2012

SparOptLib - A Testing Library of Sparse Solution Recovery Problems

Ana-Iulia Alexandrescu
Lehigh University

SparOptLib – A Testing Library of Sparse Solution Recovery Problems

by

Ana-Iulia Alexandrescu

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial and Systems Engineering

Lehigh University

January 2012

Certification of Approval

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science degree.

Date

Thesis Advisor

Chairperson of Department

Acknowledgments

This thesis represents a joint work with Dr. Katya Scheinberg , thesis advisor, and Bai Xi, colleague and Doctoral Candidate, both from the Department of Industrial and Systems Enfineering.

Table of Contents

Abstract

The Sparse Solution Recovery (SSR) problem arises in a very large number of practical applications, of which some of the most notable are compressed sensing, image and signal processing, seismic data recovery, gene sequencing, feature selection in machine learning. Given this wide array of applications that rely on effective recovery of sparse solutions from large underdetermined linear systems, developing efficient algorithms to solve the SSR problem is of paramount importance. Last ten years have witnessed an explosion of algorithms that aim to solve the SSR, most of which use a variety of different convex relaxations of the original formulation. The absence of a reference test set of problems and a proposed method to quantify the quality of the solution reached by any of these solvers prevents researchers from estimating which problems are hard and under what conditions some approaches lead to faster convergence than others. Through SparOptLib, we aim to provide researchers in the field with a collection of problems and a framework for testing sparse solution recovery algorithms. The problems are drawn from a variety of applications, including compressed sensing and signal processing, and cover a wide range of size, difficulty, and sparsity. The current version of the library contains over 300 instances provided in a standard format, which includes suggested target accuracy for optimization. It is our hope that SparOptLib will provide a universal testing framework and will enable researchers to develop improved algorithms for this class of problems.

Introduction

The Sparse Solution Recovery (SSR) problem arises in a number of practical applications. Most notably, compressed sensing and signal processing, machine learning, seismic data recovery and gene sequencing rely on recovering sparse solutions to linear underdetermined systems. Recently, there has been an explosion of interest in this special class of optimization problems, mainly because of an increased interest in machine learning and the development of efficient algorithms for machine learning. These circumstances led to a very active research interest in developing efficient algorithms for recovering sparse solutions, and a number of software packages exist today.

Collections of test problems exist in various areas of optimization, including NETLIB for linear programming problems, CUTEr for nonlinear optimization, SDPLIB for semi-definite programming, and MIPLIB for mixed-integer linear programming problems. These collections have become standard for testing algorithms, benchmarking and calibrating parameters to improve algorithm robustness and convergence speed and for providing a wide spectrum of problems in their respective areas. In turn, this has led to the development of improved algorithms and has provided insight into problem structures that can be leveraged for better algorithmic convergence.

Through SparOptLib, we aim to provide researchers with a similar collection of problems and testing framework in the area of SSR. Currently, the library contains over 300 instances drawn from a variety of applications and sources. The problems reflect a wide range of difficulty and size and we hope they provide a complex enough environment for testing the robustness of different solvers and solution approaches. The following paper provides a background on the SSR problem and its convex relaxations, a short inventory of the solver packages available to solve SSR, a detailed description of the library and a user's manual that we hope will enable researchers to benefit from SparOptLib.

The Sparse Solution Recovery Problem

The motivation behind sparse optimization is simple: sometimes, simple approximate solutions that can be easily obtainable are preferable to exact solutions that are computationally prohibitive (S. Wright 2009). This may arise because simple solutions are far easier to obtain or more robust, or because completely accurate cannot be obtained because of noise levels. In the context of large, full-rank underdetermined linear systems of the type

$$Ax = b$$

we have an infinity of solutions, but we are interested in finding sparse solutions x that satisfy the equations. In particular, if our aim is to find the sparsest solution, the sparse solution recovery problem can be modeled as follows:

$$\min \|x\|_0$$

$$s.t. Ax = b$$

where $x \in R^n$ represents the signal we are trying to recover, A is an $m - by - n$ matrix, and $b \in R^m$ represents the vector of observations. By $\|x\|_0$ we mean the 0-norm of vector x, defined as the number of non-zero components in the signal.

Convex Relaxations

The zero-norm integer formulation of the SSR problem was proved to be NP-hard by Davis et al., and thus is rarely solved in practice. Instead, convex relaxations that replace the zero-norm with the L1-norm are used. Additionally, since most solvers focus on solving the L1-relaxations, obtaining a perfectly feasible solution is often impractical, and thus we are often times satisfied with just *approximately* complying with the feasibility constraints $Ax = b$ and we use instead $\|Ax - b\|_p \leq \varepsilon_b, \varepsilon_b \ \geq 0$.

There are three L1-relaxations that are most recurring in the literature and are widely used in solver packages (S. Wright 2009):

1. Basis-Pursuit De-Noising (BPDN)

$$\min \|x\|_1$$

$$s.t. \|Ax - b\|_2 \leq \varepsilon_b$$

2. Lagrangian relaxation of the BPDN formulation (LAG)

$$\min \frac{1}{2}\|Ax - b\|_2^2 + \rho\|x\|_1$$

3. Least Absolute Shrinkage and Selection Operator (LASSO)

$$\min \frac{1}{2}\|Ax - b\|_2^2$$

$$s.t. \|x\|_1 \leq \ \sigma, \sigma > 0$$

Each of these formulations takes a different approach to solving the L1-relaxation of the SSR problem, and often reach different solution if applied to the same problem. The next section provides a brief overview of the solvers that exist currently for solving the SSR problem and the respective convex L1-reformulation each solver uses.

Review of Algorithms and Solution Approach

Following is a list of several SSR solvers, grouped by the formulation they approach. This list is by no means exhaustive – it groups several of the more popular solvers currently available.

