

# Heuristic Optimality Check and Computational Solver Comparison for Basis Pursuit

**Andreas M. Tillmann**

Research Group Optimization, TU Darmstadt, Germany



joint work with

Dirk A. Lorenz ([TU Braunschweig](#)) and Marc E. Pfetsch ([TU Darmstadt](#))

ISMP 2012

Berlin, Germany

Motivation

Infeasible-Point Subgradient Algorithm  
ISAL1

Comparison of  $\ell_1$ -Solvers  
Testset Construction and Computational Results  
Improvements with Heuristic Optimality Check

Possible Future Research

## Motivation

Infeasible-Point Subgradient Algorithm

ISAL1

Comparison of  $\ell_1$ -Solvers

Testset Construction and Computational Results

Improvements with Heuristic Optimality Check

Possible Future Research

- ▶ Seek sparsest solution to underdetermined linear system:

$$\min \|\mathbf{x}\|_0 \quad \text{s. t.} \quad \mathbf{Ax} = \mathbf{b} \quad (\mathbf{A} \in \mathbb{R}^{m \times n}, m < n)$$

- ▶ Finding minimum-support solution is  $\mathcal{NP}$ -hard.  
Convex “relaxation”:  $\ell_1$ -minimization / Basis Pursuit:

$$\min \|\mathbf{x}\|_1 \quad \text{s. t.} \quad \mathbf{Ax} = \mathbf{b} \tag{L1}$$

- ▶ Several conditions (RIP, Nullspace Property, etc) ensure “ $\ell_0$ - $\ell_1$ -equivalence”

# Solving the Basis Pursuit Problem



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ (L1) can be recast as a linear program
- ▶ Broad variety of specialized algorithms for (L1)
  - ▶ direct or primal-dual approaches
  - ▶ regularization, penalty methods
  - ▶ further relaxations (e.g.  $\|Ax - b\| \leq \delta$  instead of  $Ax = b$ )
  - ▶ ...
- ▶ Which algorithm is “the best”?



# Solving the Basis Pursuit Problem

- ▶ (L1) can be recast as a linear program
- ▶ Broad variety of specialized algorithms for (L1)
  - ▶ direct or primal-dual approaches
  - ▶ regularization, penalty methods
  - ▶ further relaxations (e.g.  $\|Ax - b\| \leq \delta$  instead of  $Ax = b$ )
  - ▶ ...



- ▶ Which algorithm is “the best”?
- ▶ A classic algorithm from nonsmooth optimization:  
**(projected) subgradient method – competitive?**

Motivation

## Infeasible-Point Subgradient Algorithm ISAL1

Comparison of  $\ell_1$ -Solvers  
Testset Construction and Computational Results  
Improvements with Heuristic Optimality Check

Possible Future Research

Problem:  $\min f(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathcal{F}$  ( $f, \mathcal{F}$  convex)

standard projected subgradient iteration

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathcal{F}} (\mathbf{x}^k - \alpha_k \mathbf{h}^k), \quad \alpha_k > 0, \quad \mathbf{h}^k \in \partial f(\mathbf{x}^k)$$

applicability: only reasonable if projection is “easy”

Problem:  $\min f(\mathbf{x})$  s.t.  $\mathbf{x} \in \mathcal{F}$  ( $f, \mathcal{F}$  convex)

standard projected subgradient iteration

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathcal{F}} (\mathbf{x}^k - \alpha_k \mathbf{h}^k), \quad \alpha_k > 0, \quad \mathbf{h}^k \in \partial f(\mathbf{x}^k)$$

