



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*, 2022, 37 (5), 10.1080/10556788.2021.1977809 . hal-03419540v2

HAL Id: hal-03419540

<https://ensta-bretagne.hal.science/hal-03419540v2>

Submitted on 19 Nov 2021

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, Sébastien BOURGUIGNON^{a*} and Jordan NININ^b

^a*LS2N, CNRS UMR 6004, École Centrale de Nantes, Nantes, France;*

^b*Lab-STICC, CNRS UMR 6285, ENSTA Bretagne, Brest, France*

ARTICLE HISTORY

Compiled August 23, 2021

ABSTRACT

Finding solutions to least-squares problems with low cardinality has found many applications, including portfolio optimization, subset selection in statistics, and inverse problems in signal processing. Although most works consider local approaches that scale with high-dimensional problems, some others addressed its global optimization via mixed integer programming (MIP) reformulations. We propose dedicated branch-and-bound methods for the exact resolution of moderate-size, yet difficult, sparse optimization problems, through three possible formulations: cardinality-constrained and cardinality-penalized least-squares, and cardinality minimization under quadratic constraints. A specific tree exploration strategy is built. Continuous relaxation problems involved at each node are reformulated as ℓ_1 -norm-based optimization problems, for which a dedicated algorithm is designed. The obtained certified solutions are shown to better estimate sparsity patterns than standard methods on simulated variable selection problems involving highly correlated variables. Problem instances selecting up to 24 components among 100 variables, and up to 15 components among 1000 variables, can be solved in less than 1000s. Unguaranteed solutions obtained by limiting the computing time to 1s are also shown to provide competitive estimates. Our algorithms strongly outperform the CPLEX MIP solver as the dimension increases, especially for quadratically-constrained problems. The source codes are made freely available online.

KEYWORDS

Sparse approximation; Subset selection; 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 [4, 13, 27, 40], sparse regularization for inverse problems [12, 18, 30, 42, 47] or compressed sensing [8, 19], and variable or subset selection in statistics [16, 31, 36, 44]. In operations research, many works addressed the cardinality-constrained problem [4, 5, 13, 27, 40]:

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

*CONTACT S. BOURGUIGNON. Author. Email: Sebastien.Bourguignon@ec-nantes.fr

where $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$ with often $n > m$, $\|\mathbf{x}\|_0 = \text{Card}\{i|x_i \neq 0\}$ (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*. Without loss of generality, the columns of \mathbf{A} are supposed to have unit ℓ_2 norm. In many applications, one may prefer solving the error-constrained problem [31, 33, 45]:

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

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

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

where $\mu > 0$ trades off between approximation error and sparsity, is also encountered in the field of inverse problems, *e.g.*, for Geophysics [30] or ultrasonic non-destructive testing [35, 47]. In the Bayesian statistical framework, the ℓ_0 -norm penalization term corresponds to a Bernoulli-Gaussian prior assumption about the unknown components in \mathbf{x} , and parameter μ then depends on both the noise level and the expected rate of non-zero values in \mathbf{x} [42].

Optimization of such problems is NP-hard due to the discrete nature of the cardinality term [5, 33]. In many applications involving high-dimensional problems (that is, essentially, high n and high K), exact resolution of such problems may not be computationally tractable and a vast literature has been dedicated to scalable sparsity-promoting approaches. On the one hand, local optimization methods have been proposed, such as the *Iterative Hard Thresholding* algorithm [6, 24], the CoSaMP local exploration strategy [34], or many heuristic-based greedy algorithms [31, 42, 43], even coupled with local tree-search methods [25]. On the other hand, many works studied continuous approximations of the discrete-valued ℓ_0 norm. In particular, using the ℓ_1 norm opens the way to many possible convex optimization methods (see for example [45] and references therein). In very particular cases as those addressed within the compressed sensing theory [19], such an approach may solve the ℓ_0 -norm problems. Concave approximations of the ℓ_0 norm, coupled with local optimization strategies, were shown to achieve better performance in practice (see for example [11, 14, 32, 38]). Iterative methods such as FOCUSS [22] or IRL1 [10] do not have any explicit cost function and rather iteratively solve a sequence of sparsity-promoting problems, and may also provide better solutions than standard ℓ_1 -norm-based optimization. All such methods, however, do not provide any strong optimality guarantee with respect to the original ℓ_0 -norm formulation. More theoretical guarantees may be obtained for alternate problem reformulations such as penalty decomposition [29], DC programming [26] or complementarity constraints [9]. However, equivalence conditions with the original problem are weak, and practical implementation may be hard to monitor.

On the other hand, the resolution of ℓ_0 -norm problems has also been addressed from a global optimization perspective. In particular, the certified optimal solutions were shown to provide better estimates than classical sparsity-promoting methods on low-dimensional problems typically encountered in variable selection [3] and signal processing [7]. Despite their small size, these problems are made difficult by the high correlation level between the columns of matrix \mathbf{A} . Exact optimization of sparse

optimization problems essentially resorts to combinatorial exploration, which can be performed *via* tree-search methods. In [3, 7], such problems were reformulated as mixed integer programs (MIPs), and then solved with a generic MIP solver.

Dedicated optimization algorithms have been proposed for solving $\mathcal{P}_{2/0}$, in the context of sparse portfolio selection and subset selection. To our knowledge, Bienstock was the first to propose a specific branch-and-cut algorithm for such problems, with additional non-negativity constraints [5]. Continuous relaxation problems involved in each iteration were solved *via* a specific convex quadratic programming algorithm. In [4], 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 [20], Lagrangian relaxation [13, 27, 40] or geometric approaches [21], were shown to give tighter bounds than the continuous relaxation. In [2], we proposed a branch-and-bound method specific to the penalized version \mathcal{P}_{2+0} . Each node was evaluated by solving an ℓ_1 -norm-penalized least-squares problem, for which an active-set method was built. We also mention the preprint [23], which proposes a specific solver for the penalized formulation with an additional ℓ_2 -norm regularization term. Up to our knowledge, problem $\mathcal{P}_{0/2}$ was never addressed with dedicated exact optimization methods.

In this paper, we address the three problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} with exact resolution methods which are specifically designed for such problems. Our goal is to outperform generic MIP solvers and to increase the size of considered problems, while maintaining the quality of solutions and global optimality guarantees. Following the works in [4, 5], our motivation lies in the fact that sparsity-enhancing least-squares problems are very specific MIPs, that could be advantageously solved by dedicated algorithms. Starting from the MIP formulation, we propose a branch-and-bound procedure whose exploration strategy is inspired by local greedy search techniques, betting on quickly finding high-quality feasible solutions. Then, we show that all continuous relaxation problems involved in the resolution are particular forms of problems involving the ℓ_1 norm. We build a dedicated algorithm for all such problems, inspired by the homotopy principle [15, 16, 36]. This algorithm is able to solve the relaxation problems involved in the resolution of any of the three problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} with similar computational burden.

The paper is organized as follows. In Section 2, we give the general architecture of our branch-and-bound procedure and we study the structure of continuous relaxation problems involved at each node, which are reformulated as ℓ_1 -norm-based problems. A dedicated optimization strategy for such problems is then built in Section 3. In Section 4, the performance of our method is evaluated through numerical experiments on simulated subset selection problems. Solutions are compared to that of standard methods, and computation times are compared to those of the CPLEX MIP solver through different possible formulations. The discussion in Section 5 closes the paper.

2. Branch-and-bound algorithm and continuous relaxations

In this section, we consider the MIP reformulations of problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} . Binary decision variables b_i are introduced to encode the nullity of the model components. 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 $-Mb_i \leq x_i \leq Mb_i$. The three problems can then be reformulated as the standard

MIPs given in Table 1 (see for example [7]).

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

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

Using a tight bound for parameter M is a crucial point, since an artificially high value for M would strongly impact on the resolution efficiency. In the overdetermined case $n < m$, valid (although possibly loose) bounds can be found by solving a convex continuous optimization problem [3]. In the more general underdetermined case, automatic tuning of M remains an open question. In practice, however, natural bounds on the model coefficients can often be obtained from prior knowledge on the problem. In this paper, the value of M is set to $1.1 \|\mathbf{A}^T \mathbf{y}\|_\infty$, where $\|\mathbf{A}^T \mathbf{y}\|_\infty$ corresponds to the maximum amplitude of 1-sparse solutions [7]. Although empirical, this tuning rule will be shown to provide valid bounds on the simplest problems addressed in Section 4.3.

In Section 2.1, we describe our branch-and-bound implementation to solve such MIPs, in particular by using specific variable selection and branching rules. Then, the mathematical properties of the continuous relaxation problems involved at the root node and at any node of the algorithm are studied in Sections 2.2 and 2.3 respectively, leading to their reformulation as ℓ_1 -norm-based optimization problems, and removing the binary variables.

2.1. Branch-and-bound implementation

We consider a resolution strategy based on a branch-and-bound procedure, as adopted by most MIP solvers. The initial problem defines the 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 as the value of the cost function at 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 binary variables to 0 and 1. The two new nodes are then added to the list.

We use *depth-first search*, and our branching rule is based on selecting the binary variable, say b_i , with the highest value in the solution of the relaxed problem. We branch *up* first, that is, we first explore the branch corresponding to the decision $b_i = 1$. This strategy, similar to the principle of greedy forward selection algorithms [31], aims at activating first the most prominent nonzero variables in \mathbf{x} , therefore focusing on

quickly finding satisfactory feasible solutions and subsequent upper bounds of good quality. It is particularly well adapted for problems $\mathcal{P}_{2/0}$, 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.

The general procedure is summarized in Algorithm 1, using the following notations:

- L is the set of active subproblems;
- $(\mathbf{b}^*, \mathbf{x}^*)$ is the best known feasible solution;
- z_U is the current upper bound (the value of the objective function at $(\mathbf{b}^*, \mathbf{x}^*)$);
- $\mathcal{Q}^{R(i)}$ is the continuous relaxation of subproblem i ;
- $(\mathbf{b}^{R(i)}, \mathbf{x}^{R(i)})$ is the solution to $\mathcal{Q}^{R(i)}$;
- $z^{R(i)} = \min \mathcal{Q}^{R(i)}$ is the lower bound on the value of subproblem i ;
- $z_L = \min_{i \in L} z^{R(i)}$ is the global lower bound on the problem.

Note that, contrary to generic MIP solvers, we do not set any duality gap or ε precision on $z_U - z_L$ as a stopping condition. Our procedure is run until all nodes have been evaluated or removed. Indeed, feasible solutions of the MIP are systematically computed throughout the procedure. More precisely, at each node, the initialization step of our continuous relaxation algorithm requires the computation of box-constrained least-squares problems on cardinality-limited subsets (see Equations 7a and 7b), which are feasible solutions for $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} . For $\mathcal{P}_{0/2}$, feasible solutions are found when the corresponding least-squares error is lower than the given parameter ϵ . Thus, our branch-and-bound algorithm converges towards an optimal solution of the problem up to machine precision.

2.2. Continuous relaxation at 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:

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

The following proposition reformulates $\mathcal{P}_{2/0}^R$ as an ℓ_1 -norm-constrained problem involving only the continuous variables \mathbf{x} .

