



HAL
open science

Techniques d'accélération d'une méthode de Branch-and-bound pour l'optimisation parcimonieuse

Gwenaël Samain, Sébastien Bourguignon, Jordan Ninin

► **To cite this version:**

Gwenaël Samain, Sébastien Bourguignon, Jordan Ninin. Techniques d'accélération d'une méthode de Branch-and-bound pour l'optimisation parcimonieuse. GRETSI'22 XXVIIIème Colloque Francophone de Traitement du Signal et des Images, Sep 2022, Nancy, France. hal-03781318

HAL Id: hal-03781318

<https://hal.science/hal-03781318>

Submitted on 20 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Techniques d'accélération d'une méthode de Branch-and-bound pour l'optimisation parcimonieuse

Gwenaël SAMAIN¹, Sébastien BOURGUIGNON¹, Jordan NININ²

¹Laboratoire des Sciences du Numérique de Nantes - Centrale Nantes
1 rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France

²Laboratoire des Sciences et Techniques de l'information de la Communication et de la Connaissance
Technopole Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France

gwenael.samain@ls2n.fr, sebastien.bourguignon@ls2n.fr
jordan.ninin@ensta-bretagne.fr

Résumé – Les problèmes d'ajustement de modèles de faible cardinalité ont trouvé de nombreuses applications en statistique, en finance et en traitement du signal. Au sein de ces problèmes, nous nous intéressons au problème de l'ajustement par moindres carrés, pénalisé par la cardinalité de la solution. Nous utilisons un algorithme branch-and-bound pour trouver l'optimum global de ce problème NP-complet. Au sein de cet algorithme, les bornes inférieures évaluées à chaque nœud sont calculées par la résolution de problèmes en norme ℓ_1 , qui disposent d'une large panoplie de méthodes dédiées. Dans cette communication, nous exposons deux techniques exploitant la dualité convexe pour, d'une part, éviter de résoudre certains problèmes de relaxation jusqu'à l'optimalité, permettant d'accélérer le calcul des bornes inférieures, et d'autre part réduire la dimension de ces problèmes par une stratégie de *screening*. Une étude expérimentale valide la pertinence de ces techniques pour réduire le temps de calcul.

Abstract – Linear sparse regression has found many applications in statistics, finance and signal processing. Among these problems, we investigate least-square regression penalized by the solution cardinality. We use a branch-and-bound algorithm to find the global optimum of this NP-complete problem. Within this algorithm, the lower bounds evaluated at each node are computed by the resolution of ℓ_1 -norm problems, for which plenty of dedicated methods have been proposed. In this communication, we study two techniques based on convex duality for, first, avoiding solving some of these relaxation problems to optimality, accelerating the lower bound computations, and, second, reducing the dimension of such problems with screening methods. An experimental study validates the efficiency of the techniques in order to reduce the computing time.

Introduction

Les problèmes de régression parcimonieuse – l'ajustement de modèle de faible cardinalité – ont trouvé de nombreuses applications, par exemple en statistique [7], finance [4] et traitement du signal [5], et peuvent s'écrire via la pseudo-norme ℓ_0 :

$$\min_{x \in \mathbb{R}^Q} \frac{1}{2} \|y - Ax\|_2^2 + \mu \|x\|_0 \quad (1)$$

avec $\|x\|_0 := \text{Card}(\{i | x_i \neq 0\})$. Le problème (1), NP-difficile (voir [8]), est souvent abordé par des méthodes heuristiques [1], des algorithmes d'optimisation convexe résolvant sa relaxation en norme ℓ_1 [1] et des méthodes locales optimisant des relaxations non-convexes [10] pour les problèmes de grande taille.

Ces méthodes ne garantissant pas de trouver le minimum global du problème (1), nous nous concentrons sur la résolution globale et exacte de celui-ci. Tandis que de précédents travaux s'appuient sur une reformulation en nombres mixtes du problème, résolue par un solveur générique (voir [5, 3]), des algorithmes de branch-and-bound dédiés, exploitant les spécificités du problème, peuvent s'avérer bien plus performants, permet-

tant de résoudre exactement des problèmes difficiles où l'on cherche quelques dizaines de variables parmi quelques centaines [2, 7, 4, 6]. Nous proposons ici des techniques visant à accélérer ce type d'algorithmes, basées sur la dualité convexe.