applicability: only reasonable if projection is “easy”

~~~ idea: replace exact projection by approximation

“infeasible” subgradient iteration

$$\mathbf{x}^{k+1} = \mathcal{P}_{\mathcal{F}}^{\varepsilon_k} (\mathbf{x}^k - \alpha_k \mathbf{h}^k), \quad \|\mathcal{P}_{\mathcal{F}}^{\varepsilon_k} - \mathcal{P}_{\mathcal{F}}\|_2 \leq \varepsilon_k$$

# ISA = Infeasible-Point Subgradient Algorithm



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ ... works for arbitrary convex objectives and constraint sets
- ▶ ... incorporates adaptive approximate projections  $\mathcal{P}_{\mathcal{F}}^{\varepsilon}$  such that  $\|\mathcal{P}_{\mathcal{F}}^{\varepsilon}(\mathbf{y}) - \mathcal{P}_{\mathcal{F}}(\mathbf{y})\|_2 \leq \varepsilon$  for every  $\varepsilon \geq 0$
- ▶ ... converges to optimality (under reasonable assumptions) whenever projection inaccuracies ( $\varepsilon_k$ ) sufficiently small,
  - ▶ for stepsizes  $\alpha_k > 0$  with  $\sum_{k=0}^{\infty} \alpha_k = \infty$ ,  $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$
  - ▶ for dynamic stepsizes  $\alpha_k = \lambda_k(f(\mathbf{x}^k) - \varphi)/\|\mathbf{h}^k\|_2^2$  with  $\varphi \leq \varphi^*$
- ▶ ... converges to  $\varphi$  with dynamic stepsizes using  $\varphi \geq \varphi^*$

# ISAL1 = Spezialization of ISA to $\ell_1$ -Minimization

- ▶  $f(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \mathcal{F} = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \}, \quad \text{sign}(\mathbf{x}) \in \partial\|\mathbf{x}\|_1$
- ▶ exact projected subgradient step for (L1):

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathcal{P}_{\mathcal{F}} (\mathbf{x}^k - \alpha_k \mathbf{h}^k) \\ &= (\mathbf{x}^k - \alpha_k \mathbf{h}^k) - \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} (\mathbf{A}(\mathbf{x}^k - \alpha_k \mathbf{h}^k) - \mathbf{b})\end{aligned}$$