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

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

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 $\mathbf{b}^1 := \frac{1}{M} |\mathbf{x}^1|$. Then, $(\mathbf{b}^1, \mathbf{x}^1)$ is clearly feasible for $\mathcal{P}_{2/0}^R$, therefore $\|\mathbf{y} - \mathbf{A}\mathbf{x}^R\|_2^2 \leq \|\mathbf{y} - \mathbf{A}\mathbf{x}^1\|_2^2$. Conversely, \mathbf{x}^R is feasible for $\mathcal{P}_{2/1}$ because $\|\mathbf{x}^R\|_1 \leq M\|\mathbf{b}^R\|_1 = M \sum_{i=1}^N b_i^R \leq KM$ and $\|\mathbf{x}^R\|_\infty \leq M\|\mathbf{b}^R\|_\infty \leq M$. Consequently, $\|\mathbf{y} - \mathbf{A}\mathbf{x}^1\|_2^2 \leq$

0. Initialization: $L \leftarrow \{\text{Initial problem}\}$; $z_U \leftarrow +\infty$.

1. Optimality: If $L = \emptyset$, then $(\mathbf{b}^*, \mathbf{x}^*)$ is the certified optimal solution.

2. Node selection: Choose subproblem i in L (depth-first search) and remove it from L .

3. Node evaluation:

```

begin
  Solve the continuous relaxation problem  $Q^{R(i)}$ .
  if  $Q^{R(i)}$  has no solution then
    | return to Optimality step;                                ▷ Pruning
  else
    | let  $(\mathbf{b}^{R(i)}, \mathbf{x}^{R(i)})$  and  $z^{R(i)}$  be the solution and the optimum value
    |   of  $Q^{R(i)}$ ;
  end
  if  $z^{R(i)} \geq z_U$  then
    | return to Optimality step;                                ▷ Pruning
  else
    | if all variables in  $\mathbf{b}^{R(i)}$  are binary then
    |   | update the best-known solution:  $(\mathbf{b}^*, \mathbf{x}^*) \leftarrow (\mathbf{b}^{R(i)}, \mathbf{x}^{R(i)})$ ;
    |   | update the upper bound  $z_U \leftarrow z^{R(i)}$ ;
    |   | remove from  $L$  all subproblems  $j$  such that  $z^{R(j)} \geq z_U$ ;    ▷ Pruning
    |   | update the global lower bound  $z_L = \min_{i \in L} z^{R(i)}$ ;
    |   | return to Optimality step;                                    ▷ Pruning
    | else
    |   | go to Branching step;
    | end
  end
end

```

4. Branching: Choose index $i_0 = \arg \max_{k \in \{1, \dots, n\}, b_k^{R(i)} < 1} b_k^{R(i)}$, subdivide problem i into two sub-problems by fixing $b_{i_0} = 1$ and $b_{i_0} = 0$, add them to L and return to Optimality step.

Algorithm 1: Branch-and-bound algorithm for the resolution of any of the three problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$ and \mathcal{P}_{2+0} .

$\|\mathbf{y} - \mathbf{A}\mathbf{x}^R\|_2^2$ and the proposition follows. \square

We note that this result was given by [4, 5] for problems with non-negativity constraints, and by [3] in our case.

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

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

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

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

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

Proof. Let $(\mathbf{b}^R, \mathbf{x}^R)$ be a minimizer of $\mathcal{P}_{0/2}^R$. We show that $|\mathbf{x}^R| = M\mathbf{b}^R$, from which the proof is straightforward. Suppose that $|x_i^R| < Mb_i^R$ for some component i . Let $\mathbf{b}' := \frac{1}{M}|\mathbf{x}^R|$, such that $b'_i < b_i^R$. Then, $(\mathbf{b}', \mathbf{x}^R)$ is feasible for $\mathcal{P}_{0/2}^R$, with $\sum_{i=1}^n b'_i < \sum_{i=1}^n b_i^R$, which contradicts the definition of $(\mathbf{b}^R, \mathbf{x}^R)$. \square

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

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

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

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

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

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

We now consider a given node of the branch-and-bound algorithm and the corresponding continuous relaxation sub-problem. Let \mathbf{A}_S denote the sub-matrix formed by all columns of the matrix \mathbf{A} indexed by S . Similarly, \mathbf{z}_S denotes the corresponding sub-vector of \mathbf{z} . Let S^0 (respectively, S^1) denote the index set of binary variables that have been set to 0 (respectively, to 1) after a series of branching operations. Then, $\forall i \in S^0$, $b_i = 0$ and $x_i = 0$, so that the variables \mathbf{x}_{S^0} can be removed from the sub-problem. Similarly, $\forall i \in S^1$, $b_i = 1$ and variables x_i are only box-constrained by M . Last, let \bar{S} index the remaining (undetermined) binary variables, for which $b_i \in [0, 1]$ and $|x_i| \leq Mb_i$.

For problem $\mathcal{P}_{2/0}$, the continuous relaxation of variables $\mathbf{b}_{\bar{S}}$ in the corresponding

sub-problem reduces to:

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

where n_1 and \bar{n} denote the size of S^1 and \bar{S} , respectively. Then, similarly to the developments in Section 2.2, one can show that $\mathcal{Q}_{2/0}^R$ and $\mathcal{Q}_{2/1}$ have the same minimum value, with:

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

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

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

Continuous relaxation problem	Equivalent problem without binary variables
$\mathcal{Q}_{2/0}^R : \min_{\mathbf{b} \in [0, 1]^n, \mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \ \mathbf{y} - \mathbf{A}\mathbf{x}\ _2^2$ $\text{s.t.} \quad \begin{cases} \sum_{i=1}^n b_i \leq K \\ \mathbf{x} \leq M\mathbf{b} \\ \mathbf{b}_{S^1} = 1 \\ \mathbf{b}_{S^0} = 0 \end{cases}$	$\mathcal{Q}_{2/1} : \min_{\mathbf{x}_{S^1} \in \mathbb{R}^{n_1}, \mathbf{x}_{\bar{S}} \in \mathbb{R}^{\bar{n}}} \frac{1}{2} \ \mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}\ _2^2$ $\text{s.t.} \quad \begin{cases} \ \mathbf{x}_{\bar{S}}\ _1 \leq M(K - n_1) \\ \ \mathbf{x}_{\bar{S}}\ _\infty \leq M \\ \ \mathbf{x}_{S^1}\ _\infty \leq M \end{cases}$
$\mathcal{Q}_{0/2}^R : \min_{\mathbf{b} \in [0, 1]^n, \mathbf{x} \in \mathbb{R}^n} \sum_{i=1}^n b_i$ $\text{s.t.} \quad \begin{cases} \frac{1}{2} \ \mathbf{y} - \mathbf{A}\mathbf{x}\ _2^2 \leq \epsilon \\ \mathbf{x} \leq M\mathbf{b} \\ \mathbf{b}_{S^1} = 1 \\ \mathbf{b}_{S^0} = 0 \end{cases}$	$\mathcal{Q}_{1/2} : \min_{\mathbf{x}_{S^1} \in \mathbb{R}^{n_1}, \mathbf{x}_{\bar{S}} \in \mathbb{R}^{\bar{n}}} \frac{1}{M} \ \mathbf{x}_{\bar{S}}\ _1 + n_1$ $\text{s.t.} \quad \begin{cases} \frac{1}{2} \ \mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}\ _2^2 \leq \epsilon \\ \ \mathbf{x}_{\bar{S}}\ _\infty \leq M \\ \ \mathbf{x}_{S^1}\ _\infty \leq M \end{cases}$
$\mathcal{Q}_{2+0}^R : \min_{\mathbf{b} \in [0, 1]^n, \mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \ \mathbf{y} - \mathbf{A}\mathbf{x}\ _2^2 + \mu \sum_{i=1}^n b_i$ $\text{s.t.} \quad \begin{cases} \mathbf{x} \leq M\mathbf{b} \\ \mathbf{b}_{S^1} = 1 \\ \mathbf{b}_{S^0} = 0 \end{cases}$	$\mathcal{Q}_{2+1} : \min_{\mathbf{x}_{S^1} \in \mathbb{R}^{n_1}, \mathbf{x}_{\bar{S}} \in \mathbb{R}^{\bar{n}}} \frac{1}{2} \ \mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}\ _2^2 + \frac{\mu}{M} \ \mathbf{x}_{\bar{S}}\ _1 + \mu n_1$ $\text{s.t.} \quad \begin{cases} \ \mathbf{x}_{\bar{S}}\ _\infty \leq M \\ \ \mathbf{x}_{S^1}\ _\infty \leq M \end{cases}$

These properties are of major interest for our work, with two main consequences:

- whatever the formulation (constrained or penalized), all continuously relaxed subproblems involved in the evaluation of each node 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 a quadratic misfit and the ℓ_1 norm has been a very active field of research in the past ten years. Many dedicated convex, non-smooth, optimization algorithms have been developed (see for example [1, 45] and references therein), for solving problems:

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

$$\text{and } \mathcal{P}_1^\lambda : \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1.$$

The homotopy method [15, 16, 36] was shown to be a particularly efficient algorithm for highly sparse solutions in moderate dimension, as addressed in this paper [1]. It computes the set of solutions to \mathcal{P}_1^λ as a function of λ , which is piecewise linear in λ . Starting from $\lambda^{(0)} = \|\mathbf{A}^T \mathbf{y}\|_\infty$ (such that the solution is identically zero $\forall \lambda > \lambda^{(0)}$), a decreasing sequence of *breakpoints* $\{\lambda^{(t)}\}_{t=1, \dots, T}$ is computed iteratively, such that the *support* of the solution (the set of non-zero components and their sign) is constant over intervals $[\lambda^{(t+1)}, \lambda^{(t)}]$. On such interval, non-zero components vary linearly with λ , which allows one to find an analytical expression of the next breakpoint. The algorithm stops at iteration T when the target value λ belongs to $[\lambda^{(T)}, \lambda^{(T-1)}]$. Since the two objectives are convex, the three problems \mathcal{P}_1^τ , \mathcal{P}_1^ϵ , and \mathcal{P}_1^λ are equivalent. Therefore, problems \mathcal{P}_1^τ and \mathcal{P}_1^ϵ can be similarly solved by stopping the algorithm when the corresponding value of τ or ϵ is reached, respectively.

