Sparse Signal Reconstruction: LASSO and Cardinality Approaches

Nikita Boyko, Gulver Karamemis, Viktor Kuzmenko, and Stan Uryasev

Abstract The paper considers several optimization problem statements for signal sparse reconstruction problems. We tested performance of AORDA Portfolio Safeguard (PSG) package with different problem formulations. We solved several medium-size test problems with cardinality functions: (a) minimize L1-error of regression subject to a constraint on cardinality of the solution vector; (b) minimize cardinality of the solution vector subject to a constraint on L1-error of regression. We compared performance of PSG and IBM CPLEX solvers on these problems. Although cardinality formulations are very appealing because of the direct control of the number of nonzero variables, large problems are beyond the reach of the tested commercial solvers. Step-down from the cardinality formulations is the formulation with the constraint on the sum of absolute values of the solution vector. This constraint is a relaxation of the cardinality constraint. Medium and large problems (from SPARCO toolbox for testing sparse reconstruction algorithms) were solved with PSG in the following formulation: minimize L1-error subject to a constraint on the sum of absolute values of the solution vector. The further

Nikita Boyko
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 32611 USA
e-mail: nikita@ufl.edu

Gulver Karamemis
Department of Information Systems and Operations Management, University of Florida, Gainesville, FL, 32611 USA
e-mail: gkaramemis@ufl.edu

Viktor Kuzmenko
V.M. Glushkov Institute of Cybernetics, Kyiv, Ukraine
e-mail: kvnu@mail.ru

Stan Uryasev
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, 32611 USA
e-mail: uryasev@ufl.edu

step-down in the sparse reconstruction problem formulations is the LASSO approach which does not have any constraints on functions. With the LASSO approach you do not know in advance the cardinality of the solution vector and you solve many problems with different regularization parameters. Then you select a solution with appropriate regression error and cardinality. Definitely, it is a time consuming process, but an advantage of LASSO approach is that optimization problems can be solved quite quickly even for very large problems. We have solved with PSG several medium and large problems from the SPARCO toolbox in LASSO formulation (minimize L2-error plus the weighted sum of absolute values of the solution vector).

1 Introduction

Problems considered in this paper are special cases of a broad family of approaches known as Compressive Sensing. The goal of Compressive Sensing is to reconstruct a sparse signal from a small number of observations. Theory of compressed sensing was first introduced by [15]. In another paper by Donoho [16], they define a vector $y = Ax$ in $\mathbb{R}^m$, where they reconstruct vector x given the $m \times n$ matrix A with $m < n \leq \phi m$. Compressed Sensing is further discussed in many other papers including [18, 11, 10]. More recent work on compressed sensing is presented and summarized in [24]. Furthermore, many resources including tutorials and papers about compressive sensing can be downloaded from the website [3]. Another resource about compressive sensing, involving mathematical programming formulations and codes can be found in the website [5]. This website provides "L1-MAGIC" collection of MATLAB routines for solving the convex optimization programs central to compressive sampling. The algorithms are based on interior-point methods, and are suitable for large-scale problems.

Let us consider a problem of reconstructing an n-dimensional vector x given the $m \times n$ matrix A and the m-dimensional vector y, which is an observation (possibly noisy) of the product Ax. The most common approach for restoration is to minimize the difference between observation y and estimation Ax

$$\min_x \frac{1}{2}\|y - Ax\|_2^2. \tag{1}$$

This problem has the analytical solution

$$x = (A'A)^{-1}A'y, \tag{2}$$

if $n \leq m$ and is degenerate if $n > m$. Here $\|v\|_2 = \sqrt{\sum_i v_i^2}$ denotes the Euclidean norm. If the noise is not normally distributed (especially if distribution has fat tails) the estimated vector x can be quite sensitive to tail observations (outliers) of y. In this case, robust statistical approaches can be used. In particular, the L1-error is much less sensitive to outliers than standardly used L2-error. Therefore the following formulation of the regression problem is a good alternative to the formulation (1).

$$\min_x \|y - Ax\|_1, \tag{3}$$

where L1 norm is defined as $\|v\|_1 = \sum_i |v_i|$. The objective is a convex piecewise linear function of x and the problem can be reduced to a linear program. This fact is important because of the availability of efficient large scale linear programming solvers and relevant numerical technologies based on linear programming.