Nous rappelons tout d'abord le principe de ce genre d'algorithmes en Section 1. Nous proposons ensuite deux techniques d'amélioration de leur performance : d'une part, l'utilisation de la dualité faible pour faciliter le rejet de nœuds en Section 2, d'autre part l'utilisation d'une méthode de screening pour réduire la dimension des problèmes convexes en Section 3. Nous terminons par des simulations numériques en Section 4, avant de conclure et d'esquisser quelques perspectives.

1 Principe de résolution

Les approches de résolution dédiées de la littérature [2, 4] utilisent un algorithme de branch-and-bound qui requiert le calcul de relaxations convexes du problème (1). Pour permettre ces relaxations convexes, nous contraignons les amplitudes de

x dans une boîte : $\|x\|_\infty \leq M$, donnant le problème :

$$\hat{x} := \arg \min_{x \in \mathbb{R}^Q} \frac{1}{2} \|y - Ax\|_2^2 + \mu \|x\|_0, \text{ s.c. } \|x\|_\infty \leq M. (\mathcal{P}_{2+0})$$

Une fois le *support* de la solution (les indices des variables non-nulles) fixé, c'est un simple problème de moindres carrés sous contrainte de boîte. Ainsi, la principale difficulté est la recherche du support optimal S^* , et nous construisons l'algorithme de branch-and-bound 1 où l'espace de recherche utilisé est celui des *supports*.

Algorithme 1 Algorithme Branch-&-Bound de base

- 1: **procédure** BRANCHANDBOUND
 - 2: $\bar{ub} \leftarrow +\infty, \bar{x} \leftarrow \text{Null}, \mathcal{L} \leftarrow \{\mathbf{N}(\emptyset, \emptyset, \{1..Q\})\}$
 - 3: **tant que** \mathcal{L} non vide **faire**
 - 4: Extraire un nœud \mathbf{N} de \mathcal{L} .
 - 5: Diviser \mathbf{N} en deux sous-nœuds \mathbf{L} et \mathbf{R} .
 - 6: **pour tout** sous-nœud \mathbf{N}_i dans $\{\mathbf{L}, \mathbf{R}\}$ **faire**
 - 7: Calculer une borne inférieure ($x_{\text{lb}}^{\mathbf{N}_i}, \text{lb}_{\mathbf{N}_i}$)
 - 8: Calculer une borne supérieure ($x^{\mathbf{N}_i}, \text{ub}_{\mathbf{N}_i}$)
 - 9: **si** $\text{ub}_{\mathbf{N}_i} < \bar{ub}$ **alors**
 - 10: $\bar{x} \leftarrow x^{\mathbf{N}_i}$ et $\bar{ub} \leftarrow \text{ub}_{\mathbf{N}_i}$.
 - 11: Retirer de \mathcal{L} les nœuds \mathbf{N}_j t.q. $\text{lb}_{\mathbf{N}_j} > \bar{ub}$.
 - 12: **si** $\text{lb}_{\mathbf{N}_i} < \bar{ub}$ **alors**
 - 13: Insérer \mathbf{N}_i dans \mathcal{L} .
 - 14: **retourner** $(\bar{ub}, \hat{x} = \bar{x})$
-

1.1 Structuration de l'espace des supports

Au sein de notre algorithme de branch-and-bound, un nœud représente une configuration du support selon les choix de branchement réalisés : on note ainsi S_0 l'ensemble des variables à 0, S_1 l'ensemble de celles sélectionnées dans le support, et \bar{S} l'ensemble de celles pour lesquelles aucune décision n'a encore été prise. On notera un nœud par $\mathbf{N}(S_1, S_0, \bar{S})$, ou \mathbf{N} en l'absence d'ambiguïté. La Figure 1 illustre le processus de branchement : diviser un nœud signifie ainsi prendre un indice $i \in \bar{S}$ et, pour un enfant (ici le gauche), inclure dans le support la variable x_i associée en imposant $i \in S_1$ et, pour l'autre enfant (ici le droit), exclure x_i du support en imposant $i \in S_0$.

Les composants de base d'une méthode de branch-and-bound sont la *stratégie de branchement*, qui définit la manière de diviser un nœud, la *stratégie d'exploration*, qui définit le prochain nœud à diviser, et les *opérateurs de borne*, qui définissent la manière de calculer les bornes supérieures et inférieures, dont nous détaillons rapidement la construction.