In this section, we generalize the homotopy method to the class of problems $\mathcal{Q}_{2/1}$, $\mathcal{Q}_{1/2}$ and \mathcal{Q}_{2+1} defined in Table 2, where *free* variables (that is, variables that are not involved in the ℓ_1 -norm term) and box constraints are included. Our algorithm still belongs to the class of *exact* methods, where the solution can be computed after a finite number of operations. More precisely, we show that i) the solution is still piecewise linear in intervals of the form $[\lambda^{(t+1)}, \lambda^{(t)}]$, ii) an analytical expression can still be obtained for the sequence of breakpoints $\lambda^{(t)}$ and iii) the solution in intervals $[\lambda^{(t+1)}, \lambda^{(t)}]$ can still be found analytically. The presence of free variables then impacts initialization and somewhat complicates all analytical expressions. Box constraints increase the number of tests that need to be computed in order to predict the next breakpoint. Note that the homotopy method with box constraints was recently proposed in [28], which also established convergence proofs. We first obtain optimality conditions for \mathcal{Q}_{2+1} in Section 3.1. Then, the algorithm is built in Section 3.2, and it is adapted to problems $\mathcal{Q}_{2/1}$ and $\mathcal{Q}_{1/2}$ in Section 3.3. Implementation details are discussed in Section 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 consider equivalently the optimization problem of

the form:

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

where index sets \mathbb{S}^1 and $\bar{\mathbb{S}}$ are defined in Section 2.3. The function J is differentiable with $\nabla J(\mathbf{x}) = -\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x})$. The subdifferentials of $h(\mathbf{x})$ and of $g_i(\mathbf{x})$ are respectively:

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

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 a minimizer of (1) if and only if there exists $\boldsymbol{\pi} \in \mathbb{R}^n$ such that $(\mathbf{x}^*, \boldsymbol{\pi})$ satisfies the Karush-Kuhn-Tucker optimality conditions applied to continuous, convex, non-differentiable functions (see for example [39]):

$$\begin{cases} 0 \in -\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^*) + \lambda \partial h(\mathbf{x}^*) + \partial \sum_{i=1}^n \pi_i g_i(\mathbf{x}^*) & (4a) \\ g_i(\mathbf{x}^*) \leq 0 \quad \forall i = 1, \dots, n & (4b) \\ \pi_i \geq 0 \quad \forall i = 1, \dots, n & (4c) \\ \pi_i g_i(\mathbf{x}^*) = 0 \quad \forall i = 1, \dots, n. & (4d) \end{cases}$$

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

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

$$|\mathbf{A}_{\bar{\mathbb{S}}_0}^T(\mathbf{y} - \mathbf{A}_{\mathbb{S}^1} \mathbf{x}_{\mathbb{S}^1}^* - \mathbf{A}_{\bar{\mathbb{S}}} \mathbf{x}_{\bar{\mathbb{S}}}^*)| < \lambda. \quad (5a)$$

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

$$-\mathbf{A}_{\bar{\mathbb{S}}_{in}}^T(\mathbf{y} - \mathbf{A}_{\mathbb{S}^1} \mathbf{x}_{\mathbb{S}^1}^* - \mathbf{A}_{\bar{\mathbb{S}}} \mathbf{x}_{\bar{\mathbb{S}}}^*) + \lambda \text{sgn}(\mathbf{x}_{\bar{\mathbb{S}}_{in}}^*) = 0. \quad (5b)$$

- (3) Let $\bar{\mathbb{S}}_{\square} := \{i \in \bar{\mathbb{S}} \mid |x_i^*| = M\}$. From equation (4c), $\boldsymbol{\pi}_{\bar{\mathbb{S}}_{\square}} \geq 0$ and using equa-

tions (3) and (4a), optimality conditions of $\mathbf{x}_{\bar{S}_\square}^*$ become:

$$-\mathbf{A}_{\bar{S}_\square}^T (\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}^*) + \lambda \text{sgn}(\mathbf{x}_{\bar{S}_\square}^*) + \boldsymbol{\pi}_{\bar{S}_\square} \odot \text{sgn}(\mathbf{x}_{\bar{S}_\square}^*) = 0, \quad \boldsymbol{\pi}_{\bar{S}_\square} \geq 0, \quad (5c)$$

where \odot denotes the Hadamard (entrywise) product.

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

$$-\mathbf{A}_{S_{in}^1}^T (\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}^*) = 0. \quad (5d)$$

- (5) Let $S_\square^1 := \{i \in S^1 \mid |\mathbf{x}_i^*| = M\}$. From equation (4c), $\boldsymbol{\pi}_{S_\square^1} \geq 0$ and using equations (3) and (4a), optimality conditions of $\mathbf{x}_{S_\square^1}^*$ become:

$$-\mathbf{A}_{S_\square^1}^T (\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}^* - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}^*) + \boldsymbol{\pi}_{S_\square^1} \odot \text{sgn}(\mathbf{x}_{S_\square^1}^*) = 0, \quad \boldsymbol{\pi}_{S_\square^1} \geq 0. \quad (5e)$$

Let us remark that equations (5b), (5c) and (5e) 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 problem (1). Let $\mathbf{r} := \mathbf{y} - \mathbf{A}_{\bar{S}_\square} \mathbf{x}_{\bar{S}_\square}^* - \mathbf{A}_{S_\square^1} \mathbf{x}_{S_\square^1}^*$, where each component in $\mathbf{x}_{\bar{S}_\square}$ and $\mathbf{x}_{S_\square^1}$ equals $\pm M$. Equations (5b) and (5d) are linear systems in $\mathbf{x}_{\bar{S}_{in}}^*$ and $\mathbf{x}_{S_{in}^1}^*$, whose solution is:

$$\begin{cases} \mathbf{x}_{\bar{S}_{in}}^* = \left(\mathbf{A}_{\bar{S}_{in}}^T (\mathbf{I} - \mathbf{P}^{S_{in}^1}) \mathbf{A}_{\bar{S}_{in}} \right)^{-1} \left(\mathbf{A}_{\bar{S}_{in}}^T (\mathbf{I} - \mathbf{P}^{S_{in}^1}) \mathbf{r} - \lambda \text{sgn}(\mathbf{x}_{\bar{S}_{in}}^*) \right), & (6a) \\ \mathbf{x}_{S_{in}^1}^* = (\mathbf{A}_{S_{in}^1}^T \mathbf{A}_{S_{in}^1})^{-1} (\mathbf{A}_{S_{in}^1}^T \mathbf{r} - \mathbf{A}_{S_{in}^1}^T \mathbf{A}_{\bar{S}_{in}} \mathbf{x}_{\bar{S}_{in}}^*), & (6b) \end{cases}$$

where $\mathbf{P}^{S_{in}^1} := \mathbf{A}_{S_{in}^1} (\mathbf{A}_{S_{in}^1}^T \mathbf{A}_{S_{in}^1})^{-1} \mathbf{A}_{S_{in}^1}^T$ and \mathbf{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 (recall that 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 (5a)–(5e) is violated. The algorithm works as follows:

- (1) First, it is clear that as $\lambda \rightarrow +\infty$, ℓ_1 -norm-penalized variables $\mathbf{x}_{\bar{S}}$ are zero. In that case, other variables \mathbf{x}_{S^1} are found by solving the least-squares problem:

$$\min_{-M \leq \mathbf{x}_{S^1} \leq M} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}\|_2^2. \quad \text{We note } \mathbf{x}^{(0)} \text{ the vector defined by}$$

$$\begin{cases} \mathbf{x}_{S^1}^{(0)} := \underset{-M \leq \mathbf{x}_{S^1} \leq M}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}\|_2^2, & (7a) \\ \mathbf{x}_{\bar{S}}^{(0)} := \mathbf{0}. & (7b) \end{cases}$$

Equation (5a) shows that $\mathbf{x}^{(0)}$ is the solution of the problem (1) as long as $\lambda \geq \lambda^{(0)}$, with:

$$\lambda^{(0)} := \|\mathbf{A}_{\bar{S}}^T(\mathbf{y} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}^{(0)})\|_{\infty}. \quad (7c)$$

- (2) As λ decreases below $\lambda^{(0)}$, indices $j \in \bar{S}$ such that $|\mathbf{a}_j^T(\mathbf{y} - \mathbf{A}_{S^1}^T \mathbf{x}_{S^1}^{(0)})| = \lambda^{(0)}$ 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:

$$\begin{cases} \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \gamma^{(k)} \mathbf{d}^{(k)} & (8a) \\ \text{and } \lambda^{(k)} = \lambda^{(k-1)} - \gamma^{(k)}, & (8b) \end{cases}$$

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 (6a) and (6b), the direction $\mathbf{d}^{(k)}$ is obtained by:

$$\begin{cases} \mathbf{d}_{\bar{S}_{in}}^{(k)} = (\mathbf{A}_{\bar{S}_{in}}^T (\mathbf{I} - \mathbf{P}^{S_{in}^1}) \mathbf{A}_{\bar{S}_{in}})^{-1} \text{sgn}(\mathbf{x}_{\bar{S}_{in}}^{(k-1)}) & (9a) \\ \mathbf{d}_{S_{in}^1}^{(k)} = -(\mathbf{A}_{S_{in}^1}^T \mathbf{A}_{S_{in}^1})^{-1} \mathbf{A}_{S_{in}^1}^T \mathbf{A}_{\bar{S}_{in}} \mathbf{d}_{\bar{S}_{in}}^{(k)} & (9b) \\ d_i^{(k)} = 0 \quad \forall i \notin \{\bar{S}_{in} \cup S_{in}^1\}, & (9c) \end{cases}$$

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 notations:

$$\begin{aligned} \mathbf{t}^{(k-1)} &:= \mathbf{y} - \mathbf{A}_{\bar{S}} \mathbf{x}_{\bar{S}}^{(k-1)} - \mathbf{A}_{S^1} \mathbf{x}_{S^1}^{(k-1)}, & \mathbf{u}^{(k)} &:= \mathbf{A}_{\bar{S}_{in}} \mathbf{d}_{\bar{S}_{in}}^{(k)} + \mathbf{A}_{S^1} \mathbf{d}_{S^1}^{(k)}, \\ \mathbf{v}^{(k-1)} &:= \mathbf{A}^T \mathbf{t}^{(k-1)}, & \mathbf{w}^{(k)} &:= \mathbf{A}^T \mathbf{u}^{(k)}. \end{aligned} \quad (10)$$

- (a) A component with index $\ell \in \bar{S}_0$ becomes nonzero when equality in equation (5a) is reached. Inserting equations (6a) and (6b) into equation (5a), one can show that it may become positive (respectively, negative) when:

$$\gamma = \frac{\lambda^{(k-1)} + v_{\ell}^{(k-1)}}{1 - w_{\ell}^{(k)}} \left(\text{respectively, when } \gamma = \frac{-\lambda^{(k-1)} + v_{\ell}^{(k-1)}}{-1 - w_{\ell}^{(k)}} \right). \quad (11a)$$

- (b) A component with index $\ell \in \bar{S}_{in}$ becomes zero. From equation (8a), this may occur when:

$$\gamma = \frac{-x_{\ell}^{(k-1)}}{d_{\ell}^{(k)}}. \quad (11b)$$

- (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 (8a), this may occur when:

$$\gamma = \frac{M \operatorname{sgn}(x_\ell^{(k-1)}) - x_\ell^{(k-1)}}{d_\ell^{(k)}}. \quad (11c)$$

- (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 (5c), which yields:

$$\gamma = \frac{\operatorname{sgn}(x_\ell^{(k-1)})\lambda^{(k-1)} - v_\ell^{(k-1)}}{\operatorname{sgn}(x_\ell^{(k-1)}) - w_\ell^{(k)}}. \quad (11d)$$

- (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 (5e), which yields:

$$\gamma = \frac{-v_\ell^{(k-1)}}{w_\ell^{(k)}}. \quad (11e)$$

The shortest step size $\gamma^{(k)}$ is then defined as the shortest positive step among all possible ones, defined by equations (11a)–(11e). 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 \mathbf{x}^* for $\lambda = \lambda^*$ is found by:

$$\mathbf{x}^* = \mathbf{x}^{(k-1)} + \gamma^* \mathbf{d}^{(k)}, \quad (12)$$

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

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

Set $k = 0$. Initialize $\mathbf{x}^{(0)}$ and $\lambda^{(0)}$ by equations (7a)–(7c).

while $\lambda^{(k)} > \lambda^*$ **do**

$k \leftarrow k + 1$.

 Update $\mathbf{d}^{(k)}$ by equations (9a)–(9b).

 Determine the step size $\gamma^{(k)}$ as the smallest positive value among all cases in equations (11a)–(11e).

 Compute accordingly $(\mathbf{x}^{(k)}, \lambda^{(k)})$ by equations (8a)–(8b).

 Update index sets $\{\bar{S}_{in}, S_{in}^1, \bar{S}_\square, S_\square^1, \bar{S}_0\}$.

end

Compute \mathbf{x}^* by equation (12).

