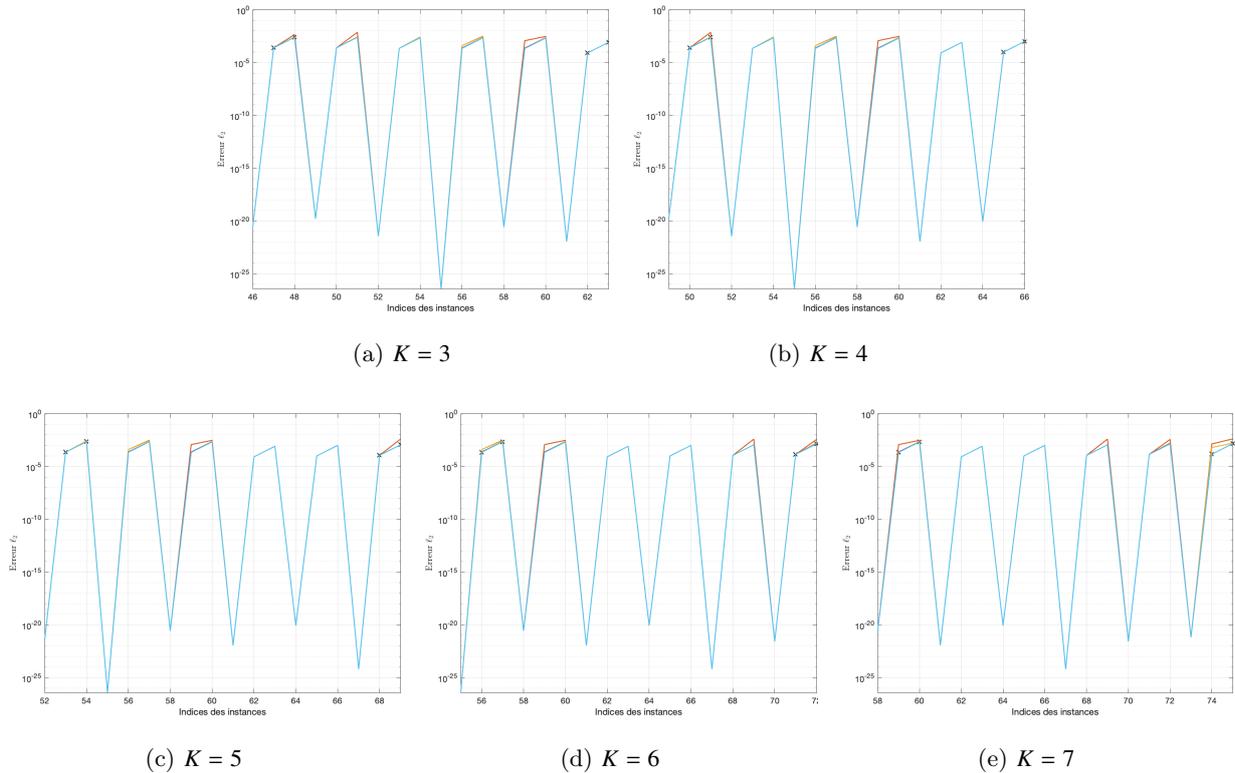


FIGURE 18: Représentation de la valeur de la norme  $\ell_2$  fonction du saut de dualité  $\delta$  - instances  $N \geq Q$  -  $K \in \llbracket 3, 7 \rrbracket$  - Méthodes : MIP\_Cplex\_DT\_x avec  $x = 4, x = 5, x = 6, x = 7$  et  $x = 8$

### 5.2.2.3 Comparaison des résultats expérimentaux obtenus par résolution avec notre branch-and-bound et par résolution du MIP

#### 5.2.2.3.1 Instances de type $N \geq Q$

L'analyse comparée des méthodes de résolutions MIP et dédiées est présentée dans la figure 20. Les profils de performance montrent que pour la majorité des instances sur-déterminées, la méthode de résolution homotopique est la plus efficace et ceci quelque soit la complexité du problème en termes de coefficient  $K$ . Nous observons également que la méthode BB\_R\_Cplex\_QP s'avère être peu efficace comparées aux autres algorithmes sur ce même ensemble de problèmes. Notons également que pour les problèmes relativement difficiles avec  $K = 6$ , la méthode FCLS est l'algorithme qui résout le plus de problème; ce constat pourra s'avérer intéressant dans le suite du projet de recherche.