1.2 Borner les nœuds

Au nœud $\mathbf{N}(S_1, S_0, \bar{S})$, on a le sous-problème suivant :

$$\begin{aligned} \min_{x \in \mathbb{R}^Q} \quad & \frac{1}{2} \|y - Ax\|_2^2 + \mu |S_1| + \mu \|x_{\bar{S}}\|_0 \\ \text{s.c.} \quad & \|x\|_\infty \leq M, x_{S_0} = 0. \end{aligned} \quad (\mathcal{P}_{2+0}^{\mathbf{N}})$$

$$\begin{aligned} S_1 &= \{4\} \\ S_0 &= \{5, 6\} \\ \bar{S} &= \{1, 2, 3, 7\} \end{aligned}$$

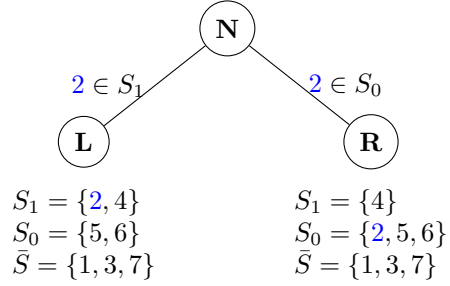


FIGURE 1 – Un exemple de branchement avec $x \in \mathbb{R}^7$, utilisant la variable x_2 . Les ensembles S_1, S_0, \bar{S} sont mis à jours dans les nœuds enfants \mathbf{L} and \mathbf{R} en fonction.

Ce problème étant également NP-difficile par le terme $\|x_{\bar{S}}\|_0$, il s'agit d'évaluer une borne inférieure et une borne supérieure sur $(\mathcal{P}_{2+0}^{\mathbf{N}})$ comme dans l'Algorithme 1.

Une borne supérieure et la solution réalisable associée peuvent être calculées comme dans [2] en considérant $x_{\bar{S}} = 0$:

$$\begin{aligned} \text{ub}_{\mathbf{N}} &:= \min_{x \in \mathbb{R}^Q} \frac{1}{2} \|y - Ax\|_2^2 + \mu |S_1| \\ \text{s.c.} \quad & \|x\|_\infty \leq M, x_{S_0} = 0, x_{\bar{S}} = 0. \end{aligned} \quad (2)$$

C'est un simple problème de moindres carrés sous contrainte de boîte en x_{S_1} . Durant toute l'exécution du branch-and-bound, $\bar{ub} = \min_{\mathbf{N}} \text{ub}_{\mathbf{N}}$ (voir l'Algorithme 1) est le meilleur candidat courant pour le minimum global du problème (\mathcal{P}_{2+0}) .

Pour la borne inférieure, une relaxation de $\|x_{\bar{S}}\|_0$ est $\frac{1}{M} \|x_{\bar{S}}\|_1$: $\|x_{\bar{S}}\|_1 := \sum_{i \in \bar{S}} |x_i| \leq \sum_{i \in \bar{S}} M = M \|x_{\bar{S}}\|_0$, donc :

$$\begin{aligned} \text{lb}_{\mathbf{N}} &:= \min_{x \in \mathbb{R}^Q} \frac{1}{2} \|y - Ax\|_2^2 + \mu |S_1| + \frac{\mu}{M} \|x_{\bar{S}}\|_1 \\ \text{s.c.} \quad & \|x\|_\infty \leq M, x_{S_0} = 0. \end{aligned} \quad (\mathcal{P}_{2+1}^{\mathbf{N}})$$

est bien une borne inférieure de la valeur optimale de $(\mathcal{P}_{2+0}^{\mathbf{N}})$, calculée par une méthode homotopique dédiée proposée dans [2], qui construit des itérés x^k associés à des $\mu^k \rightarrow \mu$ décroissants.

1.3 Stratégie de branchement et d'exploration

La stratégie de branchement utilisée est le maximum des amplitudes décrit dans [2] : étant donné l'optimum $x_{\text{lb}}^{\mathbf{N}}$ du problème $(\mathcal{P}_{2+1}^{\mathbf{N}})$, la variable $j \in \bar{S}$ ayant la plus grande valeur absolue sera sélectionnée : $j \in \arg \max_{i \in \bar{S}} |(x_{\text{lb}}^{\mathbf{N}})_i|$. La stratégie d'exploration priorise les nœuds ayant $|S_1|$ grand [2].

2 Rejet précoce

Au nœud \mathbf{N} , la valeur optimale de $(\mathcal{P}_{2+1}^{\mathbf{N}})$ donne une borne inférieure $\text{lb}_{\mathbf{N}}$ du sous-problème $(\mathcal{P}_{2+0}^{\mathbf{N}})$. Si $\text{lb}_{\mathbf{N}} \geq \bar{ub}$, le nœud peut être rejeté puisqu'il ne contient pas la solution optimale : c'est un nœud sous-optimal (ligne 11 de l'Algorithme 1).