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

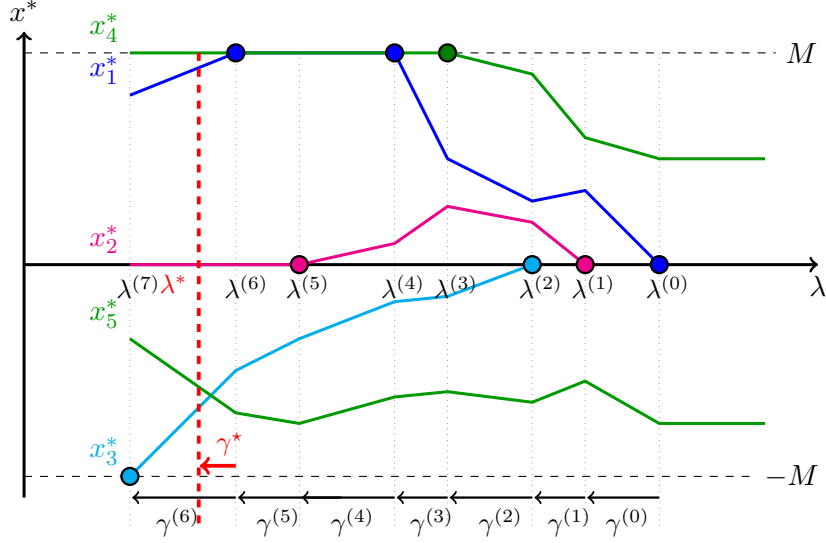


Figure 1. Example of solution path \mathbf{x}^* giving the solutions of the problem (1) as a function of λ , with 5 variables: $\bar{S} = \{1, 2, 3\}$ and $S^1 = \{4, 5\}$. Circles represent the events that cause a change in the support configuration. Vertical dotted lines represent the breakpoints.

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 $\|\mathbf{x}_{\bar{S}}^*\|_1$ is continuously increased and the least-squares function $\frac{1}{2}\|\mathbf{y} - \mathbf{A}_{\bar{S}}\mathbf{x}_{\bar{S}}^* - \mathbf{A}_{S^1}\mathbf{x}_{S^1}^*\|_2^2$ is continuously decreased. 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 such that 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 (12). By construction, there is no sign change between $\mathbf{x}^{(k-1)}$ and the optimal solution \mathbf{x}^* such that $\|\mathbf{x}^*\|_1 = \tau$. One can then easily show that the value of γ such that $\|\mathbf{x}^*\|_1 = \tau$ is:

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

- 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 (12) 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:

$$\gamma^* := \frac{\mathbf{t}^{(k-1)T} \mathbf{u}^{(k)} - \sqrt{(\mathbf{t}^{(k-1)T} \mathbf{u}^{(k)})^2 - \mathbf{u}^{(k)T} \mathbf{u}^{(k)} (\mathbf{t}^{(k-1)T} \mathbf{t}^{(k-1)} - 2\epsilon)}}{\mathbf{u}^{(k)T} \mathbf{u}^{(k)}},$$

where $\mathbf{t}^{(k-1)}$ and $\mathbf{u}^{(k)}$ are defined in equation (10).

3.4. Implementation and practical issues

Some practical remarks concerning the numerical implementation of the homotopy algorithm 2 are detailed in this section.

First, each iteration mostly consists of solving linear systems of equations (9a)–(9b), whose size respectively corresponds to the current number of variables in \bar{S}_{in} and S_{in}^1 . By construction, the support configuration only slightly changes between two breakpoints: \bar{S}_{in} and S_{in}^1 are generally modified by at most one element at each breakpoint, corresponding to the activation of one condition among equations (11a)–(11e). Therefore, the matrix inverses in equations (9a)–(9b) can be computed recursively by performing rank-one updates. In our simulations, using the *block matrix inversion formulas* [37] appeared to be the most efficient strategy.

We also note that, for each non-zero component with index $\ell \in \bar{S}_{in}$, only one of the two computations defined by equations (11b) and (11c) is necessary. In particular, if $x_\ell^{(k-1)} > 0$ and $d_\ell^{(k)} > 0$ (respectively, $d_\ell^{(k)} < 0$), then the only possible change in the support configuration is when x_ℓ hits the upper bound M (respectively, $x_\ell = 0$), corresponding to case 2c in Section 3.2 (respectively, to case 2b). A similar reasoning applies to negative components.

We conclude this section with an important remark concerning the 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 $\mathcal{P}_{0/2}$. This lower bound is compared to the global upper bound, say z^u , provided by the best-known feasible solution. Since z^u is discrete, it is clear that the node can be pruned if the corresponding lower bound exceeds $z^u - 1$. With the homotopy method, the objective function in $\mathcal{Q}_{1/2}$ monotonically increases at each iteration. Therefore, the homotopy algorithm is stopped and the node is pruned as soon as the ℓ_1 norm of the current iterate exceeds $z^u - 1$, as shown in Figure 2.

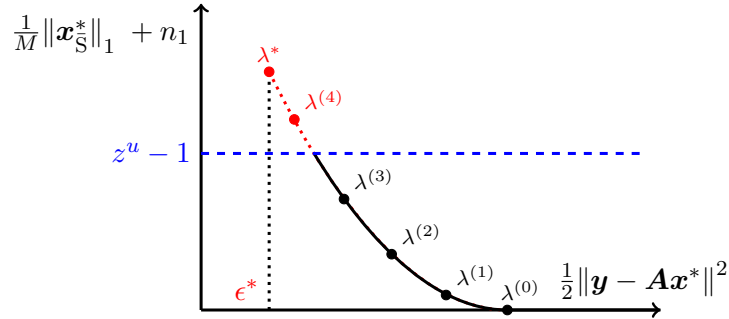


Figure 2. Set of optimal solutions in the coordinate system $(\frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}^*\|_2^2, \frac{1}{M}\|\mathbf{x}_S^*\|_1 + n_1)$ as a function of λ , and illustration of a stopping criterion of the homotopy algorithm for $\mathcal{Q}_{1/2}$. The algorithm can be stopped (and the corresponding node is pruned) as soon as $\frac{1}{M}\|\mathbf{x}_S^*\|_1 + n_1 \geq z^u - 1$, where z^u is the cardinality of the best-known feasible solution to $\mathcal{P}_{0/2}$ at a given iteration of the branch-and-bound procedure.

4. Experimental results

In this section, we evaluate the performance of the proposed algorithm, that we name $\text{B\&B}_{\text{sparse}}$, where the continuous relaxation algorithms built in Section 3 are inserted into the branch-and-bound implementation described in Section 2.1, for the resolution of the three problems $\mathcal{P}_{2/0}$, $\mathcal{P}_{0/2}$, and \mathcal{P}_{2+0} .

Test cases and implementation details are first described in Section 4.1. Then, in Section 4.2, we evaluate the quality of the solutions obtained by our algorithm, compared to several state-of-the-art sparse estimation algorithms. Finally, a comparison of computing times between our approach and the MIP resolution with CPLEX is provided in Section 4.3.

4.1. Experimental setup

We consider synthetic subset selection problems with random entries following the simulation setup proposed in [3]. The rows \mathbf{a}_i of matrix \mathbf{A} are independently drawn from an n -dimensional multivariate centered normal distribution: $\mathbf{a}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, with covariance matrix $\mathbf{\Sigma}$ whose entries equal $\sigma_{ij} = \rho^{|i-j|}$, $(i, j) \in \{1, \dots, n\}^2$, where ρ controls the correlation level between close columns of \mathbf{A} . Columns are then scaled to have unit ℓ_2 norm. We consider two values $\rho = 0.8$ and $\rho = 0.9$, corresponding to medium and high correlation levels. The higher ρ , the more difficult the estimation problem. Sparse vectors \mathbf{x}^0 are generated with equidistant non-zero coordinates (rounding non-integer values when necessary), and the corresponding amplitudes are set to one. Data are generated as $\mathbf{y} = \mathbf{A}\mathbf{x}^0 + \boldsymbol{\xi}$, where $\boldsymbol{\xi}$ contains samples of a white noise process with normal distribution $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, and σ controls the signal to noise ratio $\text{SNR} := \|\mathbf{A}\mathbf{x}^0\|_2^2 / (m\sigma^2)$, which is fixed to 7.

We set $m = 500$ and $n \in \{100; 500; 1000\}$, and the cardinality $\|\mathbf{x}^0\|_0$ varies from 3 to 50 for $\rho = 0.8$, and from 3 to 30 for $\rho = 0.9$. For problems $\mathcal{P}_{2/0}$, parameter K is set to the true value $\|\mathbf{x}^0\|_0$. 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 [7] for details): ϵ is tuned such that the probability $P(\|\boldsymbol{\xi}\|_2^2 \leq \epsilon) = 95\%$, and $\lambda = 2\sigma^2 \log(1/\phi - 1)$, where $\phi = \|\mathbf{x}^0\|_0/n$. As proposed in [7], the bound M on the absolute values of the coefficients is set to $M = 1.1\|\mathbf{A}^T \mathbf{y}\|_\infty$. B&B_{sparse} is implemented in C++ and all methods are executed on a UNIX machine equipped with 16 Go RAM and Intel i7-8650U central processing units (CPUs) clocked at 1.9 GHz. Computations are restricted to one core in order to focus on the algorithm performance, disabling parallelization capacities.

4.2. Quality of estimated solutions

In this section, we evaluate the ability of our algorithm to recover the exact solution. Only the formulation $\mathcal{P}_{2/0}$ is considered, and B&B_{sparse} is compared to the following sparse estimation methods:

- convex regularization by the ℓ_1 norm or BPDN (Basis Pursuit De-Noising), computed here by the homotopy algorithm [44];
- Subspace Pursuit (SP) [34];
- A^* Orthogonal Matching Pursuit (A^* OMP) [25];
- Continuation Single Best Replacement (CSBR) [43];
- nonconvex regularization by the CEL0 penalty [41], locally optimized by an IRL1 strategy (code provided by the author), referred to as CEL0;
- nonconvex regularization by the $\ell_{1/2}$ -norm penalty, locally optimized by the proximal algorithm provided with paper [46], referred to as $\ell_{1/2}$.

On the addressed problems, the running time of such methods time never exceeds 0.1 s for BPDN, SP, $\ell_{1/2}$ and CEL0, whereas the highest computation times for CSBR

and A^* OMP is respectively about 3s and 10s for the largest problems. Let us recall that all these algorithms are able to cope with higher-dimensional problems, but do not provide any guarantee on the global optimality of their solutions. Conversely, $B\&B_{\text{sparse}}$ is theoretically guaranteed to compute an optimal solution to $\mathcal{P}_{2/\rho}$, but requires more computation time. In order to evaluate intermediate solutions found by $B\&B_{\text{sparse}}$, we consider several values of the maximum time, say T_{max} , allowed for each resolution: 1, 10, 100, and 1000 s. If the maximum time is reached, then the current solution is considered, which is the best solution found, without any optimality guarantee.

All algorithms are tuned so that solutions have the true number of non-zero components. We focus on the capacity of the methods to correctly estimate the sparsity pattern of \mathbf{x}^0 . For each value of n , ρ and K , the exact recovery rate ER is computed, which is the average number of simulations for which the algorithm correctly identifies the true support. In the presence of noise, even the global optimum does not necessarily identify the true nonzero components of \mathbf{x}^0 . In particular, for a given value of K and ρ , as n decreases, atoms in the true solution become more correlated, therefore identifying the true support becomes harder from an informational point of view. Averaged results over 20 instances are reported in Figure 3.