If the modeling of the phenomenon assumes that x vector is sparse (i.e., the vector has many zero components) a regularization is used to "suppress" the irrelevant components of x. The regularization part $\tau\|x\|$ is added to the objective. The problem is formulated as follows:

LASSO-O

$$\min_x \frac{1}{2}\|y - Ax\|_2^2 + \tau\|x\|, \tag{4}$$

where τ sets a tradeoff between error and sparsity and is chosen based on empirical considerations. The latter case allows to handle the ill-conditioned formulation, where $n > m$ and x are sparse. Using L1 norm for regularization part is known as LASSO (least absolute shrinkage and selection operator) technique in the literature [12, 26, 27]. LASSO approach gained popularity because it provides sparse solutions due to properties of L1 norm.

The following two problems can be used instead of problem (4):

LASSO-I

$$\min_x \|y - Ax\|_2, \quad \text{s.t. } \|x\|_1 \le t. \tag{5}$$

LASSO-II

$$\min_x \|x\|_1, \quad \text{s.t. } \|y - Ax\|_2 \le \epsilon. \tag{6}$$

Here t and ϵ are some predefined parameters. By variation of parameter t in problem **LASSO-I** and parameter ϵ in problem **LASSO-II** we can get the same solution vectors x as by variation of parameter τ in problem **LASSO-O**.

The considered minimization problems are convex which makes them computationally attractive. Paper [20] efficiently solves a large size optimization model (4) with a gradient projection algorithm.

A natural approach to sparse signal reconstructing would be to bound the number of nonzero components (spikes) of vector x. Such a cardinality constraint results in nonconvex formulations. Therefore appropriate numerical tools are needed to solve such problems.

The goal of this paper is to formulate a signal reconstruction problem with cardinality constraints and test the performance of existing commercial solvers on such formulations. In our numerical experiments we generated the data set according to the procedure described in [20].

2 Problem Formulation

First, let us define cardinality function as the number of non-zero components of a vector, i.e.,

$$\mathrm{card}(x) = \sum_{i=1}^{n} I(x_i),$$

where I is an indicator function defined as:

$$I(z) = \begin{cases} 1,\ z \neq 0; \\ 0, \text{ otherwise.} \end{cases} \tag{7}$$

Next let us formulate optimization problems similar to **LASSO-I** and **LASSO-II**. We will use the robust L1 norm to quantify the difference between the observed and reconstructed outputs. The first formulation minimizes the L1-error of regression.

Cardinality-I

$$\min_x \|y - Ax\|_1 \tag{8}$$
$$\text{s.t.} \quad \mathrm{card}(x) \leq S, \tag{9}$$
$$L \leq x_i \leq U, \forall i = 1, \ldots, n, \tag{10}$$
$$x \in \mathbb{R}^n, \tag{11}$$

where S is a threshold on “sparsity”, i.e., we have an a-priori knowledge that no more than S spikes is possible in the initial signal x. L and U are upper and lower bounds on vector x components (in the considered case, $L = -1$ and $U = 1$).

The alternative formulation is as follows:

Cardinality-II

$$\min_x \mathrm{card}(x) \tag{12}$$
$$\text{s.t.} \quad \|y - Ax\|_1 \leq \epsilon \tag{13}$$
$$L \leq x_i \leq U, \forall i = 1, \ldots, n, \tag{14}$$
$$x \in \mathbb{R}^n, \tag{15}$$

Here we want to determine the signal with the minimal number of spikes that provides us with the output within predefined accuracy ϵ. These two problems, **Cardinality-I** and **Cardinality-II**, can be directly optimized without any additional programming by AORDA Portfolio Safeguard (PSG) solver. PSG includes the L1-error, L1 norm, and cardinality functions in the list of standard functions.

In order to use integer linear solvers such as IBM CPLEX, we rewrite the problems **Cardinality-I** and **Cardinality-II** as mixed integer linear programs.

