

Ensemble des solutions parcimonieuses exactes en démélange spectral : algorithme garanti et analyse des solutions

Nils FOIX-COLONIER¹ Sébastien BOURGUIGNON¹ Frédéric SCHMIDT^{2,3} Mehdi LATIF¹

¹Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

²Université Paris-Saclay, CNRS, GEOPS Orsay, 91405, France

³Institut Universitaire de France, Paris, France

Résumé – On s’intéresse à la résolution exacte de problèmes de démélange spectral linéaire parcimonieux, *i.e.* à la recherche de solutions à cardinalité limitée d’un problème de moindres carrés linéaires sous contraintes de positivité et de somme unité. L’originalité de la méthode proposée – dont le code Python est mis à disposition – repose sur son aspect multi-solutions : on renvoie l’ensemble des meilleurs supports de solution. Cette méthode est testée sur des données synthétiques et les performances obtenues sont prometteuses.

Abstract – We focus on the exact resolution of sparse spectral unmixing problems, that is, the search for cardinality-limited linear least squares solutions under non-negativity and sum-to-one constraints. The originality of the proposed method - for which the Python code is provided - lies in its multisolution nature: we return the set of supports that yield the best solutions. The method is tested on synthetic data, with promising results.

1 Introduction

Le démélange spectral de données de télédétection permet d’estimer la composition de scènes observées par un spectro-imageur. En planétologie, par exemple, le démélange de données hyperspectrales satellitaires est un levier pour caractériser la composition minéralogique de la surface martienne, difficilement accessible autrement [1]. De nombreux algorithmes de démélange existent [2], et ceux qui prennent en compte des contraintes de parcimonie (*i.e.* la solution n’est composée que de quelques spectres) sont particulièrement pertinents dans la mesure où ils intègrent une contrainte physique : en un pixel, donc sur une surface limitée au sol, peu de composants (donc de spectres) sont observés (mélangés). De nombreuses méthodes recourent à cette contrainte de manière approchée, comme dans [3, 4] par exemple. Cependant, il est rare que la parcimonie soit prise en compte de manière exacte, en imposant par exemple un nombre maximal de composantes non nulles. En effet, bien qu’elle permette d’améliorer significativement les solutions [5], cette contrainte en “norme” ℓ_0 rend le problème NP-difficile et donc potentiellement long à résoudre.

Dans le cas de problèmes de moindres carrés linéaires sous contrainte de cardinalité, des méthodes de séparation et évaluation – *Branch-and-Bound* – permettent l’optimisation exacte sous contrainte de parcimonie. Initialement, une reformulation en problème en nombres mixtes, couplée à une résolution par des solveurs génériques [6], a rendu possible l’obtention de solutions exactes tout en limitant grandement le coût de calcul par rapport à l’exploration exhaustive des solutions possibles. Il existe un exemple de mise en œuvre dans le cas du démélange spectral, où des contraintes de positivité et de somme unité sont prises en compte [5]. Depuis, il a été montré qu’une implémentation dédiée de *Branch-and-Bound*, pour les problèmes généraux d’optimisation parcimonieuse, était susceptible d’améliorer fortement les temps de calcul [7]. Cela motive donc une première contribution du présent article : un algorithme de *Branch-and-Bound* dédié au démélange.

Malgré la possibilité d’obtenir de manière garantie la

meilleure solution (*i.e.* le minimiseur global du problème d’optimisation), il arrive que cette solution manque d’interprétabilité et les spectres estimés dans le dictionnaire ne sont pas forcément ceux recherchés, notamment lorsque le niveau de bruit est fort. Comme la solution d’intérêt n’est pas forcément celle qui minimise la fonction objectif, nous proposons une approche inédite dite multi-solutions, consistant à étendre la recherche, et la garantie d’optimalité, à l’ensemble des combinaisons de spectres fournissant les meilleures solutions.