For all algorithms, the exact recovery rate naturally decreases as the sparsity level increases. When allowed $T_{\text{max}} = 1000$ s, $B\&B_{\text{sparse}}$ always gives better solutions than the other approaches, except for $n = 100$, where $\ell_{1/2}$ and CSBR slightly outperform $B\&B_{\text{sparse}}$ for the highest values of K . Among the competitors, BPDN always performs worse and CSBR performs best. On the smallest problems with $n = 100$, most methods (except BPDN and CEL0 which perform worse) have similar performance, where ER quickly decays as K increases (from $K = 15$ for $\rho = 0.8$ and from $K = 9$ for $\rho = 0.9$).

For $n \in \{500; 1000\}$, $B\&B_{\text{sparse}}$ always performs better than other methods as long as $T_{\text{max}} \geq 10$ s for $\rho = 0.8$, and $T_{\text{max}} \geq 1$ s for $\rho = 0.9$. With $n = 500$ and $\rho = 0.8$, $B\&B_{\text{sparse}}$ always finds the right support in less than 1 s up to $K = 21$ (as also does CSBR); in less than 10 s, this is the only algorithm which still achieves ER = 100% up to $K = 27$. Most of the other methods achieve ER = 100% up to $K = 15$ only. Similar conclusions can be drawn for $n = 500$ and $\rho = 0.9$, although the decrease in ER occurs at lower values of K . For the largest test problems with $n = 1000$, 100% exact recovery is achieved by $B\&B_{\text{sparse}}$ (even in 1 s) and CSBR up to $K = 21$ for $\rho = 0.8$, and up to $K = 12$ for $\rho = 0.9$. For higher K , $B\&B_{\text{sparse}}$ outperforms all competitors as long as $T_{\text{max}} \geq 10$ s (respectively $T_{\text{max}} \geq 1$ s) for $\rho = 0.8$ (respectively, for $\rho = 0.9$).

These results show that, if obtaining a proof on the global optimality is not a critical point, following our global optimization strategy, even restricted to partial tree exploration by limiting the computation time, may provide a competitive alternative to other existing sparse estimation methods (which may often converge toward a local optimum) in a reasonable amount of time—1 s to 10 s on these problem classes. Nevertheless, in order to certify the solution, it is necessary to let the algorithm run until convergence. In the next section, we focus on the computing time of *guaranteed* methods, by comparing $B\&B_{\text{sparse}}$ with the resolution of different MIP formulations by the CPLEX solver.

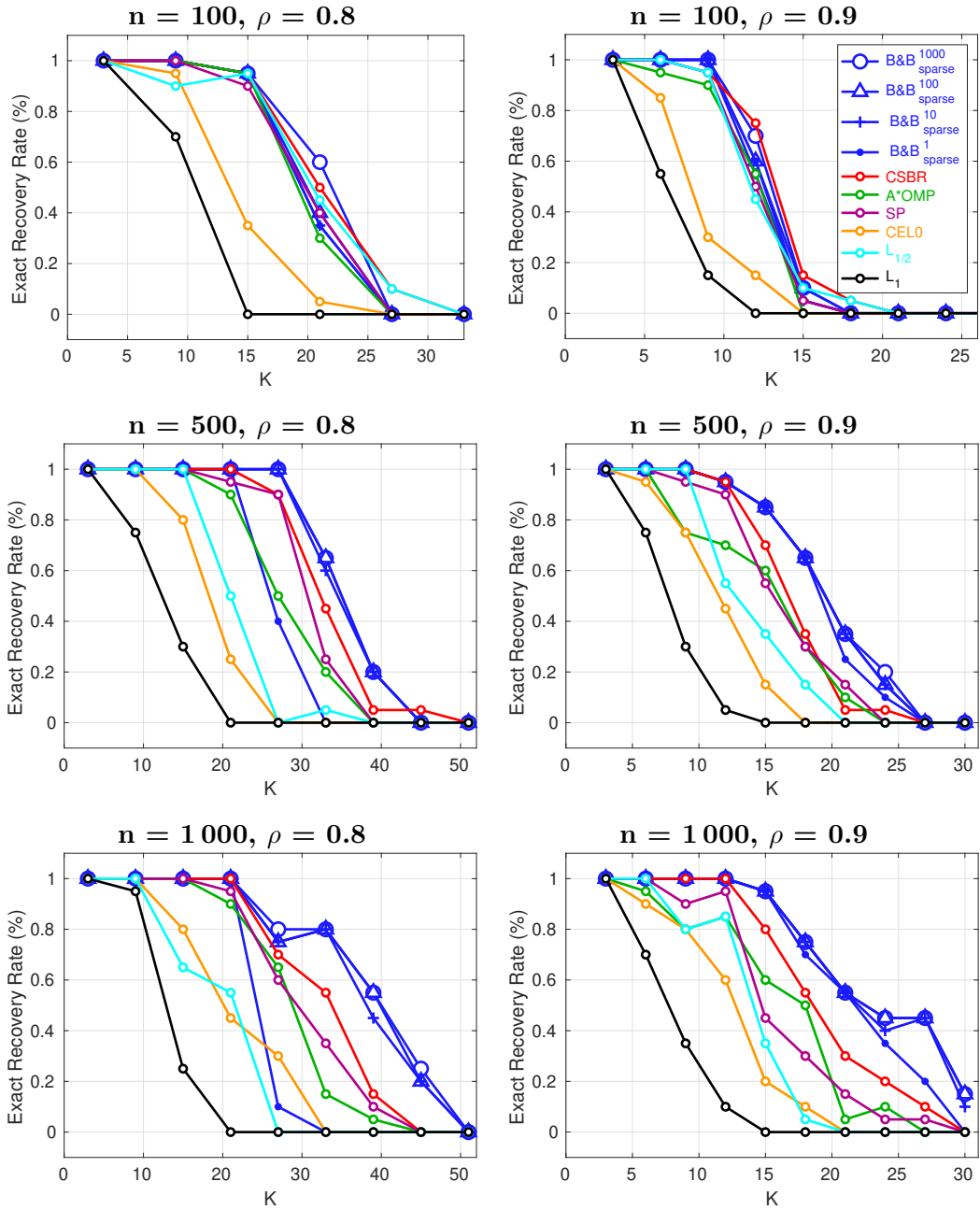


Figure 3. Quality of solutions obtained by $B\&B_{\text{sparse}}$ by limiting the computation time to 1, 10, 100 and 1000 s, compared to other state-of-the-art sparse estimation methods: exact recovery rate as a function of the cardinality of the solution for medium (left) and high (right) correlation levels of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables. Results are averaged over 20 instances.

4.3. Computing times

We finally compare the computing times of our algorithm $\text{B\&B}_{\text{sparse}}$ to other global optimization approaches based on different reformulation techniques:

- $\text{CPLEX}_{\text{bigM}}$: we use CPLEX v12.8 to solve the MIP problems based on *bigM* constraints and given in Table 1;
- $\text{CPLEX}_{\text{SOS}}$: we use CPLEX v12.8 to solve the formulation proposed in [3] based on specially ordered set (SOS) constraints. Originally, this reformulation was solved in [3] with Gurobi, which is a generic global optimization solver equivalent to CPLEX;
- $\text{CPLEX}_{\text{SOS}}^M$: we use CPLEX v12.8 to solve the SOS approach, where box constraints are added to the continuous variables with the same value M as for $\text{B\&B}_{\text{sparse}}$ and $\text{CPLEX}_{\text{bigM}}$.

The maximum time allowed for all methods is set to 1000 s, and all computations with CPLEX are run with its default settings. An additional reformulation introducing variables $\mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x}$ was necessary for solving underdetermined problems of the form $\mathcal{P}_{2/0}$ with CPLEX. For other problems, such a reformulation resulted less efficient and was therefore not considered.

Note that the only available code that was found implementing a specific solver for this kind of problems is the BBL algorithm¹ associated to [23], which considers the formulation \mathcal{P}_{2+0} with an additional quadratic regularization term. As mentioned in [23], this ℓ_2 regularization plays a key role in the resolution performance. Here, in order to solve the same global optimization problems as addressed in this paper, the quadratic weight was set to 0 and, as could be expected, BBL revealed to be poorly efficient. Therefore, we did not include it in our results.

Figures 4, 5 and 6 show the computation times of each method for 10 random instances of each problem with $n \in \{100; 500; 1000\}$ and $\rho \in \{0.8; 0.9\}$, as a function of K , until no instance could be solved in 1000s, for the three formulations $\mathcal{P}_{2/0}$, \mathcal{P}_{2+0} , and $\mathcal{P}_{0/2}$, respectively. Tables 3, 4 and 5 give the corresponding number of instances that were successfully solved in less than 1000 s, the average computation times and numbers of explored nodes, for $\text{B\&B}_{\text{sparse}}$ and $\text{CPLEX}_{\text{bigM}}$, which is the best competing approach.

As could be expected, all computation times quickly increase with K . Indeed, as K increases, the number of explored nodes increases (due to combinatorial complexity) and the computing time per node also increases, especially for $\text{B\&B}_{\text{sparse}}$ due to the structure of our relaxation algorithm: sparser solutions to the ℓ_0 -norm problems involve sparser solutions to the ℓ_1 -norm-based relaxed problems, for which the homotopy algorithm requires fewer iterations.

$\text{CPLEX}_{\text{bigM}}$ performs better than $\text{CPLEX}_{\text{SOS}}$ and $\text{CPLEX}_{\text{SOS}}^M$ on all instances. For $n = 100$, with formulations $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} , $\text{B\&B}_{\text{sparse}}$ is much faster than $\text{CPLEX}_{\text{bigM}}$ up to $K = 12$ for $\rho = 0.8$, and up to $K = 9$ for $\rho = 0.9$. Then, for higher K , $\text{CPLEX}_{\text{bigM}}$ becomes more efficient and solves more instances than $\text{B\&B}_{\text{sparse}}$. Both methods can solve some instances of $\mathcal{P}_{2/0}$ up to $K = 24$ ($\rho = 0.8$) and to $K = 12$ ($\rho = 0.9$). Averaging computing times over instances that could be solved by the two methods, $\text{B\&B}_{\text{sparse}}$ is about 30 times faster than $\text{CPLEX}_{\text{bigM}}$ for $\mathcal{P}_{2/0}$ and 50 times faster for \mathcal{P}_{2+0} . For $\mathcal{P}_{0/2}$, both methods are able to solve some instances up to $K = 15$ ($\rho = 0.8$) and to $K = 9$ ($\rho = 0.9$), but $\text{B\&B}_{\text{sparse}}$ is always much faster.