BPDN

- TFOCS – Templates for First-Order Conic Solvers (Becker, Candès and Grant 2011)
- NESTA – Nesterov's Algorithm (Becker, Bobin and Candès 2009)
- SALSA – Split Augmented Lagrangian Shrinkage Algorithm(Afonso, Bioucas-Dias and Figueiredo 2009)
- SPGL1 – Spectral-Projected Gradient Algorithm (Berg and Friedlander 2010)
- YALL1 – Your Algorithm for L1(Yang and Zhang 2009)

LAG

- TFOCS
- NESTA
- SALSA
- SPGL1
- YALL1
- IST – Iterative Shrinking Threshold(Daubechies, Defrise and Mol 2004)
- TwIST – Two Step Ietartive Shrinkage/Thresholding(Bioucas-Dias and Figueiredo 2007)
- FISTA – Fast Iterative Shrinkage/ Thresholding Algorithm (Beck and Teboulle 2009)
- FPC – Fixed-Point Continuation scheme (Hale, Yin and Zhang 2007)

- GPSR – Gradient Projection for Sparse Reconstruction (Figueiredo, Nowak and Wright 2007)
- SpaRSA – Sparse Reconstructions by Separable Approximation (Wright, Nowak and Figueiredo 2008)
- ALM – Augmented Lagrangian Method(Yang, et al. 2010)
- FALM – First-order Augmented Lagrangian Method (Aybat and Iyengar 2010)

LASSO

- SPGL1 – Spectral-Projected Gradient Algorithm
- YALL1 – Your Algorithm for L1
- IST – Iterative Shrinkage/Thresholding

It is worth reiterating that this is a very brief list and it is by no means exhaustive. Specifically, all solvers listed above use first-order methods for solving SSR. There are other convex optimization-based solvers that use interior-point methods, as well as non-optimization-based solvers (“greedy” algorithms). S. Becker’s webpage (List of sparse and Low-rank recovery algorithms) offers a more comprehensive review of the algorithms.

SparOptLib – a Collection of Sparse Solution Recovery Problems

The wide array of solvers available, the various formulations and relaxations that they tackle, and the lack of a standardized reference problem set create issues in assessing the relative difficulty of a problem and the solver performance, as well as in improving the robustness of the algorithms. We created SparOptLib to provide a standardized format for sparse solution recovery instances, which we found to be compatible with most of the solvers we came across.

Each instance represents a structure p with the following fields:

- A – m–by–n matrix of the system
- rhs –right-hand side vector of observations
- sol –true solution (provided by the authors of the problem)
- m, n –size of A (also referred to as the size of the problem)
- $noise$ –noise level
- $info$ –instance documentation

SparOptLib currently contains over 300 instances of varying size and difficulty. These problems can be grouped according to their origin, into three categories.

The first contributions came from A. Nemirovski, who provided the four problems that contain his name. Not much other information is available on these problems.

Next, some 30+ instances were generated using the Sparco Toolbox(2007). Sparco represents a collection of sparse signal recovery problems and an environment to create new problems using the suite of linear operators provided. For these instances, several problems provided in Sparco were selected to illustrate a varied range of applications. However, we restricted ourselves to the use of those problems for which a solution was provided, as this was needed to establish a reference for the target accuracy for optimization to be strived for by the solvers. In addition, we created three sizes for each instance: a “small” instance, which was the original problem provided by Sparco; a “medium”-sized version, which made each of the dimensions of the problem five times bigger than in the “small” version, and thus the problem grew in size 25 times; a “large” version, which similarly increased each of the dimensions tenfold and thus led to an instance 100 times bigger than its “small” counterpart. This procedure allowed us to introduce size variability, which has been documented to significantly impact solver performance. The naming convention kept the original Sparco ID of the problem and appended the relative size as a one-letter suffix (ex. spaco1s, sparco1m and sparco1l represent three instances of different sizes of the same problem, which has the Sparco ID 1). For readers interested in learning more about Sparco, we recommend referring to the project’s website maintained at the Computer Science Department at the University of British Columbia and to the technical report released with the toolbox, listed under the references page. A note should be made that Sparco provides the matrix A as a function handle rather than in matrix form. While this is strictly an implementation consideration and provides no different behavior in solvers, the additional use of a suite of linear operators to recover A and provide the input to the solver is needed. Please, refer below to the user’s guide for directions on how to get these operators.

The third and largest category of problems was obtained from the Sparse Exact and Approximate Recovery (SPEAR) project (2011). This project is a collaboration between the Institute for Mathematical Optimization and the Institute for Analysis and Algebra from

Technische Universität Braunschweig and it aims to develop a better understanding of the conditions under which sparse solution recovery is possible. 273 problems adapted from the L1-Test Set developed as part of the SPEAR project we re-cast in the standard format proposed and included in SparOptLib, contributing a very large proportion of the library. For the readers interested in reading more about the SPEAR project, please refer to the project website and the technical report that accompanied the L1 Test set, cited in the references.

All problems taken together provide a wide variety, which can be traced on several axes. Some of those that have been demonstrated to affect the solver behavior are listed below and provided for each problem in the "library catalog" in Appendix B and on the library webpage:

- Problem size, given by the size of the system matrix A
- Solution sparsity, provided both as the number of the non-zero components (the zero-norm) and as the relative ratio of the number of non-zero components to the size of the solution (the sparsity ratio)
- Dynamic range, defined as the ratio of the largest to the smallest non-zero components of the solution (recorded in absolute value)

Thus, it is our hope that the SparOptLib collection covers a varied enough range of problems that would render it useful for researchers developing sparse recovery algorithms.

SparOptLib – A Testing Environment for Sparse Solution Recovery Algorithms

Through SparOptLib, we aim to provide both a test set to be used as a reference, and a method to assess solution quality and/or solver performance. The difficulty in the latter comes mainly from the wide array of approaches taken to solve the problem. Not only are there multiple possible relaxations to the original sparse solution recovery problem, but there are also many solvers, each with a different approach to solving varying relaxations of the problem. Thus, it becomes difficult to evaluate whether a problem is more difficult than another, or to characterize circumstances under which a particular solution approach is better than another one. We propose a framework through which such evaluations can be more easily made, which is captured in the instances by two parameters, εb and εx, given in three respective pairs. Intuitively, εb measures relative "distance from feasibility" for the current solution, while εx represents a reasonable target accuracy for optimization. A good solution x satisfies the following relationships with respect to the solution provided, x^* :

$$\|x\|_1 \leq (1 + \varepsilon_x)\|x^*\|_1$$

$$\|Ax - b\|_p \leq \|Ax^* - b\|_p + \varepsilon_b\|b\|_p$$

The first relationship places the solution provided by the solver x within a required radius of "sparsity" with respect to the sparse solution provided with the problem x^*, while the second relationship controls the feasibility of the solution. Thus, the two parameters we introduce model relative tolerance with respect to the tradeoff between sparsity and accuracy/feasibility.