2 Problèmes de démélange

On considère que le spectre de réflectance mesuré en un pixel, $\mathbf{y} \in \mathbb{R}^N$, est issu d’un mélange linéaire des spectres des composants présents dans l’aire couverte par le pixel. On note $\mathbf{A} \in \mathbb{R}^{N \times P}$ la matrice de spectres de référence (le dictionnaire) et $\mathbf{x} \in \mathbb{R}^P$ le vecteur d’abondances, dont les coefficients non-nuls activent et pondèrent des colonnes de \mathbf{A} . Enfin, un bruit additif ε perturbe la mesure, donnant le modèle :

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon \quad (1)$$

Trouver le vecteur d’abondances \mathbf{x} à partir de la mesure \mathbf{y} et du dictionnaire \mathbf{A} est un problème inverse dont on peut trouver des solutions en maximisant la vraisemblance. En considérant que le bruit est blanc gaussien et identiquement distribué, il est équivalent de minimiser l’écart quadratique entre \mathbf{y} et $\mathbf{A}\mathbf{x}$. Cette approche donne en général de très mauvais résultats, en raison de la forte corrélation des colonnes de \mathbf{A} (si le dictionnaire est de grande taille, de nombreux spectres se ressemblent) et du mauvais conditionnement résultant de \mathbf{A} . Des contraintes physiques sont donc habituellement ajoutées : *abundance sum-to-one constraint* (ASC) et *abundance non-negativity constraint* (ANC). Le problème inverse de démélange linéaire avec les contraintes ANC et ASC est nommé FCLS (Fully Constrained Least Squares) dans la littérature [8]. Étant donné la positivité des coefficients de \mathbf{x} , contraindre la somme à 1 équivaut à contraindre la norme ℓ_1 à 1 :

$$\hat{\mathbf{x}}^{\text{FCLS}} = \arg \min_{\mathbf{x} \in [0,1]^P} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \text{ s.c. } \|\mathbf{x}\|_1 = 1 \quad (2)$$

Cependant, en pratique, la solution du problème FCLS (2) a tendance à inclure de nombreux spectres non pertinents et peut aussi en omettre. Ainsi, minimiser les moindres carrés permet d’avoir un bon ajustement de modèle ($\mathbf{A}\hat{\mathbf{x}} \approx \mathbf{y}$), mais pas nécessairement une bonne estimation des abondances, car on ajuste une solution au bruit. Il est alors intéressant de prendre en compte une hypothèse plus forte de parcimonie afin de limiter la dispersion des abondances estimées sur trop de spectres du dictionnaire [3, 4, 5]. La manière la plus rigoureuse de le faire est de contraindre la “norme” ℓ_0 de \mathbf{x} , qui compte le nombre de coefficients non-nuls, limitée à $K \in \mathbb{N}^*$:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in [0,1]^P} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \text{ s.c. } \|\mathbf{x}\|_1 = 1, \|\mathbf{x}\|_0 \leq K \quad (3)$$

Obtenir de manière garantie le minimiseur \mathbf{x}^* ci-dessus est bien plus compliqué que celui de l’équation (2) car la contrainte additionnelle discrète rend le problème de nature combinatoire. Il faut donc recourir à des algorithmes spécifiques, qui vont explorer un espace non-continu, celui des supports des solutions (le support $S \subset \llbracket 1, P \rrbracket$ d’un vecteur \mathbf{x} est l’ensemble des indices de ses coefficients non nuls). On peut donc construire un algorithme de type *Branch-and-Bound* [7]. La section suivante introduit un tel algorithme, créé sur mesure pour le démélange comme proposé récemment dans [9], et ici étendu dans un cadre multi-solutions (en termes de supports).

3 Méthode multi-solutions proposée

Soit \mathbb{S}^K l’ensemble des supports de taille limitée à K :

$$\mathbb{S}^K = \{S \subset \llbracket 1, P \rrbracket \mid |S| \leq K\} \quad (4)$$

En notant \mathbf{A}_S la matrice \mathbf{A} restreinte aux colonnes d’indices dans S , on définit le vecteur d’abondances minimisant le problème FCLS sur un support donné S :

$$\mathbf{x}_S^* = \arg \min_{\mathbf{x}_S \in [0,1]^{|S|}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_S \mathbf{x}_S\|_2^2 \text{ s.c. } \|\mathbf{x}_S\|_1 = 1 \quad (5)$$

Chaque support S fournit ainsi “sa meilleure solution”, mesurée par la valeur de la fonction objectif en \mathbf{x}_S^* , notée f^* :

$$f^*(S) = \frac{1}{2} \|\mathbf{y} - \mathbf{A}_S \mathbf{x}_S^*\|_2^2 \quad (6)$$

