ParNes: A New Algorithm for Compressed Sensing Problems

Cinna Wu
Department of Mathematics
UC Berkeley

Joint work with Ming Gu and Lek-Heng Lim

$$\min \|x\|_1 \text{ s.t. } \|Ax - b\|_2 \leq \sigma$$
Contents

• Background
  – Motivation
  – Applications
  – Solvers

• ParNes
  – The Algorithm
  – Convergence
  – Numerical Experiments
Sparse Signal Recovery

- Classical approach: Sample then compress.

\[ f = B x_0 \]

- \( B \in \mathbb{R}^{n \times n} \): compression matrix
- \( f \in \mathbb{R}^n \): sampled signal
- \( x_0 \in \mathbb{R}^n \): sparse compressed signal.

- Compressed Sensing: Sample and compress in one stage.

\[ b = M f = MBx_0 = Ax_0 \]

- \( M \in \mathbb{R}^{m \times n} \): measurement matrix with \( m < n \)
- \( b \in \mathbb{R}^m \): measurements

Can we recover \( x_0 \) given \( A \) and \( b \)?
Applications of Compressed Sensing

Compressed sensing may be useful when...

- signals are sparse in a known basis.
- measurements are expensive but computations are cheap.

- Magnetic Resonance Imaging (MRI):
  - Lengthy procedure! Needs a large number of measurements of the patient.
  - Compressed sensing can reduce the number of measurements.
  - This could reduce the procedure time or produce better images in the same amount of time.
Rice Single Pixel Camera\textsuperscript{[1]}

\[ b = Mf = MBx_0 = Ax_0 \]

- \( b \in \mathbb{R}^m \) : measurements
- \( M \in \mathbb{R}^{m \times n} \) : measurement matrix with \( m < n \), rows determined by the digital micromirror device (DMD)
- \( f \in \mathbb{R}^n \) : the image we wish to recover
- \( x_0 \in \mathbb{R}^n \) : sparse representation of \( f \) under the basis given by \( B \).

\textsuperscript{[1]} Image courtesy of Rice University.
Recovering the Sparse Signal

- We can try to recover the sparse signal with

\[
\min \|x\|_0 \quad \text{s.t.} \quad Ax = b
\]

- \(\|x\|_0\) : number of nonzero coefficients in \(x\).

- Combinatorial and NP-hard!

- Relax to the **Basis Pursuit (BP)** problem:

\[
\min \|x\|_1 \quad \text{s.t.} \quad Ax = b
\]

- This can recovers the sparse signal!

  * **Mutual coherence** of \(A\): Given \(A\), works if the signal is sufficiently sparse. (Donoho, Elad, Huo, etc)

  * Given the sparsity of the signal, depends on the **restricted isometry constants** of \(A\). (Candès, Romberg, Tao)
$\ell_1$-relaxations for Noisy Measurements

Recover the sparse vector $x$ when $Ax \approx b$.

- Basis pursuit denoise (BP$_\sigma$)
  \[
  \min \|x\|_1 \quad \text{s.t.} \quad \|Ax - b\|_2 \leq \sigma
  \]

- Penalized least squares (QP$_\lambda$)
  \[
  \min \|Ax - b\|_2^2 + \lambda \|x\|_1
  \]

- Lasso Problem (LS$_\tau$)
  \[
  \min \|Ax - b\|_2 \quad \text{s.t.} \quad \|x\|_1 \leq \tau
  \]

Solutions coincide for appropriate choices of $\sigma, \lambda, \tau$.

- Many solvers use this relationship.
Solvers

- **PDCO** - Primal-Dual IP method for Convex Objectives (Saunders, Kim):
  - Solves Basis Pursuit
    \[
    \min \|x\|_1 \quad \text{s.t.} \quad Ax = b
    \]
    by solving an equivalent Linear Program.

