



HAL
open science

Global optimization for sparse solution of least squares problems

Ramzi Ben Mhenni, Sébastien Bourguignon, Jordan Ninin

► **To cite this version:**

Ramzi Ben Mhenni, Sébastien Bourguignon, Jordan Ninin. Global optimization for sparse solution of least squares problems. Optimization Methods and Software, inPress, 10.1080/10556788.2021.1977809 . hal-03419540v1

HAL Id: hal-03419540

<https://ensta-bretagne.hal.science/hal-03419540v1>

Submitted on 8 Nov 2021 (v1), last revised 19 Nov 2021 (v2)

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.



Global Optimization for Sparse Solution of Least Squares Problems

Ramzi Ben Mhenni, Sébastien Bourguignon, Jordan Ninin

► **To cite this version:**

Ramzi Ben Mhenni, Sébastien Bourguignon, Jordan Ninin. Global Optimization for Sparse Solution of Least Squares Problems. 2019. hal-02066368

HAL Id: hal-02066368

<https://hal.archives-ouvertes.fr/hal-02066368>

Preprint submitted on 13 Mar 2019

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.

Global Optimization for Sparse Solution of Least Squares Problems

Ramzi BEN MHENNI^{a*}, Sebastien BOURGUIGNON^a and Jordan NININ^b

^aLS2N, CNRS UMR 6004, École Centrale de Nantes, 44321, Nantes Cedex 3, France;

^bLab-STICC, CNRS UMR 6285, ENSTA Bretagne, Brest Cedex, France

ARTICLE HISTORY

Compiled March 13, 2019

ABSTRACT

Finding solutions to least-squares problems with low cardinality has found many applications, including cardinality-constrained portfolio optimization, subset selection in Statistics, and many sparsity-enhancing inverse problems in signal processing. In general, this problem is NP-hard, and most works from a global optimization perspective consider a mixed integer programming (MIP) reformulation with binary variables, whose resolution is performed via branch-and-bound methods. We propose dedicated branch-and-bound algorithms for three possible formulations: cardinality-constrained and cardinality-penalized least-squares, and cardinality minimization under quadratic constraints. We show that the continuous relaxation problems involved at each node of the search tree are ℓ_1 -norm-based optimization problems. A dedicated algorithm is built, based on the homotopy continuation principle, which efficiently computes the relaxed solutions for the three kinds of problems. The performance of the resulting global optimization procedure is then shown to compete with or improve over the CPLEX MIP solvers, especially for problems involving quadratic constraints. The proposed strategies are able to exactly solve some problems involving 500 to 1 000 unknowns in less than 1 000 seconds, for which CPLEX mostly fails.

KEYWORDS

Sparse approximation; Cardinality constraint; Branch-and-bound; Continuous relaxation; Homotopy continuation.

1. Introduction

We are interested in solving optimization problems mixing a quadratic data adjustment term and a sparsity measure. Such problems arise in many application fields, among which portfolio optimization (see, *e.g.*, [2, 3, 8, 18, 27]), sparse regularization for inverse problems [6, 13, 20, 28, 31] and compressed sensing [5, 14], and variable or subset selection in statistics [11, 21, 24, 29]. In operations research, many works addressed the cardinality-constrained problem:

$$\mathcal{P}_{2/0} : \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \text{ subject to } \|\mathbf{x}\|_0 \leq K,$$

where $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$ with usually $n > m$, $\|\mathbf{x}\|_0 = \text{Card}\{i | x_i \neq 0\}$ (which will be called the ℓ_0 -“norm” in this paper) and $K \in \mathbb{N}$ is a given cardinality that is fixed *a priori*. In some applications, however, one may prefer solving the error-constrained problem:

$$\mathcal{P}_{0/2} : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \epsilon.$$

*CONTACT R. BEN MHENNI. Author. Email: ramzi.benmhenni@ls2n.fr

Such a formulation may be more relevant in signal processing and statistical problems, where parameter $\epsilon \geq 0$ controls the approximation level (or the prediction accuracy), that can be tuned according to prior knowledge on the data. One is then interested in finding the sparsest approximation compatible with some given noise level (or some prediction quality requirement). Finally, the penalized problem:

$$\mathcal{P}_{2+0} : \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu \|\mathbf{x}\|_0,$$

where $\mu > 0$ trades off between approximation error and sparsity, is also encountered in the field of inverse problems, *e.g.*, for Geophysics [20] or ultrasonic non-destructive testing [23, 31]. In the Bayesian statistical framework, the ℓ_0 -norm penalization term corresponds to a Bernoulli-Gaussian prior assumption on the unknown components in \mathbf{x} , and parameter μ then depends on the noise variance and the expected rate of non-zero values in \mathbf{x} [28].

Due to the discrete nature of the ℓ_0 norm, these three problems are essentially combinatorial. Therefore, many works in signal processing and statistics proposed local optimization methods. On the one hand, we find the substitution of the ℓ_0 norm by the ℓ_1 norm, for which the three problems resort to convex, non-smooth optimization. In statistics, the ℓ_1 -norm version of problem $\mathcal{P}_{2/0}$ is known as the Least Absolute Shrinkage and Selection Operator (LASSO [29]). On the other hand, greedy algorithms (*e. g.*, [7, 25, 28]), which can similarly solve the three forms above, have been developed. In very special cases as those addressed within the compressed sensing theory [14], it has been shown that the ℓ_1 -norm approach or greedy methods may give the solution to the ℓ_0 problem. However, in general, these approaches do not provide the global solution to the initial problem with the ℓ_0 norm [4], and the three considered optimization problems are usually NP-hard [22].

Dedicated global optimization algorithms have been proposed for solving $\mathcal{P}_{2/0}$, which is a cardinality-constrained quadratic program (CCQP), in the context of sparse portfolio selection and subset selection problems. To our knowledge, Bienstock was the first to propose a specific branch-and-cut algorithm for such problems (including positivity constraints) [3]. Continuous relaxation problems involved at each iteration were solved *via* a specific convex quadratic programming algorithm. In [2], Bertsimas and Shioda extended this work using Lemke's pivoting method to solve the continuous relaxation. When the matrix involved in the quadratic term is the sum of a diagonal positive matrix and a positive definite one, branch-and-bound techniques using perspective reformulation [15], Lagrangian relaxation [8, 18, 27] or geometric approaches [16], were shown to give tighter lower bounds than the continuous relaxation. In many problems of the form $\mathcal{P}_{2/0}$, however, variables involved in the columns of matrix \mathbf{A} (explaining variables in statistics, dictionary atoms in sparse approximation) are often highly correlated, so that matrix $\mathbf{A}^T \mathbf{A}$ is ill-conditioned and such a decomposition is not possible. Much fewer works could be found that address problems $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} from a global optimization perspective. We remark, however, that $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} both involve a quadratic cost function, therefore dedicated optimization methods developed for the former may also be adapted to the latter. This is not the case for $\mathcal{P}_{0/2}$, for which quadratic constraints may require adapted methodology.

In [4], the three problems were reformulated as mixed-integer programs (MIPs). Resolution was performed with the CPLEX solver. It was experimentally shown that exact resolution of difficult sparse approximation problems was possible in practice for small-size problems, but that computing times for solving $\mathcal{P}_{0/2}$ were far higher than for the two other ones. In this paper, we propose a branch-and-bound resolution strategy for such problems. Following the works in [2, 3], our motivation lies on the fact that sparsity-enhancing least-squares problems are very specific MIPs, that could be advantageously solved by dedicated implementation. In particular, we show that all continuous relaxation problems involved in the resolution are particular forms of ℓ_1 -norm-based

problems, for which we build a dedicated continuous optimization method, based on the homotopy principle [10, 11, 24]. Our algorithm is able to solve the relaxation problems involved in any of the three problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} . Therefore, these three problems can be solved exactly by our procedure, with similar computational burden.

The paper is organized as follows. In Section 2, we study the structure of continuous relaxation problems involved at each node of a branch-and-bound algorithm solving $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} , which are reformulated as ℓ_1 -norm-based problems. In Section 3, we build a dedicated algorithm for such problems, based on the homotopy principle. In Section 4, the performance of our method is evaluated through numerical experiments, on both sparse deconvolution and subset selection problems. The discussion in Section 5 closes the paper.

2. Continuous relaxations within a branch-and-bound algorithm

In this section, we consider the MIP reformulations of problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} , and we study/reformulate the continuous relaxation problems involved at each node of the branch-and-bound strategy.

We introduce binary decision variables b_i such that $b_i = 0 \Leftrightarrow x_i = 0$. We use the classical *bigM* formulation: assuming that solutions of interest satisfy $\forall i, |x_i| \leq M$ for some known value M , the former logical constraint reads $-Mb_i \leq x_i \leq Mb_i$. Trivial extensions of such assumption write $-M_i^{\text{inf}} \leq x_i \leq M_i^{\text{sup}}$, with $M_i^{\text{inf}}, M_i^{\text{sup}} \geq 0$, but in the following we keep $|x_i| \leq M$ to simplify notations. The three problems can then be reformulated as the standard MIPs given in Table 1 (see for example [4]).