The problem **Cardinality-I**, minimizing the error, can be equivalently rewritten as follows:

MILP I

$$\min_{x,z,t} \sum_{j=1}^{m} z_j \tag{16}$$

$$\text{s.t.} \quad -z_j \leq y_j - \sum_{i=1}^{n} a_{ji} x_i \leq z_j, \ \forall j = 1, \ldots, m, \tag{17}$$

$$L t_i \leq x_i \leq U t_i, \ \forall i = 1, \ldots, n, \tag{18}$$

$$\sum_{i=1}^{n} t_i \leq S, \tag{19}$$

$$z \geq 0, \tag{20}$$

$$x \in I\!R^n, \ z \in I\!R^m, \ t \in \{0,1\}^n \ \forall i = 1, \ldots, n. \tag{21}$$

The problem **Cardinality-II**, minimizing the number of spikes in the initial signal can be equivalently reformulated as follows:

MILP II

$$\min_{x,z,t} \sum_{i=1}^{n} t_i \tag{22}$$

$$\text{s.t.} \quad -z_j \leq y_j - \sum_{i=1}^{n} a_{ji} x_i \leq z_j, \ \forall j = 1, \ldots, m, \tag{23}$$

$$L t_i \leq x_i \leq U t_i, \ \forall i = 1, \ldots, n, \tag{24}$$

$$\sum_{j=1}^{m} z_j \leq \epsilon, \tag{25}$$

$$z \geq 0, \tag{26}$$

$$x \in I\!R^n, \ z \in I\!R^m, \ t \in \{0,1\}^n \ \forall i = 1, \ldots, n. \tag{27}$$

We also consider the relaxed version with L1-norm instead of the cardinality function, which is similar to the problem **LASSO-I** :

Relaxed III

$$\min_{x} \|y - Ax\|_1 \tag{28}$$

$$\text{s.t.} \quad \|x\|_1 \leq S, \tag{29}$$

$$L \leq x_i \leq U, \forall i = 1, \ldots, n, \tag{30}$$

$$x \in I\!R^n. \tag{31}$$

The last problem can be written as a linear program:

LP III

$$\min_{x,z,t} \sum_{j=1}^{m} z_j \tag{32}$$

$$\text{s.t.} \quad -z_j \leq y_j - \sum_{i=1}^{n} a_{ji} x_i \leq z_j, \ \forall j = 1, \ldots, m, \tag{33}$$

$$L \leq x_i \leq U, \ \forall i = 1, \ldots, n, \tag{34}$$

$$-t_i \leq x_i \leq t_i, \ t_i \geq 0 \ \forall i = 1, \ldots, n, \tag{35}$$

$$\sum_{i=1}^{n} t_i \leq S, \tag{36}$$

$$z \geq 0, \tag{37}$$

$$x, t \in I\!R^n, \ z \in I\!R^m, \ \forall i = 1, \ldots, n. \tag{38}$$

The next section compares computational performance of AORDA [2] nonlinear PSG solver and MILP solver IBM CPLEX [4].

We also consider problem formulation **Relaxed III D** which is equivalent to **Relaxed III** but all variables have lower bounds equal to 0. This fact is very important for linear programming approach because the sparse solution vector has almost all components equal to zero.

Relaxed III D formulation has the double set of variables. The positive additional variables are used instead of original variables having negative values.

Relaxed III D

$$\min_{x,z} \|y - Ax + Az\|_1 \tag{39}$$

$$\text{s.t.} \sum_{i=1}^{n}(x_i + z_i) \leq S, \tag{40}$$

$$0 \leq x_i \leq U,\ 0 \leq z_i \leq -L,\ \forall i = 1, \ldots, n, \tag{41}$$

$$x \in \mathbb{R}^n,\ z \in \mathbb{R}^n. \tag{42}$$

The **Relaxed III D** formulation was used to test Car and Tank solvers of Portfolio Safeguard on sparse problems provided by [6, 7].

3 Computational Experiments with Normally Distributed Data

This section presents numerical experiments for relatively simple test problems with matrices of samples from normal distribution. To test the performance of several optimization solvers based on quite different principals we have generated the Sparse Reconstruction Problem described in [20]. The matrix A, 1024×4096, is initially filled with independent standard Gaussian samples and then the rows were orthogonalized. Vector x contains 160 randomly placed ± 1 spikes and the observation $y = Ax + d$.

Input data and solutions for **Cardinality-I**, **Cardinality-II**, and **Relaxed-III** problems are available in Portfolio Safeguard (PSG) format. We also provide **MILP-I** and **MILP-II** problems in CPLEX LP format. These PSG and CPLEX files can be downloaded from www.ise.ufl.edu/uryasev/testproblems/case_studies/CS_Sparse_Reconstruction/CS_Sparse_Reconstruction .

