

Seuillage CEL0 pour la minimisation ℓ_2 - ℓ_0 : comparaisons avec IHT

Emmanuel SOUBIES¹, Laure BLANC-FÉRAUD¹, Gilles AUBERT²

¹Université de Nice Sophia Antipolis, CNRS, Laboratoire I3S UMR 7271, 06903 Sophia Antipolis, France

²Université de Nice Sophia Antipolis, CNRS, Laboratoire J.A. Dieudonné UMR 7351, 06100 Nice, France

{soubies, blancf, gaubert}@unice.fr

Résumé – Le problème d’optimisation ℓ_2 - ℓ_0 a fait l’objet de nombreux travaux ces dernières années du fait de son importance pour diverses applications en traitement du signal et de l’image. Dans cet article, nous nous intéressons à l’algorithme *Iterative Hard Thresholding* (IHT) qui minimise directement la fonction objectif ℓ_2 - ℓ_0 , sans approximation de la pseudo-norme ℓ_0 , et converge vers un minimiseur (local) de celle-ci. Dans ce contexte, nous proposons de modifier le seuillage dur de l’algorithme IHT par un seuillage continu CEL0 associé à la fonctionnelle relaxée ℓ_2 -CEL0 récemment introduite. Cette fonctionnelle, continue, préserve les minimiseurs globaux de la fonctionnelle initiale ainsi que certains de ses minimiseurs locaux. Ces propriétés en font une bonne alternative au critère ℓ_2 - ℓ_0 . Des expériences numériques montrent les avantages du seuillage proposé sur le seuillage dur comme sa capacité à « éviter » des minimiseurs locaux de ℓ_2 - ℓ_0 .

Abstract – The ℓ_2 - ℓ_0 optimization problem has been widely studied because of its importance in many applications in signal and image processing. In this paper, we focus on the *Iterative Hard Thresholding* (IHT) algorithm which minimizes directly the ℓ_2 - ℓ_0 criterion, without approximating the ℓ_0 pseudo-norm, and converges to a (local) minimizer. Within this framework, we propose to replace the hard thresholding step in the IHT algorithm by the continuous CEL0 thresholding associated to the relaxed functional ℓ_2 -CEL0 recently introduced. This continuous functional preserves minimizers of the initial one as well as some local minimizers. These properties make it a good alternative to the ℓ_2 - ℓ_0 criterion. Numerical experiments show the benefits of the proposed thresholding regarding to the hard thresholding such as its ability to « avoid » local minimizers of ℓ_2 - ℓ_0 .

1 Introduction

Depuis plusieurs années, notamment avec le développement de l’échantillonnage compressif, de nombreux travaux sont consacrés à l’étude de problèmes visant à déterminer une solution parcimonieuse d’un système linéaire sous-déterminé. Typiquement, cette question se rencontre dans les problèmes de sélection de variables, séparation de sources, déconvolution de train d’impulsions ou encore en échantillonnage compressif. L’idée sous-jacente est que le signal recherché peut être représenté par combinaison linéaire de seulement quelques atomes d’un dictionnaire bien choisi ou simplement qu’il est naturellement parcimonieux (e.g. train d’impulsions). Le problème peut alors être formulé comme il suit :

$$\hat{x}_\lambda = \arg \min_{x \in \mathbb{R}^N} G_{\ell_0}(x) := \frac{1}{2} \|Ax - d\|^2 + \lambda \|x\|_0, \quad (1)$$

où $A \in \mathbb{R}^{M \times N}$ contient les atomes du dictionnaire ou bien représente la matrice d’observation du signal, $d \in \mathbb{R}^M$ est un vecteur contenant les données, $\|\cdot\|_0$ est la pseudo-norme ℓ_0 comptant le nombre de coefficients non nuls du signal et $\lambda > 0$ est un paramètre de pondération. Notons que le problème (1) peut aussi être formulé sous une forme contrainte généralement non équivalente du fait de la non convexité de $\|\cdot\|_0$. Dans la suite nous considérerons la formulation (1). Bien que ce problème soit NP-difficile, de nombreuses méthodes et algorithmes ont été proposés afin d’en approcher une solution.

