A variable fixing version of the two-block nonlinear constrained Gauss-Seidel algorithm for ℓ_1-regularized least-squares

Margherita Porcelli[*]and Francesco Rinaldi[†]

November 7, 2013

Abstract

The problem of finding sparse solutions to underdetermined systems of linear equations is very common in many fields as e.g. in signal/image processing and statistics. A standard tool for dealing with sparse recovery is the ℓ_1-regularized least-squares approach that has recently attracted the attention of many researchers.

In this paper, we describe a new version of the two-block nonlinear constrained Gauss-Seidel algorithm for solving ℓ_1-regularized least-squares that at each step of the iteration process fixes some variables to zero according to a simple active-set strategy. We prove the global convergence of the new algorithm and we show its efficiency reporting the results of some preliminary numerical experiments.

Keywords Gauss-Seidel Algorithm, Active Set, Sparse Approximation, ℓ_1-regularized least-squares.

1 Introduction

We address the solution of the following $\ell_2 - \ell_1$ unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1, \tag{1}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ $(m < n)$, $\lambda \in \mathbb{R}^+$, $\|\cdot\|_2$ denotes the standard Euclidean norm and $\|\cdot\|_1$ stands for the ℓ_1 norm defined as $\|x\|_1 = \sum_{i=1}^n |x_i|$.

Problem (1) is known in the literature as *Basis Pursuit Denoising* (see e.g. [5], [9]) and is strictly related to the *Least Absolute Shrinkage and Selection Operator (LASSO)*, a widely-studied problem in statistics, described for the first time by Tibshirani in [23]:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 \\ &s.t. \quad \|x\|_1 \le \tau, \end{aligned} \tag{2}$$

[*]Università di Bologna, Dipartimento di Matematica, Piazza di Porta S. Donato 5, 40127 Bologna, Italy. Email: margherita.porcelli@unibo.it

[†]Università di Padova, Dipartimento di Matematica, Via Trieste 63, 35121 Padova, Italy. Email: rinaldi@math.unipd.it.

where τ is a nonnegative real parameter.

Since the presence of the ℓ_1 term can induce sparsity in the optimal solution of the optimization problems (1) and (2) [2], these formulations can be used to identify sparse approximate solutions to the large underdetermined linear system of equations

$$Ax = b.$$

In the last decades Problem (1) has become increasingly popular in statistics, signal/image analysis and optimization and various fast algorithms have been proposed for solving it (see e.g. [2, 4, 14, 18, 21, 28, 24] and references therein). A further interesting application of Problem (1) is *Compressed Sensing*. Compressed sensing basically consists in encoding a large sparse signal using a relatively small number of linear measurements, and minimizing the ℓ_1-norm in order to decode the signal (see e.g. [6, 8, 7, 12] and references therein).

The Basis Pursuit Denoising problem can be easily transformed into a convex quadratic problem, with linear inequality constraints, and then solved using a constrained optimization method as e.g. the Projected Conjugate Gradient method or the Interior-Point method as proposed in [21]. These general purpose techniques represent a good way to get a reliable solution with little programming efforts, but they might become inefficient when applied to large-scale problems. This is the reason why many special purpose class of methods have been proposed for solving Problem (1) as e.g. the Iterative Shrinkage-Thresholding (IST) algorithms [2, 4, 10, 28], the Decomposition methods [24] and the Augmented-Lagrangian methods [1].

Recently, *active-set* strategies have become a valid alternative approach for ℓ_1-regularized minimization due to their robustness. In e.g. [26, 27] a two-stage algorithm (called FPC-AS) has been proposed for solving problems of the form (1). In the first stage of FPC-AS, an estimation of the subset of components of x likely to be nonzero in an optimal solution is made by using a first-order iterative "shrinkage" method. Then, the ℓ_1-norm $\|x\|_1$ is reduced to a linear function of x by fixing the components of x to the estimated "active-set" and fixing their signs at their current values. This yields to the second stage, when a subspace problem involving the minimization of a smaller and smooth quadratic function is solved by means of a second-order method.

The aim of the present paper is to bring a further contribution in this research field by proposing a fast and easy to implement first-order algorithm for solving Problem (1). In particular, we first transform the original nondifferentiable unconstrained Problem (1) into a differentiable problem with simple constraints and we use a classical variable splitting approach to construct an exact penalty function for Problem (1). Then, we describe a *two-block Nonlinear Constrained Gauss-Seidel (NLCGS) algorithm* (see e.g. [3, 16, 17]) for its solution that can be interpreted as an Iterative Shrinkage-Thresholding algorithm with constant stepsize (see e.g. [2, 4, 10, 28] and references therein). Finally, we propose a variable fixing variant of the two-block NLCGS algorithm, called two-block VF-NLCGS, that belongs to the class of active-set methods. The new algorithm preserves the convergence properties of the original one but enhances its efficiency by solving reduced problems consisting in separable constrained optimization problems with closed-form solution to be solved only for a small subset of variables. In fact, at each iteration, the two-block VF-NLCGS algorithm selects a suitable subset of variables according to a simple rule defining the current active-set, keeps those variables fixed to zero and solves the resulting problem in the subspace of free (inactive) variables.

The paper is organized as follows. In Section 2, we analyze the theoretical properties of Problem (1). Furthermore, we define an exact penalty function of the problem which will be used for formally deriving the proposed algorithms and for establishing convergence results. Then, in Section 3 we introduce the two-block NLCGS algorithm and the variable fixing variant VF-NLCGS, and study their convergence properties. In Section 4, we report a numerical experience showing that the approach is competitive with other existing methods for ℓ_1-regularized minimization. Finally, in Section 5, we draw some conclusions.

2 Problem formulation

Consider the ℓ_1-regularized least-squares problem (1). As a preliminary remark, note that, since the objective function is coercive, all level sets are compact and therefore the problem admits optimal solutions.

Introducing the variable $w \in \mathbb{R}^n$, Problem (1) can be rewritten into the equivalent linearly-constrained form

$$\begin{aligned} &\min_{x\in\mathbb{R}^n, w\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T w \\ &\qquad s.t. \quad -w \le x \le w, \end{aligned} \tag{3}$$

where $e \in \mathbb{R}^n$ is the vector with all entries 1, which corresponds to the following Lagrangian function

$$L(x,w,u,v) = \frac{1}{2}\|Ax-b\|_2^2 + \lambda e^T w + u^T(-x-w) + v^T(x-w), \tag{4}$$

with $u, v \in \mathbb{R}^n$ vectors of Lagrangian multipliers. Let f denote the quadratic term in the objective function of Problem (1), i.e.

$$f(x) = \frac{1}{2}\|Ax-b\|_2^2,$$

and let g be the gradient of f, that is

$$g(x) = A^T(Ax-b).$$

Using this notation we can formulate the Karush-Kuhn-Tucker (KKT) conditions for Problem (3).

Proposition 1. *The point $(x^\star, w^\star)$ is a solution of Problem (3) if and only if there exist KKT multipliers $u^\star \in \mathbb{R}^n$ and $v^\star \in \mathbb{R}^n$ satisfying the conditions:*

$$\begin{array}{l} \nabla_x L \equiv g(x^\star) - u^\star + v^\star = 0 \\ \nabla_w L \equiv \lambda e - u^\star - v^\star = 0 \\ u^{\star T}(x^\star + w^\star) = 0 \\ v^{\star T}(x^\star - w^\star) = 0 \\ -w^\star \le x^\star \le w^\star \\ u^\star, v^\star \ge 0, \end{array} \tag{5}$$

where $\nabla_x L, \nabla_w L$ denote the partial gradients of L with respect to x and w.

By solving the KKT system (5) with respect to $u^\star, v^\star$, we obtain

$$\begin{array}{l} u^\star = \frac{1}{2}[\lambda e + g(x^\star)], \\ v^\star = \frac{1}{2}[\lambda e - g(x^\star)], \end{array} \tag{6}$$

which allows to write the optimality conditions for Problem (3) as follows:

$$\begin{array}{l} -\lambda \le g_i(x^\star) \le \lambda \\ (\lambda + g_i(x^\star))\,(x_i^\star + w_i^\star) = 0 \\ (\lambda - g_i(x^\star))\,(x_i^\star - w_i^\star) = 0 \\ -w_i^\star \le x_i^\star \le w_i^\star, \end{array} \tag{7}$$

for $i = 1, \dots, n$, or equivalently

$$\begin{array}{l} -\lambda \le g_i(x^\star) \le \lambda \\ g_i(x^\star) w_i^\star + \lambda x_i^\star = 0 \\ g_i(x^\star) x_i^\star + \lambda w_i^\star = 0 \\ -w_i^\star \le x_i^\star \le w_i^\star, \end{array} \tag{8}$$

for $i = 1, \dots, n$. Conditions (7) and (8) are crucial to draw important information on the optimal solution $x^\star$ of Problem (1). Firstly, from (8) we have that

$$\|g(x^\star)\|_\infty \le \lambda. \tag{9}$$

Secondly, again immediately from (8), we get that

$$x^\star = 0 \;\text{ if and only if }\; \|g(0)\|_\infty = \|A^T b\|_\infty \le \lambda. \tag{10}$$

Thirdly, if $\|g(0)\|_\infty > \lambda$, so that $x^\star \neq 0$, from (7) we can easily see that there must exist at least an index $i^\star$ such that $|g_{i^\star}(x^\star)| = \lambda$, and hence, by (9) we have:

$$\text{if } \|g(0)\|_\infty > \lambda, \text{ then } \|g(x^\star)\|_\infty = \lambda. \tag{11}$$

2.1 Construction of an exact penalty function

We now construct an exact penalty function for Problem (1) which will be adopted to formally construct algorithmic schemes and to study their convergence properties.