#### 5.2.2.3.2 Instances de type $N \leq Q$

L'analyse comparée des méthodes de résolutions MIP et dédiées est présentée dans la figure 21. Les profils de performances montrent que la méthode BB\_R\_Hom est celle qui résout le plus de problèmes pour les valeurs de  $K \in \{3, 5\}$ . Pour les deux valeurs restantes du coefficient de parcimonie associées à des problèmes compliqués, c'est l'algorithme MIP\_Cplex\_DT\_4 qui semble le plus performant. Cependant, rappelons le constat réalisé dans le paragraphe 5.2.2.2 et pour lequel nous avons observé que le saut de dualité élevé augmentait la rapidité des algorithmes mais au détriment de la qualité de la solution. Dès

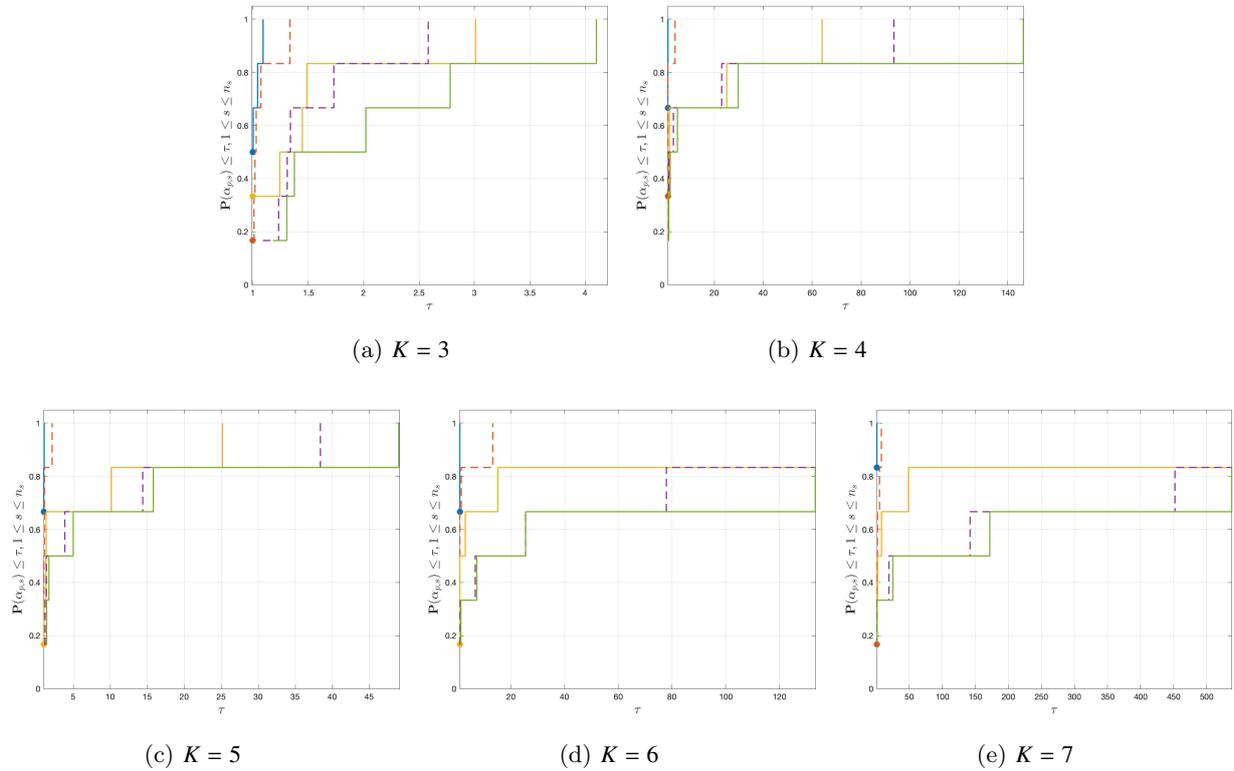


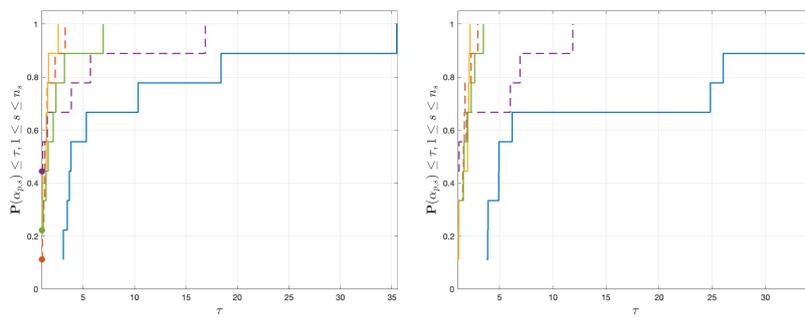
FIGURE 19: Profils de performance - critère : temps de résolution Cplex MIQP - instances  $N \leq Q$  -  $K \in \llbracket 3, 7 \rrbracket$  - Méthodes : MIP\_Cplex\_DT\_x avec  $x = 4, x = 5, x = 6, x = 7$  et  $x = 8$