For $n = 500$ and $n = 1000$, $\text{B\&B}_{\text{sparse}}$ outperforms every CPLEX formulation on

¹<https://github.com/alisaab/10bnb>

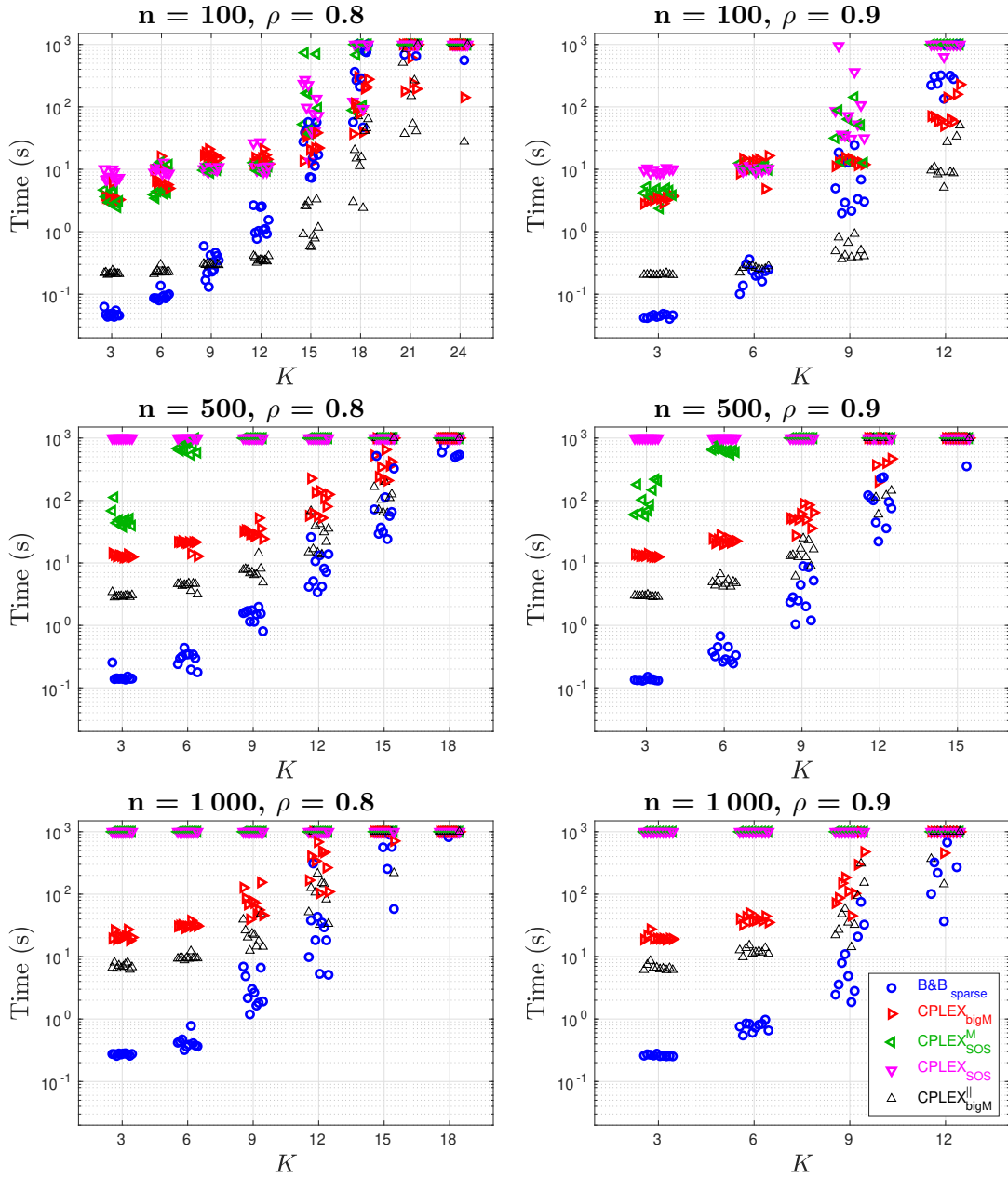


Figure 4. Exact optimization of $\mathcal{P}_{2/0}$: computation times for $B\&B_{\text{sparse}}$ and for different MIP formulations solved with CPLEX as a function of the solution cardinality, for medium (left) and high (right) correlation levels ρ of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables. Each marker corresponds to one instance, and points are split horizontally around the corresponding cardinality level.

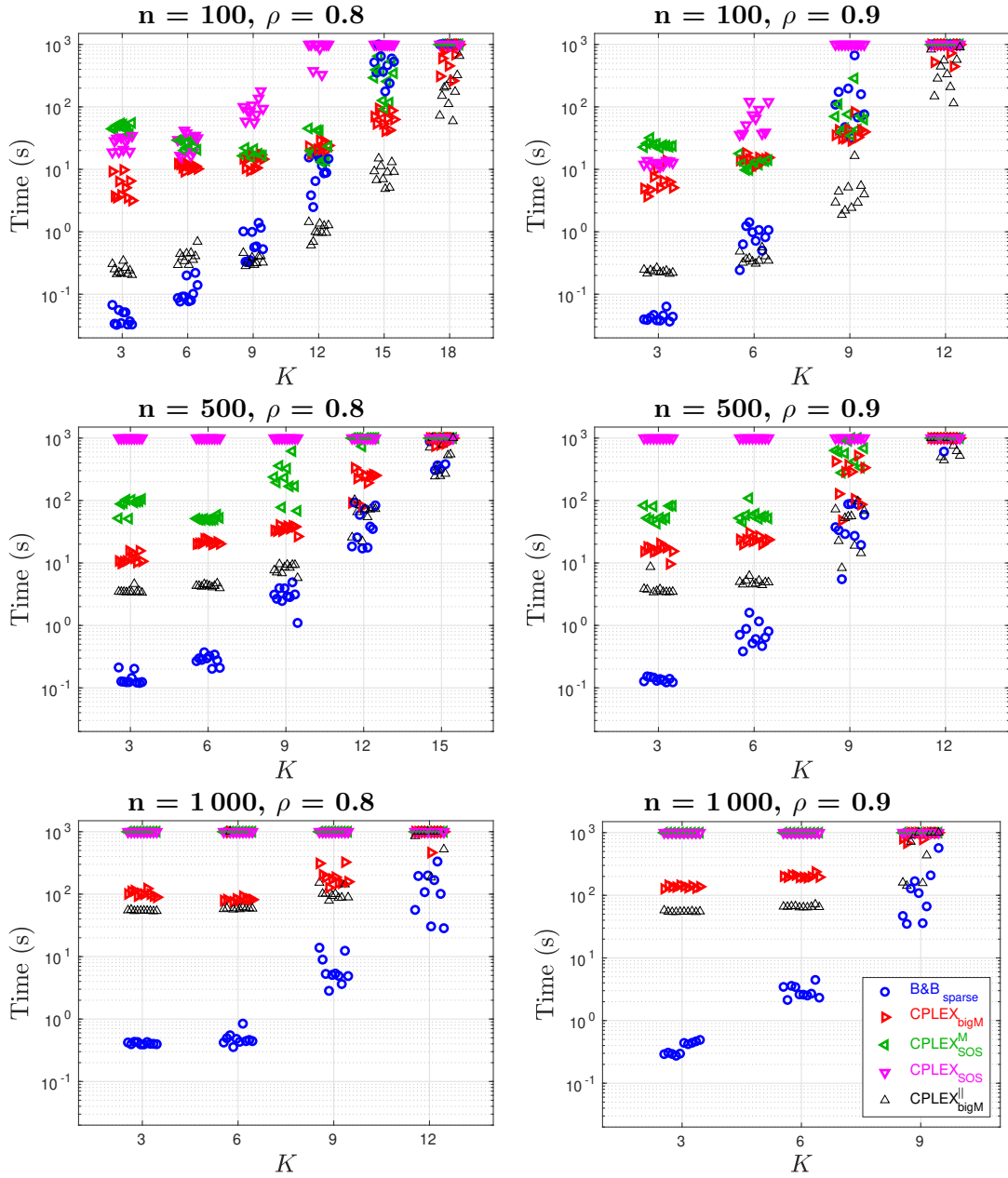


Figure 5. Exact optimization of \mathcal{P}_{2+0} : computation times for $B\&B_{\text{sparse}}$ and for different MIP formulations solved with CPLEX as a function of the solution cardinality, for medium (left) and high (right) correlation levels ρ of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables. Each marker corresponds to one instance, and points are split horizontally around the corresponding cardinality level.

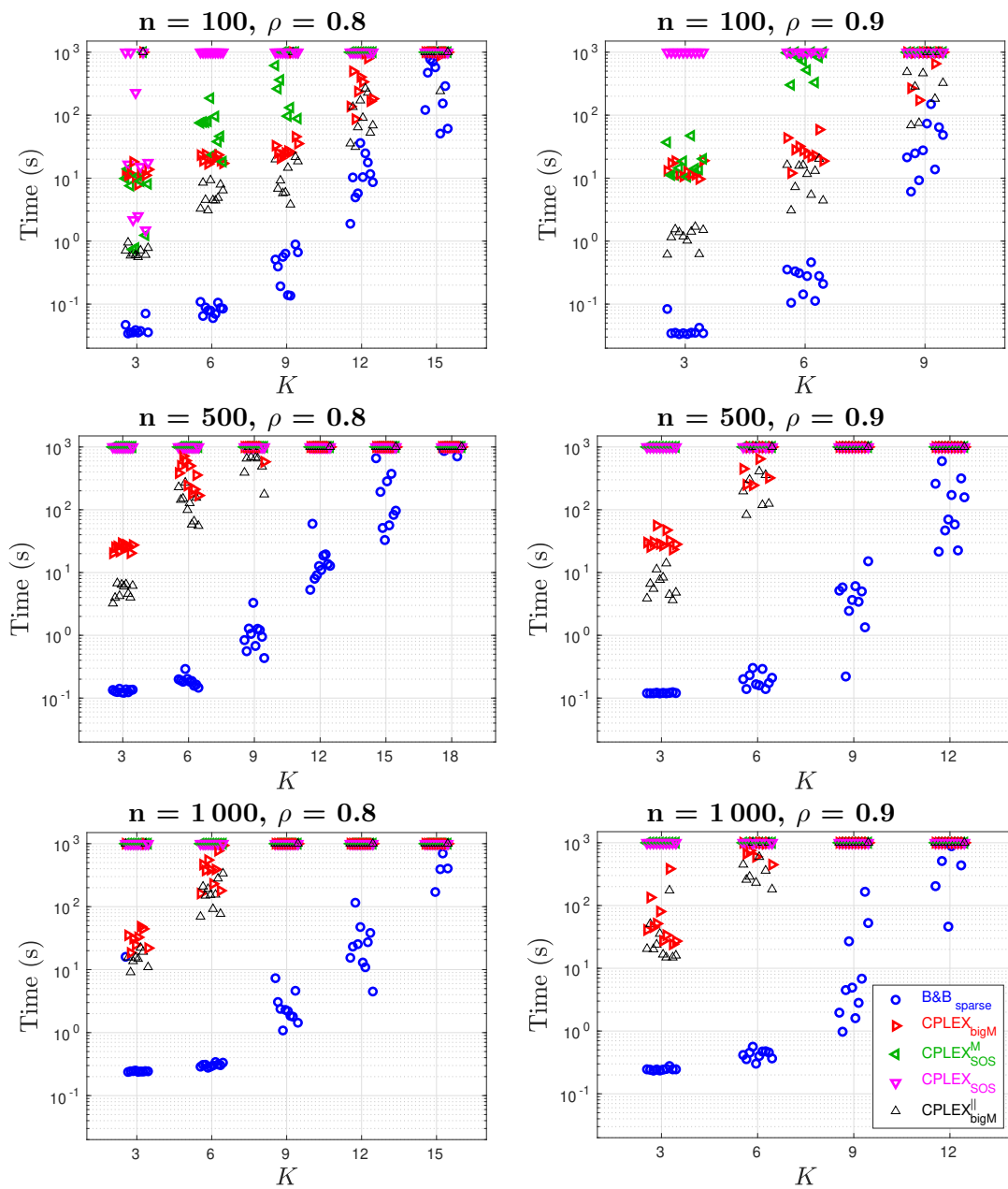


Figure 6. Exact optimization of $\mathcal{P}_{0/2}$: computation times for $B\&B_{\text{sparse}}$ and for different MIP formulations solved with CPLEX as a function of the solution cardinality, for medium (left) and high (right) correlation levels ρ of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables. Each marker corresponds to one instance, and points are split horizontally around the corresponding cardinality level.