To this purpose, we first use a variable splitting approach and we write Problem (1) into the equivalent constrained form

$$\begin{array}{c} \displaystyle\min_{x \in \mathbb{R}^n, z \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|z\|_1 \\ \\ s.t. \quad x - z = 0, \end{array} \tag{12}$$

obtained by adding the variable $z \in \mathbb{R}^n$ and the equality constraint $x - z = 0$. Then, we construct an exact penalty function for Problem (12) by means of the Augmented Lagrangian function of Hestenes-Powell (see e.g. [20, 22])

$$L_{HP}(x, z) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 + \mu(x)^T(x - z) + \frac{c}{2}\|x - z\|_2^2, \tag{13}$$

where the Lagrange multiplier is approximated as follows

$$\mu(x) = -g(x),$$

and $c \ge 0$ represents the so-called *penalty parameter*.

We notice that the function L_{HP} can also be seen as the Fletcher's Augmented Lagrangian [15] applied to Problem (12) with z fixed. Using this penalty approach, we obtain the unconstrained problem

$$\min_{x\in\mathbb{R}^n, z\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|z\|_1 - g(x)^T(x-z) + \frac{c}{2}\|x-z\|_2^2. \tag{14}$$

In this section we will prove that, for sufficiently large values of the penalty parameter c, Problem (14) is equivalent to the original problem.

By proceeding as in the previous section, we can eliminate the nondifferentiable term $\|z\|_1$ in the objective function of problem (14), by considering the equivalent problem in (x, z, w)

$$\begin{array}{cc} \displaystyle\min_{x\in\mathbb{R}^n, z\in\mathbb{R}^n, w\in\mathbb{R}^n} & \Psi(x,z,w) = \Phi(x,z) + \lambda e^T w \\ s.t. & -w \le z \le w \end{array} \tag{15}$$

where Φ denotes the quadratic function in x, z defined by

$$\Phi(x,z) = \frac{1}{2}\|Ax-b\|_2^2 - g(x)^T(x-z) + \frac{c}{2}\|x-z\|_2^2. \tag{16}$$

Trivially, the components of the gradient of Φ have the form

$$\nabla_x\Phi(x,z) = c(x-z) - A^TA(x-z) \tag{17}$$

$$\nabla_z\Phi(x,z) = A^T(Ax-b) - c(x-z) \tag{18}$$

and the Hessian matrix $\nabla^2\Phi$ is given by

$$\nabla^2\Phi(x,z) = \begin{pmatrix} cI - A^TA & A^TA - cI \\ A^TA - cI & cI \end{pmatrix}. \tag{19}$$

First, we establish that the objective function Ψ in (15) is convex for sufficiently large values of the penalty parameter c.

Proposition 2. *Let $c > \|A\|_2^2$. Then the function $\Psi(x,z,w)$ in (15) is convex in (x,z,w).*

Proof. Since the term in w is linear and $\lambda > 0$, we must prove that the quadratic function Φ is convex in (x,z). As the assumption made implies that $cI - A^TA$ is positive definite, using the Schur complement theory [19], we have that $\nabla^2\Phi 0$ is positive semi-definite if and only if

$$cI - (A^TA - cI)(cI - A^TA)^{-1}(A^TA - cI) = A^TA,$$

is positive semi-definite, which is obviously true. $\square$

From this result, it follows that the KKT conditions for problem (15) are both necessary and sufficient conditions for optimality. Therefore, we can state the following proposition.

Proposition 3. *Let $c > \|A\|_2^2$. Then the point $(x^\star, z^\star, w^\star)$ is an optimal solution of Problem* (15) *if and only if we have $x^\star = z^\star$ and $(x^\star, w^\star)$ is an optimal solution of Problem* (3).

Proof. Suppose that $(x^\star, z^\star, w^\star)$ is an optimal solution of Problem (15). Then from the KKT conditions for Problem (15), since the problem is unconstrained in x, from (17) we get immediately

$$c(x^\star - z^\star) - A^T A(x^\star - z^\star) = 0,$$

which implies, as $cI - A^T A$ is non singular, that $x^\star = z^\star$. Under this assumption, it is easy to verify that the KKT conditions Problem (15) imply the existence of KKT multipliers $u^\star, v^\star$ and of a vector $w^\star$, such that conditions (5) are satisfied, so that $x^\star$ is an optimal solution of Problem (3). Conversely, if $x^\star, w^\star$ is an optimal solution of Problem (3) then, letting $z^\star = x^\star$ we have that the the KKT conditions for Problem (15) are satisfied. □

3 The two-block Gauss-Seidel Algorithms

In this section we first describe a general two-block Gauss-Seidel Algorithm for solving Problem (3) by using the penalty function approach introduced in the previous section and review its convergence properties. Secondly, we present the two-block Gauss-Seidel Algorithm enriched by a new variable-fixing strategy and prove its global convergence.

3.1 Basic scheme

The main effect of the transformation made in Section 2 is that Problem (15) can now be (formally) solved by a two-block Nonlinear Constrained Gauss-Seidel (NLCGS) algorithm [3]. A general block Nonlinear Gauss-Seidel algorithm is usually applied for the solution of a problem of the form

$$\min_{x \in X} h(x)$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and the feasible set X is the Cartesian product of closed, nonempty and convex subsets $X_i \subseteq \mathbb{R}^{n_i}$ for $i = 1, \ldots, m$, with $\sum_{i=1}^m n_i = n$. If the vector $x \in \mathbb{R}^n$ is partitioned into m component vectors $x_i \in \mathbb{R}^{n_i}$, then a generic iteration of the method is defined as follows:

$$x_i^{k+1} = \arg \min_{y_i \in X_i} h(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y_i, x_{i+1}^k, \ldots, x_m^k),$$

which updates in turns the components of x and, starting from a given starting point $x^0 \in X$, generates a sequence $\{x^k\}$ with $x^k = (x_1^k, \ldots, x_m^k)$. This method is globally convergent for $m = 2$ and for $m > 2$ is proved to converge under suitable assumptions on the objective function h, see [16, 17].

In order to apply the method described above to Problem (15), we partition the problem variables into the two blocks x and (z, w) and we define a conceptual iterative scheme where Φ is given as in (16), see Algorithm 1.

We notice that Algorithm 1 is equivalent to the Iterative Shrinkage-Thresholding Algorithm with constant stepsize, see [2, 4, 10, 28] and references therein. Moreover, as the subproblems (20) and (21) are well defined and the objective function $\Psi(x, z, w)$ is convex in (x, z, w), it is known that the sequence generated by Algorithm 1 converges to a solution $(x^\star, z^\star, w^\star)$ of Problem (15), see [16]. Therefore, recalling Proposition 3, we can state the following result.

Theorem 1. *Let $c > \|A\|_2^2$. Then every limit point $x^\star$ of the sequence $\{x^k\}$ generated by Algorithm 1 is an optimal solution of Problem* (3).