For each problem, three pairs of ε_x and ε_b are provided. The procedure to obtain the three pairs was the following: three values were selected for the parameter ε_b, and with those, the Spectral Projected Gradient Algorithm (SPGL1) solver package developed by M. Friedlander and E. van den Berg was used to obtain corresponding values for ε_x. The values are given in two arrays, ε_b and ε_x, with entries at matching indices corresponding to the same instance. With this additional information, the quality of a solution can be measured by how small the corresponding ε -values are. For example, for a pair value $\varepsilon_b = 10^{-8}$ and $\varepsilon_x = 10^{-4}$ for a particular problem p and using the 2-norm, we mean that within the feasible set

$$\|Ax - b\|_2 \leq \sigma = \|Ax^* - b\|_2 + 10^{-8}\|b\|_2$$

SPGL1 can reach a solution within

$$\|x\|_1 \leq (1 + 10^{-4})\|x^*\|_1$$

L1-accuracy from the solution provided with the problem.

Intuitively, one can see a tradeoff between the two parameters: allowing for larger violations on feasibility has the potential to yield more accurate solutions, and vice-versa, relaxing the requirements on accuracy can produce solutions within the original feasible set.

A few remarks:

The parameter choice one makes for the solver influences its performance. To reach the values provided, we used an “out-of-the-box” version of SPGL1 – no parameter tweaking took place. It is also expected that for some applications, different values for ε_x and ε_b may be appropriate. Thus, rather than an objective reference for solution quality, the ε_x and ε_b pairs provide a way to capture the tradeoff between sparsity and feasibility, which is illustrated on a “case study” of SPGL1 that resulted in the particular values provided in the library. We leave it up to the researchers to define criteria for optimal performance and to achieve it by calibrate their solvers on the SparOptLib test set.

User's Guide

Download

The problems in SparOptLib are available for download at coral.ie.lehigh.edu/SPAROPTLIB.

Several options for download are available:

- The zip file of the entire library (approx.2.5GB)
- Corresponding subsets of problems grouped by origin or
- Individual instances

A library catalog documenting several features of the problems is provided to help users characterize and locate relevant problems for their use. These features include the size of the instance and the size of the file, the sparsity of the given solution and the dynamic range of the coordinates in the solution provided. All this information is provided with the instance as well.

Instance format

Each instance is organized in a .mat file, in a standard format using the following structure:

- A – m−by−n matrix of the system, given as a matrix or function handle
- rhs –right-hand side vector of observations
- sol −true solution (provided by the authors of the problem)
- m, n −size of A
- $\varepsilon_b \in 10^{-8}, 10^{-4}, 10^{-2}$
- $\varepsilon_x \in \varepsilon_{1,2}, \varepsilon_3$, a reasonable accuracy for an estimated signal from the true solution
- $noise$ −noise level

- $info$ –instance documentation (includes all the original information supplied with the problem, and additional fields such as sparsity, sparsity ratio and dynamic range of the solution)

This structure provides the input to the sparse solution recovery algorithms to be used. While some pre-processing may be required for individual inputs depending on the solver setup, we found that this structure complies with most of the solvers available. A file demonstrating the use of an instance with the spgl1 package is included for reference.

Matrix vs. Function Handle

A quick note should be made about the problems generated using the Sparco Toolbox. Sparco represents an environment for creating sparse signal reconstruction problems using a suite of linear operators provided. In the current version, all problems that contain "sparco" in the file name have been created using the toolbox. These problems store the information contained in 'A' as a function handle, rather than a matrix. In order to recover the information in A and comply with solver input setups, the user needs to download and install the Sparco Toolbox or the Spotbox (a lightweight version of Sparco that consists only of the linear operators needed to recover 'A' from the handle). The Spotbox is provided for download with the rest of the SparOptLib. The Sparco Toolbox is available for download on the project website(SPARCO: A toolbox for testing sparse reconstruction algorithms).

References

Afonso, Manya V., Jose M. Bioucas-Dias, and Mario A. T. Figueiredo. "An Augmented Lagrangian Approach to the contrained optimization formulation of imaging inverse problems." *IEEE Transations on Image Processing*, December 2009.

Aybat, Necdet Serhat, and Garud Iyengar. "A First-Order Augmented Lagrangian Method for Compressed Sensing." *Optimization Online*, 2010.

Beck, Amir, and Marc Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems." *SIAM Journal on Imaging Sciences*, 2009.

Becker, S. *List of sparse and Low-rank recovery algorithms.* http://www.ugcs.caltech.edu/~srbecker/algorithms.shtml (accessed 2011).

Becker, S., E. J. Candès, and M. Grant. *Templates for convex cone problems with applications to sparse signal recovery.* Technical Report, Stanford University, 2011.

Becker, S., J. Bobin, and E. J. Candès. "NESTA: A fast and accurate first-order method for sparse recovery." *SIAM Journal on Imaging Sciences*, 2009.

Berg, E. van den, and M. P. Friedlander. *Sparse Optimization with least-squares constraints.* Technical Report, Dep. of Computer Science, Univ. of British Columbia, 2010.

Berg, E. van den, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and O. Yilmaz. *SPARCO: A toolbox for testing sparse reconstruction algorithms.* http://www.cs.ubc.ca/labs/scl/sparco/ (accessed 2011).

Bioucas-Dias, J., and M. Figueiredo. "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration." *IEEE Transactions on Image Processing* (IEEE Transactions on Image Processing), 2007.

Daubechies, Ingrid, Michel Defrise, and Christine De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint." *Communications on Pure and Applied Mathematics*, 2004.

Figueiredo, Mario A. T., Robert D. Nowak, and Stephen J. Wright. "Gradient Projection for Sparse Reconstruction: application to compressed sensing and other inverse problems." *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 2007.