Parmi les plus répandus, on peut citer la *relaxation convexe* ℓ_1 , les *algorithmes gloutons* ainsi que la relaxation par *pénalités non-convexes*, généralement non-différentiables en zéro (pour favoriser la parcimonie) mais *continues* permettant ainsi l’utilisation d’algorithmes émergents d’optimisation non-convexe.

Par ailleurs il existe un algorithme itératif, connu sous le nom de *Iterative Hard Thresholding* (IHT), qui contrairement à la majorité des autres méthodes minimise directement la fonctionnelle G_{ℓ_0} . Cet algorithme a été introduit par Blumensath et Davies [2] et sa convergence vers un minimiseur (local) de G_{ℓ_0} établie pour $\|A\| \leq 1$. Cette contrainte sur l’opérateur A a récemment été relâchée par Attouch *et al* [1] en considérant un pas de descente γ comme décrit par l’algorithme 1. La preuve de convergence repose sur le fait que G_{ℓ_0} vérifie l’inégalité de *Kurdyka-Lojasiewicz* (KL)¹. Par une autre approche et sans la nécessité pour G_{ℓ_0} de vérifier l’inégalité KL, Kowalski [5] montre la convergence de l’Algorithme 1 pour n’importe quelle règle de seuillage \mathcal{S} .

Cet algorithme, de la famille *Forward-Backward Splitting* (FBS) [4], itère entre une étape de descente et une étape de seuillage. La règle de seuillage $\mathcal{S}_{\gamma\lambda}^0$ est définie par l’opérateur proximal de la pseudo-norme $\|\cdot\|_0$:

$$\mathcal{S}_{\gamma\lambda}^0(y) = \text{prox}_{\gamma\lambda\|\cdot\|_0}(y) = \left(\text{prox}_{\gamma\lambda|\cdot|_0}(y_i) \right)_{i \in \mathbb{I}_N}, \quad (2)$$

1. voir [1, Section 2.2] et les références associées pour plus de détails sur cette inégalité. La preuve de convergence proposée dans [1] rentre en fait dans un cadre plus large dont l’algorithme IHT en est un exemple.

Algorithme 1 *Iterative Hard Thresholding algorithm*

Entrées: $x^0 \in \mathbb{R}^N$, $\gamma \in (0, \frac{1}{\|A\|^2})$

- 1: **répéter**
 - 2: $x^{k+1} = \mathcal{S}_{\gamma\lambda}^0(x^k - \gamma A^T(Ax^k - d))$
 - 3: **jusqu'à** convergence
-

où $\mathbb{I}_N = \{1, \dots, N\}$ et

$$\text{prox}_{\gamma\lambda|\cdot|_0}(u) = \begin{cases} 0 & \text{si } |u| < \sqrt{2\gamma\lambda}, \\ \{0, u\} & \text{si } |u| = \sqrt{2\gamma\lambda}, \\ u & \text{sinon,} \end{cases} \quad (3)$$

également connu sous le nom de *seuillage dur* et représenté en rouge sur la Figure 1 (droite) pour $\gamma = \lambda = 1$. Un des inconvénients de l'algorithme IHT vient de sa sensibilité à l'initialisation. Une « mauvaise » initialisation peut entraîner une convergence vers un minimiseur local de G_{ℓ_0} non parcimonieux. Afin d'éviter cela les auteurs dans [2] suggèrent d'initialiser IHT avec le résultat d'autres méthodes comme par exemple l'algorithme *Matching Pursuit* (MP) [6] et montrent que IHT permet ainsi d'en améliorer les performances.

Dans ce travail nous proposons de remplacer la règle de seuillage (3), utilisée dans l'algorithme IHT, par un seuillage *continu* tout en préservant la convergence vers un minimiseur (local) de G_{ℓ_0} et en améliorant la capacité de l'algorithme à « éviter » des minimiseurs locaux de G_{ℓ_0} . Nous commençons, dans la Section 2, par présenter le seuillage proposé (CEL0) ainsi que les principaux résultats théoriques de [9] garantissant la convergence de l'algorithme de seuillage itératif (FBS) vers un minimiseur (local) de G_{ℓ_0} . Ensuite les expériences numériques menées dans [2] sont reprises dans la Section 3 afin de montrer l'avantage du seuillage CEL0 par rapport au seuillage dur.