lors, nous pouvons nous intéresser au second algorithme le plus performant, qui se trouve être la méthode homotopique.

#### 5.2.2.4 Conclusion de l'analyse comparée

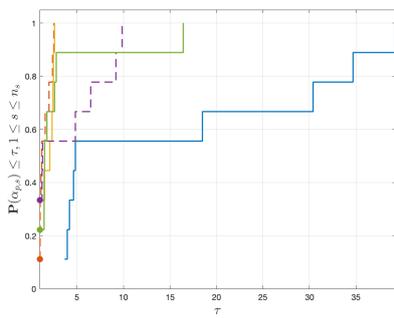
L'analyse des simulations réalisées sur les différentes instances nous permet de conclure sur l'utilité de la mise en place d'approches dédiées pour la résolution du problème de démixage parcimonieux. Ces dernières présentent des performances en termes de temps de calcul plus qu'intéressantes pour toutes les valeurs du coefficient de parcimonie  $K$  alors que les résolutions à utilisant les solveurs génériques de Cplex nous imposent de réaliser un compromis entre la qualité de solution et la vitesse de résolution, les algorithmes dédiés semblent être une approche très compétitive.

Notons que dans cette étude, nous nous sommes concentrés sur le critère du temps de calcul pour comparer les algorithmes, la suite de cette analyse pourra consister en l'étude de l'impact des algorithmes dédiés sur la qualité des solutions en termes d'erreur quadratique et de support.