Hale, E. T., W. Yin, and Y. Zhang. *A Fixed-Point Continuation Method for l1-Regularized Minimization with Applications to Compressed Sensing.* Technical Report, Houston: Department of Computational and Aoolied Mathematics, Rice University, 2007.

Lorenz, Dirk A. "Constructing test instances for Basis Pursuit Denoising." Technical Report, 2011.

Wright, Stephen J., Robert D. Nowak, and Mario A. T. Figueiredo. "Sparse Reconstruction by Separable Approximation." *IEEE International Conference on Acoustics, Speech and Signal Processing.* 2008.

Wright, Stephen. "Sparse Optimization Methods." *Conference on Advanced Methods and Perspectives in Nonlinear Optimization and Contro.* Toulouse, 2009.

Yang, Allen Y., Arvind Ganesh, Zihan Zhou, S. Shankar Sastry, and Yi Ma. "A Review of Fast l1-Minimization Algorithms for Robust Face Recognition." *SIAM Journal on Imaging Sciences*, 2010.

Yang, Junfeng, and Yin Zhang. *Alternating direction algorithms for l-1 problems in compressive sensing.* Technical Report, Department of Computational and Applied Mathematics, 2009.

Yilmaz, E. {van den} Berg and M. P. Friedlander and G. Hennenfent and F. Herrmann and R. Saab and O. *Sparco: A testing framework for sparse reconstruction.* Technical, Vancouver: University of British Columbia, Department of Computer Science, 2007.

Appendix A: Instance Structure

Each instance is provided in a standard format in a .mat file. The following structure is used:

□A – m−by−n matrix of the system, given as a matrix or function handle

□rhs –right-hand side vector of observations

□sol −true solution (provided by the authors of the problem)

□m, n −size of A

□$\varepsilon_b \in 10^{-8}, 10^{-4}, 10^{-2}$

□$\varepsilon_x \in \varepsilon_1, 2, \varepsilon_3$, a reasonable accuracy for an estimated signal from the true solution

□$noise$ −noise level

□$info$ −instance documentation (includes all the original information supplied with the problem)

Appendix B: SparOptLib Catalog

Table 1: SparOptLib Catalog

name	m	n	sparsity	sparsity_ratio	dyn_range	file_size
nemirovski1	1036	1036	16	0.0154	5.9915	8228022
nemirovski2	2062	2062	23	0.0112	7.5365	32589781
nemirovski3	2062	2062	23	0.0112	17.1223	32590114
nemirovski4	2062	4124	16	0.0039	0.0511	45559808
sparco10l	10240	10240	16	0.0016	0.4643	2623
sparco10m	5120	5120	16	0.0031	0.4643	2297
sparco10s	1024	1024	12	0.0117	0.8138	1912
sparco11m	1280	5120	32	0.0063	1.3108	101005129
sparco11s	256	1024	32	0.0313	1.3108	4044011
sparco1l	20480	40960	20483	0.5001	1.84E+20	462383
sparco1m	10240	20480	10243	0.5001	1.18E+20	230028
sparco1s	2048	4096	2050	0.5005	1.01E+19	45395
sparco2m	5120	5120	198	0.0387	2.6000	2881
sparco2s	1024	1024	77	0.0752	3.9091	2322
sparco3l	20480	40960	122	0.0030	183.7930	152398
sparco3m	10240	20480	122	0.0060	129.9610	74778
sparco3s	2048	4096	122	0.0298	58.1205	13017
sparco4l	20480	40960	20600	0.5029	9.64E+19	616163
sparco4m	10240	20480	10360	0.5059	2.38E+19	305595
sparco4s	2048	4096	2168	0.5293	6.75E+18	57541
sparco5m	1500	20480	63	0.0031	3.0000	236095776
sparco5s	300	4096	63	0.0154	3.0000	9446890
sparco6m	3000	10240	9658	0.9432	1.0713	123832
sparco6s	600	2048	1917	0.9360	1.3599	28823
sparco7m	3000	12800	20	0.0016	1.0000	590243641
sparco7s	600	2560	20	0.0078	1.0000	23615049
sparco8m	3000	12800	20	0.0016	1.0000	590268590
sparco8s	600	2560	20	0.0078	1.0000	23618987
sparco902l	2000	10000	3	0.0003	1.9941	25863
sparco902m	1000	5000	3	0.0006	1.9941	14454
sparco902s	200	1000	3	0.0030	1.9941	4635
sparco903s	1024	1024	12	0.0117	1.0320	71792
sparco9l	1280	1280	16	0.0125	0.5000	1852
sparco9m	640	640	16	0.0250	0.5000	1755
sparco9s	128	128	12	0.0938	0.8600	1617
spear1	512	1024	8	0.0078	3.2782	206800
spear10	512	1024	18	0.0176	2.3133	743914
spear100	1024	3072	22	0.0072	6.8040	10557151