2 La fonctionnelle CEL0

Tout d'abord nous commençons par présenter la pénalité CEL0 (*Continuous Exact ℓ_0 penalty*) ainsi que les résultats théoriques associés [9]. Dans la suite, les notations suivantes sont utilisées :

- $a_i \in \mathbb{R}^M$, la $i^{\text{ème}}$ colonne² de $A \in \mathbb{R}^{M \times N}$,
- $\sigma(x) = \{i \in \mathbb{I}_N : x_i \neq 0\} \subset \mathbb{I}_N$, le support de $x \in \mathbb{R}^N$,
- $\sigma^-(x) = \{i \in \sigma(x) : |x_i| \leq \sqrt{2\lambda}/\|a_i\|\}$.

La pénalité CEL0 est une pénalité *continue*, non-convexe, approchant la pseudo-norme ℓ_0 et définie par

$$\Phi_{\text{CEL0}}(A, \lambda; x) = \sum_{i \in \mathbb{I}_N} \phi(\|a_i\|, \lambda; x_i), \quad (4)$$

où

$$\phi(a, \lambda; u) = \lambda - \frac{a^2}{2} \left(|u| - \frac{\sqrt{2\lambda}}{a} \right)^2 \mathbb{1}_{\{|u| \leq \sqrt{2\lambda}/a\}}. \quad (5)$$

Cette fonction ϕ est représentée en bleu sur la Figure 1 (gauche).

On notera alors G_{CEL0} la fonction objectif ℓ_2 -CEL0 associée :

$$G_{\text{CEL0}}(x) := \frac{1}{2} \|Ax - d\|^2 + \Phi_{\text{CEL0}}(A, \lambda; x). \quad (6)$$

2. On suppose $a_i \neq 0_{\mathbb{R}^N}$, $\forall i \in \mathbb{I}_N$.

Lorsque la matrice A est orthogonale, on peut montrer [9, §3] que la fonctionnelle G_{CEL0} est la convexifiée de G_{ℓ_0} . Bien que ce résultat ne soit plus vrai lorsque A n'est pas orthogonale, la *continuité* de cette fonctionnelle et le fait qu'elle *admette les mêmes minimiseurs globaux* que G_{ℓ_0} (voir §2.1) lui confèrent un intérêt particulier pour minimiser indirectement G_{ℓ_0} .

2.1 Liens entre les minimiseurs de G_{CEL0} et G_{ℓ_0}

Nous présentons maintenant les principaux résultats théoriques concernant les liens entre les minimiseurs de G_{CEL0} et G_{ℓ_0} . Une étude plus détaillée ainsi que les preuves des théorèmes peuvent être trouvées dans [9].

Théorème 1 (Minimiseurs globaux de G_{CEL0} et G_{ℓ_0})

- (i) *L'ensemble des minimiseurs globaux de G_{ℓ_0} est inclus dans l'ensemble des minimiseurs globaux de G_{CEL0} ,*

$$\arg \min_{x \in \mathbb{R}^N} G_{\ell_0}(x) \subseteq \arg \min_{x \in \mathbb{R}^N} G_{\text{CEL0}}(x), \quad (7)$$

- (ii) *réciquement si \hat{x} est un minimiseur global de G_{CEL0} , alors \hat{x}^0 défini par*

$$\forall i \in \mathbb{I}_N, \hat{x}_i^0 = \hat{x}_i \mathbb{1}_{\{|\hat{x}_i| \geq \sqrt{2\lambda}/\|a_i\|\}}, \quad (8)$$

est minimiseur global de G_{ℓ_0} et

$$G_{\text{CEL0}}(\hat{x}) = G_{\text{CEL0}}(\hat{x}^0) = G_{\ell_0}(\hat{x}^0). \quad (9)$$

Ce résultat, basé sur la caractérisation des minimiseurs de G_{ℓ_0} établie par Nikolova [7], nous assure une équivalence en termes de minimisation entre les fonctionnelles G_{ℓ_0} et G_{CEL0} . Cependant, G_{CEL0} est également non-convexe et, à ce jour, les algorithmes d'optimisation dits « *nonsmooth nonconvex* » [1, 8] ne nous assurent, au mieux, qu'une convergence locale. C'est pourquoi il est aussi intéressant d'étudier les minimiseurs locaux de G_{CEL0} .