Algorithm 1 Two-block NLCGS

```
Require: A penalty parameter $c > \|A\|_2^2$, a point $x^0 = z^0 \in \mathbb{R}^n$.
  for $k = 1, 2, \dots,$ until convergence do
    Step 1. Fixed $z^k$, compute $x^{k+1}$ solving the problem
        $$\min_{x \in \mathbb{R}^n} \Phi(x, z^k) \qquad (20)$$
    Step 2. Fixed $x^{k+1}$, compute $z^{k+1}$ solving the problem
        $$\min_{z \in \mathbb{R}^n, w \in \mathbb{R}^n} \Psi(x^{k+1}, z, w) = \Phi(x^{k+1}, z) + \lambda e^T w \quad \text{s.t.} \; -w \le z \le w \qquad (21)$$
  end for
```

Now, we focus on the subproblems to be solved at each iteration of Algorithm 1. The solution of the subproblem (20) at Step 1 consists in the computation of a stationary point of Φ with respect to x (z^k is fixed). Recalling Proposition 3 and its proof, we simply get that

$$x^{k+1} = z^k$$

solves problem (20). Therefore, the subproblem (21) at Step 2 takes the form

$$\begin{aligned} \min_{z \in \mathbb{R}^n, w \in \mathbb{R}^n} \quad & -g(z^k)^T (z^k - z) + \frac{c}{2}\|z - z^k\|_2^2 + \lambda e^T w \\ \text{s.t.} \quad & -w \le z \le w, \end{aligned} \tag{22}$$

which can be decomposed into the n independent problems

$$\begin{aligned} \min_{z_i \in \mathbb{R}, w_i \in \mathbb{R}} \quad & -g_i(z^k)(z_i^k - z_i) + \frac{c}{2}\|z_i - z_i^k\|_2^2 + \lambda w_i \\ \text{s.t.} \quad & -w_i \le z_i \le w_i, \end{aligned} \tag{23}$$

for $i = 1, \dots, n$. For each $i \in \{1, \dots, n\}$, the solution of problem (23) can be easily calculated and has the following closed form:

$$z_i^{k+1} = \begin{cases} 0 & \text{if } \dfrac{-\lambda + g_i(z^k)}{c} \le z_i^k \le \dfrac{\lambda + g_i(z^k)}{c} \\ z_i^k - \dfrac{\lambda + g_i(z^k)}{c} & \text{if } z_i^k > \dfrac{\lambda + g_i(z^k)}{c} \\ z_i^k + \dfrac{\lambda - g_i(z^k)}{c} & \text{if } z_i^k < \dfrac{-\lambda + g_i(z^k)}{c} \end{cases}, \quad w_i^{k+1} = |z_i^{k+1}|. \tag{24}$$

3.2 A new variable-fixing strategy

Let z^k be the solution of problem (21) at the kth iteration of Algorithm 1. Consider the index sets E_0^k and E^k that represent the set of zero components at z^k and the set of zero components of z^k that satisfy the KKT conditions for Problem (3), respectively:

$$\begin{array}{rcl} E_0^k & = & E_0(z^k) = \{i \in \{1, \dots n\} \;:\; z_i^k = 0\}, \\ E^k & = & E(z^k) = \{i \in E_0^k \;:\; |g_i(z^k)| \le \lambda\}. \end{array}$$

Then, by using (24), we obtain the following relationship between the zero components of the current z^k and the new corresponding components of z^{k+1}:

$$E^k \subseteq E_0^{k+1}, \tag{25}$$

i.e. if $i \in E^k$, then $z_i^{k+1} = 0$. This result suggests the definition of a version of Algorithm 1 that, at each step, fixes to zero a suitably chosen subset of variables, thus allowing to solve subproblems of reduced dimension and, consequently, to significantly improve the algorithmic performance.

Now, taking into account that for each $i = 1, \dots, n$,

$$|g_i(z^k)| \le \|(A)_i\|_2 \cdot \|Az^k - b\|_2,$$

where $(A)_i$ denotes the ith column of A, we define the sets $I_{\xi^k}^k$ and $I_{\delta^k}^k$ as follows

$$I_{\xi^k}^k \;=\; I_{\xi^k}(z^k) = \{i \in \{1, \dots, n\} \;:\; |z_i^k| \le \xi^k\}, \tag{26}$$

$$I_{\delta^k}^k \;=\; I_{\delta^k}(z^k) = \{i \in I_{\xi^k}^k \;:\; \|(A)_i\|_2 \cdot \|Az^k - b\|_2 \le \lambda + \delta^k\}, \tag{27}$$

where $\xi^k, \delta^k > 0$. We further define the following two sets:

$$\begin{array}{rcl} I_{\delta^k,1}^k & = & \{i \in I_{\xi^k}^k \;:\; \|(A)_i\|_2 \cdot \|Az^k - b\|_2 \le \lambda - \delta^k\}, \\ I_{\delta^k,2}^k & = & \{i \in I_{\xi^k}^k \;:\; \|(A)_i\|_2 \cdot \|Az^k - b\|_2 - \lambda| \le \delta^k\}, \end{array}$$

and it is easy to see that $I_{\delta^k}^k = I_{\delta^k,1}^k \cup I_{\delta^k,2}^k$.

Now, we prove two theoretical results that will be used to define the new variant of Algorithm 1 and prove its convergence properties. In the first one, we prove that, when k is sufficiently large, the set $I_{\xi^k}^k$ contains the set

$$E_0^\star = \{i \in \{1, \dots n\} \;:\; z_i^\star = 0\}. \tag{28}$$

In the second one, we establish that for sufficiently large k, the set $I_{\delta^k,2}^k$ identifies a subset of the set variables that will be zero in the solution $z^\star$, i.e. of the set

$$E^\star = \{i \in E_0^\star \;:\; |g_i(z^\star)| \le \lambda\}. \tag{29}$$

To carry-out the convergence analysis, the use of an identification function in the construction of the set $I_{\xi^k}^k$ and $I_{\delta^k}^k$ will be considered, see [13, 27].

Definition 1. *A function $\rho(z) : \mathbb{R}^n \to \mathbb{R}_+$ is an identification function for $z^\star$ with respect to $\{z^k\}$ if $\rho(z^\star) = 0$ and*

$$\lim_{z^k \to z^\star, z^k \neq z^\star} \frac{\rho(z^k)}{\|z^k - z^\star\|_2} = +\infty.$$

From the definition, it follows that, given an identification function ρ for $z^\star$ with respect to $\{z^k\}$, we have that for k sufficiently large

$$\|z^k - z^\star\|_2 \leq \rho(z^k). \tag{30}$$

Proposition 4. *Let $\{z^k\}$ be a sequence such that $\lim_{k\to\infty} \|z^k - z^\star\|_2 = 0$. Moreover, let $\xi^k = \delta^k = \rho(z^k)$ in (26) and (27), where $\rho(z)$ is an identification function for $z^\star$ with respect to $\{z^k\}$. Then*

$$I^k_{\xi^k} \equiv E^\star_0 \tag{31}$$

for k sufficiently large.

Proof. We first prove that $I^k_{\xi^k} \subseteq E^\star_0$. Let $\nu = \min\{|z^\star_i|, i \text{ s.t } |z^\star_i| > 0\}$ and let $i \in I^k_{\xi^k}$. Since $\lim_{k\to\infty} \rho(z^k) = 0$, for k sufficiently large $\rho(z^k) \leq \nu/4$ and then

$$|z^k_i| \leq \rho(z^k) \leq \nu/4.$$

Since $\lim_{k\to\infty} \|z^k - z^\star\| = 0$, for k sufficiently large

$$|z^k_i - z^\star_i| \leq \|z^k - z^\star\|_2 \leq \nu/4,$$

and therefore we have

$$|z^\star_i| \leq |z^k_i - z^\star_i| + |z^k_i| \leq \nu/2,$$

and finally, from the definition of ν, we get $|z^\star_i| = 0$.

Now we prove that $E^\star_0 \subseteq I^k_{\xi^k}$. From (30), for all $i \in \{1, \dots, n\}$, for k sufficiently large

$$|z^k_i - z^\star_i| \leq \|z^k - z^\star\|_2 \leq \rho(z^k).$$

Let $i \in E^\star_0$, then $i \in I^k_{\xi^k}$ since

$$|z^k_i| = |z^k_i - z^\star_i| \leq \|z^k - z^\star\|_2 \leq \rho(z^k) = \xi^k.$$

□

Proposition 5. *Let $\{z^k\}$ be a sequence such that $\lim_{k\to\infty} \|z^k - z^\star\|_2 = 0$. Moreover, let $\xi^k = \delta^k = \rho(z^k)$ in (26) and (27), where $\rho(z)$ is an identification function for $z^\star$ with respect to $\{z^k\}$. Then*

$$I^k_{\delta^k,2} = \tilde{E}^\star \subseteq E^\star$$

for k sufficiently large, with $\tilde{E}^\star = \{i \in \{1, \dots n\} \ : \ z^\star_i = 0, \ \|(A)_i\|_2 \cdot \|Az^\star - b\|_2 = \lambda\}$.