We report performance of AORDA Portfolio Safeguard (PSG) 64 bit version (PSG 64 bit version runs about 50% faster than PSG 32 bit version) and IBM CPLEX 32 bit version. Computations were conducted on PC with 2.66 MHz processor.

We have considered two cases. In the first case, we assumed the absence of noise (i.e., $d = 0$). The calculation results are presented in Table 1. For Problem Type 1 and Problem Type 3 the PSG and CPLEX solvers have comparable performance (PSG solves these problems faster than CPLEX, but we used 64bit version of PSG and 32 bit version of CPLEX).

In the second case we considered Gaussian noise $d \sim N(0, 10^{-4})$. The result shown in Table 2 demonstrate that presence of noise significantly complicates computations for both PSG and CPLEX.

The Tables 1 and 2 show that **Relaxed III** problem is solved quite fast by PSG in no noise and in noisy cases (solving time is below 13 sec). Therefore, Problem **Relaxed III** is a good alternative for Problem **LASSO-O**, which

CPLEX			PSG		
Problem	time	objective	Problem	time	objective
MILP I	76.64	0	**Card. I**	42.5	3.067e-13
MILP II	*	44	**Card. II**	3234.24	160
LP III	18.74	0	**Rel. III**	12.38	1.063e-12

Table 1 Noise $d = 0$. Computation time in seconds for CPLEX and PSG (solver TANK). Problem Types: I = error minimization, II = cardinality minimization, and III = relaxed version of I. * CPLEX did not find the exact solution within 24 hours.

CPLEX			PSG		
Problem	time	objective	Problem	time	objective
MILP I	*	7.91e-5	**Card. I**	34.09	1.311e-4
MILP II	10793.9	160	**Card. II**	1925.96	160
LP III	2129.48	3.848e-5	**Rel. III**	10.66	1.315e-4

Table 2 Noise $d \sim N(0, 10^{-4})$. Computation time in seconds for CPLEX and PSG (solver TANK). Problem Types: I = error minimization, II = cardinality minimization, and III = relaxed version of I. * CPLEX crashed because of out-of-memory status.

is standardly considered in the literature. CPLEX solver showed good performance (solving time about 19 sec) for the equivalent problem **LP III** in no noise case, see Table 1. In noisy case it takes CPLEX quite long to solve (2129 sec), see Table 2. However, a faster performance of CPLEX could be achieved if we stop it when a lower precision is achieved; CPLEX minimized regression error with precision 3.848e-5, but PSG reported precision error 1.315e-4, see Table 1.

If number of spikes or upper bound on the number of spikes is known, then Problem **Cardinality-I** is a good alternative to Problems **LASSO-O** and **Relaxed III**. We knew (by construction of the problem) that number of spikes equals 160. Therefore, the constraint on the number of spikes was set to 160. PSG solves Problem **Cardinality-I** quite fast (solving time is below 43 sec), see Tables 1 and 2. Moreover, the solution can be immediately used "as it is" without eliminating small nonzero terms, as it is necessary in Problems **LASSO-O** and **Relaxed III**. The reason for this is that Problem **Cardinality-I** can be considered as the "correct" formulation of the original sparse reconstruction problem, compared to the approximate regularized Problems **LASSO-O** and **Relaxed III** which are computationally efficient "surrogates" of the original problem. CPLEX demonstrated good performance for the equivalent formulation **MILP I** in no noise case (solving time 77 sec), see Table 1, however CPLEX crashed in noisy case. The crash can probably be prevented by solving the problem with a 64 bit version of CPLEX or on some UNIX machine.

If precision of solution of the regression problem is specified in advance, then Problem **Cardinality-II** and the equivalent Problem **MILP II** can be considered. PSG solving times for Problem **Cardinality-II** are quite large (3234 sec in no noise case and 1926 in noisy case), see Tables 1 and 2. CPLEX performed poorly for the Problem **MILP II**, it did not find a solution during 24 hours for the no noise case, see Table 1, and it has found solution during 1079 sec for the noisy case, see Table 2.