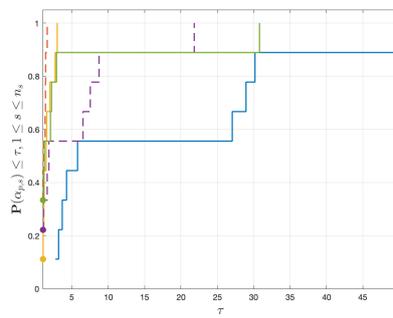


(a)  $K = 3$

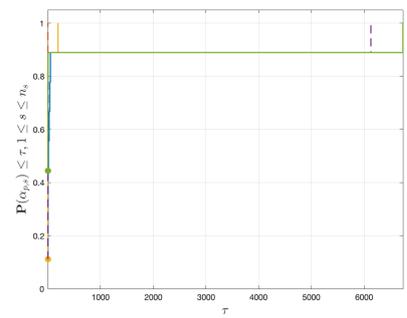
(b)  $K = 4$



(c)  $K = 5$

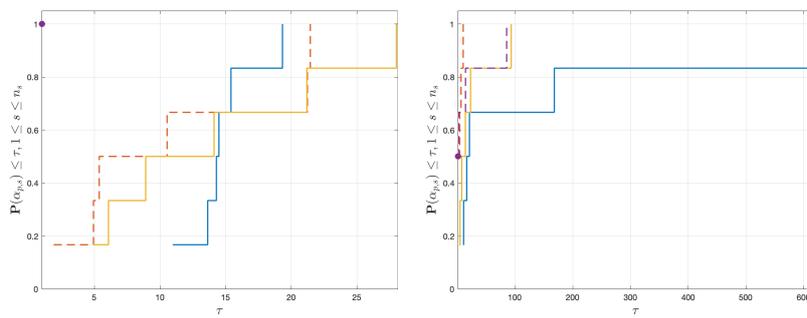


(d)  $K = 6$



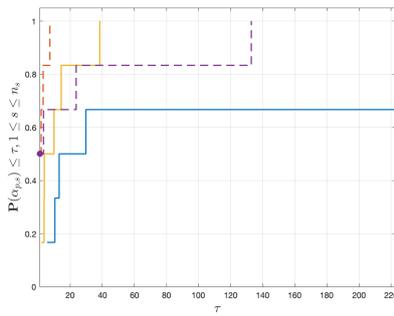
(e)  $K = 7$

FIGURE 20: Profils de performance - critère : temps de résolution branch-and-bound et MIP - instances  $N \geq Q$  -  $K \in \llbracket 3, 7 \rrbracket$  - Méthodes : [BB\\_R\\_Cplex\\_QP](#), [MIP\\_Cplex\\_DT\\_4](#), [MIP\\_Cplex\\_DT\\_7](#), [BB\\_R\\_Hom](#) et [BB\\_R\\_FCLS](#)

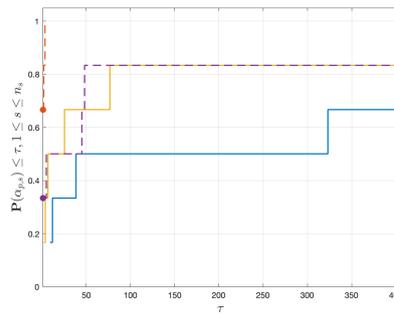


(a)  $K = 3$

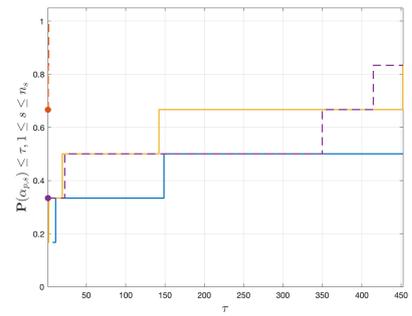
(b)  $K = 4$



(c)  $K = 5$



(d)  $K = 6$



(e)  $K = 7$

FIGURE 21: Profils de performance - critère : temps de résolution branch-and-bound et MIP - instances  $N \leq Q$  -  $K \in \llbracket 3, 7 \rrbracket$  - Méthodes : **BB\_R\_Cplex\_QP**, **MIP\_Cplex\_DT\_4**, **MIP\_Cplex\_DT\_7** et **BB\_R\_Hom**

## 6. Conclusion générale

Ces travaux de recherche ont montré l'intérêt de développer des solutions algorithmiques spécifiques pour le démantèlement parcimonieux sous contraintes de somme à 1 et de non négativité, là où les travaux antérieurs utilisaient un solveur générique.

Bien que l'optimisation exacte nécessite un temps de calcul bien plus élevé que les méthodes approchées proposées dans la littérature, nous avons montré qu'il était possible de prouver l'optimalité des solutions à l'aide d'un algorithme de branch-and-bound dédié, exploitant les spécificités mathématiques des problèmes évalués à chaque nœud de l'arbre de recherche, et finalement plus efficace en temps de calcul que le solveur MIP de CPLEX.

Ces travaux font office de point de départ à des recherches visant à améliorer l'évaluation des nœuds de notre arborescence de recherche par le développement d'algorithmes dédiés aux problèmes de type moindres carrés sous contraintes de positivité et de somme unitaire, le recours aux solveurs génériques ayant montré de moindres performances.