- **HOMOTOPY** (Osborne, Presnell, Turlach):
  - Solves a sequence of QP\(\lambda\) problems to solve BP\(\sigma\).
    \[
    \min \|x\|_1 \quad \text{s.t.} \quad \|Ax - b\|_2 \leq \sigma
    \]

- **FPC** - Fixed Point Continuation Method (Hale, Yin, Zhang):
  - Uses a version of fixed point iteration to solve QP\(\lambda\).
    \[
    \min \|Ax - b\|_2^2 + \lambda \|x\|_1
    \]
Solvers

• **SPGL1** - spectral gradient-projection method (Berg, Friedlander):

  – Solves a sequence of $LS_\tau$ problems

  \[
  \begin{align*}
  \min & \|Ax - b\|_2 \\
  \text{s.t.} & \|x\|_1 \leq \tau
  \end{align*}
  \]

  to solve $BP_\sigma$

  \[
  \begin{align*}
  \min & \|x\|_1 \\
  \text{s.t.} & \|Ax - b\|_2 \leq \sigma
  \end{align*}
  \]

• **NESTA** (Becker, Bobin, Candès):

  – Uses a method to minimize non-smooth functions proposed by Yu. Nesterov to solve $BP_\sigma$.

Our algorithm, ParNes, combines the ideas used in NESTA and SPGL1.
Comparison of Solvers

- Comparison of HOMOTOPY, PDCO, SPGL1 [2].

- Two 3GHz CPU’s, 4Gb RAM. Problems from the SPARCO toolbox.
  - ⋆ : solver failed to converge in the allowed CPU time (1 hour)
  - \( nz(x) \) : number of ”nonzero” entries of \( x \) above some tolerance
  - \( r \) : residual.

<table>
<thead>
<tr>
<th>Problem Data</th>
<th>PDCO</th>
<th>HOMOTOPY</th>
<th>SPGL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>size A</td>
<td>( | r |_2 ) ( | x |_1 ) ( nz(x) )</td>
<td>( | r |_2 ) ( | x |_1 ) ( nz(x) )</td>
</tr>
<tr>
<td>blocksig</td>
<td>1024×1024</td>
<td>3.3e-4 4.5e+2 703</td>
<td>1.0e-4 4.5e+2 246</td>
</tr>
<tr>
<td>blurrycam</td>
<td>65536×65536</td>
<td>⋆ ⋆ ⋆</td>
<td>⋆ ⋆ ⋆</td>
</tr>
<tr>
<td>blurspike</td>
<td>16384×16384</td>
<td>9.1e-3 3.4e+2 5963</td>
<td>⋆ ⋆ ⋆</td>
</tr>
<tr>
<td>cosspike</td>
<td>1024×2048</td>
<td>1.6e-4 2.2e+2 2471</td>
<td>1.0e-4 2.2e+2 500</td>
</tr>
<tr>
<td>sgnspike</td>
<td>600×2560</td>
<td>9.3e-6 2.0e+1 131</td>
<td>1.0e-04 2.0e+1 80</td>
</tr>
<tr>
<td>seismic</td>
<td>41472×480617</td>
<td>⋆ ⋆ ⋆</td>
<td>⋆ ⋆ ⋆</td>
</tr>
</tbody>
</table>

An Outline of ParNes

Combines the best features of NESTA and SPGL1 to solve BP$_{\sigma}$

\[
\min \|x\|_1 \quad \text{s.t.} \quad \|Ax - b\|_2 \leq \sigma
\]

- **SPGL1**
  - Like SPGL1, ParNes solves BP$_{\sigma}$ by solving a sequence of LS$_{\tau}$ problems.

\[
\min \|Ax - b\|_2 \quad \text{s.t.} \quad \|x\|_1 \leq \tau
\]
  - LS$_{\tau}$ and BP$_{\sigma}$ are related by the Pareto Curve.
  - In SPGL1, LS$_{\tau}$ is solved with a spectral projected gradient method.

- **NESTA**
  - Uses a method by Y. Nesterov to minimize non-smooth functions.
  - ParNes solves the LS$_{\tau}$ problems with a similar method for minimizing smooth functions.
The Pareto Curve

- Convex, continuously differentiable, and strictly decreasing.

- Graph of \((\|x_\tau\|_1, \|b - Ax_\tau\|_2)\) where \(x_\tau\) solves LS\(_\tau\).

- Also the graph of \((\|x_\sigma\|_1, \|b - Ax_\sigma\|_2)\) where \(x_\sigma\) solves BP\(_\sigma\).