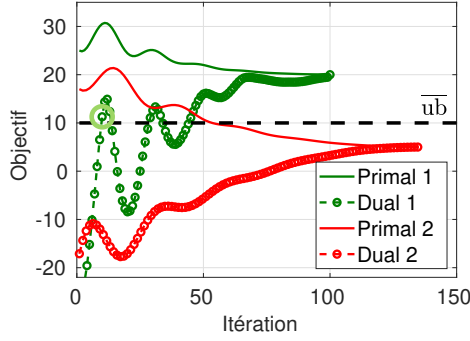


FIGURE 2 – Rejet précoce : il faut ici 100 itérations de l’algorithme d’optimisation de (\mathcal{P}_{2+1}^N) pour rejeter le nœud 1 *via* la valeur du primal, contre seulement 10 *via* celle du dual. Le nœud 2 a ici une borne inférieure trop faible pour être rejeté.

L’approche standard pour rejeter un nœud consiste à résoudre exactement (\mathcal{P}_{2+1}^N) afin de garantir la borne lb_N . Nous proposons d’utiliser la dualité convexe afin de réduire le coût calculatoire permettant de prouver la sous-optimalité d’un nœud. On peut montrer [6] que le problème dual associé à (\mathcal{P}_{2+1}^N) est :

$$\max_{w \in \mathbb{R}^N} D(w) := -f^*(w) - g^*(-A^T w), \quad (3)$$

avec $f^*(w) := \frac{1}{2}(\|w + y\|_2^2 - \|y\|_2^2)$

$$\text{et } g^*(u) := M \left(\sum_{i \in \bar{S}} \max(0, |u_i| - \frac{\mu}{M}) + \sum_{i \in S_1} |u_i| \right) - \mu |S_1|$$

En notant $P(x)$ la fonction objectif du problème (\mathcal{P}_{2+1}^N) , le saut de dualité s’écrit $G(x, w) = P(x) - D(w)$. Notons $x^* = \arg \min_x P(x)$ et $w^* = \arg \max_w D(w)$.

Nous utilisons le fait que $P(x) \geq P(x^*) = \text{lb}_N \geq D(w) \forall x, w$. Ainsi, pour un point dual donné w , si on a $D(w) \geq \bar{ub}$, alors $\text{lb}_N \geq \bar{ub}$ donc le nœud peut être rejeté. Notons $\{x^k\}_k$ une séquence d’itérés correspondant à des pénalisations μ^k produite par la méthode homotopique. L’Équation KKT $w^* \in \partial f(Ax^*)$ signifie qu’on a $w^* = Ax^* - y$, et nous calculons $D(w^k)$ sur des points $w^k = \frac{\mu^k}{\mu^k}(Ax^k - y)$, remise à l’échelle par rapport à μ qui s’avère efficace en pratique. Si on a $D(w^k) \geq \bar{ub}$, l’algorithme est arrêté et le nœud rejeté (voir Figure 2).

3 Screening

Les méthodes de *screening* fixent des variables d’un problème avant sa résolution. Pour (\mathcal{P}_{2+1}^N) les valeurs particulières sont $x_i = 0, i \in \bar{S}$, et $x_i = \pm M, i \in \bar{S} \cup S_1$. Nous utilisons le screening gap-safe [9], qui fixe les variables à leur valeur optimale pendant l’optimisation, en exploitant les conditions KKT et le fait que $\forall w, \|w - w^*\|_2 \leq r = \sqrt{2G(x, w)}$ pour (\mathcal{P}_{2+1}^N) [9], donnant les tests suivants :

$$\forall i \in \bar{S}, |A_i^T w| > \frac{\mu}{M} + r \Rightarrow x_i^* = -M \text{sign}(A_i^T w^*) \quad (4a)$$

$$\forall i \in \bar{S}, |A_i^T w| < \frac{\mu}{M} - r \Rightarrow x_i^* = 0 \quad (4b)$$

$$\forall i \in S_1, |A_i^T w| > r \Rightarrow x_i^* = -M \text{sign}(A_i^T w^*) \quad (4c)$$

Lorsque le saut de dualité diminue, la région sûre rétrécit, et les tests de screening deviennent plus performants.