Dans un premier temps, il serait envisageable d'étudier la possibilité d'étendre l'algorithme FCLS proposé par Heinz[4] au cas des systèmes sous-déterminés et d'intégrer la contrainte de parcimonie dans cette méthode d'estimation des abondances. En effet, l'analyse des simulations a montré que la résolution des sous-problèmes s'est montrée bien plus efficace dans les cas où le coefficient de parcimonie est grand ( $K \in \{6, 7\}$  sur la figure 20). Cependant, cet algorithme ne peut être appliqué que pour des systèmes surdéterminés *i.e.* où le nombre de variables est inférieur au nombre de données ce qui peut s'avérer critique puisqu'en pratique, nous aurons tendance à utiliser des dictionnaires de plus en plus fournis pour améliorer l'estimation des paramètres de notre modèle.

Le développement d'approches hybrides ou de stratégies permettant la simplification de la résolution des sous-problèmes pour l'obtention de bornes inférieures est également à considérer.

La structure de l'algorithme de branch-and-bound nous permet d'envisager une multitude de règles de branchement et d'élagage des nœuds de l'arbre *i.e.* établir un seuil d'abondance significative, ... Enfin, nous pourrions envisager l'introduction de contraintes de parcimonie plus spécifiques *e.g.* la parcimonie sur des sous-groupes (Cf. R. Ben Mhenni[3]).

Pour conclure ce mémoire, je souhaiterais encore une fois remercier Sébastien et Ramzi pour m'avoir permis de vivre cette expérience enrichissante et de m'avoir initié au domaine passionnant du traitement du signal. Je tiens également à remercier l'ensemble des membres de l'équipe pédagogique du Master ORO de l'Université de Nantes pour leur accompagnement durant ces deux années et la possibilité qu'ils m'ont offerte d'étudier la Recherche Opérationnelle et ses applications au sein de l'Université Libre de Bruxelles.

## 7. Bibliographie

- [1] M.-D. Iordache, J. M. Bioucas-Dias, A. Plaza, Sparse Unmixing of Hyperspectral Data, *IEEE Transactions on Geoscience and Remote Sensing* 49 (6) (2011) 2014–2039. doi :10.1109/TGRS.2010.2098413.
- [2] R. B. Singer, T. B. McCord, Mars-large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance, in : *Lunar and Planetary Science Conference Proceedings*, Vol. 10, 1979, pp. 1835–1848.
- [3] R. B. Mhenni, Méthodes de programmation en nombres mixtes pour l’optimisation parcimonieuse en traitement du signal, Ph.D. thesis, ECOLE CENTRALE DE NANTES (05 2020).
- [4] D. C. Heinz, et al., Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery, *IEEE transactions on geoscience and remote sensing* 39 (3) (2001) 529–545.
- [5] B. K. Natarajan, Sparse approximate solutions to linear systems, *SIAM Journal on Computing* 24 (2) (1995) 227–234.
- [6] T. T. Nguyen, C. Soussen, J. Idier, E.-H. Djermoune, Np-hardness of  $l_0$  minimization problems : revision and extension to the non-negative setting, 2019. doi :10.1109/SampTA45681.2019.9030937.