- Since \(\|x_\tau\|_1 = \tau\) and \(\|b - Ax_\sigma\|_2 = \sigma\), the Pareto curve is the graph of a function \(\phi(\tau) = \sigma\).
**Root Finding**

- \( \text{BP}_\sigma: \min \|x\|_1 \text{ s.t. } \|Ax - b\|_2 \leq \sigma \)
- \( \text{LS}_\tau: \min \|Ax - b\|_2 \text{ s.t. } \|x\|_1 \leq \tau \)

- \( \text{BP}_\sigma \) can be solved by finding a root \( \tau_\sigma \) to \( \phi(\tau) = \sigma \).
- Newton’s method can be applied to \( \phi(\tau) = \sigma \) to get \( \tau_k \rightarrow \tau_\sigma \):
  \[
  \tau_{k+1} = \tau_k + \left( \sigma - \phi(\tau_k) \right)/\phi'(\tau_k)
  \]

- Since each iteration involves solving a potentially large \( \text{LS}_{\tau_k} \) problem, an inexact Newton method is used.
Solving $\text{LS}_{\tau_k} : \min \| Ax - b \|_2 \quad \text{s.t.} \quad \| x \|_1 \leq \tau$

- Each iteration of SPGL1 involves computing:

  \[
  \tau_{k+1} = \tau_k + \frac{(\sigma - \phi(\tau_k))}{\phi'(\tau_k)}
  \]

- Let $x_{\tau_k}$ approximately solve $\text{LS}_{\tau_k}$ and $r_{\tau_k} = Ax_{\tau_k} - b$, then

  \[
  \phi(\tau_k) = \| r_{\tau_k} \|_2 \quad \text{and} \quad \phi'(\tau_k) = \frac{\| A^\top r_{\tau_k} \|_\infty}{\| r_{\tau_k} \|_2}
  \]

- Note: $\phi(\tau_k)$ and $\phi'(\tau_k)$ are the approximate solution and dual solution to $\text{LS}_{\tau_k}$, respectively.

- SPGL1 uses a SPG (Spectral Projected Gradient) method to solve $\text{LS}_{\tau_k}$.

- ParNes uses the same framework except the SPG method is replaced with a proximal gradient method.
Nesterov’s Proximal Gradient Algorithm for Smooth Minimization

- Solves:

\[
\min f(x) \text{ s.t. } x \in Q
\]

where \( Q \subseteq \mathbb{R}^n \) is closed and convex and \( f : Q \rightarrow \mathbb{R} \) is smooth, convex, and Lipschitz differentiable with Lipschitz constant \( L \).

- Computes the sequences:

\[
\begin{align*}
y_k &= \arg\min_{y \in Q} \nabla f(x_k)^\top (y - x_k) + \frac{L}{2} \|y - x_k\|_2^2, \\
z_k &= \arg\min_{z \in Q} \sum_{i=0}^{k} \frac{i+1}{2} \nabla f(x_i)^\top (z - x_i) + \frac{L}{2} \|z - c\|_2^2, \\
x_k &= \frac{2}{k+3} z_k + \frac{k+1}{k+3} y_k. \ (f(x_k) \text{ converges to the solution})
\end{align*}
\]

- \( c \) is a constant called the prox-center.
Nesterov’s Algorithm for Non-Smooth Minimization

• Solves:

\[
\min f(x) \text{ s.t. } x \in Q
\]

where \( Q \subseteq \mathbb{R}^n \) is closed and convex and \( f : Q \rightarrow \mathbb{R} \) is convex but not necessarily differentiable.