Cela permet de comparer et de classer les supports de \mathbb{S}^K : $f^*(S_1) \leq f^*(S_2) \leq \dots \leq f^*(S_{|\mathbb{S}^K|})$. Le problème (3) revient à trouver le meilleur support parcimonieux :

$$S_1 = \arg \min_{S \in \mathbb{S}^K} f^*(S) \quad (7)$$

On propose, dans ce travail, d’étendre le problème à la recherche des m meilleurs supports d’au plus K composantes :

$$\mathbb{S}^{*m} = \left\{ \begin{array}{l} S_i, i \in \llbracket 1, m \rrbracket \mid f^*(S_1) \leq \dots \leq f^*(S_m), \\ \forall S \in \mathbb{S}^K \setminus \mathbb{S}^{*m}, f^*(S) \geq f^*(S_m) \end{array} \right\} \quad (8)$$

Pour ce faire, on procède à une exploration de l’espace de recherche discret (ensemble des supports). Le problème initial (équation (3), nœud racine) est divisé en deux sous-problèmes complémentaires (enfants gauche et droit), sur la base d’une décision : la nullité ou non d’un coefficient de \mathbf{x} , le j -ème par exemple. On obtient d’une part un sous-problème (enfant gauche) où le j -ème spectre dans \mathbf{A} est inclus dans le mélange

– le vecteur d’abondances a donc pour support $S = \{j\}$ avec un coût de 1 pour la contrainte ℓ_0 – et d’autre part un sous-problème où ce spectre est exclu de la solution (support vide pour l’instant). Les autres variables sont alors non décidées, et on note l’ensemble de leurs indices (restants) $R = \llbracket 1, P \rrbracket \setminus \{j\}$. En procédant itérativement à de telles divisions, on obtient un arbre binaire de sous-problèmes, qui représente l’ensemble des supports possibles. En un nœud \mathcal{N} , le sous-problème s’écrit :

$$\mathbf{x}_{\mathcal{N}}^* = \begin{cases} \arg \min_{\mathbf{x}_{S \cup R} \in [0,1]^{|S \cup R|}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S \cup R} \mathbf{x}_{S \cup R}\|_2^2 \\ \text{s.c. } \|\mathbf{x}_{S \cup R}\|_1 = 1, \|\mathbf{x}_R\|_0 \leq K - |S| \end{cases} \quad (9)$$

En chaque nœud, on encadre la valeur optimale de la fonction objectif du problème (9). On a une borne supérieure UB, correspondant à une solution particulière, et une borne inférieure LB, donnant un minorant de la plus basse valeur de la fonction objectif sur tous les sous-problèmes issus de \mathcal{N} . Nous choisissons pour le calcul de UB de prendre \mathbf{x}_R nul, ce qui revient à résoudre un problème FCLS sur le support S :

$$\text{UB} = f^*(S) \quad (10)$$

Par construction, par les vecteurs d’abondances correspondant aux minimiseurs associés à chaque UB, on énumère l’ensemble des solutions parcimonieuses candidates pour être solution du problème (3). On constitue alors une liste de taille maximale m , qui reçoit les solutions trouvées au fur et à mesure, mise à jour afin de ne conserver que les m meilleures solutions.

La borne inférieure LB permet d’éviter de parcourir tout l’arbre tout en conservant la garantie mathématique de l’optimalité de \mathbb{S}^{*m} . En effet, si la borne inférieure d’un nœud est plus grande que la plus grande valeur parmi la liste des m meilleures solutions, alors les solutions des descendants du nœud considéré ne fourniront pas de solution meilleure, on peut donc élaguer cette branche. Usuellement, la borne inférieure est issue d’une relaxation convexe du problème (9). Sur $[0, 1]^{|R|}$, la meilleure relaxation convexe de la contrainte $\|\mathbf{x}_R\|_0 \leq K - |S|$ est la contrainte $\ell_1 : \|\mathbf{x}_R\|_1 \leq K - |S|$. C’est un résultat classique dont on omet la démonstration par manque de place. Dans notre cas, comme la norme ℓ_1 est fixée à 1 par l’ASC, cette relaxation en norme ℓ_1 disparaît car superflue, sauf quand S a pour cardinal K : on est sur une feuille de l’arbre, \mathbf{x}_R est le vecteur nul, on ne calcule donc pas de borne inférieure car il n’y pas d’enfants. La meilleure relaxation convexe possible est donc obtenue par :