Théorème 2 (Minimiseurs locaux de G_{CEL0} et G_{ℓ_0}) *Soit \hat{x} un minimiseur local de G_{CEL0} , alors \hat{x}^0 défini par (8) est un minimiseur local de G_{ℓ_0} et (9) est vérifiée.*

Le deuxième point du Théorème 1 est donc également vérifié pour les minimiseurs locaux de G_{CEL0} . Cependant, les minimiseurs locaux de G_{ℓ_0} ne le sont pas nécessairement pour G_{CEL0} . Ceci est caractérisé dans [9] et vérifié expérimentalement [9, §4.2] sur des exemples numériques où on observe qu'un nombre important de minimiseurs locaux de G_{ℓ_0} ne sont pas points critiques pour G_{CEL0} . De fait, il semblerait qu'en général G_{CEL0} admette « moins »³ de minima locaux que G_{ℓ_0} ce qui rend cette fonctionnelle d'autant plus intéressante pour la minimisation (indirecte) de G_{ℓ_0} .

Toutefois, G_{CEL0} admet également des points critiques qui ne sont pas des minimiseurs locaux de G_{CEL0} . Un algorithme peut donc potentiellement converger vers un tel point (instable) pour lequel il n'est pas possible de conclure sur l'optimalité (locale) pour G_{ℓ_0} . On a alors le lemme suivant :

3. « moins » au sens des \hat{x}^0 du Théorème 2.

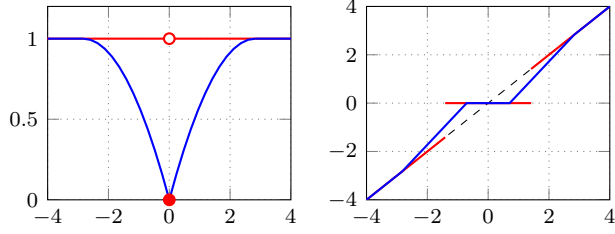


FIGURE 1 – Gauche : Pénalités (1D) ℓ_0 (rouge) et CEL0 avec $a = 0.5$ (bleu) pour $\lambda = 1$. Droite : Seuillages associés Dur (rouge) et CEL0 (bleu) pour $\gamma = 1$.

Lemme 1 *Tout point critique \hat{x} de G_{CEL0} tel que $\sigma^-(\hat{x}) = \emptyset$ est minimiseur (local) de G_{ℓ_0} et $G_{\text{CEL0}}(\hat{x}) = G_{\ell_0}(\hat{x})$.*

Ce résultat nous permet, à partir de n'importe quel algorithme convergeant vers un point critique de G_{CEL0} , de définir un macro algorithme [9, Algorithme 1] en ajoutant une boucle externe afin d'itérer d'un point critique à un autre en assurant une décroissance de la fonction objectif jusqu'à atteindre un \hat{x} tel que $\sigma^-(\hat{x}) = \emptyset$. Un tel point est donc, d'après le Lemme 1, un minimiseur (local) de G_{ℓ_0} .

2.2 Le seuillage CEL0

Utiliser l'algorithme FBS pour la minimisation de G_{CEL0} , revient à modifier la règle de seuillage $\mathcal{S}_{\gamma\lambda}^0$ de l'algorithme 1 par l'opérateur proximal de Φ_{CEL0} donné par

$$\mathcal{S}_{\gamma\lambda}^{\text{CEL0}}(y) = \left(\text{prox}_{\gamma\phi(\|a_i\|, \lambda; \cdot)}(y_i) \right)_{i \in \mathbb{I}_N}, \quad (10)$$

avec $\text{prox}_{\gamma\phi(a, \lambda; \cdot)}(u)$ défini comme suit :

$$\begin{cases} \text{sign}(u) \min \left(|u|, \frac{(|u| - \sqrt{2\lambda\gamma}a)_+}{1 - a^2\gamma} \right) & \text{si } a^2\gamma < 1, \quad (11a) \\ u \mathbb{1}_{\{|u| > \sqrt{2\lambda\gamma}\}} + \{0, u\} \mathbb{1}_{\{|u| = \sqrt{2\lambda\gamma}\}} & \text{si } a^2\gamma \geq 1. \quad (11b) \end{cases}$$