Proof. Let k be a sufficiently large value, so that Proposition 4 holds. We first prove that $\tilde{E}^\star \subseteq I^k_{\delta^k,2}$. Let $i \in \tilde{E}^\star$, then we have:

$$\begin{aligned} \big|\|(A)_i\|_2 \cdot \|Az^k - b\|_2 - \lambda\big| &= \big|\|(A)_i\|_2 \cdot \|Az^k - b\|_2 - \|(A)_i\|_2 \cdot \|Az^\star - b\|_2\big| \\ &= \|(A)_i\|_2 \big|\|Az^k - b\|_2 - \|Az^\star - b\|_2\big|. \end{aligned}$$

By (30) we get that for k sufficiently large

$$\big|\|(A)_i\|_2 \cdot \|Az^k - b\|_2 - \lambda\big| \leq \|(A)_i\|_2 \|A\|_2 \|z^k - z^\star\| \leq \rho(z^k) = \delta^k,$$

and then by (31) we have that $i \in I^k_{\delta^k,2}$.

We now prove that $I^k_{\delta^k,2} \subseteq \tilde{E}^\star$ for k sufficiently large. Let $i \in I^k_{\delta^k,2}$. By Proposition 4, we have $|z^\star_i| = 0$. Then, we assume, by contradiction, that $\|(A)_i\|_2 \cdot \|Az^\star - b\|_2 \neq \lambda$. Since $\|Az^k - b\|_2 \to \|Az^\star - b\|_2$, for k sufficiently large $\|(A)_i\|_2 \cdot \|Az^k - b\|_2 \neq \lambda$. Hence we have $i \notin I^k_{\delta^k,2}$. □

As a consequence of Proposition 5, we can define a modified version of the Algorithm 1 where, at each iteration, we fix to zero the variables corresponding to the indices in $I^k_{\delta^k}$. We report in Algorithm 2 the variable fixing version of Algorithm 1.

We would like to highlight that

- the definition of the active-set $I^k_{\delta^k}$ at Step 2 does not require the computation of the gradient $g(x) = A^T(Ax - b)$;
- the solution of the subproblem at Step 4 only involves the computation of the components of g corresponding to indices in the set $I^k_{\delta^k}$ (which is expected to be very small for k sufficiently large).

Further, we notice that problem (33) is very similar to the one to be solved at Step 2 of Algorithm 1 (namely, problem (21)) , except that some of the variables are fixed to zero. Thus, we can still get a closed-form solution to problem (33), similar to the one we described in (24) for problem (21), where the variables with indices in $I^k_{\delta_k}$ are set equal to zero.

Algorithm 2 Two-block VF-NLCGS

```
Require: A penalty parameter $c > \|A\|_2^2$, parameters $\xi^0, \delta^0 > 0$, a point $x^0 = z^0 \in \mathbb{R}^n$.
  for $k = 1, 2, \ldots,$ until convergence do
    Step 1. Fixed $z^k$, compute $x^{k+1}$ solving the problem
            $$\min_x \Phi(x, z^k) \qquad (32)$$
    Step 2. Compute $I^k_{\delta^k}$ by (27).
    Step 3. Set $P^k = \{(z, w) \in \mathbb{R}^n \times \mathbb{R}^n : -w \le z \le w,\ w_i = 0 \text{ with } i \in I^k_{\delta^k}\}$.
    Step 4. Fixed $x^{k+1}$, compute $z^{k+1}$ solving the problem
            $$\min_{z \in \mathbb{R}^n, w \in \mathbb{R}^n} \ \Psi(x^{k+1}, z, w) = \Phi(x^{k+1}, z) + \lambda e^T w \qquad (33)$$
            $$\text{s.t. } (z, w) \in P^k,$$
    Step 5. Suitably reduce $\delta^k$ and $\xi^k$.
  end for
```

Now, we state some theoretical results that will be useful to prove the convergence of the sequence generated by Algorithm 2.

Proposition 6. *Let $c > \|A\|_2^2$. Then the function $\Psi(x, z, w)$ defined in (15) is coercive in the feasible set*

$$P = \{(x, z, w) \in \mathbb{R}^{3n} : \ -w \le z \le w\}.$$

Proof. Recalling that $\Psi(x,z,w)$ has the form

$$\Psi(x,z,w) = \frac{1}{2}\|Ax-b\|_2^2 - g(x)^T(x-z) + \frac{c}{2}\|x-z\|_2^2 + \lambda e^T w,$$

it is easy to see that when $\|x\|_2 \to \infty$ or $\|z\|_2 \to \infty$, then $\Psi(x,z,w) \to \infty$. Consider now the case when both $\|x\|_2$ and $\|z\|_2 \to \infty$. We set, just for our convenience,

$$u = Ax - b, \quad v = x - z,$$

and, as $g(x) = A^T(Ax-b)$, we can write

$$\begin{aligned}\Psi(x,z,w) &= \frac{1}{2}\|Ax-b\|^2 - g(x)^T(x-z) + \frac{c}{2}\|x-z\|_2^2 + \lambda e^T w \\ &\geq \frac{1}{2}\|Ax-b\|_2^2 - \|A\|_2\|Ax-b\|_2\|x-z\|_2 + \frac{c}{2}\|x-z\|_2^2 + \lambda e^T w \\ &= \frac{1}{2}\|u\|_2^2 - \|A\|_2\|u\|_2\|v\|_2 + \frac{c}{2}\|v\|_2^2 + \lambda e^T w.\end{aligned}$$

We have three different cases:

a) $\|u\|_2 \leq M$ and $\|v\|_2 \leq M$ for some $M > 0$. As $\|z\|_2 \to \infty$ and $-w \leq z \leq w$, it is easy to see that $\|w\|_2 \to \infty$. Then, we have $\Psi(x,z,w) \to \infty$.

b) $\|u\|_2 \to \infty$ (or $\|v\|_2 \to \infty$). It is easy to see that $\Psi(x,z,w) \to \infty$.

c) $\|u\|_2 \to \infty$ and $\|v\|_2 \to \infty$. By assumption, $c > \|A\|_2^2$, then we have

$$\begin{pmatrix} 1 & -\|A\|_2 \\ -\|A\|_2 & c \end{pmatrix}$$

is positive definite and $\Psi(x,z,w) \to \infty$.

We can conclude that $\Psi(x,z,w)$ is coercive in P. □

Proposition 7. *Let $c > \|A\|_2^2$. Then the sequence $\{(x^k, z^k, w^k)\}$ generated by Algorithm 2 admits a limit point and*

$$\lim_{k\to\infty} \|z^{k+1} - z^k\|_2 = 0.$$

Proof. As the function is coercive in P, it has compact level sets. Furthermore, the sequence $\{(x^k, z^k, w^k)\}$ generated by Algorithm 2 is such that

$$\Psi(x^{k+1}, z^{k+1}, w^{k+1}) \leq \Psi(x^{k+1}, z^k, w^k) \leq \Psi(x^k, z^k, w^k). \tag{34}$$

Then the sequence $\{(x^k, z^k, w^k)\}$ admits a limit point and there exists a subsequence of indices $K \subseteq \{0, 1, \ldots\}$ such that the sequence $\{(x^k, z^k, w^k)\}_K$ converges to a point $(x^\star, w^\star, z^\star)$.
The convergence of $\{(x^k, z^k, w^k)\}_K$ and the continuity of Ψ imply that the sequence $\{\Psi(x^k, z^k, w^k)\}$ has a subsequence converging to $\Psi(x^\star, w^\star, z^\star)$. As $\{\Psi(x^k, z^k, w^k)\}$ is nonincreasing, we have that $\{\Psi(x^k, z^k, w^k)\}$ is bounded from below and converges to $\Psi(x^\star, w^\star, z^\star)$. Furthermore, by (34) we have that the sequence $\{\Psi(x^{k+1}, z^k, w^k)\}$ converges to $\Psi(x^\star, w^\star, z^\star)$.
Now, we define the following function:

$$\Theta(x,w,z) = \frac{1}{2}\|Ax-b\|_2^2 - g(x)^T(x-z) + \frac{\|A\|_2^2 + d}{2}\|x-z\|_2^2 + \lambda e^T w,$$

with $d = \dfrac{c - \|A\|_2^2}{2}$. By (34) and the fact that $x^{k+1} = z^k$, we can write

$$\Theta(x^{k+1}, z^k, w^k) - \Theta(x^k, z^k, w^k) \leq 0 \tag{35}$$

and

$$\Theta(x^{k+1}, z^{k+1}, w^{k+1}) - \Theta(x^{k+1}, z^k, w^k) \leq -\frac{d}{2}\|z^{k+1} - z^k\|_2^2. \tag{36}$$

Hence, proceeding as in the case of Ψ, we obtain

$$\lim_{k\to\infty} \Theta(x^{k+1}, z^{k+1}, w^{k+1}) - \Theta(x^k, z^k, w^k) = 0$$

and, taking the limit in (36) for $k \to \infty$, we have

$$\lim_{k\to\infty} \|z^{k+1} - z^k\|_2 = 0.$$

□

Theorem 2. *Let $c > \|A\|_2^2$, let $\{x^k\}$ be the sequence generated by Algorithm 2, and let $\xi^k = \delta^k = \rho(x^k)$ in $I^k_{\xi^k}$ and $I^k_{\delta^k}$, with $\rho(x)$ satisfying Definition 1. Then every limit point $x^\star$ of the sequence $\{x^k\}$ is an optimal solution of Problem (3).*