$$\text{LB} = f^*(S \cup R) \quad (11)$$

Le parcours de l’arbre, et la manière dont est opérée la division en sous-problèmes, restent à choisir. Soient S et R les ensembles du nœud courant \mathcal{N} et $\mathbf{x}_{S \cup R}^*$ la solution de (11) associée. Nous choisissons l’indice de branchement j correspondant à la plus grande composante de $\mathbf{x}_{S \cup R}^*$ parmi l’ensemble d’indices R . Ainsi, l’enfant gauche – noté $\mathcal{N}^{(g)}$ – porte le sous-problème donné par :

$$S^{(g)} = S \cup \{j\} \text{ et } R^{(g)} = R \setminus \{j\} \quad (12)$$

Tandis que pour le nœud droit, $\mathcal{N}^{(d)}$, on a :

$$S^{(d)} = S \text{ et } R^{(d)} = R \setminus \{j\} \quad (13)$$

Cette création de nœuds est nommée **diviser** dans le pseudo-code de l’Algorithme 1, qui résume la méthode proposée. Par construction, le nœud droit hérite de la borne supérieure de son parent. Par soucis d’unicité, on ne considère donc que des

nœuds gauches comme solutions potentielles.

Pour le parcours de l’arbre, on ajoute les nœuds créés dans une pile en mettant le nœud gauche en dernier, de sorte à le traiter en premier. Cela correspond à une stratégie de parcours en profondeur d’abord qui, en combinaison avec le choix de la variable de branchement, va explorer prioritairement les solutions que l’on suppose meilleures.

La majorité du temps de calcul est consacrée à la détermination des valeurs UB et LB pour les nœuds (équations (10) et (11)), qui sont des problèmes de type FCLS, évalués sur différents supports. Ces problèmes, formulés comme des programmes quadratiques sous contrainte de positivité et de somme unité, sont transmis au solveur *Open Source* de type points intérieurs Clarabel [10], qui s’est avéré le plus efficace parmi les différents logiciels libres évalués.

Algorithme 1 : BnB- m

Entrées : \mathbf{A} , \mathbf{y} , K , m

- 1 $\mathcal{N} = \text{Nœud}(S = \emptyset, R = \llbracket 1, P \rrbracket)$
- 2 $L = \{\mathcal{N}\}$ # pile initialisée avec la racine
- 3 $L_{\text{sol}} = \emptyset$
- 4 $\text{UB}^{*m} = +\infty$
- 5 **tant que** $L \neq \emptyset$ **faire**
- 6 $\mathcal{N} = \text{dépiler } L$ # charge implicitement S et R
- 7 LB = calculée par équation (11)
- 8 **si** $\text{LB} < \text{UB}^{*m}$ **alors**
- 9 UB = calculée par équation (10)
- 10 **si** $\text{UB} < \text{UB}^{*m}$ **alors**
- 11 **si** $|L_{\text{sol}}| < m$ et \mathcal{N} est nœud gauche **alors**
- 12 ajouter \mathcal{N} à L_{sol}
- 13 **sinon**
- 14 retirer le dernier élément de L_{sol}
- 15 ajouter \mathcal{N} à L_{sol}
- 16 trier L_{sol} sur les UB des nœuds
- 17 $\text{UB}^{*m} = \text{UB}$ du m^{e} élément de L_{sol}
- 18 **si** $|S| < K$ et $|R| > 0$ **alors**
- 19 $\{\mathcal{N}^{(g)}, \mathcal{N}^{(d)}\} = \text{diviser}(\mathcal{N})$
- 20 empiler $\mathcal{N}^{(d)}$ sur L
- 21 empiler $\mathcal{N}^{(g)}$ sur L
- 22 # sinon la branche sous-optimale est coupée
- 23 **fin**