4 Computational Experiments with SPARCO Problems

This section presents performance of AORDA PSG solvers (64 bit) for a set of real life problems from the SPARCO website [6, 7]. Computations were conducted on PC with 2.83 MHz processor.

Compared to the previous section, mostly, these are real life problems in imaging, compressed sensing, geophysics, information compressing, etc. References to the problem sources are included in the calculation results tables.

To access initial data we used software from [6, 7]. This site provides also a set of operators to deal with data. For the relatively small problems we converted all data to PSG format. The problems in PSG can be solved in PSG MATLAB or PSG RunFile environments. Large problems were solved with the External Function tool of PSG in MATLAB environment to avoid generating full matrix for the regressions problem (and to save computational time and memory).

We think that the cardinality problem formulations, such as **Cardinality-I** or **Cardinality-II** are much better formulations than **LASSO**-type formulations which do not directly control the cardinality of the decision vector. With the non- **Cardinality** formulations we need to solve the problem many times and adjust parameters of the problem until we find the solution with acceptable cardinality. Although the **Cardinality**-type formulations are preferable, it may be quite difficult to solve them for very large dimensions.

This section solves the problems with the **Relaxed III D** and **LASSO-O** formulations. Results of solving problems in **Relaxed III D** formulation with different values of upper bound S are shown in the Table 3. For some instances we used linearization of objective (39) to obtain fully linearized form of **Relaxed III D**. The table shows that the problems with several thousands of variables and several thousands of scenarios can be solved quite fast when the parameter S bounding the total absolute value of nonzero variables is small. However, for large values of S, the solving time may be quite large. The data for the problems and codes, and solutions in PSG format and references to the original sources of information are placed to this website www.ise.ufl.edu/uryasev/testproblems/case_studies/CS_Sparse_Reconstruction_SPARCO/CS_Sparse_Reconstruction_SPARCO.

For the really large problems, we have found that **LASSO-O** is the most preferable problem formulation, at least with the solvers included in PSG. Table 4 shows the calculation results for the large problems using the External Function tool of PSG in MATLAB environment. This table presents the runs with different value of the regularization coefficient in objective. MATLAB *.m files to prepare data, convert them to PSG format and solve problems using either fully converting to PSG format or External Function tool of PSG can be downloaded from www.ise.ufl.edu/uryasev/testproblems/case_studies/CS_Sparse_Reconstruction_SPARCO_matlab/CS_Sparse_Reconstruction_SPARCO_matlab.

5 Conclusion

We have proposed an approach for solving signal sparse reconstruction problems using nonconvex formulations with cardinality functions. In the first group of experiments, we have used test problems with matrices of samples from normal distribution. In order to test the performance of several optimization solvers, the Sparse Reconstruction Problem was generated which was described in [20]. Using different principals, the problems have been solved with AORDA Portfolio Safeguard (PSG) optimization package. Correspondent MILP formulations have been used to solve equivalent problems with IBM CPLEX optimization package and compare solvers' performance and results were obtained. Computational experiments show that the optimization package specialized for dealing with cardinality functions (PSG) has better performance than general MILP package on the considered test problems.

Second group of experiments were conducted using real-life medium and large scale problems that were downloaded from the SPARCO website [6, 7]. The problems have been solved with PSG optimization package. For the really large problems, the most preferable formulation was founded to be **LASSO-O**. Based on the computational experiments, we can conclude that PSG is a reasonable alternative to the specially developed algorithms for the sparse reconstruction problems, considered, for instance in [20]. The main advantages of using PSG versus other softwares are the simplicity of the PSG codes which have only several lines for even large-scale problems and the transparency of formulations where pre-coded mathematical functions are used to solve the problems.

References

1. Database of Creative Commons licensed sounds (2007). URL http://freesound.iua.upf.edu/
2. American Optimal Decisions (AORDA), Portfolio Safeguard (PSG) (2009). http://www.aorda.com
3. Compressed Sensing Resources (2009). http://www.dsp.ece.rice.edu/cs/
4. IBM CPLEX 11.2 (2009). http://www.ilog.com/products/cplex/
5. L1-magic (2009). http://www.acm.caltech.edu/l1magic/