- ▶  $f(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \mathcal{F} = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \}, \quad \text{sign}(\mathbf{x}) \in \partial\|\mathbf{x}\|_1$
- ▶ exact projected subgradient step for (L1):

$$\mathbf{y}^k \leftarrow \mathbf{x}^k - \alpha_k \mathbf{h}^k$$

$$\mathbf{z}^k \leftarrow \text{Solution of } \mathbf{A}\mathbf{A}^\top \mathbf{z} = \mathbf{A}\mathbf{y}^k - \mathbf{b}$$

$$\mathbf{x}^{k+1} \leftarrow \mathbf{y}^k - \mathbf{A}^\top \mathbf{z}^k = \mathcal{P}_{\mathcal{F}}(\mathbf{y}^k)$$

$\mathbf{A}\mathbf{A}^\top$  is s.p.d.  $\Rightarrow$  may employ CG to solve equation system

- ▶  $f(\mathbf{x}) = \|\mathbf{x}\|_1, \quad \mathcal{F} = \{ \mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b} \}, \quad \text{sign}(\mathbf{x}) \in \partial\|\mathbf{x}\|_1$
- ▶ inexact projected subgradient step for (L1):

$$\mathbf{y}^k \leftarrow \mathbf{x}^k - \alpha_k \mathbf{h}^k$$

$$\mathbf{z}^k \leftarrow \text{Solution of } \mathbf{A}\mathbf{A}^\top \mathbf{z} \approx \mathbf{A}\mathbf{y}^k - \mathbf{b}$$

$$\mathbf{x}^{k+1} \leftarrow \mathbf{y}^k - \mathbf{A}^\top \mathbf{z}^k = \mathcal{P}_{\mathcal{F}}^{\varepsilon_k}(\mathbf{y}^k)$$

$\mathbf{A}\mathbf{A}^\top$  is s.p.d.  $\Rightarrow$  may employ CG to solve equation system

- ▶ Approximation: Stop after a few CG iterations  
(CG residual norm  $\leq \sigma_{\min}(\mathbf{A}) \cdot \varepsilon_k \Rightarrow \mathcal{P}_{\mathcal{F}}^{\varepsilon_k}$  fits ISA framework)

# Why simple subgradient scheme?



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Drawbacks of standard subgradient algorithms can often be alleviated by **bundle methods**, especially concerning “excessive” parameter tuning

Experiments for (L1) with two bundle method implementations  