Lorsque $a^2\gamma < 1$, on obtient ainsi un seuillage *continu* (11a) qui est présenté sur la Figure 1 (droite) en bleu pour $a = 0.5$ et $\lambda = \gamma = 1$. Sinon on retrouve le seuillage dur (11b). En suivant [1], étant donné que $\frac{1}{2}\|Ax - d\|^2$ est une fonction polynomiale et que Φ_{CEL0} est polynomiale par morceaux, G_{CEL0} est semi-algébrique et vérifie donc l'inégalité KL. Il en suit que l'algorithme FBS convergera vers un point critique de G_{CEL0} si $\gamma \in (0, 1/\|A\|^2)$. Or il est clair que

$$\|A\| = \sup_{\|x\|=1} \|Ax\| \geq \max_{i \in \mathbb{I}_N} \|a_i\|, \quad (12)$$

ce qui implique que

$$\forall i \in \mathbb{I}_N, \|a_i\|^2\gamma < 1, \quad (13)$$

et $\mathcal{S}_{\gamma\lambda}^{\text{CEL0}}$ est alors réduit à la partie *continue* (11a) pour chacune des variables. L'algorithme FBS résultant est donc donné par le schéma IHT où le seuillage dur (3) est remplacé par (11a).

Notons qu'à travers ce seuillage on retrouve le seuillage Firm introduit en 1995 par Bruce et Gao [3] comme un compromis

entre les seuillages doux et dur. Ce seuillage est défini par deux paramètres (λ_1, λ_2) et on peut identifier à partir de (11a) que le seuillage CEL0 correspond à $\lambda_1 = \sqrt{2\lambda\gamma}a$ et $\lambda_2 = \sqrt{2\lambda}/a$ où $a = \|a_i\|$ est ici différent pour chacune des composantes du signal $i \in \mathbb{I}_N$. Bien sûr, le fait de considérer la pénalité CEL0 permet aussi d'envisager l'utilisation d'autres algorithmes émergents d'optimisation « nonsmooth nonconvex », par exemple [8], pour minimiser G_{CEL0} et donc G_{ℓ_0} .

La section suivante présente les résultats d'expériences numériques visant à comparer IHT et sa version modifiée avec le seuillage CEL0 (11a). Nous avons vu que, afin de garantir la convergence vers un minimiseur (local) de G_{ℓ_0} , il était nécessaire d'ajouter une boucle externe à l'algorithme (macro-algo [9, Algorithme 1]). Cependant, afin de garder la simplicité du schéma FBS, nous ne considérerons pas le macro-algorithme dans la suite. Cette boucle externe permet de « sortir » de points critiques de G_{CEL0} qui ne seraient pas des minimiseurs locaux. Nous avons pu remarquer, sur des exemples pratiques, que le macro algorithme ne réalisait que quelques, voire aucune, itérations. Dans ce dernier cas, l'algorithme FBS avait alors convergé vers un point critique de G_{CEL0} vérifiant les hypothèses du Lemme 1, c'est-à-dire un minimiseur (local) de G_{ℓ_0} .

3 Résultats numériques

Afin de comparer les seuillages dur (3) et CEL0 (11a) (avec $a = \|a_i\|$ pour la composante $i \in \mathbb{I}_N$), nous reprenons les expériences numériques effectuées dans le cadre de l'algorithme IHT [2]. Tout d'abord nous nous intéressons à la capacité de l'algorithme à minimiser G_{ℓ_0} [2, §4.1]. Pour ce faire, on génère 1000 dictionnaires $A \in \mathbb{R}^{128 \times 256}$ avec des éléments (colonnes) distribués uniformément sur la sphère unité. Pour chaque dictionnaire, un signal $x^* \in \mathbb{R}^{256}$ est construit avec des coefficients générés selon une loi normale centrée réduite. Cette étape de génération, identique à [2, §4.1], est réalisée de telle sorte qu'en moyenne, sur les 1000 réalisations, $\sigma(x^*) = K \in \mathbb{N}$ et que $\forall i \in \sigma(x^*), |x_i^*| > \sqrt{2\lambda}$. Ainsi la valeur de λ est fixée en fonction de la génération du signal et sera la même pour chacune des méthodes testées par la suite. Enfin, pour chaque dictionnaire, différentes valeurs de K (taille du support de x^*) et différents niveaux de bruit (gaussien) sont considérés. Les données observées s'écrivent alors

$$d = Ax^* + \epsilon \text{ avec } \epsilon \sim \mathcal{N}(0, I\sigma^2). \quad (14)$$