Problem	MIP Reformulation
$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \frac{1}{2} \ \mathbf{y} - \mathbf{Ax}\ _2^2 \\ \text{s. t.} \quad & \ \mathbf{x}\ _0 \leq K \\ & \ \mathbf{x}\ _\infty \leq M \end{aligned}$	$\begin{aligned} \min_{\mathbf{b} \in \{0,1\}^n, \mathbf{x} \in \mathbb{R}^n} \quad & \frac{1}{2} \ \mathbf{y} - \mathbf{Ax}\ _2^2 \\ \text{s. t.} \quad & \sum_{i=1}^n b_i \leq K \\ & \mathbf{x} \leq M\mathbf{b} \end{aligned}$
$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \ \mathbf{x}\ _0 \\ \text{s. t.} \quad & \frac{1}{2} \ \mathbf{y} - \mathbf{Ax}\ _2^2 \leq \epsilon \\ & \ \mathbf{x}\ _\infty \leq M \end{aligned}$	$\begin{aligned} \min_{\mathbf{b} \in \{0,1\}^n, \mathbf{x} \in \mathbb{R}^n} \quad & \sum_{i=1}^n b_i \\ \text{s. t.} \quad & \frac{1}{2} \ \mathbf{y} - \mathbf{Ax}\ _2^2 \leq \epsilon \\ & \mathbf{x} \leq M\mathbf{b} \end{aligned}$
$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \frac{1}{2} \ \mathbf{y} - \mathbf{Ax}\ _2^2 + \mu \ \mathbf{x}\ _0 \\ \text{s. t.} \quad & \ \mathbf{x}\ _\infty \leq M \end{aligned}$	$\begin{aligned} \min_{\mathbf{b} \in \{0,1\}^n, \mathbf{x} \in \mathbb{R}^n} \quad & \frac{1}{2} \ \mathbf{y} - \mathbf{Ax}\ _2^2 + \mu \sum_{i=1}^n b_i \\ \text{s. t.} \quad & \mathbf{x} \leq M\mathbf{b} \end{aligned}$

Table 1. Initial problems (left) and their MIP reformulations (right).

We consider a resolution strategy based on a branch-and-bound procedure, as adopted by most MIP solvers. The initial problem defines a root node. At each iteration of the algorithm, one node is selected from the list of subproblems that have not been processed yet, and a lower bound for the node is computed *via* the continuous relaxation of the binary variables. If this bound is greater than the current upper bound, defined by the best feasible solution found, then the subproblem is discarded. Otherwise, two children of this node are built through the addition of constraints fixing one of the relaxed variables b_i to 0 and 1. The two new nodes are then added to the list. In the following, we focus on the node evaluation step by continuous relaxation.

2.1. Continuous relaxation of the root node

At the root node, no decision has been made concerning any binary variable. The continuous relaxation of binary variables in $\mathcal{P}_{2/0}$ then reads:

$$\mathcal{P}_{2/0}^R : \min_{\mathbf{b} \in [0,1]^n, \mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 \quad \text{s. t.} \quad \begin{cases} \sum_{i=1}^n b_i \leq K \\ |\mathbf{x}| \leq M\mathbf{b} \end{cases} .$$

Proposition 1. Let $\mathcal{P}_{2/1}$ be the following problem:

$$\mathcal{P}_{2/1} : \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 \quad \text{s. t.} \quad \begin{cases} \|\mathbf{x}\|_1 \leq KM \\ \|\mathbf{x}\|_\infty \leq M \end{cases} .$$

Then, $\mathcal{P}_{2/0}^R$ and $\mathcal{P}_{2/1}$ have the same minimum value.

Proof. Let $(\mathbf{b}^{*R}, \mathbf{x}^{*R})$ be a minimizer of $\mathcal{P}_{2/0}^R$ and let \mathbf{x}^{*1} be a minimizer of $\mathcal{P}_{2/1}$. Let $\tilde{\mathbf{b}} = \frac{1}{M} |\mathbf{x}^{*1}|$. Then, $(\tilde{\mathbf{b}}, \mathbf{x}^{*1})$ is a feasible solution of $\mathcal{P}_{2/0}^R$, therefore $\|\mathbf{y} - \mathbf{Ax}^{*R}\|_2^2 \leq \|\mathbf{y} - \mathbf{Ax}^{*1}\|_2^2$. Conversely, consider $\mathbf{b}_{2/0}^{R'} = \frac{1}{M} |\mathbf{x}^{*R}|$. Then, \mathbf{x}^{*R} is a feasible solution of $\mathcal{P}_{2/1}$ because $\|\mathbf{x}^{*R}\|_1 = M \|\tilde{\mathbf{b}}\|_1 \leq KM$ and $\|\mathbf{x}^{*R}\|_\infty \leq M$. Consequently, $\|\mathbf{y} - \mathbf{Ax}^{*1}\|_2^2 \leq \|\mathbf{y} - \mathbf{Ax}^{*R}\|_2^2$. \square

We note that the same result was given in [2, 3] for problems with positivity constraints, and in [1] in our case.

A similar result holds for $\mathcal{P}_{0/2}^R$, the continuous relaxation of $\mathcal{P}_{0/2}$:

$$\mathcal{P}_{0/2}^R : \min_{\mathbf{b} \in [0,1]^n, \mathbf{x} \in \mathbb{R}^n} \sum_{i=1}^n b_i \quad \text{s. t.} \quad \begin{cases} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 \leq \epsilon \\ |\mathbf{x}| \leq M\mathbf{b} \end{cases}$$

Proposition 2. Let $\mathcal{P}_{1/2}$ be the following problem:

$$\mathcal{P}_{1/2} : \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{x}\|_1 \quad \text{s. t.} \quad \begin{cases} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 \leq \epsilon \\ \|\mathbf{x}\|_\infty \leq M \end{cases} .$$

Then, $\mathcal{P}_{0/2}^R$ and $\mathcal{P}_{1/2}$ have the same minimum value.

Proof. Let $(\mathbf{b}^{*R}, \mathbf{x}^{*R})$ an optimal solution of $\mathcal{P}_{0/2}^R$. Suppose that $|x_i^{*R}| < Mb_i^{*R}$ for some component i . Consider $\tilde{\mathbf{b}} = \frac{1}{M} |\mathbf{x}^{*R}|$, such that $\tilde{b}_i < b_i^{*R}$. Then, $(\tilde{\mathbf{b}}, \mathbf{x}^{*R})$ is a feasible solution for $\mathcal{P}_{0/2}^R$ with $\sum_{i=1}^n \tilde{b}_i < \sum_{i=1}^n b_i^{*R}$, which contradicts the definition of $(\mathbf{b}^{*R}, \mathbf{x}^{*R})$. Indeed, $|\mathbf{x}^{*R}| = M\mathbf{b}^{*R}$ and $\|\mathbf{x}^{*R}\|_1 = M \sum_{i=1}^n b_i^{*R}$. Thus $(\mathbf{b}^{*R}, \mathbf{x}^{*R})$ is an optimal solution of $\mathcal{P}_{1/2}$. \square

Finally, consider the continuous relaxation of binary variables in the penalized problem \mathcal{P}_{2+0} as follows:

$$\mathcal{P}_{2+0}^R : \min_{\mathbf{b} \in [0,1]^n, \mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \mu \sum_{i=1}^n b_i \quad \text{s. t.} \quad |\mathbf{x}| \leq M\mathbf{b} .$$

Proposition 3. Let \mathcal{P}_{2+1} be the following problem:

$$\mathcal{P}_{2+1} : \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \frac{\mu}{M} \|\mathbf{x}\|_1 \quad \text{s. t.} \quad \|\mathbf{x}\|_\infty \leq M.$$

Then, \mathcal{P}_{2+0}^R and \mathcal{P}_{2+1} have the same minimum value.

Proof. The proof is similar to that of Proposition 2. \square

The three continuously relaxed problems can therefore be reformulated without binary variables, and correspond to ℓ_1 optimization problems with box constraints.

2.2. Continuous relaxation in the branch-and-bound algorithm

We now consider a given node in the branch-and-bound algorithm and the corresponding continuous relaxation sub-problem. Let S^0 (respectively, S^1) denote the index set of binary variables that are set to 0 (respectively, to 1), and let \bar{S} collect all remaining indices:

$$\begin{aligned} \forall i \in S^1, & \quad b_i = 1 \text{ and } |x_i| \leq M \\ \forall i \in S^0, & \quad b_i = 0 \text{ and } x_i = 0 \\ \forall i \in \bar{S}, & \quad b_i \in [0, 1] \text{ and } |x_i| \leq Mb_i \end{aligned}$$

Let A_S denote the sub-matrix formed by all columns of matrix A indexed by S . Similarly, u_S denotes the corresponding sub-vector of u .