Problem $m \times n$	S	Objective	Card $\lvert x_i\rvert \geq 1$	Card $\lvert x_i\rvert \geq$1e-3	Max $\lvert x_i\rvert$	Time sec	Solver	Ref.
problem 2	100	1.22E+00	9	9	29.4	0.5	C	[8, 14, 17]
1024×1024	200	6.63E-01	16	16	29.4	0.6	C	
	400	4.42E-02	42	44	29.4	4.5	C	
	500	1.24E-14	67	71	43.1	2.8	C	
problem 3	100	6.38E-01	2	2	69.0	0.5	T	[6, 7]
1024×2048	140	8.03E-02	6	6	90.5	4.1	T	
	220	2.27E-03	35	119	91.5	130.3	T	
	240	4.58E-14	34	1004	90.5	460.3	Cfl	
problem 5	100	1.00E+00	3	3	56.4	0.5	C	[6, 7]
300×2048	140	2.71E-01	4	39	66.5	8.0	T	
	170	5.79E-02	15	127	67.6	41.1	Cfl	
	200	1.91E-14	15	299	68.1	3.6	Cfl	
problem 6	170	1.41E+02	14	14	35.1	1.4	T	[13]
600×2048	800	6.62E+01	64	85	57.5	56.5	T	
	1700	1.59E+00	115	423	76.6	185.9	Cfl	
	2000	3.77E-12	285	601	78.2	30.9	Cfl	
problem 7	3	5.81E-02	0	13	0.65	2.3	C	[9]
600×2560	10	3.39E-02	0	20	0.79	7.9	C	
	17	1.02E-02	0	20	0.94	12.8	C	
	20	1.37E-13	9	20	1.00	1.6	C	
problem 8	5	4.94E-02	0	19	0.63	6.0	T	[9]
600×2560	15	1.64E-02	0	20	0.87	11.9	T	
	19	3.29E-03	0	20	0.97	11.6	T	
	20	1.36E-13	12	20	1.00	1.2	T	
problem 9	15	3.14E-01	5	7	4.3	0.01	C	[14, 17]
128×128	30	9.30E-02	9	10	5.0	0.02	C	
	40	7.81E-03	12	12	5.0	0.02	C	
	45	2.92E-13	12	12	5.0	0.04	C	
problem 10	600	1.32E-01	10	10	121.5	0.5	C	[14, 17]
1024×1024	800	6.10E-02	11	11	121.5	0.6	C	
	950	1.15E-02	12	12	131.7	0.7	C	
	1000	1.64E-12	14	14	149.2	143.1	C	
problem 11	15	1.22E+00	8	28	1.94	3.1	C	[6, 7]
256×1024	20	5.18E-01	10	45	2.10	6.5	T	
	23	1.29E-01	11	58	2.17	35.5	Cfl	
	25	2.53E-14	11	221	2.18	31.6	Cfl	
problem 601	100	6.48E+00	15	26	23.2	81.4	C	[25]
3200×4096	200	2.78E+00	46	173	23.2	1965.1	V	
	300	8.42E-01	63	836	23.2	2894.0	V	
	400	9.18E-08	7	3478	23.2	509.2	V	
problem 602	200	6.80E+00	40	209	15.9	1874.2	V	[25]
3200×4096	400	2.99E+00	85	728	16.1	2906.6	V	
	600	3.60E-01	130	1060	16.3	2982.0	V	
	700	9.71E-05	142	3964	16.2	272.6	V	
problem 603	50	3.35E-01	4	4	24.4	2.2	C	[20]
1024×4096	100	1.75E-01	12	16	30.9	10.7	C	
	200	4.16E-02	30	294	31.0	1150.4	Cfl	
	300	2.32E-14	39	1019	32.4	391.0	Cfl	
problem 902	0.2	2.32E-02	0	1	0.20	0.07	C	[21, 22, 23]
200×1000	0.5	1.86E-02	0	3	0.40	0.12	C	
	1.5	3.60E-03	0	3	0.78	0.12	C	
	2.0	3.12E-14	0	5	0.87	26.30	C	
problem 903	5	6.17E-01	3	6	1.18	0.6	C	[19]
1024×1024	7	4.06E-01	3	11	1.47	1.4	C	
	12	9.92E-02	6	48	2.19	29.7	T	
	13	9.18E-05	6	17	2.19	10.7	C	