Proof. First, we assume that there exists an index $\bar{k}$ such that $I^k_{\delta^k} = \emptyset$ for all $k \geq \bar{k}$. In this case, Algorithm 2 is equivalent to Algorithm 1 and convergence follows from Theorem 1. Let us now assume that there exists a subsequence of indices $K \subseteq \{0, 1, \ldots\}$ such that the sequence $\{(x^k, z^k, w^k)\}_K$ converges to a point $(x^\star, w^\star, z^\star)$ and $I^k_{\delta^k} \neq \emptyset$ for all $k \in K$. By Proposition 7, we have

$$\lim_{k\to\infty} \|z^{k+1} - z^k\|_2 = 0,$$

which implies

$$\lim_{k\to\infty} (x^k, w^k, z^k) = (x^\star, w^\star, z^\star),$$
$$\lim_{k\to\infty} (x^{k+1}, w^k, z^k) = (x^\star, w^\star, z^\star),$$

$x^\star = z^\star$ and $w^\star = |z^\star|$ .We can have three different cases:

1. $i \in I^k_{\delta^k,2}$ for k sufficiently large. By Proposition 5, we have

   $$I^k_{\delta^k,2} = \tilde{E}^\star \subseteq E^\star$$

   and the corresponding variable z_i^k will stay fixed to zero until convergence. Thus we have

   $$|g_i(z^\star)| \leq \|(A)_i\|_2 \cdot \|Az^\star - b\|_2 = \lambda. \tag{37}$$

2. There exists a subsequence $H \subseteq K$ such that $i \in I^k_{\delta^k,1}$ for all $k \in H$. By the definition of $I^k_{\delta^k,1}$, we have that

   $$|g_i(z^k)| \leq \|(A)_i\|_2 \cdot \|Az^k - b\|_2 \leq \lambda - \delta^k \tag{38}$$

   for each $k \in H$. Taking the limit for $k \to \infty$, we have

   $$|g_i(z^\star)| \leq \|(A)_i\|_2 \cdot \|Az^\star - b\|_2 \leq \lambda. \tag{39}$$

   Furthermore, by Proposition 4, we have $z_i^\star = 0$ and $i \in E^\star$.

3. There exists an index $\bar{k}$ such that for all $k \geq \bar{k}$ $i \notin I^k_{\delta^k}$. Then we explicitly solve the problem (23) with respect to z_i.

Thus, it is easy to see that KKT conditions of Problem (15) are satisfied and point $x^\star$ is a solution of Problem (3). $\square$

We conclude this section remarking that the proposed variable-fixing strategy can be employed to define variants of other existing methods for ℓ_1-regularized least-squares, e.g. FISTA [2], TwiST [4].

4 Numerical results

In this section, we report the preliminary numerical results related to the two-block VF-NLCGS algorithm described in the previous section. Our aim is to analyze the effects of the variable fixing strategy in the considered decomposition framework. To this purpose, firstly we performed a brief analysis to tune the parameters involved in Algorithms 2, secondly we assess the performance of the proposed algorithm in comparison with the standard two-block NLCGS approach on one side, and with two state-of-the-art algorithms for ℓ_1-regularized least-squares problems, namely FPC-AS [26] and SpaRSA [28], on the other side. The comparison of the overall computational effort is carried-out by using the performance profiles proposed by Dolan and Moré in [11] plotting graphs in a logarithmic (base 2) scale.

4.1 The testing set

In the experiments, we considered a set of testing problems of the form (1) that comprises 5 random problems (denoted as R1-R5) widely used for software benchmarking [26].

In particular, we generated artificial signals of dimensions $n \in \{2^{11}, 2^{12}, 2^{13}\}$ and with a number of observations $m = n/2$ and we set the number of nonzeros $T = round(\rho m)$ with $\rho \in \{0.6\%, 1.25\%, \ 2.5\%, \ 5\%, \ 10\%\}$. This procedure resulted in a total of 75 different random problems where the matrix A was generated as follows. Let $\bar{A}$ be the Gaussian matrix whose elements are generated independent and identically distributed from the normal distribution $N(0, 1)$, then A is obtained by extracting m random rows from these matrices:

R1) the matrix given by the $\bar{A}$ row orthogonalization by singular value decomposition;

R2) the matrix given by the $\bar{A}$ row orthogonalization by QR decomposition;

R3) the Bernoulli matrix with ± 1 elements with equal probability;

R4) the Hadamard matrix whose entries are ± 1 and whose rows are mutually orthogonal;

R5) the Discrete Cosine Transform (DCT) matrix.

The true signal $x^\star$ was built as a vector with T randomly placed ± 1 spikes, with zero in the other components.

Moreover, in order to have matrices A with 2-norm equal to 1, we scaled the matrices in R1, R3, R4 by the largest eigenvalue of $A^T A$ as in [26]. Finally, for all problems, the vector of observations b was chosen as $b = Ax^\star + \eta$, where η is a Gaussian white vector with variance 10^{-4} and we set $\lambda = 0.1\|A^T b\|_\infty$ as in [14, 28].

4.2 Implementation details

We implemented Algorithms 1 and 2 in Matlab codes denoted NLCGS and VF-NLCGS respectively, and we used the Matlab implementations of FPC-AS and SpaRSA available at the authors web sites. All the tests were performed on an Intel Core i5 M520 2.4 GHz, 4GB RAM using Matlab 7.14 (R2012a) and machine precision $\epsilon_m \approx 2 \times 10^{-16}$.

In the experiments, convergence was declared when the following condition [28] on the relative size of the step just taken was met

$$\|x^k - x^{k-1}\|_2 \leq 10^{-4}\|x^k\|_2, \tag{40}$$

and any run performing more than 1000 iterations is considered a failure.

Moreover, to verify that the comparison among the solvers was independent on the stopping rule used for each approach, we further ran SpaRSA with its default stopping rule to obtain a target objective function value, then ran the other algorithms until each of them reached the given target. We do not report the results obtained with this strategy since they were comparable to those obtained with the stopping criterion (40) described above.

The null vector is used as starting guess in all the compared codes and all matrices were stored explicitly. All other parameters in SpaRSA and FPC-AS were set at their default values. Finally, in all the experiments, we performed 10 runs for each problem for a total of 750 runs.

4.3 Experiments on parameter updating strategies

The implementation of Algorithms 2 depends on the choice of three parameters: the penalty parameter c in (14) and the parameters ξ^k ad δ^k included in (26) and (27) which are responsible for the active set strategy.

Following the theoretical analysis performed in the previous section, a possible choice for defining ξ^k ad δ^k is considering the following identification function

$$\rho(x^k, \xi^k) = \sqrt{\|(|g(x^k)| - \lambda)_{A_{\xi^k}}\|_\infty + \|x^k \odot (|g(x^k)| - \lambda)\|_2} \tag{41}$$

where

$$A_{\xi^k} = \{i \; : \; |x_i^k| > \xi^k\} \cup \{i \; : \; |x_i^k| \leq \xi^k \text{ and } |g_i(x^k)| \geq \lambda\}$$

and $\odot$ denotes the componentwise product. This function was proposed in [13] for nonlinear programming and has also been used in [26]. The definition of ρ is based on the amount that the current iterate x^k violates the optimality conditions for (1).