For problem $\mathcal{P}_{2/0}$, the continuous relaxation of variables $b_{\bar{S}}$ in the corresponding sub-problem reduces to:

$$\mathcal{Q}_{2/0}^R : \quad \min_{\substack{\mathbf{x}_{S^1} \in \mathbb{R}^{n_1} \\ \mathbf{b}_{\bar{S}} \in [0, 1]^{\bar{n}} \\ \mathbf{x}_{\bar{S}} \in \mathbb{R}^{\bar{n}}}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \quad \text{s. t.} \quad \begin{cases} \sum_{i \in \bar{S}} b_i \leq K - n_1 \\ \|\mathbf{x}_{\bar{S}}\| \leq M \mathbf{b}_{\bar{S}} \\ \|\mathbf{x}_{S^1}\|_\infty \leq M \end{cases},$$

where n_1 and \bar{n} denote the size of S^1 and \bar{S} . Then, similarly to the developments in Section 2.1, one can show that $\mathcal{Q}_{2/0}^R$ and $\mathcal{Q}_{2/1}$ have the same minimum value:

$$\mathcal{Q}_{2/1} : \quad \min_{\mathbf{x}_{S^1}, \mathbf{x}_{\bar{S}}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}\|_2^2 \quad \text{s. t.} \quad \begin{cases} \|\mathbf{x}_{\bar{S}}\|_1 \leq M(K - n_1) \\ \|\mathbf{x}_{\bar{S}}\|_\infty \leq M \\ \|\mathbf{x}_{S^1}\|_\infty \leq M \end{cases}.$$

Applying a similar reasoning to the two other formulations, we finally obtain the equivalent problems summarized in Table 2.

Two important comments arise from these results:

- Whatever the formulation (constrained or penalized), all continuously relaxed subproblems involved in the evaluation of the nodes in the branch-and-bound algorithm can be reformulated without binary variables.
- They all reduce to optimization problems mixing a least-squares function, ℓ_1 -norm terms involving only a part of the variables, and box constraints.

In Section 3, we build a dedicated algorithm which solves the three problems $\mathcal{Q}_{2/1}$, $\mathcal{Q}_{1/2}$ and \mathcal{Q}_{2+1} .

3. A dedicated homotopy continuation algorithm for relaxed problems

Optimization involving quadratic misfits and ℓ_1 -norm terms has been a very active field of research in the past ten years. Many dedicated convex, non-smooth, optimization algorithms have been

Continuous relaxation problem	Equivalent ℓ_1 -norm-based problem
$\mathcal{Q}_{2/0}^R : \min_{\mathbf{b} \in [0,1]^n, \mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \ \mathbf{y} - \mathbf{A}\mathbf{x}\ _2^2$ $\text{s.t.} \quad \sum_{i=1}^n b_i \leq K$ $\ \mathbf{x}\ \leq M\mathbf{b}$ $\mathbf{b}_{S^1} = 1$ $\mathbf{b}_{S^0} = 0$	$\mathcal{Q}_{2/1} : \min_{\mathbf{x}_{S^1}, \mathbf{x}_{S^0}} \frac{1}{2} \ \mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1} - \mathbf{A}_{S^0}\mathbf{x}_{S^0}\ _2^2$ $\text{s.t.} \quad \ \mathbf{x}_{S^0}\ _1 \leq M(K - n_1)$ $\ \mathbf{x}_{S^0}\ _\infty \leq M$ $\ \mathbf{x}_{S^1}\ _\infty \leq M$
$\mathcal{Q}_{0/2}^R : \min_{\mathbf{b} \in [0,1]^n, \mathbf{x} \in \mathbb{R}^n} \sum_{i=1}^n b_i$ $\text{s.t.} \quad \frac{1}{2} \ \mathbf{y} - \mathbf{A}\mathbf{x}\ _2^2 \leq \epsilon$ $\ \mathbf{x}\ \leq M\mathbf{b}$ $\mathbf{b}_{S^1} = 1$ $\mathbf{b}_{S^0} = 0$	$\mathcal{Q}_{1/2} : \min_{\mathbf{x}_{S^1}, \mathbf{x}_{S^0}} \frac{1}{M} \ \mathbf{x}_{S^0}\ _1 + n_1$ $\text{s.t.} \quad \frac{1}{2} \ \mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1} - \mathbf{A}_{S^0}\mathbf{x}_{S^0}\ _2^2 \leq \epsilon$ $\ \mathbf{x}_{S^0}\ _\infty \leq M$ $\ \mathbf{x}_{S^1}\ _\infty \leq M$
$\mathcal{Q}_{2+0}^R : \min_{\mathbf{b} \in [0,1]^n, \mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \ \mathbf{y} - \mathbf{A}\mathbf{x}\ _2^2 + \mu \sum_{i=1}^n b_i$ $\text{s.t.} \quad \ \mathbf{x}\ \leq M\mathbf{b}$ $\mathbf{b}_{S^1} = 1$ $\mathbf{b}_{S^0} = 0$	$\mathcal{Q}_{2+1} : \min_{\mathbf{x}_{S^1}, \mathbf{x}_{S^0}} \frac{1}{2} \ \mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1} - \mathbf{A}_{S^0}\mathbf{x}_{S^0}\ _2^2$ $+ \frac{\mu}{M} \ \mathbf{x}_{S^0}\ _1 + \mu n_1$ $\text{s.t.} \quad \ \mathbf{x}_{S^0}\ _\infty \leq M$ $\ \mathbf{x}_{S^1}\ _\infty \leq M$

Table 2. Continuous relaxation problems at any node in the branch-and-bound procedure (left), and equivalent problem without binary variables involving the ℓ_1 norm (right), for the three considered formulations.

developed (see for example [14, 30] and references therein), for solving problems:

$$\mathcal{P}_1^\tau : \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}\|_1 \leq \tau;$$

$$\mathcal{P}_1^\epsilon : \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t.} \quad \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \epsilon;$$

$$\mathcal{P}_1^\mu : \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu \|\mathbf{x}\|_1.$$

Most works addressed the penalized form, which resorts to unconstrained optimization. Since the two objectives are convex, the three problems are equivalent, that is, for any $\tau \geq 0$, there exists $\mu^{(\tau)} \geq 0$ such that \mathcal{P}_1^τ and $\mathcal{P}_1^{\mu^{(\tau)}}$ have the same solution, and reciprocally. Similarly, for any $\epsilon \geq 0$, there exists $\mu^{(\epsilon)} \geq 0$ such that \mathcal{P}_1^ϵ and $\mathcal{P}_1^{\mu^{(\epsilon)}}$ are equivalent. However, in general, there is no explicit mapping between the three parameters.

In this section, we propose to solve the three problems $\mathcal{Q}_{2/1}$, $\mathcal{Q}_{1/2}$ and \mathcal{Q}_{2+1} by a generalization of the homotopy continuation principle [10, 11, 24]. This choice is motivated by the following reasons:

- First and foremost, the same algorithm can solve the three problems with the same computational efficiency.
- Then, it can naturally incorporate specificities of the problems $\mathcal{Q}_{2/1}$, $\mathcal{Q}_{1/2}$ and \mathcal{Q}_{2+1} , such as box constraints and partial ℓ_1 -norm terms.
- In addition, it is an exact algorithm, for which the solution is obtained in a finite number of iterations.
- Last, its efficiency increases with the sparsity level in the sought solution.

The homotopy continuation method considers the penalized form \mathcal{P}_1^μ , and exploits the fact that the solution path is piecewise linear as a function of μ . Starting from $\mu^{(0)} = \|\mathbf{A}^T \mathbf{y}\|_\infty$ (such that the solution is identically zero $\forall \mu > \mu^{(0)}$), it iteratively computes all solutions by continuously

decreasing parameter μ until the target value is reached. Therefore, it can similarly solve problems \mathcal{P}_1^τ or \mathcal{P}_1^ϵ , by stopping when the corresponding value of τ or ϵ is reached, respectively. Figure 1 shows a typical solution path (left part) and the corresponding evolution on the Pareto front (right part).

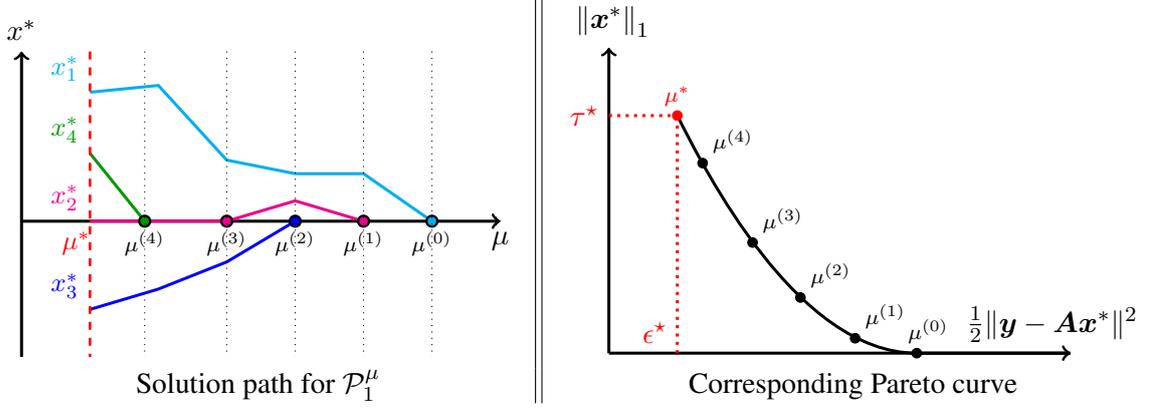


Figure 1. Homotopy method: example of solution path $\mathbf{x}^*(\mu) = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \mu \|\mathbf{x}\|_1$ as a function of μ (left), and corresponding set $(\frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}^*\|_2^2, \|\mathbf{x}^*\|_1)$ as a function of μ (right).