Table 3 Calculation results for medium-size SPARCO problems in **Relaxed III D** form. S = upper bound; Card $|x_i| \geq 1$ = number of variables with absolute value greater than 1; Card $|x_i| \geq$1e-3 = number of variables with absolute value greater than 1e-3; Max $|x_i|$ = maximum value of $|x_i|$; Solver: C = Car PSG solver, T = Tank PSG Solver, V=Van PSG Solver; Cfl = Car PSG sovler with full linearization; Tfl = Tank PSG solver with full linearization

Problem $m \times n$	τ	Objective	Card $\|x_i\| \geq 1$	Card $|x_i| \geq$1e-3	Max $\|x_i\|$	Time sec	Ref.
problem 3	10	1.308E+03	2	2	80.2	0.02	[6, 7]
1024×2048	1	1.780E+02	5	38	89.3	0.05	
	0.1	2.164E+01	29	111	90.3	0.18	
	0.01	2.217E+00	35	121	90.5	2.57	
problem 5	10	1.226E+03	3	3	57.2	0.04	[6, 7]
300×2048	1	1.580E+02	4	33	66.8	0.17	
	0.1	1.778E+01	18	109	67.8	1.67	
	0.01	1.810E+00	23	156	67.9	19.24	
problem 6	10000	9.652E+06	35	37	57.1	0.8	[13]
600×2048	1000	1.586E+06	102	175	74.3	3.6	
	100	1.722E+05	118	405	77.0	36.8	
	10	1.745E+04	121	555	77.6	385.6	
problem 7	0.2	2.252E+00	0	14	0.64	0.07	[9]
600×2560	0.1	1.562E+00	0	20	0.82	0.10	
	0.05	8.905E-01	0	20	0.91	0.11	
	0.02	3.825E-01	0	20	0.96	0.15	
problem 8	0.2	4.381E+00	0	13	0.37	0.09	[9]
600×2560	0.1	3.799E+00	0	20	0.68	0.11	
	0.05	3.156E+00	0	20	0.84	0.13	
	0.01	2.470E+00	0	20	0.97	0.23	
problem 401	2	2.431E+02	1	9	1.45	1.2	[1]
29166×57344	1	2.273E+02	6	121	2.43	3.1	
	0.5	1.849E+02	25	497	2.93	8.2	
	0.2	1.152E+02	37	1713	3.40	23.5	
problem 402	2	2.643E+02	1	10	1.45	1.8	[1]
29166×86016	1	2.473E+02	6	137	2.43	4.4	
	0.5	2.007E+02	26	545	2.93	13.9	
	0.2	1.248E+02	37	1898	3.40	32.1	
problem 403	10	1.124E+04	57	65	10.6	2.7	[6, 7]
196608×196608	3.3	6.174E+03	119	122	20.2	3.0	
	0.33	1.136E+03	277	970	25.5	10.0	
	0.1	4.802E+02	449	5591	25.8	35.2	
problem 601	10000	8.474E+05	4	4	22.7	420.7	[25]
3200×4096	1000	1.948E+05	26	60	23.1	598.0	
	500	1.187E+05	38	124	23.1	808.1	
	200	5.741E+04	54	288	23.1	1460.1	
problem 602	1000	3.165E+05	37	132	16.1	543.6	[25]
3200×4096	500	2.086E+05	68	335	16.2	848.9	
	200	1.049E+05	95	717	16.2	1117.9	
	100	5.745E+04	110	859	16.2	2038.8	
problem 603	2	1.893E+02	5	5	25.6	0.20	[20]
1024×4096	1	1.210E+02	7	9	29.3	0.20	
	0.1	2.145E+01	28	169	32.0	0.79	
	0.01	2.544E+00	38	605	32.2	6.64	
problem 701	10	6.911E+03	48	49	13.5	0.59	[20]
65536×65536	5	4.336E+03	55	55	18.6	0.67	
	2	2.110E+03	68	87	21.5	0.75	
	1	1.198E+03	107	144	22.3	0.88	
problem 702	0.05	2.484E+00	0	4	0.23	0.30	[20]
16384×16384	0.04	2.470E+00	0	26	0.47	0.55	
	0.02	2.087E+00	0	169	0.76	1.30	
	0.01	1.332E+00	0	194	0.88	1.88	

Table 4 Calculation results for medium and large size SPARCO problems in **LASSO-O** form using External Function tool of PSG. τ = regularization coefficient in objective; Card $|x_i| \geq 1$ = number of variables with absolute value greater than 1; Card $|x_i| \geq$1e-3 = number of variables with absolute value greater than 1e-3; Max $|x_i|$ = maximum value of $|x_i|$.