Ensuite, l'algorithme Matching Pursuit (MP) [6] est exécuté et stoppé lorsque la fonction objectif G_{ℓ_0} commence à croître. La solution obtenue, \hat{x}_{MP} , est utilisée comme initialisation pour l'algorithme FBS avec les seuillages dur (IHT) et CEL0. Pour chaque solution obtenue \hat{x}_{dur} et \hat{x}_{CEL0} , nous calculons le Rapport de l'énergie du Signal sur la valeur de la Fonction Coût,

$$\text{RSFC}(\hat{x}) = \|Ax^*\|^2 / G_{\ell_0}(\hat{x}). \quad (15)$$

4. cette inégalité est une condition nécessaire d'optimalité globale pour G_{ℓ_0} [7].

Cette grandeur nous permet d’observer la capacité de l’algorithme à minimiser la fonction coût G_{ℓ_0} tout en étant « normalisée » par rapport à l’énergie du signal qui diffère d’une réalisation à une autre. La Figure 2 présente la différence

$$10\log_{10}(\overline{\text{RSFC}}(\hat{x})) - 10\log_{10}(\overline{\text{RSFC}}(\hat{x}_{\text{MP}})), \quad (16)$$

exprimé en décibels (dB) où la notation « barre » désigne la moyenne sur les 1000 réalisations. On peut noter l’amélioration qu’apporte le seuillage CEL0 par rapport au seuillage dur. En effet, les résultats de la Figure 2 montrent que, pour une même initialisation \hat{x}_{MP} , le seuillage CEL0 est plus performant que le seuillage dur en terme de minimisation de G_{ℓ_0} . Cela peut s’expliquer par le fait que G_{CEL0} a généralement moins de minimiseurs locaux que G_{ℓ_0} . Ces résultats sont en accord avec ceux des expériences numériques réalisées dans [9].

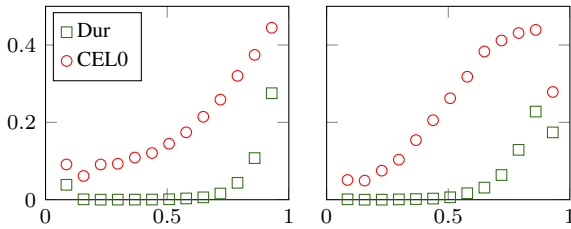


FIGURE 2 – Différence (16) avec $\hat{x} = \hat{x}_{\text{dur}}$ (carrés) et $\hat{x} = \hat{x}_{\text{CEL0}}$ (cercles), pour un SNR = 80 (gauche) et SNR = 0 (droite). L’axe des abscisses représente différents niveaux de parcimonie ($K/128$).

Nous nous intéressons maintenant au problème de reconstruction exacte d’un signal à partir d’un nombre limité de mesures [2, §4.2]. Les données sont générées comme précédemment sans ajouter de bruit. La stratégie d’initialisation de FBS est la même que pour l’expérience précédente (MP) et, pour chaque réalisation, l’optimisation est répétée pour un ensemble de valeurs de λ . Ensuite le nombre de Bonnes Détections (BD) du support ainsi que le nombre de Fausses Alarmes (FA) est déterminé.