In the following, we generalize the homotopy method to the class of problems $\mathcal{Q}_{2/1}$, $\mathcal{Q}_{1/2}$ and \mathcal{Q}_{2+1} , where the ℓ_1 norm only operates on some of the variables and box constraints are included. Note that the homotopy method with box constraints was recently proposed in [19], which also established convergence proofs. Including partial ℓ_1 -norm penalization mainly impacts initialization and additional tests to be performed at each iteration. We first derive optimality conditions in § 3.1. Then, the algorithm is built in § 3.2. Implementation details are discussed in § 3.4.

3.1. Optimality conditions

We first focus on problem \mathcal{Q}_{2+1} in Table 2, that is, the relaxed problem involved in the cardinality-penalized form. We remove from the problem the variables \mathbf{x}_{S^0} which are fixed to zeros and consider equivalently the optimization problem of the form:

$$\begin{aligned} \min_{\mathbf{x}} F(\mathbf{x}) &:= J(\mathbf{x}) + \lambda h(\mathbf{x}) \text{ s. t. } g_i(\mathbf{x}) \leq 0 \quad \forall i = 1, \dots, n, \\ \text{with} \quad J(\mathbf{x}) &:= \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S^c} \mathbf{x}_{S^c} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}\|_2^2 \\ h(\mathbf{x}) &:= \|\mathbf{x}_{S^c}\|_1 \\ g_i(\mathbf{x}) &:= |x_i| - M \end{aligned} \quad (1)$$

The Lagrangian function is:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\pi}) = J(\mathbf{x}) + \lambda h(\mathbf{x}) + \sum_{i=1}^n \pi_i g_i(\mathbf{x}),$$

with $\boldsymbol{\pi} \in \mathbb{R}^n$ the vector of Lagrange multipliers associated with the bound constraints $g_i(\mathbf{x}) \leq 0$. The function J is differentiable with $\nabla J(\mathbf{x}) = -\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x})$. The subdifferentials of $h(\mathbf{x}) =$

$\|\mathbf{x}_{\bar{S}}\|_1$ and of $g_i(\mathbf{x}) = |x_i| - M$ are respectively:

$$\partial h(\mathbf{x}) = \left\{ \mathbf{z} \in \mathbb{R}^n \left| \begin{array}{ll} z_i = 0 & \text{if } i \in S^1 \cup S^0 \\ z_i = \text{sgn}(x_i) & \text{if } i \in \bar{S} \text{ and } x_i \neq 0 \\ z_i \in [-1, 1] & \text{if } i \in \bar{S} \text{ and } x_i = 0 \end{array} \right. \right\} \quad (2)$$

and

$$\partial g_i(\mathbf{x}) = \left\{ \mathbf{z} \in \mathbb{R}^n \left| \begin{array}{ll} z_j = 0 & \text{for } j \neq i \\ z_i = \text{sgn}(x_i) & \text{if } x_i \neq 0 \\ z_i \in [-1, 1] & \text{if } x_i = 0 \end{array} \right. \right\}. \quad (3)$$

The vector \mathbf{x}^* is optimal if and only if there exists $\boldsymbol{\pi}$ such that $(\mathbf{x}^*, \boldsymbol{\pi})$ satisfies the Karush-Kuhn-Tucker optimality conditions applied to continuous convex non-differentiable functions (see for example [26]):

$$\begin{cases} 0 \in -\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^*) + \lambda \partial h(\mathbf{x}^*) + \partial \sum_{i=1}^n \pi_i g_i(\mathbf{x}^*) & (4) \\ g_i(\mathbf{x}^*) \leq 0 \quad \forall i = 1, \dots, n & (5) \\ \pi_i \geq 0 \quad \forall i = 1, \dots, n & (6) \\ \pi_i g_i(\mathbf{x}^*) = 0 \quad \forall i = 1, \dots, n. & (7) \end{cases} \quad (\mathcal{P}_{\text{KKT}})$$

Particular points are those which activate the bound constraint ($x_i^* = \pm M$ for $i \in \bar{S} \cup S^1$) or non-differentiability points ($x_i^* = 0$ for $i \in \bar{S}$). We now split the variable indices into the five possible cases.

- Case 1. Let $\bar{S}_0 = \{i \in \bar{S} \mid |x_i^*| = 0\}$. From equation (7), $\pi_{\bar{S}_0} = 0$ and using equations (2) and (4), optimality conditions of $\mathbf{x}_{\bar{S}_0}^*$ become:

$$|\mathbf{A}_{\bar{S}_0}^T(\mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^*)| < \lambda. \quad (8)$$

- Case 2. Let $\bar{S}_{in} = \{i \in \bar{S} \mid 0 < |x_i^*| < M\}$. From equation (7), $\pi_{\bar{S}_{in}} = 0$ and using equations (2) and (4), optimality conditions of $\mathbf{x}_{\bar{S}_{in}}^*$ become:

$$-\mathbf{A}_{\bar{S}_{in}}^T(\mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^*) + \lambda \text{sgn}(\mathbf{x}_{\bar{S}_{in}}^*) = 0. \quad (9)$$

- Case 3. Let $\bar{S}_{\square} = \{i \in \bar{S} \mid |x_i^*| = M\}$. From equation (6), $\pi_{\bar{S}_{\square}} \geq 0$ and using equations (3) and (4), optimality conditions of $\mathbf{x}_{\bar{S}_{\square}}^*$ become:

$$-\mathbf{A}_{\bar{S}_{\square}}^T(\mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^*) + \lambda \text{sgn}(\mathbf{x}_{\bar{S}_{\square}}^*) + \boldsymbol{\pi}_{\bar{S}_{\square}} \odot \text{sgn}(\mathbf{x}_{\bar{S}_{\square}}^*) = 0, \quad \boldsymbol{\pi}_{\bar{S}_{\square}} \geq 0, \quad (10)$$

where \odot denotes the Hadamard (entrywise) product.

- Case 4. Let $S_{in}^1 = \{i \in S^1 \mid 0 \leq |x_i^*| < M\}$. From equation (7), $\pi_{S_{in}^1} = 0$ and using equations (2) and (4), optimality conditions of $\mathbf{x}_{S_{in}^1}^*$ become:

$$-\mathbf{A}_{S_{in}^1}^T(\mathbf{y} - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^*) = 0. \quad (11)$$

- Case 5. Let $S_{\square}^1 = \{i \in S^1 \mid |x_i^*| = M\}$. From equation (6), $\pi_{S_{\square}^1} \geq 0$ and using equations (3) and (4), optimality conditions of $x_{S_{\square}^1}^*$ become:

$$-A_{S_{\square}^1}^T(\mathbf{y} - A_{S^1}x_{S^1}^* - A_{\bar{S}}x_{\bar{S}}^*) + \pi_{S_{\square}^1} \odot \text{sgn}(x_{S_{\square}^1}^*) = 0, \quad \pi_{S_{\square}^1} \geq 0. \quad (12)$$

Let us remark that equations (9), (10) and (12) concern non-zero variables, therefore the corresponding sign function is well-defined.

3.2. Homotopy continuation algorithm

We now build the homotopy algorithm that solves the problem (1). Let $\mathbf{r} = \mathbf{y} - A_{\bar{S}_{\square}}x_{\bar{S}_{\square}}^* - A_{S_{\square}^1}x_{S_{\square}^1}^*$, where each component in $x_{\bar{S}_{\square}}^*$ and $x_{S_{\square}^1}^*$ equals $\pm M$. Equations (9) and (11) are linear systems in $x_{\bar{S}_{in}}^*$ and $x_{S_{in}^1}^*$, whose the solution is:

$$\begin{cases} x_{\bar{S}_{in}}^* = \left(A_{\bar{S}_{in}}^T (I - P^{S_{in}^1}) A_{\bar{S}_{in}} \right)^{-1} \left(A_{\bar{S}_{in}}^T (I - P^{S_{in}^1}) \mathbf{r} - \lambda \text{sgn}(x_{\bar{S}_{in}}^*) \right), & (13) \\ x_{S_{in}^1}^* = (A_{S_{in}^1}^T A_{S_{in}^1})^{-1} (A_{S_{in}^1}^T \mathbf{r} - A_{S_{in}^1}^T A_{\bar{S}_{in}} x_{\bar{S}_{in}}^*), & (14) \end{cases}$$

where $P^{S_{in}^1} = A_{S_{in}^1} (A_{S_{in}^1}^T A_{S_{in}^1})^{-1} A_{S_{in}^1}^T$ and I is the identity matrix of appropriate size. These equations show that, in a given configuration of the index sets $\{\bar{S}_{in}, S_{in}^1, \bar{S}_{\square}, S_{\square}^1, \bar{S}_0\}$ (which we call the support configuration), the solution of the problem (1) is linear in λ . Indeed, variables in \bar{S}_{\square} and S_{\square}^1 are fixed to $\pm M$, and variables in \bar{S}_0 are zero. The homotopy method then constructs the *solution path* (the set of all solutions as a function of λ) by identifying iteratively the different *breakpoints* that lead to changes in the support configuration. These breakpoints will occur at specific values of λ , for which (at least) one of the conditions in equations (8)–(12) is violated. The algorithm works as follows:

- (1) First, it is clear that as $\lambda \rightarrow +\infty$, ℓ_1 -norm-penalized variables $x_{\bar{S}}$ are zero. In that case, other variables x_{S^1} are found by solving the least-squares problem: $\min_{-M \leq x_{S^1} \leq M} \frac{1}{2} \|\mathbf{y} - A_{S^1} x_{S^1}\|_2^2$.