- [7] J. A. Tropp, S. J. Wright, Computational Methods for Sparse Solution of Linear Inverse Problems, *Proceedings of the IEEE* 98 (6). doi :10.1109/JPROC.2010.2044010.
- [8] S. Bourguignon, J. Ninin, H. Carfantan, M. Mongeau, Exact sparse approximation problems via mixed-integer programming : Formulations and computational performance, *IEEE Transactions on Signal Processing* 64 (6) (2016) 1405–1419.
- [9] R. B. Mhenni, S. Bourguignon, M. Mongeau, J. Ninin, H. Carfantan, Algorithme branch-and-bound pour l’optimisation exacte en norme  $l_0$ , in : *XXVIIème Colloque francophone de traitement du signal et des images (GRETSI 2019)*, Lille, France, 2019.  
URL <https://hal.archives-ouvertes.fr/hal-02280754>
- [10] R. B. Mhenni, S. Bourguignon, J. Ninin, F. Schmidt, Spectral unmixing with sparsity and structuring constraints, in : *2018 9th Workshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing (WHISPERS)*, 2018, pp. 1–5.
- [11] A. H. Land, A. G. Doig, An Automatic Method of Solving Discrete Programming Problems, *Econometrica : Journal of the Econometric Society* 28 (3) (1960) 497–520. doi :10.2307/1910129.  
URL <https://www.jstor.org/stable/1910129>
- [12] L. A. Wolsey, G. L. Nemhauser, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1999.
- [13] P. Moulin, J. Liu, Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors, *IEEE Transactions on Information Theory* 45 (3) (1999) 909–919.
- [14] M. Lai, J. Wang, An unconstrained  $\ell_q$  minimization with  $q \leq 1$  for sparse solution of underdetermined linear systems, *SIAM Journal on Optimization* 21 (1) (2011) 82–101. doi :10.1137/090775397.  
URL <https://epubs.siam.org/doi/10.1137/090775397>
- [15] I. Rish, G. Grabarnik, *Sparse Modeling : Theory, Algorithms, and Applications*, 1st Edition, CRC Press, Inc., USA, 2014.
- [16] R. Tibshirani, Regression Shrinkage and Selection Via the Lasso, *Journal of the Royal Statistical Society, Series B* 58 (1996) 267–288.
- [17] M. R. Osborne, B. Presnell, B. Turlach, A new approach to variable selection in least squares problems, *IMA Journal of Numerical Analysis*.
- [18] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, *The Annals of Statistics* 32 (2) (2004) 407–499.
- [19] D. L. Donoho, Y. Tsaig, Fast Solution of  $\ell_1$  Norm Minimization Problems When the Solution May Be Sparse, *IEEE Transactions on Information Theory* 54 (11) (2008) 4789–4812. doi :10.1109/TIT.2008.929958.
- [20] R. Clark, G. Swayze, R. Wise, K. Livo, T. Hoefen, R. Kokaly, S. Sutley, USGS digital spectral library splib05a, US Geological Survey, Digital Data Series 231.
- [21] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Mathematical Programming* 91 (2) (2002) 201–213. doi :10.1007/s101070100263.