Table 3. Exact optimization of $\mathcal{P}_{2/0}$: computation times for B&B_{sparse} and CPLEX_{bigM} as a function of the cardinality of the solution for medium (left) and high (right) correlation levels ρ of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables, averaged over 10 instances. The column ‘S’ indicates the number of instances that were successfully solved in less than 1000 s. Computing times and numbers of explored nodes are averaged over instances which could be solved in less than 1000 s.

Problem $\mathcal{P}_{2/0}$	$\rho = 0.8$						$\rho = 0.9$					
	B&B _{sparse}			CPLEX _{bigM}			B&B _{sparse}			CPLEX _{bigM}		
	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)
	$n = 100$											
$K = 3$	10	0.05	0.01	10	3.71	0	10	0.04	0.01	10	3.27	0
$K = 6$	10	0.09	0.01	10	7.50	0.01	10	0.22	0.04	10	12.3	0.04
$K = 9$	10	0.32	0.05	10	16.3	0.06	10	8.10	1.52	10	13.2	1.27
$K = 12$	10	1.51	0.22	10	14.6	0.31	8	347	56.3	10	96.0	118
$K = 15$	10	28.3	4.17	10	25.5	8.25	–	–	–	–	–	–
$K = 18$	8	347	50.8	10	145	206	–	–	–	–	–	–
$K = 21$	2	671	108	4	306	535	–	–	–	–	–	–
$K = 24$	1	558	80.1	1	142	241	–	–	–	–	–	–
	Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 27.3$						Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 32.9$					
	$n = 500$											
$K = 3$	10	0.15	0.01	10	13.0	0	10	0.14	0.01	10	13.0	0
$K = 6$	10	0.30	0.02	10	19.9	0.02	10	0.37	0.04	10	22.8	0.05
$K = 9$	10	1.48	0.12	10	32.3	0.15	10	3.90	0.41	10	56.4	0.49
$K = 12$	10	9.57	0.69	10	104	0.91	10	107	9.61	4	358	4.86
$K = 15$	10	127	7.22	8	370	3.82	1	353	27.3	–	–	–
$K = 18$	5	582	34.5	–	–	–	–	–	–	–	–	–
	Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 37.5$						Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 44.0$					
	$n = 1000$											
$K = 3$	10	0.27	0.01	10	21.3	0.00	10	0.26	0.01	10	20.3	0.00
$K = 6$	10	0.43	0.02	10	31.4	0.02	10	0.76	0.05	10	39.7	0.06
$K = 9$	10	3.28	0.15	10	77.4	0.18	10	16.3	1.00	10	251	1.06
$K = 12$	10	51.3	2.04	9	336	1.23	6	270	14.6	1	454	2.62
$K = 15$	5	469	17.1	1	709	2.51	–	–	–	–	–	–
$K = 18$	1	820	28.7	–	–	–	–	–	–	–	–	–
	Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 36.5$						Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 36.8$					

all test problems. It also successfully solves more instances in 1000 s. For $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} , B&B_{sparse} is faster than CPLEX_{bigM} by a factor of 35 to 50, and of even more than 100 for \mathcal{P}_{2+0} and $n = 1000$. This ratio increases for more correlated problems ($\rho = 0.9$) and for larger problems ($n = 1000$). Results are even much more contrasted

Table 4. Exact optimization of \mathcal{P}_{2+0} : computation times for $B\&B_{\text{sparse}}$ and $CPLEX_{\text{bigM}}$ as a function of the cardinality of the solution for medium (left) and high (right) correlation levels ρ of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables, averaged over 10 instances. The column ‘S’ indicates the number of instances that were successfully solved in less than 1000 s. Computing times and numbers of explored nodes are averaged over instances which could be solved in less than 1000 s.

Problem	$\rho = 0.8$						$\rho = 0.9$					
	$B\&B_{\text{sparse}}$			$CPLEX_{\text{bigM}}$			$B\&B_{\text{sparse}}$			$CPLEX_{\text{bigM}}$		
	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)
	$n = 100$											
$K = 3$	10	0.04	0.01	10	5.48	0.00	10	0.04	0.01	10	6.78	0.00
$K = 6$	10	0.12	0.04	10	11.0	0.07	10	0.87	0.37	10	14.4	0.55
$K = 9$	10	0.72	0.29	10	13.4	0.74	10	157	54.4	10	41.0	44.4
$K = 12$	10	10.3	3.19	10	22.1	7.15	–	–	–	3	563	1556
$K = 15$	9	432	109	10	64.8	94.3	–	–	–	–	–	–
$K = 18$	–	–	–	7	562	1389	–	–	–	–	–	–
	Average $T_{CPLEX_{\text{bigM}}}/T_{B\&B_{\text{sparse}}} = 48.6$						Average $T_{CPLEX_{\text{bigM}}}/T_{B\&B_{\text{sparse}}} = 57.8$					
	$n = 500$											
$K = 3$	10	0.14	0.01	10	12.0	0.00	10	0.14	0.01	10	16.3	0.01
$K = 6$	10	0.29	0.07	10	21.4	0.10	10	0.78	0.26	10	23.6	0.30
$K = 9$	10	3.10	0.85	10	35.3	1.29	10	47.4	8.43	10	260	11.9
$K = 12$	10	46.0	10.8	10	222	17.7	1	607	181	–	–	–
$K = 15$	6	539	119	3	777	82.2	–	–	–	–	–	–
	Average $T_{CPLEX_{\text{bigM}}}/T_{B\&B_{\text{sparse}}} = 35.4$						Average $T_{CPLEX_{\text{bigM}}}/T_{B\&B_{\text{sparse}}} = 51.8$					
	$n = 1000$											
$K = 3$	10	0.41	0.01	10	103	0.01	10	0.37	0.01	10	137	0.01
$K = 6$	10	0.49	0.08	9	82.1	0.09	10	2.98	0.43	10	202	0.45
$K = 9$	10	6.72	1.37	10	197	1.81	9	152	22.0	3	752	5.93
$K = 12$	9	135	23.0	1	462	7.07	–	–	–	–	–	–
	Average $T_{CPLEX_{\text{bigM}}}/T_{B\&B_{\text{sparse}}} = 113$						Average $T_{CPLEX_{\text{bigM}}}/T_{B\&B_{\text{sparse}}} = 147$					

with problems $\mathcal{P}_{0/2}$, where $CPLEX_{\text{bigM}}$ only solves the sparsest problems: almost no instance is solved for $K > 6$, whereas $B\&B_{\text{sparse}}$ can solve instances up to $K = 15$ ($\rho = 0.8$) and to $K = 12$ ($\rho = 0.9$).

For $\mathcal{P}_{2/0}$, the number of explored nodes is in general smaller for $B\&B_{\text{sparse}}$ than for $CPLEX_{\text{bigM}}$, revealing the efficiency of our tree exploration strategy. A similar conclusion holds for \mathcal{P}_{2+0} and $\mathcal{P}_{0/2}$, although the number of explored nodes for $B\&B_{\text{sparse}}$ (and therefore the computing time) is often superior to those obtained with $\mathcal{P}_{2/0}$ for the same problem complexity. One possible explanation holds in the fact that the exploration strategy, which first branches on active variables ($b_i = 1$), more efficiently limits the depth of the tree for cardinality-constrained problems. Most of all, on all instances, thanks to our specific relaxation algorithm, the time dedicated to each node

Table 5. Exact optimization of $\mathcal{P}_{0/2}$: computation times for B&B_{sparse} and CPLEX_{bigM} as a function of the cardinality of the solution for medium (left) and high (right) correlation levels ρ of the columns of \mathbf{A} , and for problems involving $n = 100$ (top), $n = 500$ (center) and $n = 1000$ (bottom) variables, averaged over 10 instances. The column ‘S’ indicates the number of instances that were successfully solved in less than 1000 s. Computing times and numbers of explored nodes are averaged over instances which could be solved in less than 1000 s.

Problem $\mathcal{P}_{0/2}$	$\rho = 0.8$						$\rho = 0.9$					
	B&B _{sparse}			CPLEX _{bigM}			B&B _{sparse}			CPLEX _{bigM}		
	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)	S	Time (s)	Nodes ($\times 10^3$)
	$n = 100$											
$K = 3$	9	0.04	0.01	9	11.9	0.19	10	0.04	0.01	10	13.5	0.95
$K = 6$	10	0.08	0.02	10	20.3	7.32	10	0.26	0.08	10	29.0	4.18
$K = 9$	9	0.46	0.13	9	28.9	4.58	10	43.9	12.1	4	519	191
$K = 12$	10	13.2	3.27	10	380	113	–	–	–	–	–	–
$K = 15$	10	416	94.7	1	882	179	–	–	–	–	–	–
	Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 126$						Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 154$					
	$n = 500$											
$K = 3$	10	0.13	0.01	10	25.0	0.19	10	0.12	0.01	10	32.9	0.56
$K = 6$	10	0.19	0.02	10	386	20.0	10	0.20	0.03	5	382	26.1
$K = 9$	10	1.16	0.15	1	580	10.8	10	4.82	0.91	–	–	–
$K = 12$	10	17.0	1.83	–	–	–	10	172	21.9	–	–	–
$K = 15$	9	204	19.2	–	–	–	–	–	–	–	–	–
$K = 18$	2	787	60.3	–	–	–	–	–	–	–	–	–
	Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 908$						Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 1086$					
	$n = 1000$											
$K = 3$	10	1.81	0.14	8	32.1	0.07	10	0.25	0.01	10	84.7	0.33
$K = 6$	10	0.31	0.01	10	445	1.50	10	0.43	0.03	5	685	1.60
$K = 9$	10	2.80	0.17	–	–	–	10	26.8	2.16	–	–	–
$K = 12$	10	32.1	1.37	–	–	–	6	509	24.2	–	–	–
$K = 15$	4	417	14.9	–	–	–	–	–	–	–	–	–
	Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 736$						Average $T_{\text{CPLEX}_{\text{bigM}}}/T_{\text{B\&B}_{\text{sparse}}} = 976$					

evaluation is reduced by several orders of magnitude. For $\mathcal{P}_{0/2}$, both the number of nodes explored by CPLEX_{bigM} and the computing time per node become much higher than with the two other formulations, and largely exceed those of B&B_{sparse}, revealing the inefficiency of CPLEX_{bigM} to tackle quadratically-constrained problems. Note that this analysis is not always reflected in the results of Tables 4 to 6, because averages are performed only over instances which could be solved in less than 1000 s by each algorithm.

The formulation CPLEX_{SOS}^M is always less efficient than CPLEX_{bigM}, except for the smallest problems ($n = 100$, low K), where the two approaches perform rather

similarly. For $n = 500$, it only solves instances of $\mathcal{P}_{2/0}$ (respectively, of \mathcal{P}_{2+0}) up to $K = 6$ (respectively, $K = 9$), and no instance of $\mathcal{P}_{0/2}$ could be solved. For $n = 1\,000$, no instance could be solved by $\text{CPLEX}_{\text{SOS}}^{\text{M}}$. Although $\text{CPLEX}_{\text{bigM}}$ and $\text{CPLEX}_{\text{SOS}}^{\text{M}}$ solve exactly the same MIP problem, introducing the box constraints under the *bigM* formulation ($|x_i| \leq M|b_i|$) is therefore more efficient than SOS constraints, especially when the dimension of the problem increases.

Finally, as could be expected, $\text{CPLEX}_{\text{SOS}}$ is even less efficient than $\text{CPLEX}_{\text{SOS}}^{\text{M}}$. It is only able to solve the sparsest instances of problems with $n = 100$, up to $K = 18$ for $\mathcal{P}_{2/0}$ and $\rho = 0.8$, and to $K = 12$ otherwise. No instance with $n = 500$ and $n = 1\,000$ could be solved by $\text{CPLEX}_{\text{SOS}}$. We note that in all our experiments where $\text{CPLEX}_{\text{SOS}}$ could be run successfully, the same solution was found than in the box-constrained case, thus proving the validity of the value of M (solving the bounded problem did not affect the solution).