We note $\mathbf{x}^{(0)}$ the vector defined by

$$\begin{cases} x_{S^1}^{(0)} := \underset{-M \leq x_{S^1} \leq M}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - A_{S^1} x_{S^1}\|_2^2, & (15) \\ x_{\bar{S}}^{(0)} := \mathbf{0}. & (16) \end{cases}$$

Equation (8) shows that $\mathbf{x}^{(0)}$ is the solution of the problem (1) as long as $\lambda \geq \lambda^{(0)}$, with:

$$\lambda^{(0)} := \|A_{\bar{S}}^T(\mathbf{y} - A_{S^1}x_{S^1}^{(0)})\|_{\infty}. \quad (17)$$

- (2) As λ decreases below $\lambda^{(0)}$, indices $j \in \bar{S}$ such that $|a_j^T(\mathbf{y} - A_{S^1}x_{S^1}^{(0)})| = \lambda^{(0)}$ leave \bar{S}_0 to form the new subset \bar{S}_{in} . This new support configuration remains valid for any $\lambda \in [\lambda^{(1)}, \lambda^{(0)}]$, where $\lambda^{(1)}$ defines the next breakpoint, *etc.* A monotonically decreasing sequence $\{\lambda^{(k)}\}_k$ is built iteratively, by testing all possible changes that can occur to the support configuration, and selecting the one(s) corresponding to the smallest decrease in λ . Then, the support configuration is updated, and a new breakpoint in λ is searched. Since the solution path is piecewise linear as a function of λ , the solution $\mathbf{x}^{(k)}$ at the k -th breakpoint

reads:

$$\begin{cases} \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \gamma^{(k)} \mathbf{d}^{(k)} \\ \text{and } \lambda^{(k)} = \lambda^{(k-1)} - \gamma^{(k)}, \end{cases} \quad (18)$$

$$\quad (19)$$

where $\mathbf{d}^{(k)}$ represents the vector of slope changes and $\gamma^{(k)} > 0$ represents the length of interval $[\lambda^{(k)}, \lambda^{(k-1)}]$. From equations (13) and (14), the direction $\mathbf{d}^{(k)}$ is obtained by:

$$\begin{cases} \mathbf{d}_{\bar{S}_{in}}^{(k)} = (\mathbf{A}_{\bar{S}_{in}}^T (\mathbf{I} - \mathbf{P}^{S_{in}^1}) \mathbf{A}_{\bar{S}_{in}})^{-1} \text{sgn}(\mathbf{x}_{\bar{S}_{in}}^{(k-1)}) \\ \mathbf{d}_{S_{in}^1}^{(k)} = -(\mathbf{A}_{S_{in}^1}^T \mathbf{A}_{S_{in}^1})^{-1} \mathbf{A}_{S_{in}^1}^T \mathbf{A}_{\bar{S}_{in}} \mathbf{d}_{\bar{S}_{in}}^{(k)} \\ d_i^{(k)} = 0 \forall i \notin \{\bar{S}_{in} \cup S_{in}^1\}, \end{cases} \quad (20)$$

$$\quad (21)$$

$$\quad (22)$$

where the last equality concerns variables that are fixed to zero or to $\pm M$. The step size $\gamma^{(k)}$ is obtained as the smallest positive value $\gamma > 0$ such that $\mathbf{x}^{(k-1)} + \gamma \mathbf{d}^{(k)}$ reaches a new breakpoint. Five different cases can occur, which are detailed hereafter. We introduce the following notation, where exponentiation by k is removed to simplify it.

$$\mathbf{c} = \mathbf{A}^T (\mathbf{y} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}^{(k-1)} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}^{(k-1)}) \quad \text{and} \quad \mathbf{u} = \mathbf{A}^T (\mathbf{A}_{\bar{S}_{in}} \mathbf{d}_{\bar{S}_{in}}^{(k)} + \mathbf{A}_{S^1} \mathbf{d}_{S^1}^{(k)}).$$

- (a) A component with index $\ell \in \bar{S}_0$ becomes nonzero when equality in equation (8) is reached. Inserting equations (13) and (14) into equation (8), one can show that x_ℓ become positive (respectively, negative) if:

$$\gamma = \frac{\lambda^{(k-1)} + c_\ell}{1 - u_\ell} \left(\text{respectively, when } \gamma = \frac{-\lambda^{(k-1)} + c_\ell}{-1 - u_\ell} \right). \quad (23)$$

- (b) A component with index $\ell \in \bar{S}_{in}$ becomes zero. From equation (18), this occur if:

$$\gamma = \frac{-x_\ell^{(k-1)}}{d_\ell^{(k)}}. \quad (24)$$

- (c) A component with index $\ell \in \bar{S}_{in}$ or S_{in}^1 yields the bound M or $-M$, depending on its current sign. From (18), this occur if:

$$\gamma = \frac{M \text{sgn}(x_\ell^{(k-1)}) - x_\ell^{(k-1)}}{d_\ell^{(k)}}. \quad (25)$$

- (d) The bound constraint for some component with index $\ell \in \bar{S}_\square$ becomes inactive. This may occur when the corresponding Lagrange multiplier $\pi_\ell = 0$ in equation (10), which yields:

$$\gamma = \frac{\text{sgn}(x_\ell^{(k-1)}) \lambda^{(k-1)} - c_\ell}{\text{sgn}(x_\ell^{(k-1)}) - u_\ell}. \quad (26)$$

- (e) The bound constraint for some component with index $\ell \in S_\square^1$ becomes inactive. This may occur when the corresponding Lagrange multiplier $\pi_\ell = 0$ in equation (12), which

yields:

$$\gamma = \frac{-c_\ell}{u_\ell}. \quad (27)$$

The shortest step size $\gamma^{(k)}$ is then defined as the shortest positive step among all possible ones, defined by equations (23)–(27). In theory, $\gamma^{(k)}$ may be obtained by several conditions above simultaneously; should this happen, the support configuration is updated correspondingly.

- (3) The algorithm stops when the target λ , say λ^* , is reached, that is, after iteration k such that $\lambda^* \in [\lambda^{(k)}, \lambda^{(k-1)}]$. Then, the optimal solution is found by:

$$\mathbf{x}^* = \mathbf{x}^{(k-1)} + \gamma^* \mathbf{d}^{(k)}, \quad (28)$$

with $\gamma^* = \lambda^* - \lambda^{(k)}$.

The homotopy algorithm is summarized in Algorithm 1. Figure 2 shows the solution path for a toy example with 5 variables: $\bar{S} = \{1, 2, 3\}$ and $S^1 = \{4, 5\}$.

Algorithm 1 Homotopy algorithm for solving the problem \mathcal{Q}_{2+1} in Table 2, reformulated as the problem (1) with $\lambda = \lambda^*$.

- 1: Set $k = 0$. Initialize $\lambda^{(0)}, \mathbf{x}^{(0)}$ by equations (15)–(17)
 - 2: **while** not stop condition **do**
 - 3: $k \leftarrow k + 1$
 - 4: Update $\mathbf{d}^{(k)}$ by equations (20)–(21)
 - 5: Determine the step size $\gamma^{(k)}$ as the smallest positive value among all cases of equations (23)–(27)
 - 6: Compute accordingly $(\mathbf{x}^{(k)}, \lambda^{(k)})$ by equations (18)–(19)
 - 7: Update index sets $\{\bar{S}_{in}, S_{in}^1, \bar{S}_\square, S_\square^1, \bar{S}_0\}$
 - 8: If $\lambda^{(k)} \leq \lambda^*$, then stop.
 - 9: **return** \mathbf{x}
-

3.3. Solutions to constrained problems $\mathcal{Q}_{2/1}$ and $\mathcal{Q}_{1/2}$

As λ is continuously decreased, the ℓ_1 norm of the penalized variables $\|\mathbf{x}_{\bar{S}}^*\|_1$ continuously increases and the least squares function $\frac{1}{2}\|\mathbf{y} - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^* - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^*\|_2^2$ continuously decreases. Therefore, the homotopy method can also solve the constrained problems $\mathcal{Q}_{2/1}$ and $\mathcal{Q}_{1/2}$ in Table 2. More precisely:

- For $\mathcal{Q}_{2/1}$, iterations stop at the first breakpoint when the ℓ_1 norm of the penalized variables $\|\mathbf{x}_{\bar{S}}^{(k)}\|_1$ exceeds the threshold value $\tau := M(K - n_1)$. Then, in the corresponding interval $[\lambda^{(k)}, \lambda^{(k-1)}]$, the solution is given by equation (28). By construction, there is no sign change between $\mathbf{x}^{(k-1)}$ and \mathbf{x}^* , and the value of γ is obtained when $\|\mathbf{x}^*\|_1 = \tau$, which happens if:

$$\gamma = \frac{\tau - \|\mathbf{x}^{(k-1)}\|_1}{\text{sgn}(\mathbf{x}^{(k-1)})^T \mathbf{d}^{(k)}}.$$

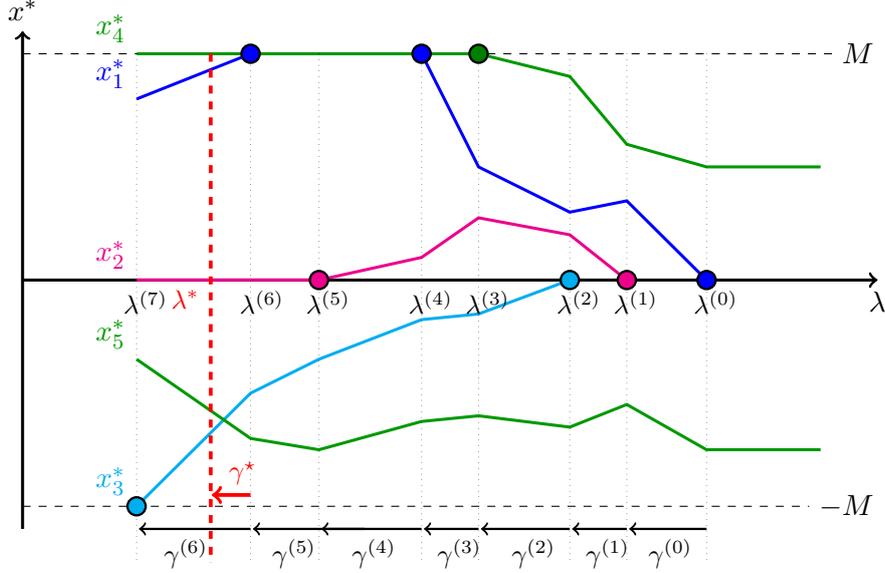


Figure 2. Example of solution path \mathbf{x}^* giving the solutions of the problem (1) as a function of λ , with 5 variables: $\bar{S} = \{1, 2, 3\}$ and $S^1 = \{4, 5\}$. Circles represent the events that cause a change in the support configuration. Vertical dotted lines represent the breakpoints.

- Similarly, for $\mathcal{Q}_{1/2}$, iterations stop at the first breakpoint such that $\frac{1}{2}\|\mathbf{y} - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^{(k)} - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^{(k)}\|_2^2 \leq \epsilon$. Substituting equation (28) in the least squares expression, the value of γ such that $\frac{1}{2}\|\mathbf{y} - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^* - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^*\|_2^2 = \epsilon$ can be found by solving a (scalar) quadratic equation, whose solution is:

$$\gamma = \frac{\mathbf{r}^T \mathbf{v} - \sqrt{(\mathbf{r}^T \mathbf{v})^2 - \mathbf{v}^T \mathbf{v} (\mathbf{r}^T \mathbf{r} - 2\epsilon)}}{\mathbf{v}^T \mathbf{v}},$$

with $\mathbf{r} = \mathbf{y} - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^{(k-1)} - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^{(k-1)}$ and $\mathbf{v} = \mathbf{A}_{\bar{S}_{in}}\mathbf{d}_{\bar{S}_{in}}^{(k)} + \mathbf{A}_{S^1}\mathbf{d}_{S^1}^{(k)}$.

3.4. Implementation and practical issues

Some practical remarks concerning the numerical implementation of the homotopy algorithm 1 are detailed in this section.

First, each iteration mostly consists of solving linear systems of equations (20) and (21), whose size respectively correspond to the current number of variables in \bar{S}_{in} and S_{in}^1 . Since the support configuration only slightly changes between two breakpoints¹, the matrix inverses of equations (20) and (21) can be computed recursively by performing rank-one updates. In our simulations, using the *block matrix inversion formulas* appeared to be the most efficient strategy.

We also remark that, for a given non-zero component, all computations of equations (20)–(21) are not necessary. In particular, if $x_\ell^{(k-1)} > 0$ and $d_\ell^{(k)} > 0$ (or respectively, $d_\ell^{(k)} < 0$), then the only possible breakpoint that can occur for component ℓ is when x_ℓ reaches the upper bound M , i.e. the case 2c in §3.2 (or respectively, 0, i.e. the case 2b). A similar reasoning is applied to negative components if $x_\ell^{(k-1)} < 0$.

¹Although several support configuration changes may happen simultaneously, in practice, \bar{S}_{in} and S_{in}^1 are usually modified by at most one element at each breakpoint, corresponding to the activation of one condition among equations (23)–(27).

We conclude this section with an important remark concerning the use of the homotopy algorithm in the branch-and-bound resolution of the error-constrained problem $\mathcal{P}_{0/2}$. At each node, the continuous relaxation brought by the solution to problems such as $\mathcal{Q}_{1/2}$ in Table 2 provides a lower bound on the global optimum value of $\mathcal{P}_{0/2}$, which is compared to the best known feasible solution. Since the objective function in $\mathcal{Q}_{1/2}$ increases within the homotopy iterations, the homotopy algorithm is stopped as soon as the ℓ_1 norm of the current iterate exceeds the global upper bound. In this case, the node of the branch-and-bound tree is pruned.

4. Experimental results

We now insert the continuous relaxation algorithms built in Section 3 into dedicated branch-and-bound procedures for the resolution of the three problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$, and \mathcal{P}_{2+0} . We use *depth-first search* and we choose to branch on the continuous variable x_k with the maximum absolute value in the solution of the relaxed problem. We branch *up* first, that is, we first explore the branch corresponding to the decision $x_k \neq 0$. This strategy is well adapted for problem $\mathcal{P}_{2/0}$, in which the depth limit is imposed by the cardinality constraint. It also allows one to quickly find feasible solutions for problems \mathcal{P}_{2+0} and $\mathcal{P}_{0/2}$ with limited depth search, since we know in advance that their solutions are sparse. For problem \mathcal{P}_{2+0} , we also exploit the fact that the initialization step in the homotopy algorithm at each node evaluation (see § 3.2) produces a feasible solution, which is used in order to improve the best solution found.

In this section, we compare our homotopy-based branch-and-bound implementation, named $\text{B\&B}_{\text{R-HOM}}$, to:

- the same branch-and-bound exploration strategy, where the continuous relaxation problems $\mathcal{Q}_{2/0}^R$, $\mathcal{Q}_{0/2}^R$ and \mathcal{Q}_{2+0}^R in Table 2 are solved with the CPLEX 12.8 quadratic programming solver ($\text{B\&B}_{\text{R-Cplex}}$);
- the resolution of the MIP problems in Table 1 with the CPLEX 12.8 MIP solver ($\text{MIP}_{\text{Cplex}}$).

All methods are implemented in C++ and executed on a UNIX machine equipped with 31.1Go RAM and four Intel Core i7-6600U central processing units (CPUs) clocked at 2.60 GHz. Computations are restricted to one core in order to focus on the algorithm performance, disabling parallelization capacities. All experiments are run by using the default CPLEX settings, and the maximum CPU time allowed was set to 1000 s.

In § 4.1, we study the performance of our algorithm on simulated sparse deconvolution problems typically encountered in signal processing. Then, simulated subset selection problems with random entries are considered in § 4.2.

4.1. Sparse deconvolution problems

Sparse deconvolution is a classical signal processing problem [17, Chapter 5], which aims to estimate a sparse sequence \boldsymbol{x} from filtered and noisy observations. Collecting sampled data in vector \boldsymbol{y} and the unknown sparse sequence in vector \boldsymbol{x} , we obtain a model of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$, where \boldsymbol{A} is a convolution matrix and \boldsymbol{n} is a random term representing noise and model errors. Deconvolution is an ill-posed inverse problem [17], whose resolution with sparsity-enhancing regularization is often addressed through suboptimal ℓ_1 -norm-based or greedy methods. It was shown in [4] that exact resolution of ℓ_0 -norm problems (with the CPLEX MIP solver) achieves better solutions, obviously with higher computing times. From a practical point of view, the three problems \mathcal{P}_{2+0} , $\mathcal{P}_{2/0}$ and $\mathcal{P}_{0/2}$ are of interest, depending on the available information for tuning their respective

parameter. We consider the problem instances² proposed in [4], with $\mathbf{y} \in \mathbb{R}^{120}$, $\mathbf{A} \in \mathbb{R}^{100 \times 120}$, noise samples are independent, identically and normally distributed with signal-to-noise ratio SNR = 10 dB and $M = 1.1 \|\mathbf{A}^T \mathbf{y}\|_\infty$. Columns of \mathbf{A} have been normalized. The cardinality varies from $K = 5$ to $K = 9$ for $\mathcal{P}_{2/0}$. For problems $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} , the respective parameters ϵ and λ are tuned from statistical rules accounting for the noise level and the sparsity degree (see [4] for details): if \mathbf{n} denotes the noise vector, ϵ is tuned such that the probability $P(\|\mathbf{n}\|^2 \leq \epsilon) = 95\%$, and $\lambda = 2\sigma^2 \log(1/\rho - 1)$, where σ^2 is the noise variance and $\rho = K/n$.