## 8. Annexes

### Table des figures

1	Principe de l’imagerie spectrale pour l’observation au sol - Source Iordache [1] . . . . .	5
2	Fonctions de la variable scalaire $\varphi(x_i)$ intervenant dans la définition de différentes normes $\ell_p$ , avec $\ \mathbf{x}\ _p^p = \sum_i \varphi( x_i )$ , pour $p = 0, 0.5, 1$ et $2$ - Source : R.Ben Mhenni[3] . . . . .	12
3	Représentation de l’optimisation de $\mathcal{P}_q$ réalisée en augmentant le rayon de la boule $\ell_q$ centrée à l’origine jusqu’à l’intersection avec l’ensemble des points réalisables $\mathbf{y} = \mathbf{H}\mathbf{x}$ . Rish & Grabarnik[15] . . . . .	12

4	Représentation de la norme $\ell_1$ comme l'enveloppe convexe de l'intersection des boules $\ell_0$ et $\ell_\infty$ .	13
5	Exemple de branchement sur la variable $b_{i_0}$ et création des nœuds fils : fils gauche $\hat{\mathcal{P}}_{2/1}^1$ et fils droit $\hat{\mathcal{P}}_{2/1}^0$ .	14
6	Représentation graphique de l'admissibilité d'une solution $(\bar{\mathbf{b}}_{\bar{S}}^0, \mathbf{x}_{\bar{S}}, \mathbf{x}_{S_1})$ au problème relâché $\hat{\mathcal{P}}_{2/1}$	18
7	Exemple de branchement sur la variable $b_q$ et création des nœuds fils : fils gauche $\hat{\mathcal{P}}_{2/1}^{(g)}$ et fils droit $\hat{\mathcal{P}}_{2/1}^{(d)}$ .	23
8	Arbre de branch-and-bound - exemple d'exécution sur un problème avec $K=3$ - nœuds de types <a href="#">racine</a> , <a href="#">feuille</a> , <a href="#">exploration</a> .	26
9	Solutions obtenues en fonction des résolutions de l'arbre de branch-and-bound. Les atomes mentionnés dans les commentaires sont $q_0, q_1, q_2$ et $q_3$ ; les atomes $\bar{q} \in \bar{S}$ sont colorés en noir.	27
10	Evolution des bornes primale et duale - exécution sur une instance ( $Q = 20, K = 5$ ). Types des nœuds : <a href="#">racine</a> , <a href="#">feuille</a> , <a href="#">exploration</a> , la borne primale est représentée en noire et la borne duale en <a href="#">violet</a> .	32
11	Arbre de branch-and-bound - exécution sur une instance ( $Q = 20, K = 5$ )	33
12	Représentation comparée des bornes inférieures obtenues avec les solveurs QP de Cplex(-) et Matlab(- -)	37
13	Instances $N \geq Q$ - Représentation du temps de résolution fonction du bruit - <a href="#">BB_R_Cplx_QP_RC</a> , <a href="#">BB_R_Cplx_QP_PC</a> , <a href="#">BB_R_Cplx_QP_RC_PC</a>	38
14	Instances $N \leq Q$ - Représentation du temps de résolution fonction du bruit- <a href="#">BB_R_Cplx_QP_RC</a> , <a href="#">BB_R_Cplx_QP_PC</a> , <a href="#">BB_R_Cplx_QP_RC_PC</a>	39
15	Exemple de chemin de solution donné par la solution $\mathbf{x}^*$ du problème (30) en fonction de $\lambda$ , avec 5 variables : $\bar{S} = \{1, 2, 3\}$ et $S_1 = \{4, 5\}$ . Chaque plan correspond à une nouvelle partie du chemin $\lambda \in [\lambda^{(k)}, \lambda^{(k-1)}]$ . Les cercles représentent les événements provoquant un changement dans la configuration du support. Les lignes pointillées verticales représentent les points de rupture. La ligne rouge en pointillé représente l'itération à laquelle $\lambda_c$ est atteint. Source : R.Ben Mhenni[3]	40
16	Représentation des spectres associés à la Calcite extraits de la base de données USGS - Clark & Swayze[20]	42
17	Représentation de l'atome $\mathbf{h}_{10}$ observé avec différents niveaux de bruit extrait de la base de données USGS - Clark & Swayze[20]	44
18	Représentation de la valeur de la norme $\ell_2$ fonction du saut de dualité $\delta$ - instances $N \geq Q$ - $K \in \llbracket 3, 7 \rrbracket$ - Méthodes : MIP_Cplex_DT_x avec $x = 4, x = 5, x = 6, x = 7$ et $x = 8$	47
19	Profils de performance - critère : temps de résolution Cplex MIQP - instances $N \leq Q$ - $K \in \llbracket 3, 7 \rrbracket$ - Méthodes : MIP_Cplex_DT_x avec $x = 4, x = 5, x = 6, x = 7$ et $x = 8$	48
20	Profils de performance - critère : temps de résolution branch-and-bound et MIP - instances $N \geq Q$ - $K \in \llbracket 3, 7 \rrbracket$ - Méthodes : <a href="#">BB_R_Cplx_QP</a> , <a href="#">MIP_Cplex_DT_4</a> , <a href="#">MIP_Cplex_DT_7</a> , <a href="#">BB_R_Hom</a> et <a href="#">BB_R_FCLS</a>	49
21	Profils de performance - critère : temps de résolution branch-and-bound et MIP - instances $N \leq Q$ - $K \in \llbracket 3, 7 \rrbracket$ - Méthodes : <a href="#">BB_R_Cplx_QP</a> , <a href="#">MIP_Cplex_DT_4</a> , <a href="#">MIP_Cplex_DT_7</a> et <a href="#">BB_R_Hom</a>	50

## Liste des tableaux

1	Tableau récapitulatif de l'impact des choix de branchement en fonction de la valeur de la variable $x_{i_0}$ à l'optimum de $\hat{\mathcal{P}}_{2/1}$	20
---	---	----