4 Evaluation expérimentale

Nous considérons des problèmes synthétiques à l’instar de [7, 2, 3] les lignes A_i de la matrice A sont tirées aléatoirement par $A_i \sim \mathcal{N}(0, \Sigma)$, avec Σ défini par $\Sigma_{ij} = \rho^{|i-j|}$ et les colonnes de A sont normalisées en norme ℓ_2 . Nous considérons deux niveaux de corrélation $\rho \in \{0.8, 0.9\}$. La dimension de y est $N = 500$, celle de x est $Q = 100$. Malgré leur petite taille, ces problèmes sont difficiles à résoudre [2]. Dix instances pour chaque ρ sont construites en générant des vecteurs parcimonieux x_{vrai} de support aléatoire avec $K = 9$ composantes égales à 1. Les données correspondantes sont générées par $y = Ax_{\text{vrai}} + \xi$, avec le vecteur de bruit $\xi \sim \mathcal{N}(0, \sigma^2 I)$, σ contrôlant le niveau de bruit dans les données, fixé à $\|Ax_{\text{vrai}}\|_2^2 / (N\sigma^2) = 6$. La valeur de M est fixée empiriquement et sans a priori sur les amplitudes de la solution x_{vrai} par $M = 1.1 \max_i |A_i^T y|$ [5]. Le paramètre μ du problème (\mathcal{P}_{2+0}) est réglé empiriquement pour chaque \hat{x} de sorte que $\|\hat{x}\|_0 = \|x_{\text{vrai}}\|_0$. Le code de calcul `mimosa-solver` (et les données test `mimosa-oms-dataset`) sont disponibles ici : <https://gitlab.univ-nantes.fr/samain-g/>.

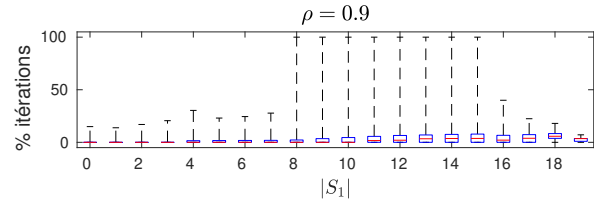
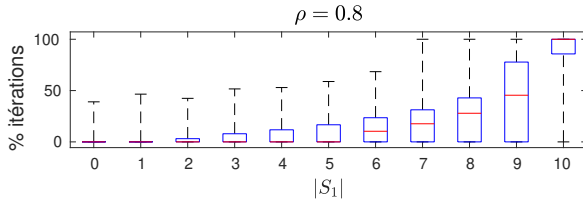
4.1 Rejet précoce

Nous évaluons ici la performance du rejet précoce décrit en Section 2. Pour chaque instance, à chaque nœud N , on considère I_N^p le nombre d’itérations pour résoudre le problème (\mathcal{P}_{2+1}^N) , le nombre d’itérations I_N^d pour obtenir $D(w) \geq \bar{ub}$, et le gain relatif en itérations économisées, $(I_N^p - I_N^d) / I_N^p$.

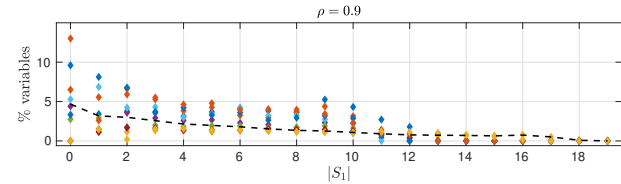
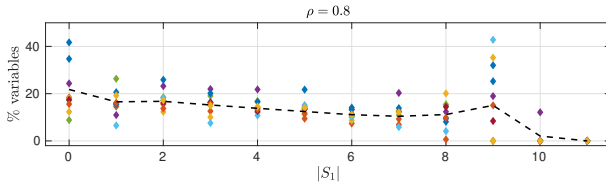
La Figure 3A (gauche) montre les gains obtenus avec $\rho = 0.8$. Si le bénéfice est très faible pour de petites valeurs de $|S_1|$, il augmente quand $|S_1| \approx K$. Ainsi, pour $|S_1| = K$, plus de 50% des itérations sont épargnées sur 50% des instances. En effet, notre stratégie de branchement (voir Section 1.3) privilégie la sélection, dans S_1 , des variables les plus contributives au modèle, améliorant ainsi la qualité des relaxations lorsque S_1 grandit. La Figure 3A (droite) montre les résultats pour $\rho = 0.9$. Malgré une tendance similaire, la performance chute, avec seulement quelques pourcents d’économisés. Cette évolution est attendue : il est bien connu que la qualité de la relaxation ℓ_1 des problèmes ℓ_0 se dégrade pour les problèmes plus difficiles. Lorsque ρ augmente, la plupart des relaxations calculées sont inefficaces (voir le nœud 2 en Figure 2). Dans ce cas, le rejet précoce est inopérant. Notons que, lorsque $|S_1| \geq K$, quelques nœuds sont quand même élagués dès la première itération de la méthode homotopique (gain relatif de 100%).