• Assume there is a convex set \( Q_d \subseteq \mathbb{R}^p \) and \( W \in \mathbb{R}^{p \times n} \) where \( f \) can be written as

\[
f(x) = \max_{u \in Q_d} \langle u, Wx \rangle.
\]

• Replace \( f(x) \) with the smooth approximation

\[
f_\mu(x) = \max_{u \in Q_d} \langle u, Wx \rangle - \frac{\mu}{2} \|u\|_2^2
\]

• Apply Nesterov’s algorithm for smooth minimization to \( f_\mu(x) \).
Convergence of Nesterov’s Algorithms

• Convergence of Smooth Version:
  – Let $x^*$ be the optimal solution to:

  \[
  \begin{align*}
  \min f(x) \text{ s.t. } x \in Q
  \end{align*}
  \]

  – The iterates $y_k$ satisfy:

  \[
  f(y_k) - f(x^*) \leq \frac{2L}{(k+1)(k+2)} \|x^* - c\|_2^2 = O\left(\frac{L}{k^2}\right)
  \]

  implying $O\left(\sqrt{\frac{L}{\epsilon}}\right)$ iterations bring $f(y_k)$ within $\epsilon$ of the optimal value.

• Convergence of Non-Smooth Version:
  – $\nabla f_\mu$ has Lipschitz constant $L_\mu = 1/\mu$.

  – Assuming $\mu$ is chosen to be proportional to $\epsilon$, $O\left(\frac{1}{\epsilon}\right)$ iterations bring $f(y_k)$ within $\epsilon$ of the optimal value.
Nesterov-LASSO

- NESTA uses Nesterov’s algorithm for non-smooth minimization to solve $BP_\sigma$.

$$\min \|x\|_1 \text{ s.t. } \|Ax - b\|_2 \leq \sigma$$

- ParNes uses the smooth version to solve $LS_{\tau_k}$ in each iteration

$$\min \|Ax - b\|_2 \text{ s.t. } \|x\|_1 \leq \tau$$

- The sequences in Nesterov’s smooth algorithm simplify to:

$$y_k = \text{proj}_1(x_k - \nabla f(x_k)/L, \tau),$$
$$z_k = \text{proj}_1\left(c - \frac{1}{L} \sum_{i=0}^{k} \frac{i+1}{2} \nabla f(x_i), \tau\right),$$
$$x_k = \frac{2}{k+3}z_k + \frac{k+1}{k+3}y_k. \quad (f(x_k) \text{ converges to the solution})$$

where $\text{proj}_1(s, \tau) := \text{argmin} \|s - x\|_2 \text{ s.t. } \|x\|_1 \leq \tau$. 
Each iteration of Nesterov-LASSO involves two solves of

$$\text{proj}_1(s, \tau) := \arg\min \|s - x\|_2 \ \text{s.t.} \ \|x\|_1 \leq \tau$$

Assume the coefficients of $s$ are positive and ordered from largest to smallest.

The solution $x^*$ is given by

$$x^*_i = \max\{0, s_i - \eta\} \text{ with } \eta = \frac{\tau - (s_1 + \ldots + s_k)}{k}$$

where $k$ is the largest index such that $\eta \leq s_k$. (Duchi, Shalev-Schwartz, Berg, etc.)

Each solve costs $O(n \log n)$ in the worst case and much less in practice.
Convergence of Nesterov-LASSO

- Recall minimizing \( f \) with Nesterov’s method gives \( (x^* = \arg\min_{x \in Q} f(x)) \)

\[
f(y_k) - f(x^*) \leq \frac{2L}{(k+1)(k+2)} \|x^* - c\|_2^2 = O\left(\frac{L}{k^2}\right)
\]

- Assume \( x^* \) is unique. Since \( x_k \to y_k \), updating \( c \) with \( x_k \) should speed up the convergence.

- In ParNes, Nesterov-LASSO is restarted every \( q \) iterations with \( c = x_{k\text{current}} \).

- \( q \) can be chosen in an optimal manner if
  1. the solution \( x^* \) is \( s \)-sparse,
  2. the iterates \( x_k \) are \( s \)-sparse,
  3. \( A \) satisfies the restricted isometry property of order \( 2s \): \( \exists \delta_{2s} \in (0,1) \) s.t

\[
(1 - \delta_{2s}) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_{2s}) \|x\|_2^2
\]

whenever \( x \) is \( 2s \)-sparse.
Convergence Results of Nesterov-LASSO

• Let $x_{p,q}$ represent the $q$-th iterate after the $p$-th prox-center change.