spear101	1024	3072	11	0.0036	2.7045	1170671
spear102	1024	3072	13	0.0042	0.6990	1171638
spear103	1024	3072	9	0.0029	0.1138	547577
spear104	1024	3072	14	0.0046	1.1371	552094
spear105	1024	3072	7	0.0023	0.3924	2601208
spear106	1024	3072	36	0.0117	1.2936	2599132
spear107	1024	3072	13	0.0042	1.2284	10617302
spear108	1024	3072	15	0.0049	0.4652	10617335
spear109	1024	3072	26	0.0085	1.9015	4064484
spear11	512	1024	26	0.0254	4.0145	177889
spear110	1024	3072	27	0.0088	0.4315	4064503
spear111	1024	3072	33	0.0107	0.0895	1019617
spear112	1024	3072	34	0.0111	0.6143	1019635
spear113	1024	3072	33	0.0107	1.2845	12954888
spear114	1024	3072	33	0.0107	0.8868	12954897
spear115	1024	3072	27	0.0088	10.5576	1020641
spear116	1024	3072	27	0.0088	0.5648	1020727
spear117	1024	3072	26	0.0085	0.7848	1644463
spear118	1024	3072	27	0.0088	1.0710	1644471
spear119	1024	3072	33	0.0107	1.0093	24181444
spear12	512	1024	27	0.0264	0.3385	177910
spear120	1024	3072	33	0.0107	4.7446	24181440
spear121	1024	3072	26	0.0085	0.8330	24181188
spear122	1024	3072	27	0.0088	0.9370	24181193
spear123	1024	4096	15	0.0037	1.4816	648028
spear124	1024	4096	23	0.0056	7.4313	648271
spear125	1024	4096	11	0.0027	0.4447	433200
spear126	1024	4096	25	0.0061	1.5292	434274
spear127	1024	4096	11	0.0027	0.4986	1559269
spear128	1024	4096	11	0.0027	0.2149	1559258
spear129	1024	4096	8	0.0020	1.3032	8709779
spear13	512	1024	26	0.0254	0.4850	2060523
spear130	1024	4096	11	0.0027	0.7262	8709779
spear131	1024	4096	9	0.0022	0.0185	10655874
spear132	1024	4096	29	0.0071	1.4022	10656072
spear133	1024	4096	26	0.0063	1.7313	5416001
spear134	1024	4096	27	0.0066	0.0059	5416004
spear135	1024	4096	31	0.0076	6.7643	1356534
spear136	1024	4096	31	0.0076	0.9842	1356527
spear137	1024	4096	30	0.0073	17.0804	21385030
spear138	1024	4096	30	0.0073	0.8304	21385037
spear139	1024	4096	26	0.0063	2.2113	1357206
spear14	512	1024	26	0.0254	19.7935	2060531

spear140	1024	4096	26	0.0063	13.5399	1357182
spear141	1024	4096	25	0.0061	0.3975	2191588
spear142	1024	4096	26	0.0063	0.6167	2191603
spear143	1024	4096	30	0.0073	0.4676	32238665
spear144	1024	4096	31	0.0076	1.1337	32238677
spear145	1024	4096	26	0.0063	0.8963	32238878
spear146	1024	4096	26	0.0063	0.9506	32238867
spear147	1024	8192	9	0.0011	1.2450	11438708
spear148	1024	8192	20	0.0024	1.4967	11438831
spear149	2048	4096	8	0.0020	7.5037	283643
spear15	512	1024	28	0.0273	5.6510	2522199
spear150	2048	4096	9	0.0022	0.1078	283732
spear151	2048	4096	58	0.0142	0.7877	88849
spear152	2048	4096	71	0.0173	0.9430	89805
spear153	2048	4096	8	0.0020	4.8569	27546
spear154	2048	4096	224	0.0547	0.9753	38127
spear155	2048	4096	9	0.0022	0.6817	23977
spear156	2048	4096	264	0.0645	0.9493	26903
spear157	2048	6144	11	0.0018	9.3580	179134
spear158	2048	6144	110	0.0179	1.1108	184698
spear159	2048	6144	9	0.0015	0.9363	99698
spear16	512	1024	29	0.0283	4.9360	2522227
spear160	2048	6144	208	0.0339	0.9774	102025
spear161	2048	6144	12	0.0020	0.2006	221302
spear162	2048	6144	11	0.0018	0.7236	221150
spear163	2048	6144	11	0.0018	149.7670	33562
spear164	2048	6144	440	0.0716	1.0413	42344
spear165	2048	8192	11	0.0013	192.2050	188278
spear166	2048	8192	142	0.0173	1.0333	195916
spear167	2048	8192	14	0.0017	0.1597	309098
spear168	2048	8192	284	0.0347	1.0005	319014
spear169	2048	8192	10	0.0012	25.2401	306485
spear17	512	1024	18	0.0176	9.4087	177890
spear170	2048	8192	139	0.0170	1.1669	308098
spear171	2048	8192	10	0.0012	0.4744	109021
spear172	2048	8192	216	0.0264	1.0408	112710
spear173	2048	12288	11	0.0009	36.5289	400885
spear174	2048	12288	148	0.0120	1.0318	404845
spear175	8192	16384	7	0.0004	16.0282	1950941
spear176	8192	16384	11	0.0007	4.9828	1951580
spear177	8192	16384	113	0.0069	1.2017	1172551
spear178	8192	16384	121	0.0074	1.0379	1173269
spear18	512	1024	19	0.0186	0.8017	177889

spear180	8192	16384	597	0.0364	1.1009	145249
spear181	8192	16384	9	0.0005	0.0038	101171
spear182	8192	16384	665	0.0406	1.0267	113726
spear183	8192	24576	14	0.0006	2.1062	1536934
spear184	8192	24576	10	0.0004	0.5184	1536760
spear185	8192	24576	11	0.0004	4.0093	1221275
spear186	8192	24576	183	0.0074	1.0253	1231989
spear187	8192	24576	11	0.0004	4.8949	896265
spear188	8192	24576	14	0.0006	1.0558	896243
spear189	8192	24576	11	0.0004	5.5748	137871
spear19	512	1024	18	0.0176	0.8077	289849
spear190	8192	24576	1180	0.0480	0.9846	160809
spear191	8192	32768	13	0.0004	0.1632	1573159
spear192	8192	32768	323	0.0099	0.9006	1590835
spear193	8192	32768	14	0.0004	9.0781	1230481
spear194	8192	32768	1015	0.0310	1.0043	1263418
spear195	8192	32768	9	0.0003	0.0316	2051773
spear196	8192	32768	47	0.0014	0.4788	2052508
spear197	8192	32768	13	0.0004	0.4797	1257638
spear198	8192	32768	424	0.0129	1.0873	1266816
spear199	8192	49152	11	0.0002	0.0983	2426522
spear2	512	1024	9	0.0088	54.2664	207343
spear20	512	1024	18	0.0176	1.4005	289843
spear200	8192	49152	313	0.0064	0.9808	2438013
spear201	512	1024	51	0.0498	0.6657	208839
spear202	512	1024	50	0.0488	0.9805	52565
spear203	512	1024	51	0.0498	0.3476	548538
spear204	512	1024	51	0.0498	0.9910	47652
spear205	512	1024	51	0.0498	0.8083	744223
spear206	512	1024	51	0.0498	1.1588	178128
spear207	512	1024	51	0.0498	1.1229	2060769
spear208	512	1024	51	0.0498	0.4792	2522422
spear209	512	1024	51	0.0498	1.4167	178217
spear21	512	1024	27	0.0264	2.2016	4033091
spear210	512	1024	51	0.0498	1.9500	290165
spear211	512	1024	51	0.0498	2.6711	4033319
spear212	512	1024	51	0.0498	0.5106	4033531
spear213	512	1536	51	0.0332	0.4176	2560689
spear214	512	1536	51	0.0332	1.1284	311791
spear215	512	1536	51	0.0332	0.1350	149759
spear216	512	1536	51	0.0332	1.2572	559343
spear217	512	1536	51	0.0332	1.4920	2562232
spear218	512	1536	51	0.0332	1.1442	1114206