(E. Hübner's and ConicBundle):

- ▶ approach 1: choose  $B$  s.t.  $A_B$  regular, then with  $d := A_B^{-1}b$ ,  $D := A_B^{-1}A_{[n] \setminus B}$ ,

$$(L1) \Leftrightarrow \min \|z\|_1 + \|d - Dz\|_1$$

- ▶ approach 2: handle constraint implicitly by using conditional subgradients
- ▶ tried various parameter settings (bundle size, periodic restarts)

**Surprise:** very often, these bundle solvers did not reach a solution (but ISA did)

# Outline

Motivation

Infeasible-Point Subgradient Algorithm  
ISAL1

Comparison of  $\ell_1$ -Solvers  
Testset Construction and Computational Results  
Improvements with Heuristic Optimality Check

Possible Future Research

- ▶ 100 matrices  $\mathbf{A}$  (74 dense, 26 sparse)
  - ▶ dense: 512  $\times$   $\{1024, 1536, 2048, 4096\}$
  - ▶ 1024  $\times$   $\{2048, 3072, 4096, 8192\}$
  - ▶ sparse: 2048  $\times$   $\{4096, 6144, 8192, 12288\}$
  - ▶ 8192  $\times$   $\{16384, 24576, 32768, 49152\}$
- ▶ random (e.g., partial Hadamard, random signs, ...)
- ▶ concatenations of dictionaries (e.g., [Haar, ID, RST], ...)
- ▶ columns normalized
- ▶ 4 or 6 vectors  $\mathbf{x}^*$  per matrix such that each resulting (L1) instance (with  $\mathbf{b} \coloneqq \mathbf{Ax}^*$ ) has unique optimum  $\mathbf{x}^*$

548 instances with known, unique solution vectors  $\mathbf{x}^*$ :

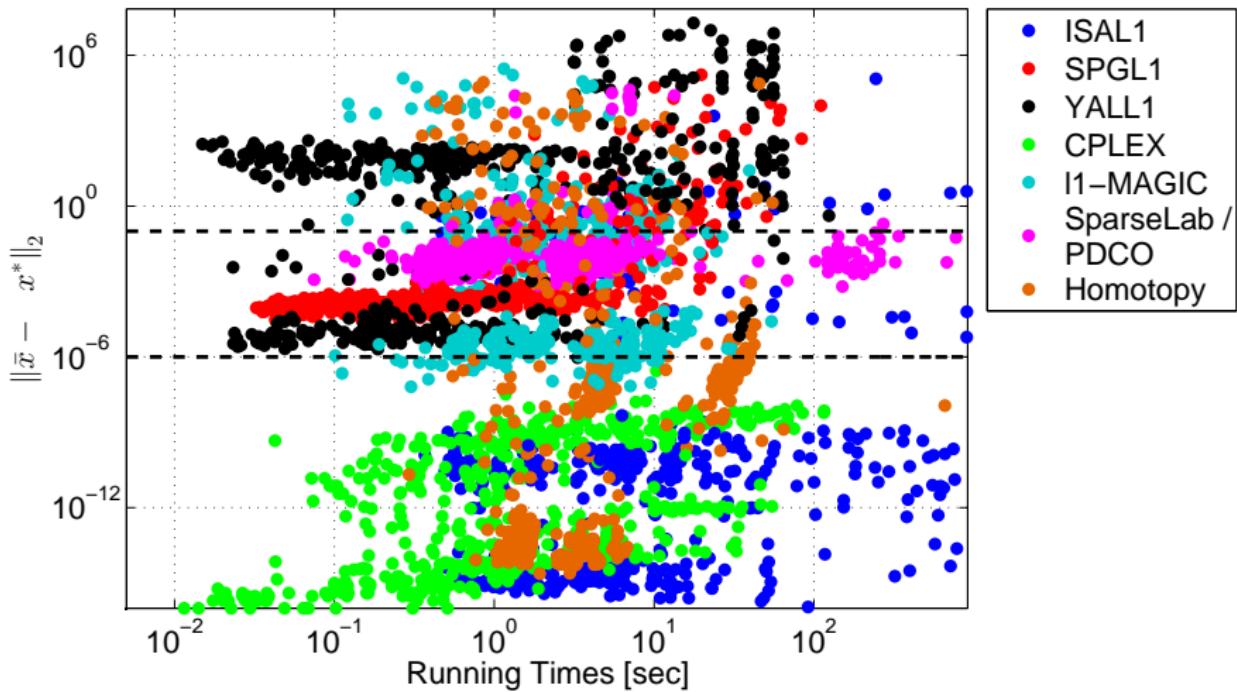
- ▶ For each matrix  $\mathbf{A}$ , choose support  $S$  which obeys

$$ERC(\mathbf{A}, S) := \max_{j \notin S} \|\mathbf{A}_S^\dagger \mathbf{a}_j\|_1 < 1.$$

1. pick  $S$  at random, and
  2. try increasing some  $S$  by repeatedly adding the resp. arg max
  3. For dense  $\mathbf{A}$ 's, use L1TestPack to construct another unique solution support (via optimality condition for (L1))
- ▶ Entries of  $\mathbf{x}_S^*$  random with
    - i) high dynamic range ( $-10^5, 10^5$ )