• With the assumptions on the previous slide, we have the following results:

  – Let $e$ be the base of the natural logarithm and

    \[
    q_{opt} = e \sqrt{\frac{L}{\delta_{2s}}} \text{ and } p_{tot} = -\log \varepsilon
    \]

    Then the total number of iterations, $p_{tot} \times q_{opt}$, to get $\|x_{p,q} - x^*\|_2 \leq \varepsilon$ is minimized with these choices of $q_{opt}$ and $p_{tot}$.

  – For each $p$,

    \[
    \|x_{p,q_{opt}} - x^*\|_2 \leq 1/e \|x_{p,1} - x^*\|_2
    \]

  – Nesterov-LASSO is linearly convergent under the previous assumptions!
ParNes: Experiment Details

• To gain a good comparison, we repeat some of the experiments done in the NESTA paper (Becker, Bobin, Candès) using their code.

• Tests some of the most competitive algorithms using hard, realistic problems.

• The next two experiments recover an $s$-sparse signal with $n = 262144$, $m = n/8$, $s = m/5$, and noise level $\sigma = 0.1$.

  – Tests dynamic range values (ratio of the largest and smallest non-zero coefficients of the unknown signal) of $d = 20, 40, 60, 80, 100$ dB.

  – $A$ is a randomly subsampled discrete cosine transform.

  – Let $x_{\text{NES}}$ be NESTA’s solution when the relative variation of the objective function is less than $10^{-7}$. The stopping rule is

    $$\|x_k\|_1 \leq \|x_{\text{NES}}\|_1 \quad \text{and} \quad \|b - Ax_k\|_2 \leq 1.05 \|b - Ax_{\text{NES}}\|_2.$$
Numerical Experiments: Speed

- Table gives the number of function calls.
- DNC if calls to $A$ or $A^\top$ exceeds 20,000.

<table>
<thead>
<tr>
<th>Method</th>
<th>20 dB</th>
<th>40 dB</th>
<th>60 dB</th>
<th>80 dB</th>
<th>100 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARNES</td>
<td>122</td>
<td>172</td>
<td>214</td>
<td>470</td>
<td>632</td>
</tr>
<tr>
<td>NESTA</td>
<td>383</td>
<td>809</td>
<td>1639</td>
<td>4341</td>
<td>15227</td>
</tr>
<tr>
<td>NESTA + CT</td>
<td>483</td>
<td>513</td>
<td>583</td>
<td>685</td>
<td>787</td>
</tr>
<tr>
<td>GPSR</td>
<td>64</td>
<td>622</td>
<td>5030</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>GPSR + CT</td>
<td>271</td>
<td>219</td>
<td>357</td>
<td>1219</td>
<td>11737</td>
</tr>
<tr>
<td>SPARSA</td>
<td>323</td>
<td>387</td>
<td>465</td>
<td>541</td>
<td>693</td>
</tr>
<tr>
<td>SPGL1</td>
<td>58</td>
<td>102</td>
<td>191</td>
<td>374</td>
<td>504</td>
</tr>
<tr>
<td>FISTA</td>
<td>69</td>
<td>267</td>
<td>1020</td>
<td>3465</td>
<td>12462</td>
</tr>
<tr>
<td>FPC-AS</td>
<td>209</td>
<td>231</td>
<td>299</td>
<td>371</td>
<td>287</td>
</tr>
<tr>
<td>FPC-AS (CG)</td>
<td>253</td>
<td>289</td>
<td>375</td>
<td>481</td>
<td>361</td>
</tr>
<tr>
<td>FPC</td>
<td>474</td>
<td>386</td>
<td>478</td>
<td>1068</td>
<td>9614</td>
</tr>
<tr>
<td>FPC-BB</td>
<td>164</td>
<td>168</td>
<td>206</td>
<td>278</td>
<td>1082</td>
</tr>
<tr>
<td>BREGMAN-BB</td>
<td>211</td>
<td>223</td>
<td>309</td>
<td>455</td>
<td>1408</td>
</tr>
</tbody>
</table>
Numerical Experiments: Accuracy