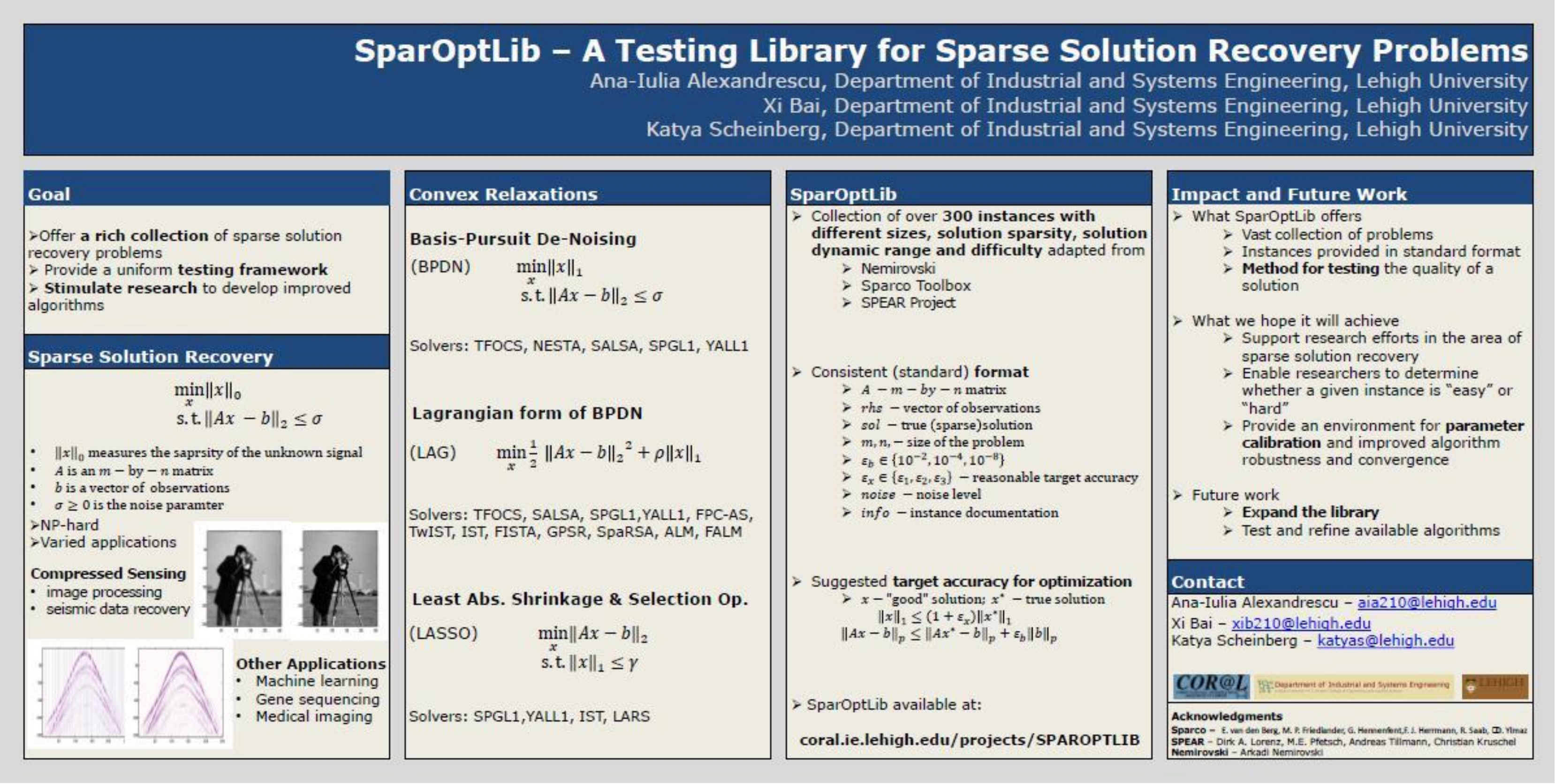
spear219	512	1536	51	0.0332	0.4727	264834
spear22	512	1024	27	0.0264	0.8178	4033097
spear220	512	1536	51	0.0332	0.7437	3164664
spear221	512	1536	51	0.0332	0.8440	265109
spear222	512	1536	51	0.0332	0.8660	433082
spear223	512	1536	51	0.0332	2.9712	6047595
spear224	512	1536	51	0.0332	0.7599	6047747
spear225	512	2048	51	0.0249	1.0640	187733
spear226	512	2048	51	0.0249	0.7605	157107
spear227	512	2048	51	0.0249	1.1278	414075
spear228	512	2048	51	0.0249	1.1132	2186201
spear229	512	2048	51	0.0249	1.4674	2573036
spear23	512	1024	18	0.0176	1.3513	4033213
spear230	512	2048	51	0.0249	0.9409	1483120
spear231	512	2048	51	0.0249	0.9105	351343
spear232	512	2048	51	0.0249	0.9697	5243166
spear233	512	2048	51	0.0249	1.1211	351578
spear234	512	2048	51	0.0249	0.3291	575270
spear235	512	2048	51	0.0249	0.4196	8061805
spear236	512	2048	51	0.0249	1.0678	8061979
spear237	512	4096	51	0.0125	0.6801	2817355
spear238	1024	2048	102	0.0498	1.0080	784777
spear239	1024	2048	84	0.0410	1.7338	205289
spear24	512	1024	18	0.0176	0.4507	4033213
spear240	1024	2048	102	0.0498	0.7628	2563403
spear241	1024	2048	102	0.0498	0.7501	167406
spear242	1024	2048	102	0.0498	0.9300	2713021
spear243	1024	2048	102	0.0498	0.8910	683300
spear244	1024	2048	102	0.0498	0.9573	8415015
spear245	1024	2048	102	0.0498	1.1386	10455141
spear246	1024	2048	102	0.0498	0.8476	683649
spear247	1024	2048	102	0.0498	0.9758	1100891
spear248	1024	2048	102	0.0498	0.7333	16124457
spear249	1024	2048	102	0.0498	0.6899	16124426
spear25	512	1536	14	0.0091	0.2562	2560316
spear250	1024	3072	102	0.0332	1.9621	10557940
spear251	1024	3072	102	0.0332	0.8580	1172889
spear252	1024	3072	102	0.0332	0.7242	555218
spear253	1024	3072	102	0.0332	1.1362	2602227
spear254	1024	3072	102	0.0332	1.4274	10618195
spear255	1024	3072	102	0.0332	1.1589	4065220
spear256	1024	3072	102	0.0332	1.1529	1020295
spear257	1024	3072	102	0.0332	1.0109	12955546