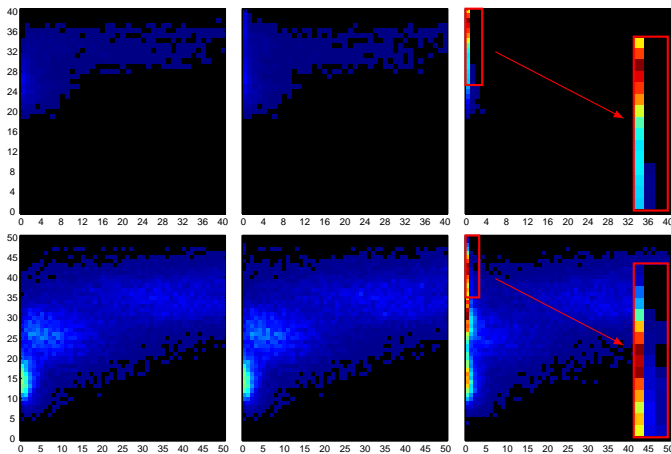


FIGURE 3 – BD (ord) Vs FA (abs) pour $K = 40$ (haut) et 50 (bas). Pour 1000 réalisations de A , x^* et différents λ , les nombres de BD et FA sont déterminés et la position du plan correspondante est incrémentée. De gauche à droite : MP, IHT et FBS-CEL0. Colorbar : bleu (faible) → rouge (fort).

La Figure 3 présente les diagrammes BD Vs FA obtenus.

Pour une position donnée cette représentation exprime le nombre de fois, sur les 1000 réalisations et les différentes valeurs de λ utilisées, où le signal estimé comportait les nombres de FA et de BD indiqués respectivement en abscisses et ordonnées. Plus de grandes valeurs sont concentrées sur la partie haut-gauche du diagramme, plus la méthode permet de retrouver, en moyenne, un nombre important d’éléments du support tout en ayant peu de FA. Ainsi, on peut apprécier la capacité du seuillage CEL0 (droite) à éliminer, pour $K = 40$, les FA présentes à l’initialisation (*i.e.* la solution MP) tout en ajoutant un nombre important de BD ce qui n’est pas le cas avec le seuillage dur (centre). Cette supériorité du seuillage CEL0 s’observe également pour $K = 50$ où le seuillage dur n’améliore que très légèrement les résultats de MP. Pour des valeurs de K plus grandes, on observe que le gain par rapport à MP obtenu avec CEL0 devient moins important.

4 Conclusion

Dans cet article, nous nous sommes intéressés à l’algorithme FBS pour la minimisation de la fonctionnelle $\ell_2\text{-}\ell_0$. L’étape de seuillage de l’algorithme peut alors être effectuée par le seuillage dur (algorithme IHT) ou par le seuillage continu CEL0 minimisant une fonctionnelle relaxée G_{CEL0} préservant les minimiseurs de la fonction objectif initiale G_{ℓ_0} . La comparaison de ces deux règles de seuillage sur des applications numériques a montré l’avantage du seuillage CEL0 sur le seuillage dur pour la minimisation de G_{ℓ_0} avec l’algorithme FBS. Enfin, il est à noter que la continuité de la pénalité CEL0 permet également l’utilisation d’algorithmes récents d’optimisation non convexe, autres que FBS, pour minimiser indirectement G_{ℓ_0} .

Références

- [1] H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems : proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods. *Mathematical Programming*, 137(1-2) :91–129, 2013.
- [2] T. Blumensath and M.E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and App.*, 14(5-6) :629–654, 2008.
- [3] A.G. Bruce and H-Y. Gao. Waveshrink : shrinkage functions and thresholds. In *International Symposium on Optical Science, Engineering, and Instrumentation*, pages 270–281. SPIE, 1995.
- [4] P.L. Combettes and V.R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. & Simul.*, 4(4) :1168–1200, 2005.
- [5] M. Kowalski. Thresholding rules and iterative shrinkage/thresholding algorithm : A convergence study. In *ICIP*. IEEE, 2014.
- [6] S. G Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans on Signal Processing*, 41(12) :3397–3415, 1993.
- [7] M. Nikolova. Description of the minimizers of least squares regularized with ℓ_0 -norm. Uniqueness of the global minimizer. *SIAM Journal on Imaging Sciences*, 6(2) :904–937, 2013.
- [8] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock. On Iteratively Reweighted Algorithms for Nonsmooth Nonconvex Optimization in Computer Vision. *SIAM Journal on Imaging Sciences*, 8(1) :331–372, 2015.
- [9] E. Soubies, L. Blanc-Féraud, and G. Aubert. A Continuous Exact ℓ_0 penalty (CEL0) for least squares regularized problem. *To appear in SIAM Journal on Imaging Sciences*, 2015.