4.2 Screening

Nous étudions maintenant l’efficacité de la méthode de screening décrite en Section 3. Pour chaque instance générée I , on



A. Rejet précoce : statistiques sur les itérations économisées selon le nombre de variables non nulles dans le nœud. Chaque boîte délimite les premier et troisième quartiles et le trait rouge central indique la médiane. Les tirets noirs marquent les valeurs extrêmes.



B. Screening : réduction du nombre de variables selon le nombre de variables non nulles dans le nœud, pour chaque instance (marqueurs colorés) et en moyenne (ligne discontinue).

FIGURE 3 – Performances obtenues pour les accélérations reposant sur le rejet précoce (haut) et le screening (bas), pour deux niveaux de corrélation de la matrice A .

considère le nombre de nœuds ayant une cardinalité de S_1 donnée, noté $N_I^{|S_1|}$, et $V_I^{|S_1|}$ le nombre total de variables fixées par le screening dans ces nœuds. On définit alors le taux moyen de variables fixées, par instance : $\mathcal{F}_I^{|S_1|} = V_I^{|S_1|}/N_I^{|S_1|}$ et globalement : $\overline{\mathcal{F}}^{|S_1|} = \sum_I V_I^{|S_1|} / \sum_I N_I^{|S_1|}$. Dans nos expériences, seuls des cas de variables $x_i, i \in S$ fixées à 0 ont été rencontrés.

Les résultats sont présentés en Figure 3B. À l'inverse du rejet précoce, l'efficacité du screening tend à diminuer avec le nombre de variables non-nulles fixées dans le nœud.

Conclusion

Deux techniques ont été explorées pour accélérer le calcul des relaxations impliquées dans chaque nœud d'un algorithme branch-and-bound dédié à la régression parcimonieuse, d'intérêt complémentaire : le rejet précoce s'est avéré plus efficace dans les nœuds où le support de la solution courante est déjà bien rempli, tandis que le screening fonctionne mieux pour des petits supports. Sans surprise, les performances de ces techniques se dégradent lorsque les problèmes sont plus difficiles.

Afin d'intégrer efficacement ces techniques, et que le gain résultant compense au mieux le (léger) surcoût calculatoire induit, une question ouverte est de construire des heuristiques décidant quand appliquer le rejet précoce et/ou le screening, afin de ne les utiliser que pour les nœuds favorables.

Références

[1] F. BACH, R. JENATTON, J. MAIRAL et G. OBOZINSKI. *Optimization with Sparsity-Inducing Penalties*. now publishers Inc, 2011.

[2] R. BEN MHENNI, S. BOURGUIGNON et J. NININ. Global optimization for sparse solution of least squares problems. *Optimization Methods and Software*, 2021.

[3] D. BERTSIMAS, A. KING et R. MAZUMDER. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 2016.

[4] D. BERTSIMAS et R. SHIODA. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 2009.

[5] S. BOURGUIGNON, J. NININ, H. CARFANTAN et M. MONGEAU. Exact sparse approximation problems via mixed-integer programming : formulations and computational performance. *IEEE Transactions on Signal Processing*, 2016.

[6] T. GUYARD, C. HERZET et C. ELVIRA. Node-screening tests for ℓ_0 -penalized least-squares problem with supplementary material. working paper or preprint, oct. 2021. URL : <https://arxiv.org/abs/2110.07308v2>.

[7] H. HAZIMEH, R. MAZUMDER et A. SAAB. Sparse regression at scale : branch-and-bound rooted in first-order optimization. *Math. Program.*, 2021.

[8] B. K. NATARAJAN. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 1995.

[9] E. NDIAYE, O. FERCOQ, A. GRAMFORT et J. SALMON. Gap safe screening rules for sparsity enforcing penalties. *NIPS*, 2016.

[10] E. SOUBIES, L. BLANC-FÉRAUD et G. AUBERT. A unified view of exact continuous penalties for ℓ_2 - ℓ_0 minimization. *SIAM Journal on Optimization*, 2017.