Sorties : L_{sol}

4 Expériences numériques

L’algorithme proposé est comparé à d’autres méthodes sur des données synthétiques. Une implémentation en Python, ainsi que les données utilisées, sont mises à disposition ici : gitlab.com/nils_fc/bnb-gretsi-2025. Les données ont été créées avec un dictionnaire \mathbf{A} ayant $N = 113$ bandes spectrales, issu de l’*United States Geological Survey* [11] et adapté pour simuler l’instrument hyperspectral OMEGA observant la planète Mars [1]. Le protocole expérimental consiste en la génération de 100 problèmes de démixage par triplet de paramètres (K, P, SNR) , pour $K \in \{3, 4, 5\}$ spectres mélangés, sélectionnés dans un dictionnaire de $P \in \{50, 100, 150, 200\}$ spectres, avec un rapport signal sur bruit $\text{SNR} \in \{30, 35, 40, 50, 55, 60\}$ (dB), où $\text{SNR} = 10 \log(\|\mathbf{A}\mathbf{x}\|_2^2 / (N\sigma^2))$. Les abondances sont tirées uniformément dans $\{\mathbf{x} \in [0.05, 1]^K \mid \|\mathbf{x}\|_1 = 1\}$ et un bruit blanc gaussien de puissance σ^2 est ajouté, correspondant au SNR

moyen voulu. L’échelle de SNR peut sembler inhabituelle mais résulte de la positivité de l’ensemble des variables. Ainsi, 30 dB correspond à un bruit assez fort.

Chaque problème est alors résolu avec quatre méthodes. “KFCLS” consiste à prendre les K plus grandes abondances de la solution FCLS [8] obtenue en résolvant (2). “Backward” est une heuristique par déflation [4], supprimant itérativement les plus petites composantes dans la solution FCLS jusqu’au niveau de parcimonie désiré. “BnB” correspond à la version mono-solution de la méthode proposée : on fixe m à 1. “BnB-10” est la méthode proposée, qui renvoie un ensemble de $m = 10$ solutions. L’évaluation d’un vecteur d’abondances estimé $\hat{\mathbf{x}}$ par rapport au vrai vecteur \mathbf{x} est faite à l’aide de deux métriques. La première est notée SI (Support Identifié) et vaut 1 quand les supports de $\hat{\mathbf{x}}$ et de \mathbf{x} sont les mêmes, 0 sinon. La seconde est le SRE (*Signal to Reconstruction Error ratio*) :

$$\text{SRE (dB)} = 10 \log_{10} \left(\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2} \right) \quad (14)$$

La Table 1 compare^(*) les performances moyennées sur les 100 instances et sur K et P . Sans surprise, le minimiseur global affiche de meilleures performances que ses concurrents KFCLS et Backward. Surtout, nous constatons l’intérêt de l’approche multi-solutions, qui permet de trouver un ensemble contenant la vraie solution dans des cas où celle-ci n’est pas obtenue par le minimiseur global, notamment lorsque le niveau de bruit augmente. Ainsi, pour $\text{SNR} = 30$ dB, si la vraie composition est retrouvée dans 28 % des cas par BnB, inclure les 10 meilleurs supports permet d’obtenir une probabilité de 55% qu’elle soit dans l’ensemble des solutions.

TABLE 1 : Performances comparées de quatre méthodes, dont l’algorithme proposé, moyennées sur K et P .

	SNR	Metric	KFCLS	Backward	BnB	BnB-10
30 dB		SI (%)	10.0	14.6	28.2	55.0
		SRE (dB)	7.2	8.7	13.1	18.9
40 dB		SI (%)	33.2	44.6	74.8	91.4
		SRE (dB)	18.3	21.9	31.6	35.7
50 dB		SI (%)	63.4	79.7	94.5	99.1
		SRE (dB)	34.7	40.8	46.0	47.3
60 dB		SI (%)	92.9	97.9	99.5	100.0
		SRE (dB)	54.6	56.7	57.4	57.5

Les résultats des expériences numériques sont détaillés en fonction de K , P et SNR, en Figure 1. L’apport de l’optimisation globale, tout comme celui d’une recherche multi-solutions, est d’autant plus significatif que les problèmes sont compliqués. Pour les problèmes les plus difficiles, beaucoup de solutions sont compatibles avec le niveau de bruit, il faudrait donc augmenter m pour augmenter la probabilité de retrouver le vrai support. L’intérêt que revêt le choix de m ressort également en Figure 2, qui représente la probabilité de trouver le vrai support en fonction de m et du SNR, pour $K = 5$ et $P = 50$.