    - ii) low dynamic range ( $-1, 1$ )

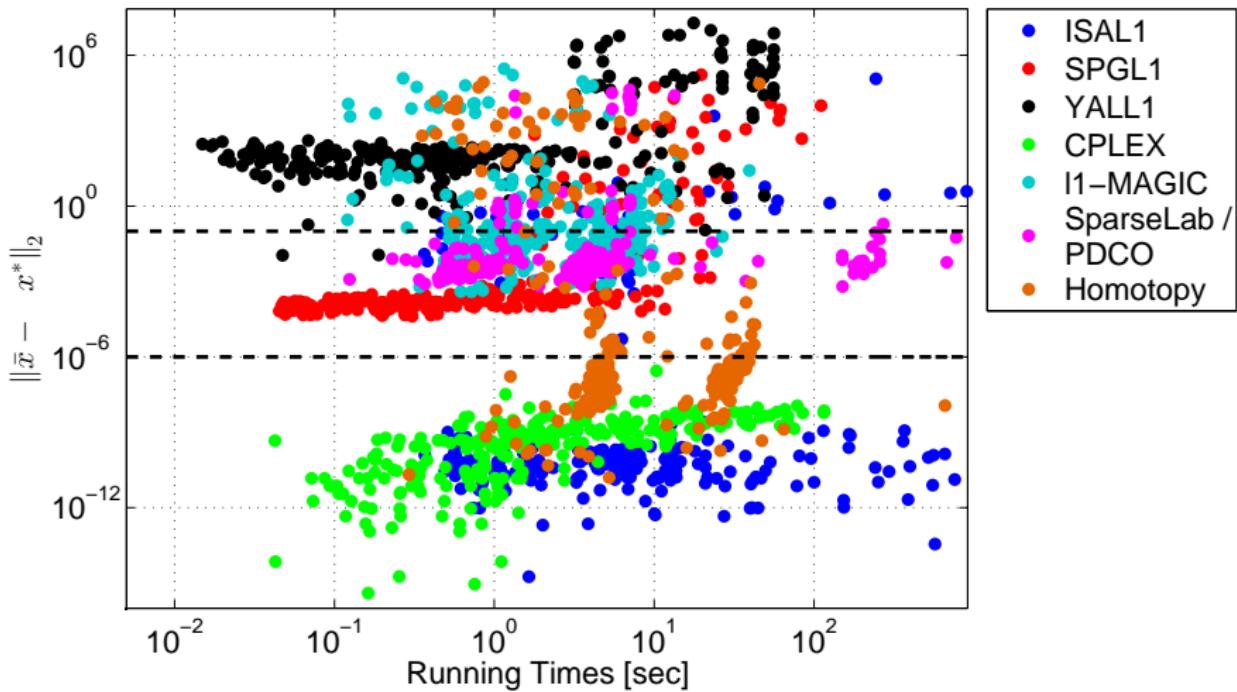
- ▶ Only **exact** solvers for (L1):  $\min \|\mathbf{x}\|_1$  s. t.  $\mathbf{Ax} = \mathbf{b}$
- ▶ Tested algorithms:  
ISAL1, SPGL1, YALL1,  $\ell_1$ -Magic, SolveBP (SparseLab),  $\ell_1$ -Homotopy, CPLEX (Dual Simplex)
- ▶ Use default settings (black box usage)
- ▶ Solution  $\bar{\mathbf{x}}$  “optimal”, if  $\|\bar{\mathbf{x}} - \mathbf{x}^*\|_2 \leq 10^{-6}$
- ▶ Solution  $\bar{\mathbf{x}}$  “acceptable”, if  $\|\bar{\mathbf{x}} - \mathbf{x}^*\|_2 \leq 10^{-1}$

# Running Time vs. Distance from Unique Optimum (whole testset)



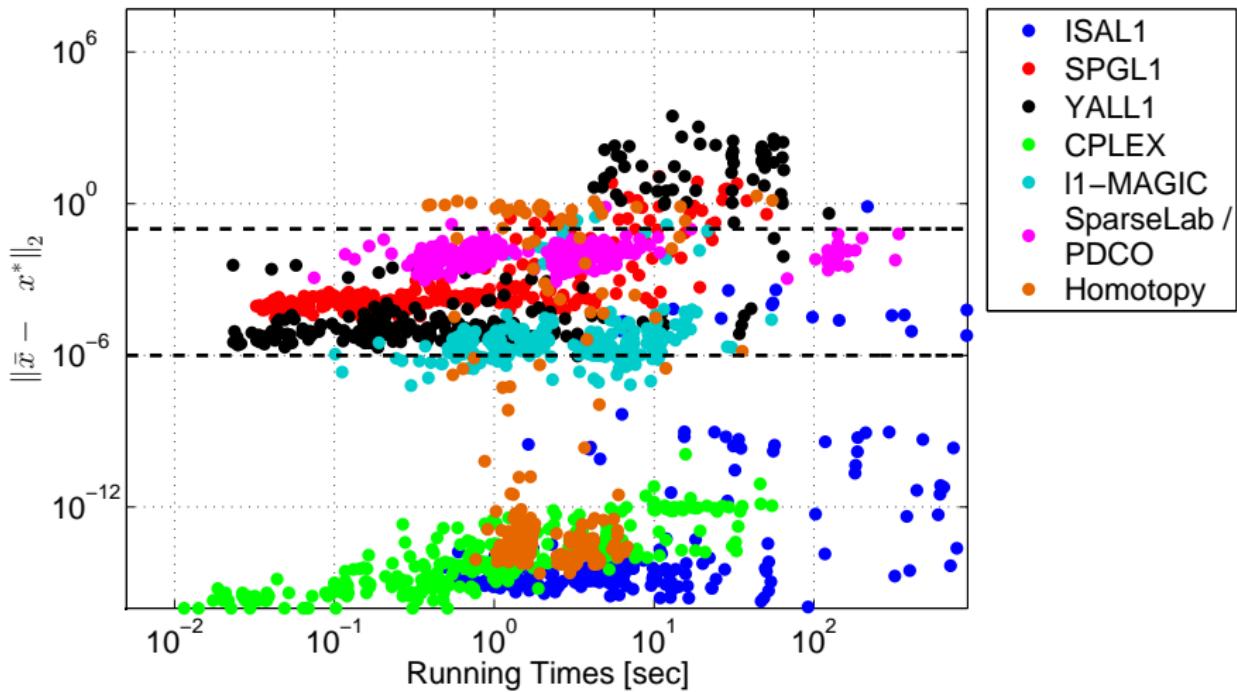
# Running Time vs. Distance from Unique Optimum

(high dynamic range)



# Running Time vs. Distance from Unique Optimum

(low dynamic range)



## Results: Average Performances

- ▶ CPLEX (Dual Simplex): most reliable solver
- ▶ SPGL1: apparently very fast with usually acceptable solutions
- ▶ ISAL1: mostly very accurate, but pretty slow
- ▶ SolveBP: mostly produces acceptable solutions, but rather slow
- ▶  $\ell_1$ -Magic: fast for sparse **A**, but results often unacceptable when solution has high dynamic range
- ▶ YALL1: very fast, but almost always unacceptable for high dynamic range
- ▶  $\ell_1$ -Homotopy: usually accurate (not always acceptable), not really fast

Can we achieve better performances without changing default (tolerance) settings?

# New Optimality Check for $\ell_1$ -Minimization

- ▶ Optimality criterion for (L1):

$$\mathbf{x}^* \in \arg \min_{\mathbf{x}: \mathbf{Ax}=\mathbf{b}} \|\mathbf{x}\|_1 \quad \Leftrightarrow \quad \partial \|\mathbf{x}^*\|_1 \cap \text{Im}(\mathbf{A}^T) \neq \emptyset$$

(Frequent) exact evalution too expensive

# New Optimality Check for $\ell_1$ -Minimization

- ▶ Optimality criterion for (L1):