6. Berg, E.v., Friedlander, M.P.: SPARCO: A toolbox for testing sparse reconstruction algorithms (2008). URL http://www.cs.ubc.ca/labs/scl/sparco/
7. Berg, E.v., Friedlander, M.P., Hennenfent, G., Herrmann, F., Saab, R., Yılmaz, Ö.: Sparco: A testing framework for sparse reconstruction. Tech. Rep. TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver (2007)
8. Buckheit, J., Donoho, D.L.: Wavelets and Statistics, chap. Wavelab and reproducible research. Springer-Verlag, Berlin, New York (1995). URL http://citeseer.ist.psu.edu/article/buckheit95wavelab.html
9. Candès, E., Romberg, J.: L1-magic. http://www.l1-magic.org/ (2007)
10. Candes, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory **52**(2), 489509 (2006)
11. Candes, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics **59**(8), 1207–1223 (2006)
12. Candes, E., Tao, T.: The dantzig selector: Statistical estimation when p is much larger than n. Annals of Statistics **35**, 2313 (2007). URL doi:10.1214/009053606000001523
13. Candés, E.J., Romberg, J.: Practical signal recovery from random projections. In: Wavelet Applications in Signal and Image Processing XI, Proc. SPIE Conf. 5914. (2004)
14. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998). URL http://epubs.siam.org/SISC/volume-20/art30401.html
15. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 12891306 (2006)
16. Donoho, D.L.: For most large underdetermined systems of equations, the minimal ℓ_1-norm near-solution approximates the sparsest near-solution. Communications on Pure and Applied Mathematics **59**(7), 907–934 (2006)
17. Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. Biometrika **81**(3), 425–455 (1994). URL http://citeseer.ist.psu.edu/donoho93ideal.html
18. Donoho, D.L., Tsaig, Y.: Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. Tech. rep., Institute for Computational and Mathematical Engineering, Stanford University (2006). http://www-stat.stanford.edu/ donoho/
19. Dossal, C., Mallat, S.: Sparse spike deconvolution with minimum scale. In: Proceedings of Signal Processing with Adaptive Sparse Structured Representations, pp. 123–126. Rennes, France (2005). URL http://spars05.irisa.fr/ACTES/PS2-11.pdf
20. Figueiredo, M., Nowak, R., Wright, S.: Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. Selected Topics in Signal Processing, IEEE Journal of **1**(4), 586–597 (2007). DOI 10.1109/JSTSP.2007.910281. URL http://www.lx.it.pt/ mtf/GPSR
21. Hennenfent, G., Herrmann, F.J.: Sparseness-constrained data continuation with frames: Applications to missing traces and aliased signals in 2/3-D. In: SEG International Exposition and 75th Annual Meeting (2005). URL http://slim.eos.ubc.ca/Publications/Public/Conferences/SEG/hennenfent05seg.pdf
22. Hennenfent, G., Herrmann, F.J.: Simply denoise: wavefield reconstruction via coarse nonuniform sampling. Tech. rep., UBC Earth & Ocean Sciences (2007)
23. Herrmann, F.J., Hennenfent, G.: Non-parametric seismic data recovery with curvelet frames. Tech. rep., UBC Earth & Ocean Sciences Department (2007). URL http://slim.eos.ubc.ca/Publications/Public/Journals/CRSI.pdf. TR-2007-1

24. Kutyniok, G.: Theory and applications of compressed sensing (2012)
25. Takhar, D., Laska, J.N., Wakin, M., Duarte, M., Baron, D., Sarvotham, S., Kelly, K.K., Baraniuk, R.G.: A new camera architecture based on optical-domain compression. In: Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Computational Imaging, vol. 6065 (2006)
26. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B **58**, 267–288 (1994)
27. Wang, L., Gordon, M.D., Zhu, J.: Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In: ICDM '06: Proceedings of the Sixth International Conference on Data Mining, pp. 690–700. IEEE Computer Society, Washington, DC, USA (2006). DOI http://dx.doi.org/10.1109/ICDM.2006.134