Notons enfin que les bonnes performances de la méthode proposée doivent être mises en regard d’un coût de calcul

^(*)Les résultats pour la méthode BnB-10 ne sont pas immédiatement comparables : la méthode renvoie un ensemble de solutions, la probabilité affichée est celle que le support initial fasse partie de l’ensemble des 10 solutions. Le SRE est calculé sur la vraie solution si elle fait partie de l’ensemble des solutions, sinon il est calculé sur la première (le minimiseur global).

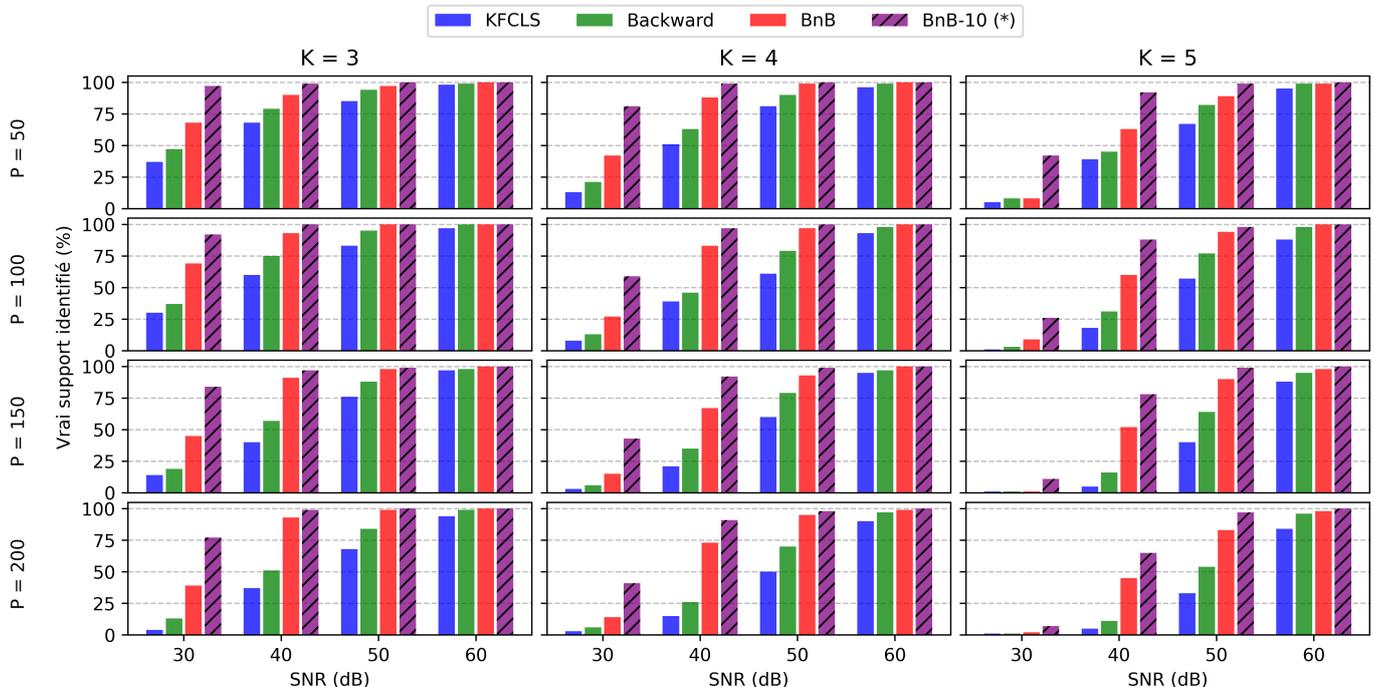


FIGURE 1 : Pourcentages de réussite à retrouver le support initial dans des problèmes avec divers triplets (K , P , SNR).