- DNC if calls to $A$ or $A^\top (N_A)$ exceeds 20,000.
- Dynamic range is $d = 100$ dB.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$N_A$</th>
<th>$|x|_1$</th>
<th>$\frac{|x-x^<em>|_1}{|x^</em>|_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARNES</td>
<td>632</td>
<td>942197.606</td>
<td>0.000693</td>
</tr>
<tr>
<td>NESTA</td>
<td>15227</td>
<td>942402.960</td>
<td>0.004124</td>
</tr>
<tr>
<td>NESTA + CT</td>
<td>787</td>
<td>942211.581</td>
<td>0.000812</td>
</tr>
<tr>
<td>GPSR</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>GPSR + CT</td>
<td>11737</td>
<td>942211.377</td>
<td>0.001420</td>
</tr>
<tr>
<td>SPARSA</td>
<td>693</td>
<td>942197.785</td>
<td>0.000783</td>
</tr>
<tr>
<td>SPGL1</td>
<td>504</td>
<td>942211.520</td>
<td>0.001326</td>
</tr>
<tr>
<td>FISTA</td>
<td>12462</td>
<td>942211.540</td>
<td>0.000363</td>
</tr>
<tr>
<td>FPC-AS</td>
<td>287</td>
<td>942210.925</td>
<td>0.000672</td>
</tr>
<tr>
<td>FPC-AS (CG)</td>
<td>361</td>
<td>942210.512</td>
<td>0.000671</td>
</tr>
<tr>
<td>FPC</td>
<td>9614</td>
<td>942211.540</td>
<td>0.001422</td>
</tr>
<tr>
<td>FPC-BB</td>
<td>1082</td>
<td>942209.854</td>
<td>0.001378</td>
</tr>
<tr>
<td>BREGMAN-BB</td>
<td>1408</td>
<td>942286.656</td>
<td>0.000891</td>
</tr>
</tbody>
</table>
Numerical Experiments: Speed

An approximately sparse signal (obtained from permuting the Haar wavelet coefficients of a $512 \times 512$ image) is recovered with the same stopping rule as before.

- The measurement vector $b$ consists of $m = n/8 = 512^2/8 = 32,768$ random discrete cosine measurements, and the noise level is set to 0.1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARNES</td>
<td>838</td>
<td>810</td>
<td>1038</td>
<td>1098</td>
<td>654</td>
</tr>
<tr>
<td>NESTA</td>
<td>8817</td>
<td>10867</td>
<td>9887</td>
<td>9093</td>
<td>11211</td>
</tr>
<tr>
<td>NESTA + CT</td>
<td>3807</td>
<td>3045</td>
<td>3047</td>
<td>3225</td>
<td>2735</td>
</tr>
<tr>
<td>GPSR</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>GPSR + CT</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>SPARSA</td>
<td>2143</td>
<td>2353</td>
<td>1977</td>
<td>1613</td>
<td>DNC</td>
</tr>
<tr>
<td>SPGL1</td>
<td>916</td>
<td>892</td>
<td>1115</td>
<td>1437</td>
<td>938</td>
</tr>
<tr>
<td>FISTA</td>
<td>3375</td>
<td>2940</td>
<td>2748</td>
<td>2538</td>
<td>3855</td>
</tr>
<tr>
<td>FPC-AS</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>FPC-AS (CG)</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>FPC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
<td>DNC</td>
</tr>
<tr>
<td>FPC-BB</td>
<td>5614</td>
<td>7906</td>
<td>5986</td>
<td>4652</td>
<td>6906</td>
</tr>
<tr>
<td>BREGMAN-BB</td>
<td>3288</td>
<td>1281</td>
<td>1507</td>
<td>2892</td>
<td>3104</td>
</tr>
</tbody>
</table>
Software Download

- Resources used in paper and talk
  - NESTA - http://www.acm.caltech.edu/ nesta/
  - SPGL1 - http://www.cs.ubc.ca/labs/scl/index.php/Main/Spgl1
  - SparseLab - http://sparselab.stanford.edu/
  - FPC-AS - http://www.caam.rice.edu/ optimization/L1/FPC_AS/
  - FPC - http://www.caam.rice.edu/ optimization/L1/fpc/
  - GSPR - http://www.lx.it.pt/ mtf/GPSR/
  - SpaRSA - http://www.lx.it.pt/ mtf/SpaRSA/
- Many other resources available at - http://www-dsp.rice.edu/cs