spear258	1024	3072	102	0.0332	0.9041	1021443
spear259	1024	3072	102	0.0332	0.8995	1645188
spear26	512	1536	16	0.0104	1.2646	2560345
spear260	1024	3072	102	0.0332	1.2678	24182107
spear261	1024	3072	102	0.0332	0.8998	24181910
spear262	1024	4096	102	0.0249	1.5494	655554
spear263	1024	4096	97	0.0237	1.0021	441516
spear264	1024	4096	102	0.0249	1.9002	1561264
spear265	1024	4096	102	0.0249	0.8890	8710738
spear266	1024	4096	100	0.0244	2.0608	10656820
spear267	1024	4096	102	0.0249	1.1091	5416737
spear268	1024	4096	102	0.0249	0.6626	1357235
spear269	1024	4096	102	0.0249	0.9564	21385743
spear27	512	1536	9	0.0059	61.7803	310233
spear270	1024	4096	102	0.0249	1.1172	1357945
spear271	1024	4096	102	0.0249	1.1970	2192346
spear272	1024	4096	102	0.0249	0.7919	32239361
spear273	1024	4096	102	0.0249	0.8897	32239612
spear274	1024	8192	101	0.0123	0.9317	11439704
spear28	512	1536	9	0.0059	10.6386	310277
spear29	512	1536	9	0.0059	0.5508	145972
spear3	512	1024	6	0.0059	0.0001	51324
spear30	512	1536	10	0.0065	2.8056	146941
spear31	512	1536	9	0.0059	1.1289	556572
spear32	512	1536	25	0.0163	0.0272	559060
spear33	512	1536	10	0.0065	15.1530	2561797
spear34	512	1536	11	0.0072	8.8127	2561829
spear35	512	1536	17	0.0111	18.1778	1113852
spear36	512	1536	18	0.0117	0.9098	1113876
spear37	512	1536	22	0.0143	0.4134	264548
spear38	512	1536	23	0.0150	2.0016	264550
spear39	512	1536	22	0.0143	1.6941	3164386
spear4	512	1024	106	0.1035	0.5887	52679
spear40	512	1536	23	0.0150	4.3579	3164402
spear41	512	1536	18	0.0117	5.6320	264774
spear42	512	1536	18	0.0117	0.4907	264757
spear43	512	1536	18	0.0117	0.0245	432748
spear44	512	1536	18	0.0117	0.1728	432751
spear45	512	1536	21	0.0137	0.1367	6047301
spear46	512	1536	21	0.0137	2.0709	6047307
spear47	512	1536	17	0.0111	0.6174	6047400
spear48	512	1536	18	0.0117	2.9450	6047427
spear49	512	2048	11	0.0054	8.4620	184154

spear5	512	1024	9	0.0088	61.9793	546166
spear50	512	2048	14	0.0068	2.1178	183868
spear51	512	2048	11	0.0054	0.0017	154079
spear52	512	2048	14	0.0068	0.2895	154231
spear53	512	2048	8	0.0039	0.0406	411874
spear54	512	2048	9	0.0044	0.0115	412467
spear55	512	2048	7	0.0034	5.6093	2182398
spear56	512	2048	7	0.0034	12.3847	2185757
spear57	512	2048	11	0.0054	0.0738	2572640
spear58	512	2048	22	0.0107	1.3926	2572758
spear59	512	2048	18	0.0088	1.6978	1482791
spear6	512	1024	14	0.0137	1.9758	544467
spear60	512	2048	18	0.0088	1.7958	1482781
spear61	512	2048	20	0.0098	2.0007	351010
spear62	512	2048	21	0.0103	2.4694	351048
spear63	512	2048	20	0.0098	1.0683	5242878
spear64	512	2048	20	0.0098	0.6231	5242868
spear65	512	2048	17	0.0083	0.9928	351237
spear66	512	2048	17	0.0083	2.0323	351238
spear67	512	2048	17	0.0083	2.7142	574930
spear68	512	2048	17	0.0083	0.1672	574934
spear69	512	2048	20	0.0098	25.8108	8061509
spear7	512	1024	34	0.0332	0.4546	47469
spear70	512	2048	21	0.0103	0.0282	8061518
spear71	512	2048	17	0.0083	0.6278	8061648
spear72	512	2048	17	0.0083	2.5331	8061638
spear73	512	4096	9	0.0022	0.1404	2813167
spear74	512	4096	10	0.0024	19.8497	2816775
spear75	1024	2048	12	0.0059	0.5362	783182
spear76	1024	2048	12	0.0059	0.0126	783214
spear77	1024	2048	5	0.0024	3.7962	202664
spear78	1024	2048	169	0.0825	0.9796	205823
spear79	1024	2048	11	0.0054	0.0056	2562480
spear8	512	1024	34	0.0332	0.7321	47484
spear80	1024	2048	17	0.0083	0.2453	2557757
spear81	1024	2048	50	0.0244	2.2424	166927
spear82	1024	2048	50	0.0244	2.2833	166926
spear83	1024	2048	27	0.0132	14.4704	2712312
spear84	1024	2048	28	0.0137	0.4834	2712306
spear85	1024	2048	39	0.0190	0.6477	682708
spear86	1024	2048	40	0.0195	0.5846	682709
spear87	1024	2048	39	0.0190	1.8983	8414435
spear88	1024	2048	40	0.0195	0.6513	8414443