We conclude this section by a comment regarding parallelization. Figures 4, 5 and 6 also show the computing times obtained by solving the bigM formulations in Table 1, allowing parallelization of CPLEX over 8 cores (recall that, up to now, $\text{B\&B}_{\text{sparse}}$ and all CPLEX implementations were restricted to one core). The computing times of $\text{CPLEX}_{\text{bigM}}$ are obviously reduced, especially for $n = 100$. For formulations $\mathcal{P}_{2/0}$ and \mathcal{P}_{2+0} and for $n = 100$, the multicore version $\text{CPLEX}_{\text{bigM}}^{\parallel}$ outperforms $\text{B\&B}_{\text{sparse}}$ for $K \geq 9$. In all other cases, however, $\text{B\&B}_{\text{sparse}}$ is still more—and often much more—efficient, although it does not exploit any parallelization.

5. Conclusion

We proposed exact optimization algorithms for least squares problems with low cardinality, based on a dedicated branch-and-bound strategy. It was shown that such algorithms can be designed without resorting to MIP reformulations, removing the corresponding binary variables. An algorithm was built for solving the relaxation problems involved at any node of the search tree, which were recast as convex non-smooth optimization problems, involving the ℓ_1 norm of some components and box constraints. This algorithm can be applied with similar efficiency to the relaxation problems involved in the resolution of the three addressed formulations: cardinality-constrained and cardinality-penalized least-squares, and cardinality minimization under quadratic constraints.

Depending on the problem formulation and on the correlation level between the columns of the dictionary \mathbf{A} , solutions were successfully computed with optimality proof in less than 1 000 seconds on simulated noisy subset selection problems with $m = 500$ data points, up to cardinality $K \in [9, 24]$ for $n = 100$ variables, $K \in [12, 18]$ for $n = 500$, and $K \in [9, 15]$ for $n = 1\,000$. Our $\text{B\&B}_{\text{sparse}}$ strategy was compared with the resolution of the same MIP formulations with the generic solver CPLEX, under the same box-constrained assumption—which resulted much more efficient than solving SOS-based formulations. For cardinality-constrained problems (the most frequently encountered formulation in the exact optimization literature) and $n = 100$, CPLEX performs quite competitively and it is more efficient than $\text{B\&B}_{\text{sparse}}$ for higher cardinalities. For $n \in \{500; 1\,000\}$, however, $\text{B\&B}_{\text{sparse}}$ always outperforms CPLEX: it can solve instances with higher cardinality, and it is more than 30 times faster in average. Similar conclusions hold for cardinality-penalized problems. In particular, $\text{B\&B}_{\text{sparse}}$ is more than 100 times faster than CPLEX for $n = 1\,000$. Although this formulation may

be of lower interest in practice, the proposed algorithm can still provide a benchmarking reference for evaluating the quality of local optimization methods which specifically consider the penalized form. For the quadratically constrained formulation, which is of major interest in many applications, results are even more in favor of $B\&B_{\text{sparse}}$, where CPLEX could only solve problems with very small cardinality, on which $B\&B_{\text{sparse}}$ was 100 to 1 000 times faster.

This work therefore contributes to pushing the limits of computationally tractable low-cardinality least-squares problems that can be addressed from a global optimization perspective. Although the computational complexity of the proposed algorithms is incomparably higher than that of standard sparse estimation methods, which limits their applicability to moderate-size problems, our results also show that most computational effort is dedicated to *proving* optimality. In particular, combining a deep-first-search strategy with variable selection rules which first activate the most explanatory variables, quickly guides the search to good feasible solutions. Therefore, the optimal solution may be found at a much earlier step. Solutions obtained by limiting our algorithm to a smaller computing time, although not guaranteed, can also represent competitive alternatives to standard sparse estimation methods.

The superiority of $B\&B_{\text{sparse}}$ over generic MIP resolution was achieved by exploiting mathematical properties of the problem, which are not considered by a generic solver. Further works may be developed following the same guideline. Building tighter relaxations, *e.g.* based on Lagrangian relaxation or on non-convex relaxations of the ℓ_0 norm, developing more sophisticated tree search strategies, for example exploiting greedy local exploration algorithms [42] or building dedicated cutting-plane methods are some possibilities. The three formulations considered in the paper could also be tackled jointly from a multi-objective optimization perspective, involving dedicated branch-and-bound techniques [17]. Finally, parallelization may also represent an important lever for accelerating our method. In this paper, multicore architecture was not exploited in the implementation of $B\&B_{\text{sparse}}$, whereas it was shown to significantly reduce the computing times of the CPLEX solver. Adapting different possible parallelization strategies to the specificity of sparse optimization problems is also a perspective of interest.

Data availability statement

The C++ code of our solver for the resolution of the three formulations is publicly available at <https://github.com/ramzi-benmhenni/BBsparse>.

Problem instances and results corresponding to Section 4.2 are available at <https://box.ec-nantes.fr/index.php/s/CwgPyHcftipJgMD>.

Problem instances and results corresponding to Section 4.3 are available at <https://box.ec-nantes.fr/index.php/s/fmYWYnEDWRqBLYn>.

Acknowledgements

The authors would like to thank Emmanuel Soubies for sharing his IRL1 code for optimization of the CEL0-regularized least-squares cost function.

Funding

This work has been partially funded by the French national research agency (ANR), project ANR-16-CE33-0005.

References

- [1] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, *Optimization with sparsity-inducing penalties*, Foundations and Trends in Machine Learning (2011).
- [2] R. Ben Mhenni, S. Bourguignon, M. Mongeau, J. Ninin, and H. Carfantan, *Sparse Branch and Bound for Exact Optimization of L_0 -Norm Penalized Least Squares*, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 5735–5739.
- [3] D. Bertsimas, A. King, and R. Mazumder, *Best subset selection via a modern optimization lens*, The Annals of Statistics 44 (2016), pp. 813–852.
- [4] D. Bertsimas and R. Shioda, *Algorithm for cardinality-constrained quadratic optimization*, Computational Optimization and Applications 43 (2009), pp. 1–22.
- [5] D. Bienstock, *Computational study of a family of mixed-integer quadratic programming problems*, Mathematical Programming 74 (1996), pp. 121–140.
- [6] T. Blumensath and M. Davies, *Iterative thresholding for sparse approximations*, Journal of Fourier Analysis and Applications 14 (2008), pp. 629–654.
- [7] 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.
- [8] 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.
- [9] O. Burdakov, C. Kanzow, and A. Schwartz, *On a Reformulation of Mathematical Programs with Cardinality Constraints*, in *Advances in Global Optimization*, D. Gao, N. Ruan, and W. Xing, eds., Cham. Springer International Publishing, 2015, pp. 3–14.
- [10] E. Candès, M. Wakin, and S. Boyd, *Enhancing sparsity by reweighted l_1 minimization*, Journal of Fourier Analysis and Applications 14 (2007), pp. 877–905.
- [11] R. Chartrand and Wotao Yin, *Iteratively reweighted algorithms for compressive sensing*, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 3869–3872.
- [12] S.S. Chen, D.L. Donoho, and M.A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing 20 (1998), pp. 33–61.
- [13] 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.
- [14] D. DiLorenzo, G. Liuzzi, F. Rinaldi, F. Schoen, and M. Sciandrone, *A concave optimization-based approach for sparse portfolio selection*, Optimization Methods and Software 27 (2012), pp. 983–1000.
- [15] 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), pp. 4789–4812.
- [16] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, *Least angle regression*, The Annals of Statistics 32 (2004), pp. 407–499.
- [17] M. Ehrgott, *Multicriteria Optimization*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [18] M. Elad, *Sparse and Redundant Representations. From Theory to Applications in Signal and Image Processing*, Springer-Verlag New York, 2010.
- [19] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.

- [20] A. Frangioni and C. Gentile, *Perspective cuts for a class of convex 0–1 mixed integer programs*, Mathematical Programming 106 (2006), pp. 225–236.
- [21] J. Gao and D. Li, *Optimal cardinality constrained portfolio selection*, Operations Research 61 (2013), pp. 745–761.
- [22] I.F. Gorodnitsky and B.D. Rao, *Sparse signal reconstruction from limited data using FO-CUSS: a re-weighted minimum norm algorithm*, IEEE Transactions on Signal Processing 45 (1997), pp. 600–616.
- [23] H. Hazimeh, R. Mazumder, and A. Saab, *Sparse regression at scale: Branch-and-bound rooted in first-order optimization*, arXiv preprint, arXiv:2004.06152 (2020).
- [24] K. Herrity, A. Gilbert, and J. Tropp, *Sparse approximation via iterative thresholding*, in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 3. IEEE, 2006, pp. 624–627.
- [25] N.B. Karahanoglu and H. Erdogan, *A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery*, Digital Signal Processing 22 (2012), pp. 555 – 568.
- [26] H. Le Thi, T. Pham Dinh, H. Le, and X. Vo, *Dc approximation approaches for sparse optimization*, European Journal of Operational Research 244 (2015), pp. 26 – 46.
- [27] 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.
- [28] X. Liang and Y. Wang, *Homotopy algorithm for box-constrained LASSO and its convergence*, International Journal of Pure and Applied Mathematics 112 (2017), pp. 333–340.
- [29] Z. Lu and Y. Zhang, *Sparse approximation via penalty decomposition methods*, SIAM Journal on Optimization 23 (2013), pp. 2448–2478.
- [30] J.M. Mendel, *Optimal Seismic Deconvolution*, Academic Press, 1983.
- [31] A. Miller, *Subset selection in regression*, Chapman and Hall/CRC, 2002.
- [32] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, *A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm*, IEEE Transactions on Signal Processing 57 (2009), pp. 289–301.
- [33] B.K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing 24 (1995), pp. 227–234.
- [34] D. Needell and J.A. Tropp, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Applied and Computational Harmonic Analysis 26 (2009), pp. 301–321.
- [35] M.S. O’Brien, A.N. Sinclair, and S.M. Kramer, *Recovery of a sparse spike time series by $L1$ norm deconvolution*, IEEE Transactions on Signal Processing 42 (1994), pp. 3353–3365.
- [36] 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).
- [37] K.B. Petersen and M.S. Pedersen, *The matrix cookbook* (2012). Available at <http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html>, Version 20121115.
- [38] F. Rinaldi, F. Schoen, and M. Sciandrone, *Concave programming for minimizing the zero-norm over polyhedral sets*, Computational Optimization and Applications 46 (2008), pp. 467–486.
- [39] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
- [40] 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.
- [41] E. Soubies, L. Blanc-Féraud, and G. Aubert, *A continuous exact ℓ_0 penalty (CEL0) for least squares regularized problem*, SIAM Journal on Imaging Science 8 (2015), pp. 1607–1639.
- [42] 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.
- [43] C. Soussen, J. Idier, J. Duan, and D. Brie, *Homotopy based algorithms for ℓ_0 -regularized least-squares*, IEEE Transactions on Signal Processing 63 (2015), pp. 3301–3316.
- [44] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society, Series B 58 (1996), pp. 267–288.

- [45] 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.
- [46] F. Wen, L. Chu, P. Liu, and R. Qiu, *A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning*, IEEE Access 6 (2018), pp. 69883–69906.
- [47] C. Zala, *High-resolution inversion of ultrasonic traces*, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control 39 (1992), pp. 458–463.