Taking into account (41), we set the following values for the sequences ξ^k and δ^k

$$\xi^0 = 10^{-3}, \delta^0 = 10 \;\text{ and }\; \xi^{k+1} = \delta^{k+1} = \rho(x^k, \xi^k) \quad k = 0, \ldots \tag{42}$$

and we experimentally verified on our testing set that the sequence ξ^k, δ^k rapidly decrease to zero and that our algorithm converges to a solution of (1). We observed that the computation of $\rho(x^k, \xi^k)$ involves the evaluation of all the components of the current gradient $g(x^k)$ and in particular of components in the set I_{δ^k} which are not required in Algorithm 2. Clearly, this reduces the algorithm performance in terms of CPU-time per iteration. Therefore, we propose to use a simpler and cheaper heuristic updating rules for the parameters ξ^k ad δ^k consisting in starting from a given value and then slowly decreasing the parameters by products with a

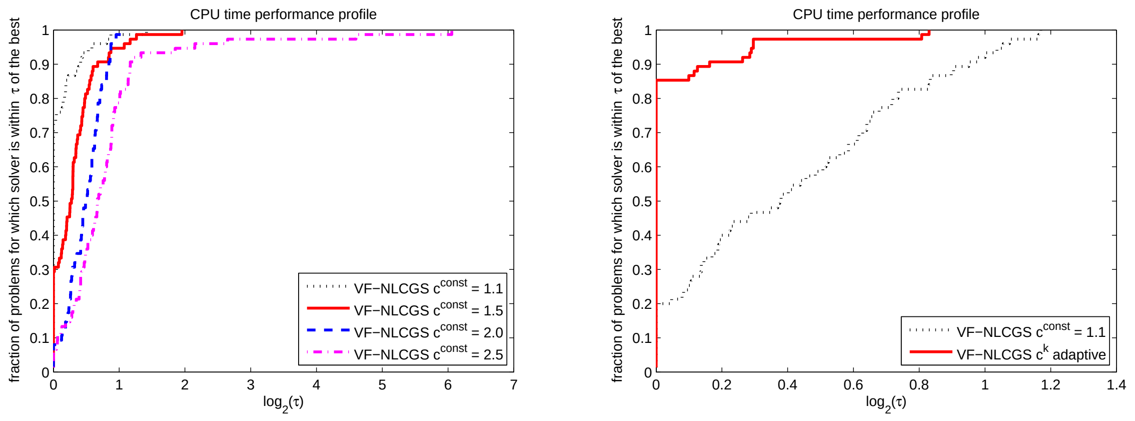

Figure 1: The CPU time performance profile of VF-NLCGS varying the penalty parameter updating rule.

fixed factor in $(0, 1)$. The aim of this strategy is to generate sequences ξ^k ad δ^k that hopefully decrease slower than $\rho(x^k, \xi^k)$ so that condition (30) is guaranteed.

We found out that the following updating rule

$$\xi^0 = 10^{-3},\ \xi^{k+1} = 0.99\,\xi^k \quad \text{and} \qquad \delta^0 = 10,\ \delta^{k+1} = 0.99\,\delta^k, \qquad k = 0, \dots \tag{43}$$

works well in practice.

Concerning the choice of the penalty parameter c in (14), in principle one should choose a value such that the equivalence between the penalty and the original problem is guaranteed (see Proposition 3). This value obviously depends on the maximum eigenvalue of $A^T A$, which is not always known or computable. Furthermore, if the chosen value c is too large, the performance of the algorithm are somehow reduced due to the fact that the penalty function obtained is difficult to be minimized.

Then, a good strategy is starting with a small value of the penalty parameter and updating it when some specific condition is met. This is very common when dealing with an exact penalty approach and is closely related, in our case, to IST methods with backtracking (see e.g. [2]). In our experiments, we started from a value $c^0 < \|A\|_2^2$ and we set

$$c^{k+1} = 1.001 c^k, \tag{44}$$

every time conditions (35) and (36) were not satisfied.

In order to analyze in depth the effects of our heuristic updating strategy (44), we tested the performance of VF-NLCGS with the updating strategy (44) against VF-NLCGS with constant penalty parameter taking into account that for all our test problems $\|A\|_2 = 1$. We report in Figure 1 two CPU performance profiles:

- the one on the left is related to the comparison of different versions of VF-NLCGS where we keep the penalty parameter fixed (namely, $c = c^{const}$ and $c^{const} = 1.1,\ 1.5,\ 2,\ 2.5$);
- the one on the right shows the comparison between VF-NLCGS with our adaptive updating strategy starting from $c^0 = 0.5$ and the best version of VF-NLCGS with fixed penalty parameter, i.e. with $c^{const} = 1.1$.

From the plots it is clear that the performance of VF-NLCGS with constant penalty parameter deteriorates as c^{const} increases, and that VF-NLCGS with an adaptive c outperforms the best version of VF-NLCGS with constant c as the former is the most efficient for the 85% of runs.

The numerical experiments presented in the next section are all obtained using the adaptive updating rule (44) for the penalty parameter c and the updating (43) for ξ^k ad δ^k.

4.4 Numerical comparison

In this section, we firstly show the beneficial effect of employing our variable fixing strategy in the two-block NLCGS method and, secondly, we show that the proposed approach, despite its simplicity, is competitive also in comparison with other available procedures.

In Figure 2, we report the CPU time performance profiles of NLCGS and VF-NLCGS. From both plots, it is clear that the use of the variable fixing strategy considerably enhances the performance of the standard NLCGS. Indeed, VF-NLCGS guarantees better results both in terms of efficiency and robustness. In particular, it is the best in the 85.3% of the runs on the considered testing set. Moreover, as expected, the ratio of the time performed by NLCGS over the time performed by VF-NLCGS to solve a problem increases as the sparsity of the original signal increases. In fact, in this case, the active-set procedure is fully exploited since subproblems of small dimensions are solved.

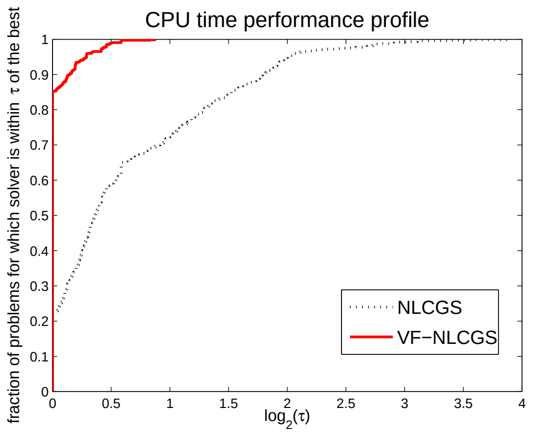

Figure 2: The CPU time performance profile: NLCGS and VF-NLCGS.

The CPU performance profile of Figure 3 summarizes the comparison among SpaRSA, VF-NLCGS and FPC-AS. As we can see, VF-NLCGS is competitive with both SpaRSA and FPC-AS. In particular, VF-NLCGS results the most efficient for around the 74.9% of the runs and, on the other side, FPC-AS is within a factor 2 in the 92.6% of the runs. All the three algorithms are successful in all runs and we encountered problems of "false termination" with the criterion (40) only in 3 over 750 runs of FPC-AS (we did not take into account these rare occurrences in the profiles).

Tables 1-3 in the Appendix collect the detailed results of the experiments referred to the performance profiles in Figures 2 and 3.

We conclude this section by showing in Figure 4 the typical trend of the curve related to the size of the estimated active set as the iterations progress. The figure refers to problem R5 with

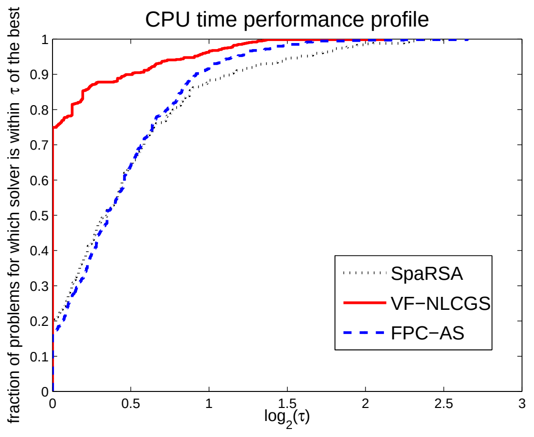

Figure 3: The CPU time performance profile: SpaRSA, VF-NLCGS and FPC-AS.

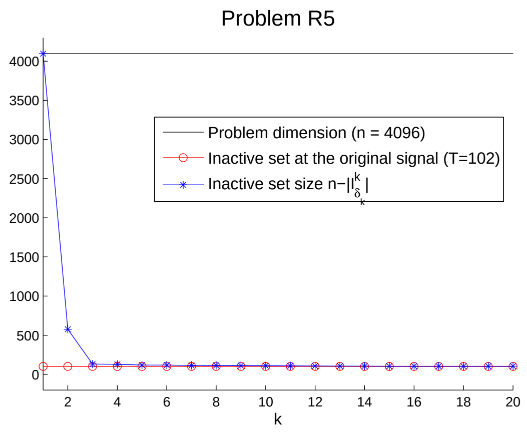

Figure 4: Plot of the size of the inactive set during the iterations.

$n = 4096$ and $T = 102$ and represents the most common situation: the size of the inactive set (blu line) decreases quickly in the first iterations and the inactive set at the solution is detected (red line).

5 Conclusions

In this paper, we presented a simple active-set strategy to enhance the practical performance of a first-order method for ℓ_1-regularized least-squares belonging to the class of Iterative Shrinkage-Thresholding algorithms.

In this context, we provided both theoretical and practical results. In particular, we first analyzed the ℓ_1-regularized least-squares problem from a different perspective, by combining a classical variable splitting approach with a suitable exact penalty function. Then, starting from the penalized problem, we described a two-block NLCGS algorithm for its solution. Finally, we introduced a suitable rule for fixing at each step a subset of variables to zero, thus obtaining a variable fixing variant of the original algorithm.