spear89	1024	2048	43	0.0210	0.8935	10454606
spear9	512	1024	18	0.0176	0.6374	743907
spear90	1024	2048	46	0.0225	0.8747	10454612
spear91	1024	2048	27	0.0132	3.7895	682933
spear92	1024	2048	27	0.0132	0.4532	682929
spear93	1024	2048	27	0.0132	2.7565	1100177
spear94	1024	2048	27	0.0132	3.2594	1100163
spear95	1024	2048	39	0.0190	1.0291	16123851
spear96	1024	2048	40	0.0195	1.4544	16123859
spear97	1024	2048	27	0.0132	1.2995	16123701
spear98	1024	2048	27	0.0132	0.1612	16123701
spear99	1024	3072	18	0.0059	180.5720	10557103

Appendix C: SparOptLib Poster



SparOptLib – A Testing Library for Sparse Solution Recovery Problems

Ana-Iulia Alexandrescu, Department of Industrial and Systems Engineering, Lehigh University
Xi Bai, Department of Industrial and Systems Engineering, Lehigh University
Katya Scheinberg, Department of Industrial and Systems Engineering, Lehigh University

Goal

- Offer **a rich collection** of sparse solution recovery problems
- Provide a uniform **testing framework**
- **Stimulate research** to develop improved algorithms

Sparse Solution Recovery

$$\min_x \|x\|_0 \quad \text{s.t. } \|Ax - b\|_2 \le \sigma$$

- $\|x\|_0$ measures the saprsity of the unknown signal
- A is an m – by – n matrix
- b is a vector of observations
- $\sigma \ge 0$ is the noise paramter

- NP-hard
- Varied applications

Compressed Sensing

- image processing
- seismic data recovery

Other Applications

- Machine learning
- Gene sequencing
- Medical imaging

Convex Relaxations

Basis-Pursuit De-Noising

(BPDN) $$\min_x \|x\|_1 \quad \text{s.t. } \|Ax - b\|_2 \le \sigma$$

Solvers: TFOCS, NESTA, SALSA, SPGL1, YALL1

Lagrangian form of BPDN

(LAG) $$\min_x \frac{1}{2}\|Ax - b\|_2{}^2 + \rho\|x\|_1$$

Solvers: TFOCS, SALSA, SPGL1,YALL1, FPC-AS, TwIST, IST, FISTA, GPSR, SpaRSA, ALM, FALM

Least Abs. Shrinkage & Selection Op.

(LASSO) $$\min_x \|Ax - b\|_2 \quad \text{s.t. } \|x\|_1 \le \gamma$$

Solvers: SPGL1,YALL1, IST, LARS

SparOptLib

- Collection of over **300 instances with different sizes, solution sparsity, solution dynamic range and difficulty** adapted from
  - Nemirovski
  - Sparco Toolbox
  - SPEAR Project
- Consistent (standard) **format**
  - $A - m - by - n$ matrix
  - rhs – vector of observations
  - sol – true (sparse)solution
  - $m, n,$ – size of the problem
  - $\varepsilon_b \in \{10^{-2}, 10^{-4}, 10^{-8}\}$
  - $\varepsilon_x \in \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$ – reasonable target accuracy
  - $noise$ – noise level
  - $info$ – instance documentation
- Suggested **target accuracy for optimization**
  - x – "good" solution; x^* – true solution

$$\|x\|_1 \le (1 + \varepsilon_x)\|x^*\|_1$$
$$\|Ax - b\|_p \le \|Ax^* - b\|_p + \varepsilon_b\|b\|_p$$

- SparOptLib available at:

coral.ie.lehigh.edu/projects/SPAROPTLIB

Impact and Future Work

- What SparOptLib offers
  - Vast collection of problems
  - Instances provided in standard format
  - **Method for testing** the quality of a solution
- What we hope it will achieve
  - Support research efforts in the area of sparse solution recovery
  - Enable researchers to determine whether a given instance is "easy" or "hard"
  - Provide an environment for **parameter calibration** and improved algorithm robustness and convergence
- Future work
  - **Expand the library**
  - Test and refine available algorithms

Contact

Ana-Iulia Alexandrescu – aia210@lehigh.edu
Xi Bai – xib210@lehigh.edu
Katya Scheinberg – katyas@lehigh.edu

Acknowledgments

Sparco – E. van den Berg, M. P. Friedlander, G. Hennenfent,F. J. Herrmann, R. Saab, Ö. Yilmaz
SPEAR – Dirk A. Lorenz, M.E. Pfetsch, Andreas Tillmann, Christian Kruschel
Nemirovski – Arkadi Nemirovski

Figure 1: Poster Selected for the Final Round at INFORMS Annual Meeting, Charlotte, 2011

Vita

Ana-Iulia Alexandrescu is a graduate student in the Industrial and Systems Engineering Department at Lehigh University. Born and raised in Bucharest, Romania, she came to Lehigh as a Bostiber Scholar in 2006, where she enrolled in the Integrated Business and Engineering Honors Program. While an undergraduate, she pursued a major in Information and Systems Engineering and a minor in Applied Mathematics. She was awarded the Information and Systems Engineering Student of the Year title all three years she was in the program. In her junior year, Ana was selected to represent her department in the engineering honors and service society, the Rossin Junior Fellows, where she served as a secretary of the executive board and was twice awarded the recognition of Excellence in Service. The summer after her junior year, she spent ten weeks performing comparative field studies in sustainability, culture, and ethics and human rights in the Mediterranean space. Ana graduated with honors in September 2010 and continued on as a Presidential Scholar, pursuing her MS degree in Industrial and Systems Engineering. As a graduate student, she held the position of Vice President of the INFORMS Student Chapter at Lehigh for one year, and she is now acting as an advisor to the current executive board of the chapter.