Computational results are reported in Table 3.

Problem		Branch-and-bound						MIP solver CPLEX		
		B&B _{R-HOM}			B&B _{R-Cplex}			CPLEX 12.8		
		Time	Nodes	F	Time	Nodes	F	Time	Nodes	F
		(s)	(10 ³)		(s)	(10 ³)		(s)	(10 ³)	
$\mathcal{P}_{2/0}$	$K = 5$	1.9	1.28	0	7.7	1.28	0	3.0	1.71	0
	$K = 7$	26.1	17.89	0	141.9	17.89	2	16.6	21.51	0
	$K = 9$	124.3	57.37	16	448.1	57.46	30	53.8	72.04	6
\mathcal{P}_{2+0}	$K = 5$	3.9	2.01	0	37.7	2.02	0	3.2	1.98	0
	$K = 7$	26.3	10.20	0	187.9	10.22	7	7.4	9.61	0
	$K = 9$	91.1	31.80	14	470.7	31.87	28	17.3	23.74	2
$\mathcal{P}_{0/2}$	$K = 5$	3.4	1.98	0	102.4	2.05	0	25.7	6.71	0
	$K = 7$	9.4	8.26	0	274.4	8.25	13	52.7	20.48	2
	$K = 9$	21.3	13.14	7	485.2	13.15	35	111.3	28.36	17

Table 3. Computational results for simulated sparse deconvolution problems. For each algorithm, columns “Time”, “Nodes”, and “F” respectively give the average computing time in seconds, the average number of explored nodes, and the number of problems that could not be solved in 1000s. 50 instances of each problem were considered. Averages are performed only over instances which could be solved in less than 1000s by the three algorithms.

For all problems, B&B_{R-HOM} and B&B_{R-Cplex} explore the same number of nodes, since they use the same branch-and-bound strategy (slight differences are observed, though, that may be due to numerical issues). However, B&B_{R-HOM} requires significantly less execution time: about a factor between 3.6 and 9 for $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} , and about a factor between 20 and 30 for $\mathcal{P}_{0/2}$. It also successfully solves more instances in 1000s.

Unsurprisingly, MIP_{Cplex} performs better than B&B_{R-HOM} on $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} : it runs generally 2 to 3 times faster (up to 5.3 on the most difficult \mathcal{P}_{2+0} problem with $K = 9$), and both the resolution strategy (number of nodes) and the continuous relaxation efficiency (computing time per node) are favorable to MIP_{Cplex}. This was expected, since the CPLEX MIP solver benefits from many additional developments that are not implemented in our branch-and-bound algorithm, such as cutting planes, heuristics, warm start, etc..

However, our algorithm outperforms MIP_{Cplex} on all instances of $\mathcal{P}_{0/2}$, both in terms of number of nodes (2 to 3 times less) and CPU time. In particular, the computing time per node is reduced

²Data are available online at pagesperso.lis2n.fr/~bourguignon-s/download_MIP.html

by more than 2 with the homotopy algorithm. As a result, $\text{B\&B}_{\text{R-HOM}}$ runs 4 to 8 times faster than $\text{MIP}_{\text{Cplex}}$ on these instances, even with a rather basic branch-and-bound strategy.

We also note that the relative efficiency of $\text{B\&B}_{\text{R-HOM}}$ compared to other tested algorithms increases as K decreases. Indeed, when the global optimum is sparse, the solutions to the continuous relaxation problems are sparse, too. Therefore, the homotopy algorithm becomes more efficient. Indeed, the computing time increases with the number of non-zero components in the solution of the relaxed problem.

In Figure 3, the performance profiles, defined by Dolan and Moré in [9], are visual tools to benchmark algorithms. The fraction of solved problems is represented as a function of the performance ratio: for each instance, the ratio between the computing time of each algorithm and the minimum computing time is computed. Then, the performance profiles draw the cumulative distribution functions for such ratios.

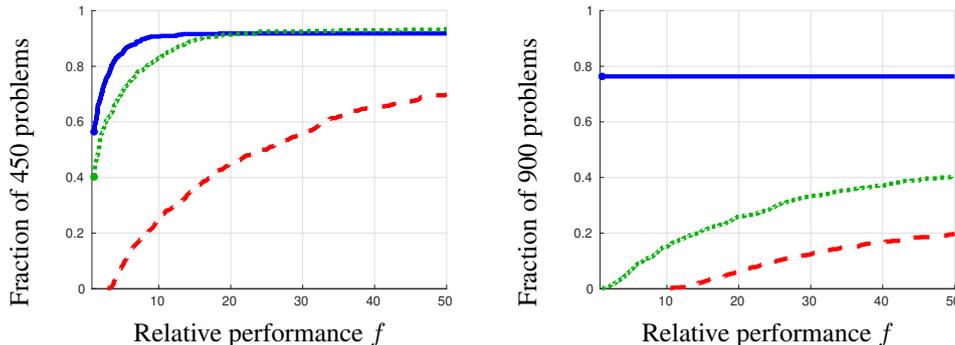


Figure 3. Performance profiles obtained on 450 sparse deconvolution (left, see § 4.1) and 900 random subset selection problems (right, see § 4.2), for the three algorithms: $\text{B\&B}_{\text{R-HOM}}$ (full line), $\text{MIP}_{\text{Cplex}}$ (dotted line) and $\text{B\&B}_{\text{R-Cplex}}$ (dashed line).

Performance profiles obtained over the 450 considered instances (50 instances for each of the 9 problems defined in Table 3) are shown in the left part of Figure 3, from which we have the following conclusions:

- The performance of $\text{B\&B}_{\text{R-HOM}}$ is far above that of $\text{B\&B}_{\text{R-Cplex}}$, confirming the results in Table 3.
- $\text{MIP}_{\text{Cplex}}$ solves more problems in 1000 s and is globally the fastest when considering all instances (the green curve achieves the higher value on the top-right corner).
- Nevertheless, the performance of $\text{B\&B}_{\text{R-HOM}}$ is above that of $\text{MIP}_{\text{Cplex}}$ in most part of the profile. For example, $\text{B\&B}_{\text{R-HOM}}$ requires 6.7 times more time than the best algorithm for solving 90% of the problems, whereas $\text{MIP}_{\text{Cplex}}$ requires 12.5 times more time (see the horizontal lines in Figure 3). Similarly, in 5 times the best algorithm time, $\text{B\&B}_{\text{R-HOM}}$ solves 86% of problems, whereas $\text{MIP}_{\text{Cplex}}$ only solves 76% of them (see the vertical line in Figure 3).

4.2. Simulated Subset selection problems

We now study algorithmic performance on artificial subset selection problems with random entries (see [2, 32] for similar simulations). We consider a dictionary $\mathbf{A} \in \mathbb{R}^{m \times n}$, with $n = 2 \times m$, composed of independent and identically distributed random entries, which are drawn from a centered, unit-variance, normal distribution (columns have been normalized). Sparse vectors are generated with random non-zero positions, and corresponding amplitudes are drawn as $u + \text{sgn}(u)$, where u follows a centered, unit variance, Gaussian distribution. Noise is added such that $\text{SNR} = 10$ dB. The problem cardinality varies from $K = 5$ to $K = 15$. Parameters M and formulation-specific

parameters K , ϵ and μ are tuned similarly as in § 4.1. Note that the random nature of the dictionary makes problems easier to solve, which allows us to increase the problem dimension: we consider here $n = 500$ and $n = 1000$. Indeed, the compressed sensing theory (see for example [14]) shows that, if the correlation between columns in \mathbf{A} is sufficiently low, then, with high probability, the ℓ_1 -norm relaxation may solve the original ℓ_0 -norm problem.

Computational results are reported in Table 4. As in § 4.1, $\text{B\&B}_{\text{R-HOM}}$ and $\text{B\&B}_{\text{R-Cplex}}$ explore the same number of nodes. However, $\text{B\&B}_{\text{R-HOM}}$ requires significantly less execution time: about a factor of 10 to 100 for $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} , and about a factor of 100 to 230 for $\mathcal{P}_{0/2}$. It also successfully solves more instances in less than 1 000s.