$$\mathbf{x}^* \in \arg \min_{\mathbf{x}: \mathbf{A}\mathbf{x}=\mathbf{b}} \|\mathbf{x}\|_1 \quad \Leftrightarrow \quad \partial \|\mathbf{x}^*\|_1 \cap \text{Im}(\mathbf{A}^T) \neq \emptyset$$

(Frequent) exact evalution too expensive

- ▶ Heuristic Optimality Check (HOC):

Estimate support  $S$  of given  $\mathbf{x}$  and (approximately) solve

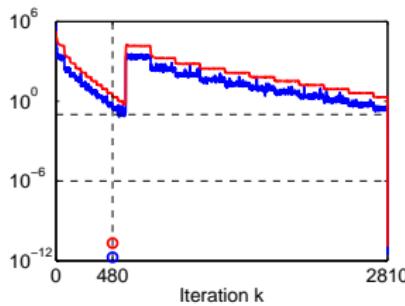
$$\mathbf{A}_S^T \mathbf{w} = \text{sign}(\mathbf{x}_S).$$

If  $\mathbf{w}$  is dual-feasible ( $\|\mathbf{A}^T \mathbf{w}\|_\infty \leq 1$ ), compute  $\bar{\mathbf{x}}$  from  $\mathbf{A}_S \bar{\mathbf{x}}_S = \mathbf{b}$ .  
If  $\bar{\mathbf{x}}$  is primal-feasible, it is optimal if  $\mathbf{b}^T \mathbf{w} = \|\bar{\mathbf{x}}\|_1$ .

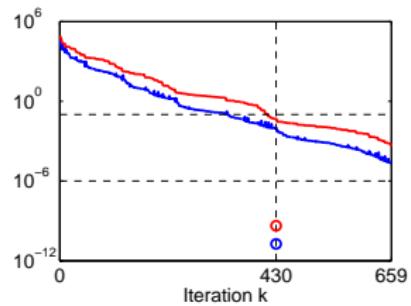
Allows safe “jumping” to optimum (also from infeasible points)

# Impact of Heuristic Optimality Check (Example)

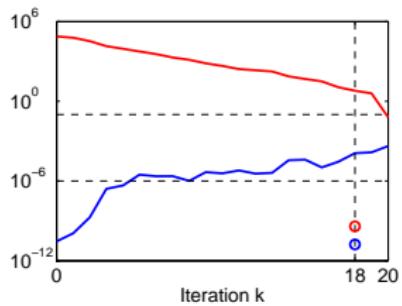
HOC in ISAL1



HOC in SPGL1



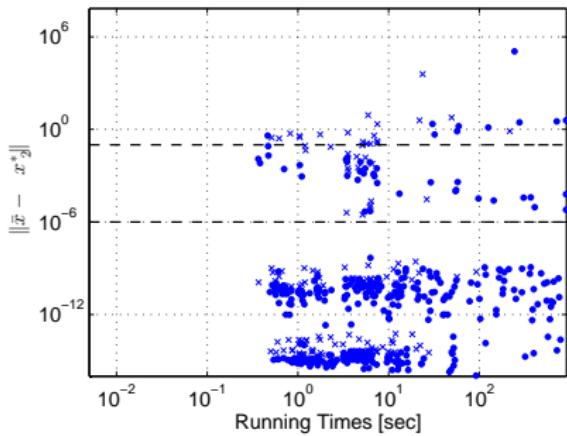
HOC in  $\ell_1$ -Magic



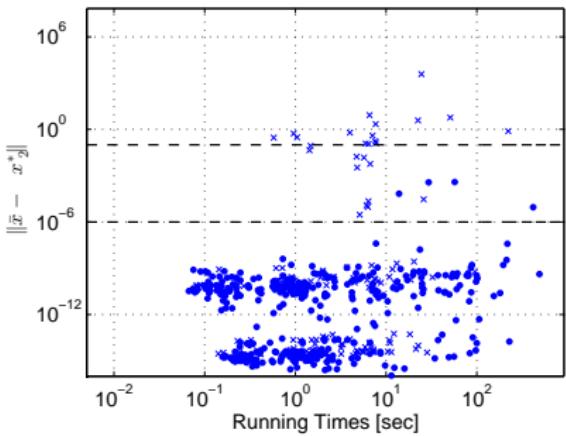
(red curves: distance to known optimum, blue curves: feasibility violation)

# Improvements with HOC – ISAL1

ISAL1 without HOC:

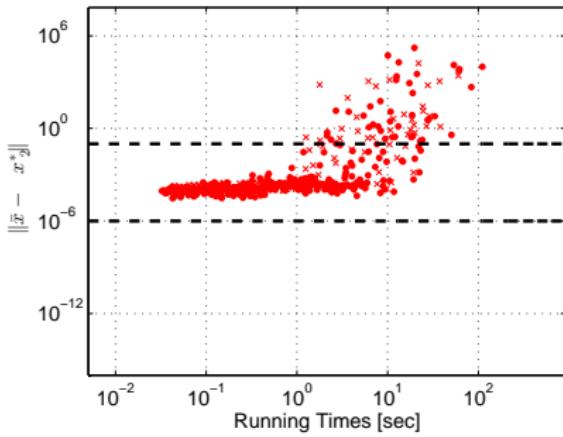


ISAL1 with HOC:

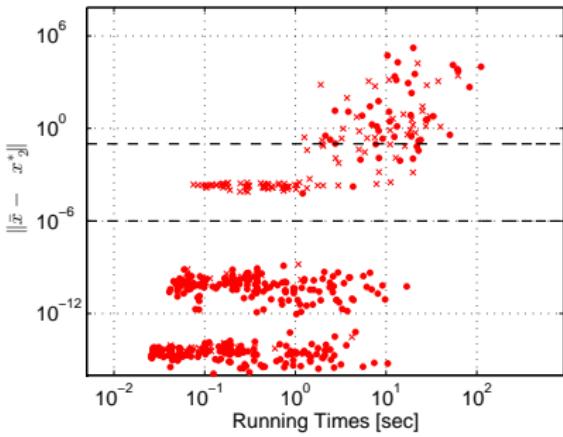


# Improvements with HOC – SPGL1

SPGL1 without HOC:



SPGL1 with HOC:

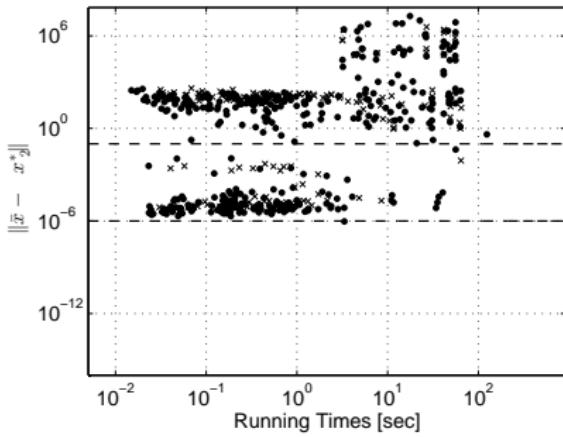


# Improvements with HOC – YALL1

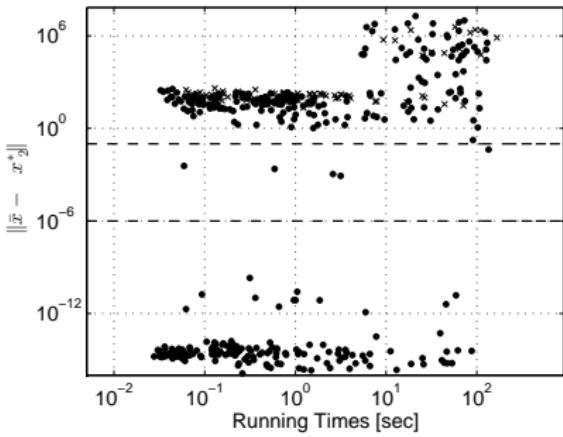


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

YALL1 without HOC:



YALL1 with HOC:

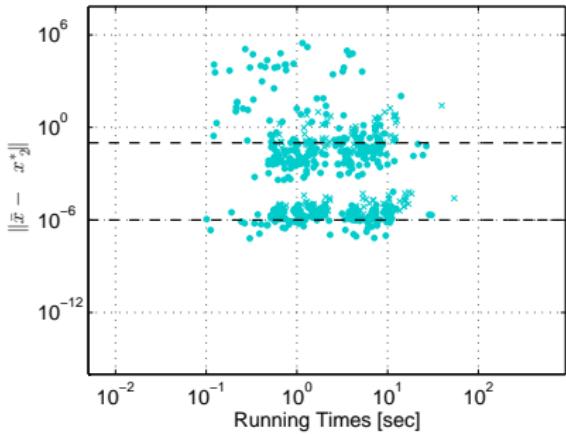


# Improvements with HOC – $\ell_1$ -Magic

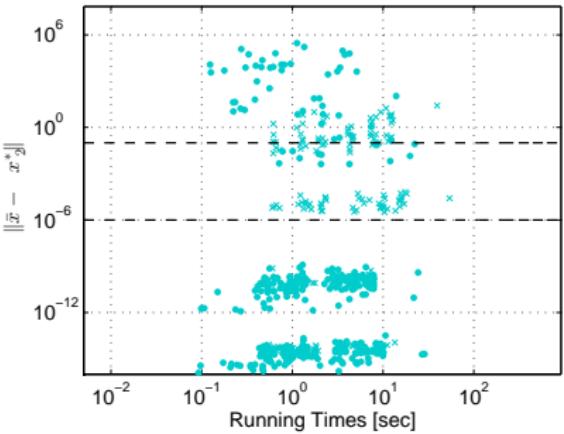


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$\ell_1$ -Magic without HOC:



$\ell_1$ -Magic with HOC:

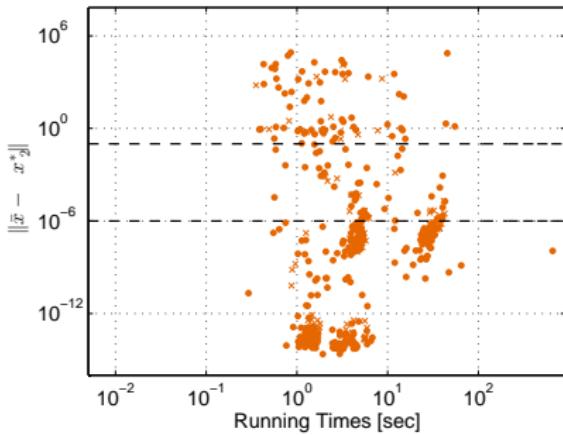


# Improvements with HOC – $\ell_1$ -Homotopy

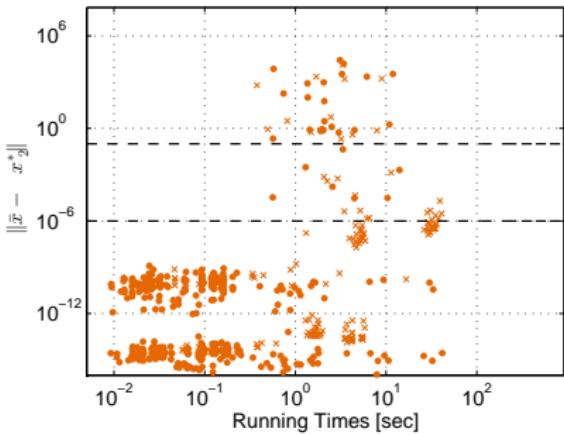


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

$\ell_1$ -Homotopy without HOC:



$\ell_1$ -Homotopy with HOC:



# Improvements with HOC (numbers)



| (400+148 instances)   | ISAL1   | SPGL1 | YALL1 | $\ell_1$ -Mag. | $\ell_1$ -Hom. |
|-----------------------|---------|-------|-------|----------------|----------------|
| solved faster w/ HOC  | 399/467 | −/0   | −/1   | 41/42          | 344/403        |
| high dyn. range       | 184/212 | −/0   | −/0   | −/0            | 161/181        |
| low dyn. range        | 215/255 | −/0   | −/1   | 41/42          | 183/222        |
| solved only w/ HOC    | 51      | 391   | 196   | 337            | 87             |
| high dyn. range       | 42      | 191   | 13    | 170            | 56             |
| low dyn. range        | 9       | 200   | 183   | 167            | 31             |
| improved (ERC-based)  | 98.3%   | 88.5% | 46.3% | 85.0%          | 92.8%          |
| high dyn. range       | 99.5%   | 88.0% | 6.5%  | 78.0%          | 91.0%          |
| low dyn. range        | 97.0%   | 89.0% | 86.0% | 92.0%          | 94.5%          |
| improved (other part) | 38.5%   | 25.0% | 7.4%  | 25.7%          | 54.1%          |
| high dyn. range       | 36.5%   | 20.3% | 0.0%  | 18.9%          | 74.3%          |
| low dyn. range        | 40.5%   | 29.7% | 14.9% | 32.4%          | 33.8%          |

# Improvements with HOC (numbers)



| (400+148 instances)   | ISAL1 | SPGL1 | YALL1 | $\ell_1$ -Mag. | $\ell_1$ -Hom. |
|-----------------------|-------|-------|-------|----------------|----------------|
| improved (ERC-based)  | 98.3% | 88.5% | 46.3% | 85.0%          | 92.8%          |
| improved (other part) | 38.5% | 25.0% | 7.4%  | 25.7%          | 54.1%          |