Furthermore, we performed preliminary numerical experiments to assess the reliability and efficiency of the variable fixing strategy. The reported results seem to indicate that the proposed approach can substantially improve the performance of the two-block NLCGS algorithm, making this simple algorithm competitive with other state-of-the-art techniques.

As a final remark, we highlight that the proposed variable fixing strategy is a general paradigm that could be employed to improve the performance of other existing methods for ℓ_1-regularized least-squares.

References

[1] M. V. Afonso, J. M. Bioucas-Dias and M. A. T. Figueiredo, *Fast image recovery using variable splitting and constrained optimization,* IEEE Transactions on Image Processing, VOL. 19(9), 2–45, 2010.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-threshold algorithm for linear inverse problems,* SIAM Journal on Imaging Sciences 2, pp. 183-202, 2009.

[3] D.P. Bertsekas, *Nonlinear Programming*, 2nd edn., Athena Scientific, 1999.

[4] J. Bioucas-Dias and M. Figueiredo, *A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration*, IEEE Transactions on Image Processing, 16:2992–3004, 2007.

[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(1), pp 34–81, 2009.

[6] E. Candès, J. Romberg and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. on Information Theory, 52(2) pp. 489 – 509, February 2006.

[7] E. Candès and J. Romberg, *Quantitative robust uncertainty principles and optimally sparse decompositions*, Foundations of Comput. Math., 6(2), pp. 227 - 254, April 2006.

[8] E. Candès, J. Romberg and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics, Vol. 59, No. 8. , pp. 1207–1223, August 2006.

[9] S.S. Chen, D.L. Donoho and M.A. Saunders, *Atomic decomposition basis pursuit*, SIAM Rev., 43, pp. 129–159, 2001.

[10] I. Daubechies and M. Defrise and C. D. Mol , *An iterative thresholding algorithm for linear inverse problems with sparsity constraints*, Communications on Pure and Applied Mathematics, 57, pp. 1413–1457, 2004.

[11] E.D., Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. 91, 201221 (2002).

[12] D. Donoho, *Compressed sensing*, IEEE Trans. on Information Theory, 52(4), pp. 1289 - 1306, 2006.

[13] F. Facchinei, A. Fischer and C. Kanzow. *On the accurate identification of active constraints.* SIAM Journal on Optimization 9, 1999, pp. 14-32.

[14] M. Figueiredo, R. Nowak and S. Wright, *Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems*, IEEE Journal on Selected Topics in Signal Processing, vol. 1, pp. 586598, 2007.

[15] R. Fletcher, *A class of methods for nonlinear programming with termination and convergence prop- erties, in Integer and Nonlinear Programming*, J. Abadie, ed., North-Holland, The Netherlands, pp. 157175, 1970.

[16] L. Grippo and M. Sciandrone., *Globally convergent block-coordinate techniques for unconstrained optimization.*, Optimization Methods and Software, Vol. 10 (4), pp.587-637, 1999.

[17] L. Grippo and M. Sciandrone., *On the convergence of the block nonlinear Gauss-Seidel method under convex constraints.*, Operations Research Letters, Vol. 26 (3), pp.127-136, 2000.

[18] E. T. Hale, W. Yin and Y. Zhang,*A numerical study of fixed-point continuation applied to compressed sensing.* Journal of Computational Mathematics, 28(2):170194, 2010.

[19] E. V. Haynsworth,*On the Schur Complement.*, Basel Mathematical Notes, BNB 20, June 1968.

[20] M.R. Hestenes,*Multiplier and gradient methods.* Journal of Optinization Methods and Applications, 4, pag. 303-320, 1969.

[21] S.-J. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, *An Interior-Point Method for Large-Scale l1-Regularized Least Squares*, IEEE Journal on Selected Topics in Signal Processing, 1(4):606–617, December 2007.

[22] M.J.D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, pag. 283-298, 1969.

[23] R. Tibshirani, *Regression shrinkage and selection via the Lasso*, J. Royal Statist. Soc. Ser. B, Vol. 58(1), pp. 267–288, 1996.

[24] P. Tseng and S. Yun , *A coordinate gradient descent method for nonsmooth separable minimization*, Mathematical Programming, Vol. 117(1), pp 387–423, 2009.

[25] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab and Ö. Yilmaz, *Algorithm 890: Sparco: A testing framework for sparse reconstruction.* ACM Trans. Math. Software, 35 (2009), pp. 116.

[26] Z. Wen, W. Yin, D. Goldfarb and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation*, SIAM J. Sci. Comput., 32(4), pp. 1832–1857, 2010.

[27] Z. Wen, W. Yin, H. Zhang and D. Goldfarb, *On the convergence of an active-set method for ℓ_1 minimization*, 27(6), pp. 1127–1146, 2012.

[28] S. Wright, R. Nowak and M. Figueiredo, Sparse reconstruction by separable approximation , IEEE Transactions on Signal Processing 57, pp. 2479-2493, 2009.

Appendix

In Tables 1-3, we report the average CPU time (Av-CPU), the average relative error (Av-rel.err.) and the average number on nonzero components of the computed solution (Av-nnz), corresponding to runs summarized in the performance profiles in Figures 2 and 3. Problems where the false termination of FPC-AS with the criterion (40) occurs 1 over 10 runs is denoted with the symbol '*' and the average values reported in the Tables do not take into account these cases.

n = 2048		Av-CPU				Av-rel.err.				Av-rel.err.			
		SpaRSA	NLCGS	VF-NLCGS	FPC-AS	SpaRSA	NLCGS	VF-NLCGS	FPC-AS	SpaRSA	NLCGS	VF-NLCGS	FPC-AS
T = 6	R1	0.10	0.08	0.08	0.10	1.15E-02	1.15E-02	1.15E-02	1.15E-02	6	6	6	6
	R2	0.10	0.09	0.07	0.10	1.17E-02	1.17E-02	1.17E-02	1.17E-02	6	6	6	6
	R3	0.12	0.24	0.11	0.12	1.14E-02	1.14E-02	1.14E-02	1.14E-02	6	6	6	6
	R4	0.11	0.08	0.09	0.11	1.10E-02	1.10E-02	1.10E-02	1.10E-02	6	6	6	6
	R5	0.28	0.26	0.09	0.18	1.03E-02	1.03E-02	1.03E-02	1.04E-02	6	6	6	6
T = 13	R1	0.11	0.10	0.09	0.12	1.31E-02	1.31E-02	1.31E-02	1.31E-02	13	13	13	13
	R2	0.11	0.11	0.09	0.12	1.32E-02	1.32E-02	1.32E-02	1.32E-02	13	13	13	13
	R3	0.14	0.25	0.13	0.14	1.37E-02	1.37E-02	1.37E-02	1.37E-02	13	13	13	13
	R4	0.12	0.10	0.08	0.12	1.25E-02	1.25E-02	1.25E-02	1.25E-02	13	13	13	13
	R5	0.24	0.27	0.09	0.16	1.20E-02	1.20E-02	1.20E-02	1.20E-02	13	13	13	13
T = 26	R1	0.12	0.10	0.10	0.12	1.59E-02	1.58E-02	1.58E-02	1.58E-02	26	26	26	26
	R2	0.11	0.12	0.11	0.14	1.55E-02	1.55E-02	1.55E-02	1.55E-02	26	26	26	26
	R3	0.16	0.28	0.18	0.15	1.81E-02	1.81E-02	1.81E-02	1.80E-02	26	26	26	26
	R4	0.13	0.10	0.10	0.14	1.45E-02	1.44E-02	1.44E-02	1.44E-02	26	26	26	26
	R5	0.28	0.55	0.12	0.24	2.39E-02	2.38E-02	2.38E-02	2.37E-02	26.4	26.4	26.4	26.4
T = 51	R1	0.14	0.13	0.11	0.14	1.90E-02	1.90E-02	1.90E-02	1.90E-02	51	51	51	51
	R2	0.14	0.12	0.11	0.14	1.99E-02	1.99E-02	1.99E-02	1.99E-02	51	51	51	51
	R3	0.17	0.31	0.21	0.16	2.71E-02	2.70E-02	2.70E-02	2.70E-02	51	51	51	51
	R4	0.14	0.14	0.12	0.15	1.92E-02	1.92E-02	1.92E-02	1.92E-02	51	51	51	51
	R5*	0.29	0.62	0.17	0.26	3.85E-02	3.84E-02	3.85E-02	3.84E-02	52.3	52.1	52	52.2
T = 102	R1	0.17	0.15	0.13	0.18	2.85E-02	2.85E-02	2.85E-02	2.85E-02	102.1	102.1	102.1	102.1
	R2	0.17	0.17	0.13	0.16	2.99E-02	2.99E-02	2.99E-02	2.99E-02	102	102	102	102
	R3	0.19	0.41	0.31	0.21	4.11E-02	4.11E-02	4.59E-02	4.11E-02	106.1	106.2	108.3	106.5
	R4	0.15	0.15	0.14	0.15	2.85E-02	2.85E-02	2.85E-02	2.85E-02	102	102	102	102
	R5	0.29	1.09	0.33	0.36	1.07E-01	1.06E-01	1.05E-01	1.06E-01	110	109.3	108.6	109.2