Problem			Branch-and-bound						MIP solver CPLEX		
			$\text{B\&B}_{\text{R-HOM}}$			$\text{B\&B}_{\text{R-Cplex}}$			CPLEX 12.8		
			Time	Nodes	F	Time	Nodes	F	Time	Nodes	F
			(s)	(10^3)		(s)	(10^3)		(s)	(10^3)	
$n = 500$	$(\mathcal{P}_{2/0})$	$K = 5$	1.1	0.03	0	23.5	0.03	0	16.8	0.04	0
		$K = 10$	16.2	0.26	0	249.4	0.26	12	107.0	0.51	5
		$K = 15$	39.5	0.59	38	538.7	0.59	48	161.4	1.09	45
	(\mathcal{P}_{2+0})	$K = 5$	2.2	0.14	0	203.6	0.12	1	19.8	0.15	0
		$K = 10$	10.7	0.33	0	460.1	0.29	21	68.8	0.90	3
		$K = 15$	23.2	0.63	40	626.9	0.63	49	179.4	3.28	47
	$(\mathcal{P}_{0/2})$	$K = 5$	0.8	0.03	2	184.9	0.03	7	177.5	4.10	17
		$K = 10$	3.7	0.13	6	533.7	0.13	46	-	-	50
		$K = 15$	482.7	8.71	44	-	-	50	-	-	50
$n = 1000$	$(\mathcal{P}_{2/0})$	$K = 5$	2.9	0.02	0	109.2	0.02	0	70.7	0.03	0
		$K = 10$	18.2	0.06	0	437.7	0.06	13	371.7	0.31	5
		$K = 15$	117.6	0.38	15	-	-	50	665.3	0.72	42
	(\mathcal{P}_{2+0})	$K = 5$	2.6	0.08	0	275.9	0.06	10	130.1	0.19	0
		$K = 10$	13.8	0.20	0	601.9	0.13	32	247.9	0.60	3
		$K = 15$	81.8	0.52	17	-	-	50	665.3	2.15	40
	$(\mathcal{P}_{0/2})$	$K = 5$	1.3	0.01	0	159.5	0.01	4	360.9	0.28	28
		$K = 10$	7.8	0.05	2	695.9	0.05	40	-	-	50
		$K = 15$	442.3	2.58	30	-	-	50	-	-	50

Table 4. Computational results for Simulated Subset selection problems. 30 instances of each problem were considered. Averages are performed only over instances which could be solved in less than 1000 s.

Compared to $\text{MIP}_{\text{Cplex}}$, $\text{B\&B}_{\text{R-HOM}}$ performs better on most problems in terms of execution time: it runs generally more than 10 times faster (up to 276 times on the most difficult problems $\mathcal{P}_{0/2}$ with $K = 5$ and $n = 1000$). For all problems, both the number of nodes and the continuous relaxation

efficiency (computing time per node) are favorable to $\text{B\&B}_{\text{R-HOM}}$.

We also remark that the CPLEX MIP solver performs very badly on most $\mathcal{P}_{0/2}$ instances. In particular, the number of explored nodes in $\text{MIP}_{\text{Cplex}}$ is much bigger than with the proposed branch-and-bound strategy. The computing time per node with $\text{MIP}_{\text{Cplex}}$ is also much higher than with $\text{B\&B}_{\text{R-HOM}}$, confirming the improvement brought by the homotopy-based relaxation computations.

Performance profiles obtained over the 900 considered instances (50 instances for each of the 18 problems defined in Table 3) are shown in right part of Figure 3, from which we have the following conclusions:

- The three algorithms generally solve more instances in 1000 s for $(m, n) = (500, 1000)$ than for $(m, n) = (250, 500)$. This somehow counterintuitive result can still be explained by the compressed sensing theory [14]: as the dimension increases, the columns of \mathbf{A} , composed of random numbers, become less correlated, making the problems easier.
- The $\text{B\&B}_{\text{R-HOM}}$ algorithm clearly outperforms $\text{B\&B}_{\text{R-Cplex}}$ and $\text{MIP}_{\text{Cplex}}$. It is the fastest and most efficient in all solved problems.

5. Conclusion

We have investigated exact optimization algorithms for least squares problems with sparsity constraint. By studying the problem properties, we proposed a dedicated branch-and-bound procedure, which binary variables that are usually used for MIP reformulation. In particular, an algorithm was built for solving the continuous relaxation problems involved at any node of the search tree, which were recast as continuous non-smooth optimization problems, involving the ℓ_1 norm. Inspired by the homotopy principle, this algorithm can be applied with similar efficiency for the three addressed formulations (cardinality-constrained and cardinality-penalized least-squares, and cardinality minimization under quadratic constraints). In particular, the resulting strategy was shown to efficiently solve the quadratically constrained formulation, which is of major interest in many applications, whereas the CPLEX MIP solver resulted to be much less efficient.

The method is even more efficient as the solution searched is sparse. The computational burden of the homotopy algorithm increases almost linearly with the cardinality of the relaxed solutions. In cases where the cardinality increases, other ℓ_1 -norm-based methods may be investigated, which could benefit from warm starting when evaluating the continuous relaxation at two close nodes. Lagrangian relaxation was also shown to achieve tighter relaxation than the continuous one, for such problems where the Gram matrix $\mathbf{A}^T \mathbf{A}$ admits a specific decomposition. Studying dedicated algorithms in this case, in particular for the two less familiar formulations $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} , would also deserve further works. Finally, the three formulations could be tackled jointly from a multi-objective optimization perspective, involving dedicated branch-and-bound techniques [12].

References

- [1] D. Bertsimas, A. King, and R. Mazumder, *Best subset selection via a modern optimization lens*, The Annals of Statistics 44 (2016), pp. 813–852.
- [2] D. Bertsimas and R. Shioda, *Algorithm for cardinality-constrained quadratic optimization*, Computational Optimization and Applications 43 (2009), pp. 1–22.
- [3] D. Bienstock, *Computational study of a family of mixed-integer quadratic programming problems*, Mathematical Programming 74 (1996), pp. 121–140.
- [4] S. Bourguignon, J. Ninin, H. Carfantan, and M. Mongeau, *Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance*, IEEE Transactions on Signal Processing 64 (2016), pp. 1405–1419.

- [5] A.M. Bruckstein, D.L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM review 51 (2009), pp. 34–81.
- [6] S.S. Chen, D.L. Donoho, and M.A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput. 20 (1998), pp. 33–61.
- [7] S. Chen, S.A. Billings, and W. Luo, *Orthogonal least squares methods and their application to non-linear system identification*, International Journal of control 50 (1989), pp. 1873–1896.
- [8] X. Cui, X. Zheng, S. Zhu, and X. Sun, *Convex relaxations and miqcqp reformulations for a class of cardinality-constrained portfolio selection problems*, Journal of Global Optimization 56 (2013), pp. 1409–1423.
- [9] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming 91 (2002), pp. 201–213.
- [10] D.L. Donoho and Y. Tsaig, *Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse*, IEEE Transactions on Information Theory 54 (2008).
- [11] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, *et al.*, *Least angle regression*, The Annals of statistics 32 (2004), pp. 407–499.
- [12] M. Ehrgott, *Multicriteria Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [13] M. Elad, *Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing*, Springer-Verlag New York, 2010.
- [14] Y. Elad and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.
- [15] A. Frangioni and C. Gentile, *Perspective cuts for a class of convex 0–1 mixed integer programs*, Mathematical Programming 106 (2006), pp. 225–236.
- [16] J. Gao and D. Li, *Optimal cardinality constrained portfolio selection*, Operations research 61 (2013), pp. 745–761.
- [17] J. Idier, *Bayesian Approach to Inverse Problems*, ISTE Ltd and John Wiley & Sons Inc, 2018.
- [18] D. Li, X. Sun, and J. Wang, *Optimal lot solution to cardinality constrained mean–variance formulation for portfolio selection*, Mathematical Finance 16 (2006), pp. 83–101.
- [19] X. Liang and Y. Wang, *Homotopy algorithm for box-constrained LASSO and its convergence*, International Journal of Pure and Applied Mathematics 112 (2017).
- [20] J.M. Mendel, *Optimal Seismic Deconvolution*, Academic Press, 1983.
- [21] A. Miller, *Subset selection in regression*, Chapman and Hall/CRC, 2002.
- [22] B.K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM journal on computing 24 (1995), pp. 227–234.
- [23] M.S. O’Brien, A.N. Sinclair, and S.M. Kramer, *Recovery of a sparse spike time series by L1 norm deconvolution*, IEEE Trans. Signal Processing 42 (1994), pp. 3353–3365.
- [24] M. Osborne, B.P. B, and B.T. BAD, *A new approach to variable selection in least squares problems*, IMA Journal of Numerical Analysis (2000).
- [25] Y. Pati, R. Rezaifar, and P.S. Krishnaprasad, *Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition*, in *Asilomar Conference on Signals, Systems and Computers*. 1993, pp. 40–44 vol.1.
- [26] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
- [27] D.X. Shaw, S. Liu, and L. Kopman, *Lagrangian relaxation procedure for cardinality-constrained portfolio optimization*, Optimization Methods and Software 23 (2008), pp. 411–420.
- [28] C. Soussen, J. Idier, D. Brie, and J. Duan, *From bernoulli gaussian deconvolution to sparse signal restoration*, IEEE Transactions on Signal Processing 59 (2011), pp. 4572–4584.
- [29] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society, Series B 58 (1996), pp. 267–288.
- [30] J.A. Tropp and S.J. Wright, *Computational methods for sparse solution of linear inverse problems*, Proceedings of the IEEE 98 (2010), pp. 948–958.
- [31] C. Zala, *High-resolution inversion of ultrasonic traces*, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 39 (1992), pp. 458–463.
- [32] X. Zheng, X. Sun, D. Li, and J. Sun, *Successive convex approximations to cardinality-constrained convex programs: a piecewise-linear dc approach*, Computational Optimization and Applications 59 (2014), pp. 379–397.