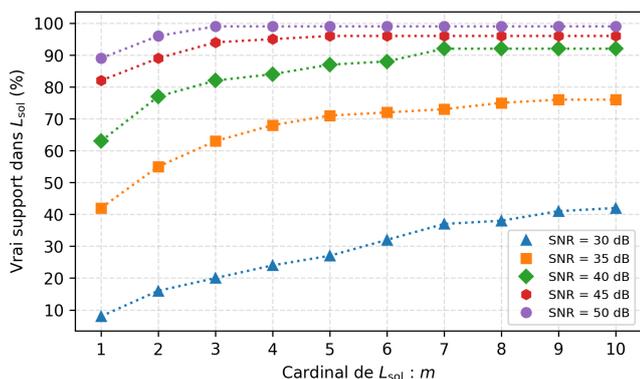


FIGURE 2 : Histogramme cumulé des indices de solutions dans L_{sol} sur les exemples à $K = 5$ et $P = 50$.

bien plus élevé que pour les estimateurs classiques et qui varie fortement avec la complexité du problème. Par exemple, sur un cœur de processeur i7-13700H, pour ($K = 3$, $P = 50$, $SNR = 30$), le temps de résolution moyen est de 0,28 secondes pour “BnB” et de 1,41 secondes pour “BnB-10”. Pour ($K = 3$, $P = 100$, $SNR = 40$) on a respectivement 1,18 et 11,4 secondes. Enfin, pour les problèmes les plus lourds, le temps de calcul moyen peut atteindre la demi-heure.

5 Conclusion et perspectives

Il peut y avoir un net intérêt à trouver des ensembles de solutions en exploitant les possibilités offertes par l’exploration de l’espace des supports par un algorithme de *Branch-and-Bound*, appliqué ici au démelange mais généralisable à d’autres problèmes. Outre le réglage du paramètre m , pour l’instant empirique, une question centrale est de savoir comment extraire la bonne solution de cet ensemble, ou de déterminer ce que peut nous apprendre l’ensemble de solutions en lui-même. Une perspective directe de ces travaux est d’exploiter ces ensembles

de solutions pour l’analyse d’un cube de données hyperspectrales, où la prise en compte de régularité spatiale doit pouvoir permettre de rechercher des mélanges parcimonieux cohérents tout en conservant des propriétés d’optimalité globale.

6 Bibliographie

- [1] F. Schmidt, G. Salomon, S. Bourguignon, J. Gurgurewicz, and D. Mège, “Mineralogy analysis using linear unmixing under group constraint,” in *2023 13th Workshop on Hyperspectral Imaging and Signal Processing : Evolution in Remote Sensing (WHISPERS)*, pp. 1–5, 2023.
- [2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, “Hyperspectral unmixing overview : Geometrical, statistical, and sparse regression-based approaches,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.
- [3] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, “Sparse Unmixing of Hyperspectral Data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2014–2039, 2011.
- [4] J. B. Greer, “Sparse demixing of hyperspectral images,” *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 219–228, 2012.
- [5] B. Mhenni Ramzi, S. Bourguignon, J. Ninin, and F. Schmidt, “Spectral Unmixing with Sparsity and Structuring Constraints,” in *2018 9th Workshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing (WHISPERS)*, pp. 1–5, 2018.
- [6] D. Bertsimas, A. King, and R. Mazumder, “Best subset selection via a modern optimization lens,” *The Annals of Statistics*, vol. 44, no. 2, pp. 813–852, 2016.
- [7] R. Ben Mhenni, S. Bourguignon, and J. Ninin, “Global optimization for sparse solution of least squares problems,” *Optimization Methods and Software*, vol. 37, no. 5, pp. 1740–1769, 2022.
- [8] D. Heinz and Chein-I-Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529–545, 2001.
- [9] M. Latif, G. Samain, N. Foix-Colonier, and S. Bourguignon, “Branch-and-bound algorithm for exact ℓ_0 -norm sparse spectral unmixing,” in *EUSIPCO 2025*, (Palermo, Italy), EURASIP, 2025.
- [10] P. J. Goulart and Y. Chen, “Clarabel : An interior-point solver for conic programs with quadratic objectives,” 2024.
- [11] R. N. Clark *et al.*, “USGS digital spectral library splib05a,” *US Geological Survey, Digital Data Series*, vol. 231, 2003.