Table 1: Results on random problems (n = 2048, T = 6, 13, 26, 51, 102).

n = 4096		Av-CPU				Av-rel.err.				Av-rel.err.			
		SpaRSA	NLCGS	VF-NLCGS	FPC-AS	SpaRSA	NLCGS	VF-NLCGS	FPC-AS	SpaRSA	NLCGS	VF-NLCGS	FPC-AS
T = 13	R1	0.29	0.22	0.26	0.29	1.21E-02	1.21E-02	1.21E-02	1.21E-02	13	13	13	13
	R2	0.30	0.23	0.25	0.31	1.22E-02	1.22E-02	1.22E-02	1.22E-02	13	13	13	13
	R3	0.37	0.82	0.28	0.39	1.24E-02	1.24E-02	1.24E-02	1.24E-02	13	13	13	13
	R4	0.31	0.24	0.25	0.32	1.18E-02	1.18E-02	1.18E-02	1.18E-02	13	13	13	13
	R5	0.90	0.86	0.26	0.54	1.05E-02	1.05E-02	1.05E-02	1.04E-02	13	13	13	13
T = 26	R1	0.33	0.26	0.27	0.33	1.35E-02	1.35E-02	1.35E-02	1.35E-02	26	26	26	26
	R2	0.34	0.26	0.26	0.36	1.36E-02	1.36E-02	1.36E-02	1.36E-02	26	26	26	26
	R3	0.40	0.84	0.37	0.38	1.60E-02	1.60E-02	1.60E-02	1.60E-02	26	26	26	26
	R4	0.34	0.27	0.28	0.33	1.28E-02	1.28E-02	1.28E-02	1.28E-02	26	26	26	26
	R5	0.83	1.52	0.31	0.66	1.31E-02	1.31E-02	1.31E-02	1.29E-02	26.2	26.2	26.2	26.2
T = 51	R1	0.36	0.30	0.28	0.37	1.58E-02	1.58E-02	1.58E-02	1.58E-02	51	51	51	51
	R2	0.36	0.30	0.29	0.39	1.60E-02	1.60E-02	1.60E-02	1.60E-02	51	51	51	51
	R3	0.45	0.92	0.48	0.45	1.89E-02	1.89E-02	1.89E-02	1.89E-02	51	51	51	51
	R4	0.37	0.28	0.30	0.37	1.62E-02	1.62E-02	1.62E-02	1.62E-02	51	51	51	51
	R5*	0.84	1.92	0.38	0.76	2.11E-02	2.12E-02	2.12E-02	2.12E-02	51.5	51.5	51.5	51.5
T = 102	R1	0.42	0.38	0.37	0.40	2.13E-02	2.13E-02	2.13E-02	2.13E-02	102	102	102	102
	R2	0.41	0.37	0.36	0.40	2.14E-02	2.14E-02	2.14E-02	2.14E-02	102	102	102	102
	R3	0.50	1.08	0.72	0.47	2.93E-02	2.93E-02	2.93E-02	2.93E-02	102.2	102.2	102.2	102.2
	R4	0.40	0.36	0.35	0.41	2.11E-02	2.11E-02	2.11E-02	2.11E-02	102	102	102	102
	R5	0.93	2.49	0.82	0.98	5.27E-02	5.27E-02	5.27E-02	5.26E-02	104.7	104.7	104.2	104.6
T = 205	R1	0.47	0.48	0.44	0.46	3.25E-02	3.25E-02	3.25E-02	3.25E-02	205.1	205.1	205.1	205.1
	R2	0.46	0.48	0.48	0.48	3.03E-02	3.03E-02	3.03E-02	3.03E-02	205	205	205	205
	R3	0.58	1.50	1.34	0.64	4.74E-02	4.74E-02	5.38E-02	4.74E-02	215	214.9	219.3	214.8
	R4	0.47	0.49	0.47	0.48	3.17E-02	3.17E-02	3.17E-02	3.17E-02	205.2	205.2	205.2	205.2
	R5*	1.00	4.03	2.19	1.20	1.03E-01	1.02E-01	1.02E-01	1.01E-01	216.8	215.1	214.8	215

Table 2: Results on random problems (n = 4096, T = 13, 26, 51, 102, 205).

n = 8192		Av-CPU				Av-rel.err.				Av-rel.err.			
		SpaRSA	NLCGS	VF-NLCGS	FPC-AS	SpaRSA	NLCGS	VF-NLCGS	FPC-AS	SpaRSA	NLCGS	VF-NLCGS	FPC-AS
T = 26	R1	1.06	0.78	0.67	1.09	1.22E-02	1.22E-02	1.22E-02	1.22E-02	26	26	26	26
	R2	1.05	0.77	0.65	1.10	1.24E-02	1.24E-02	1.24E-02	1.24E-02	26	26	26	26
	R3	1.42	2.98	0.84	1.35	1.35E-02	1.35E-02	1.35E-02	1.35E-02	26	26	26	26
	R4	1.09	0.82	0.64	1.07	1.24E-02	1.24E-02	1.24E-02	1.24E-02	26	26	26	26
	R5	3.18	3.96	0.81	2.19	1.98E-02	1.99E-02	1.99E-02	1.98E-02	26.2	26.2	26.2	26.2
T = 51	R1	1.20	0.91	0.75	1.22	1.42E-02	1.42E-02	1.42E-02	1.42E-02	51	51	51	51
	R2	1.25	0.91	0.79	1.21	1.50E-02	1.50E-02	1.50E-02	1.50E-02	51	51	51	51
	R3	1.46	3.18	1.20	1.37	1.55E-02	1.55E-02	1.55E-02	1.55E-02	51	51	51	51
	R4	1.27	0.95	0.76	1.22	1.44E-02	1.44E-02	1.44E-02	1.44E-02	51	51	51	51
	R5	2.98	4.46	0.96	1.86	2.31E-02	2.31E-02	2.31E-02	2.31E-02	51.3	51.3	51.3	51.3
T = 102	R1	1.34	1.07	0.91	1.33	1.67E-02	1.67E-02	1.67E-02	1.67E-02	102	102	102	102
	R2	1.34	1.06	0.91	1.34	1.69E-02	1.69E-02	1.69E-02	1.69E-02	102	102	102	102
	R3	1.62	3.48	1.66	1.56	2.02E-02	2.03E-02	2.03E-02	2.02E-02	102	102	102	102
	R4	1.35	1.08	0.92	1.36	1.77E-02	1.77E-02	1.77E-02	1.77E-02	102	102	102	102
	R5	4.02	9.94	1.80	2.92	3.49E-02	3.47E-02	3.47E-02	3.54E-02	104.4	103.9	103.7	104.3
T = 205	R1	1.52	1.34	1.14	1.48	2.33E-02	2.32E-02	2.32E-02	2.32E-02	205	205	205	205
	R2	1.49	1.34	1.12	1.48	2.27E-02	2.27E-02	2.27E-02	2.27E-02	205	205	205	205
	R3	1.87	4.21	2.48	1.75	3.14E-02	3.14E-02	3.14E-02	3.14E-02	205	205	205	205
	R4	1.59	1.41	1.18	1.52	2.18E-02	2.17E-02	2.17E-02	2.17E-02	205	205	205	205
	R5	3.64	12.27	3.21	4.50	6.51E-02	6.51E-02	6.51E-02	6.47E-02	209.6	209.4	209.1	209.3
T = 410	R1	1.75	1.79	1.52	1.70	3.28E-02	3.28E-02	3.28E-02	3.28E-02	410.4	410.4	410.2	410.4
	R2	1.75	1.77	1.47	1.69	3.30E-02	3.30E-02	3.33E-02	3.31E-02	410.2	410.2	410.1	410.2
	R3	2.21	5.89	4.65	2.22	5.13E-02	5.14E-02	5.82E-02	5.13E-02	432.4	432.5	439.4	433.5
	R4	1.76	1.87	1.48	1.71	3.10E-02	3.10E-02	3.10E-02	3.10E-02	410.1	410.1	410.1	410.1
	R5	3.99	18.22	7.71	6.73	1.08E-01	1.08E-01	1.08E-01	1.08E-01	433.2	431	430.2	431.3

Table 3: Results on random problems (n = 8192, T = 26, 51, 102, 205, 410).