Explanation for higher HOC success rate on ERC-based testset:

$$\text{ERC} \quad \Rightarrow \quad \mathbf{w}^* = (\mathbf{A}_{S^*}^\top)^\dagger \text{sign}(\mathbf{x}_{S^*}^*) \text{ satisfies } \mathbf{A}^\top \mathbf{w}^* \in \partial \|\mathbf{x}^*\|_1.$$

HOC implementation approximates  $\mathbf{w} = (\mathbf{A}_S^\top)^\dagger \text{sign}(\mathbf{x}_S)$  for estimated support  $S$

# HOC: Speed-up and Overhead



Overhead for HOC usually not too high  $\Rightarrow$  Speed-up on average

|                         | ISAL1           | SPGL1           | YALL1           | $\ell_1$ -Mag.  | $\ell_1$ -Hom.  |
|-------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| avg. rel. speed-up      | 60.10%          | 15.31%          | -57.84%         | 9.39%           | 66.70%          |
| high dynamic range      | 62.67%          | 9.01%           | -88.80%         | 7.07%           | 67.15%          |
| low dynamic range       | 57.53%          | 21.61%          | -26.88%         | 11.72%          | 66.25%          |
| w/ HOC faster (./548)   | 456<br>(83.21%) | 375<br>(68.43%) | 132<br>(24.09%) | 415<br>(75.73%) | 452<br>(82.48%) |
| avg. speed-up if faster | 74.59%          | 25.10%          | 46.30%          | 13.09%          | 81.84%          |
| avg. overhead if slower | 11.72%          | 5.92%           | 90.88%          | 1.93%           | 4.59%           |

# Rehabilitation of $\ell_1$ -Homotopy

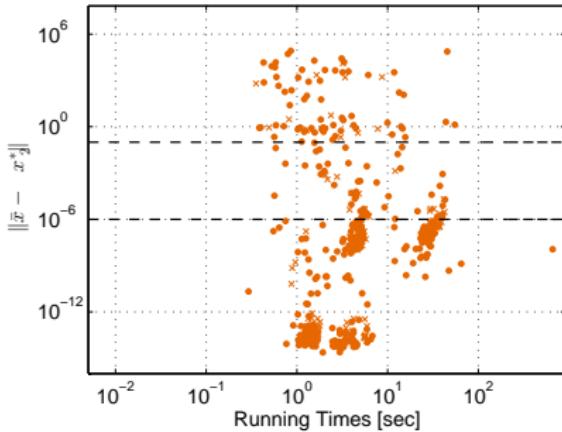


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

The Homotopy method **provably** solves (L1) via a sequence of problems

$\min \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2 + \lambda \|\mathbf{x}\|_1$  for parameters  $\lambda \geq 0$  decreasing to **zero**. Also: Provably fast for suff. sparse solutions.

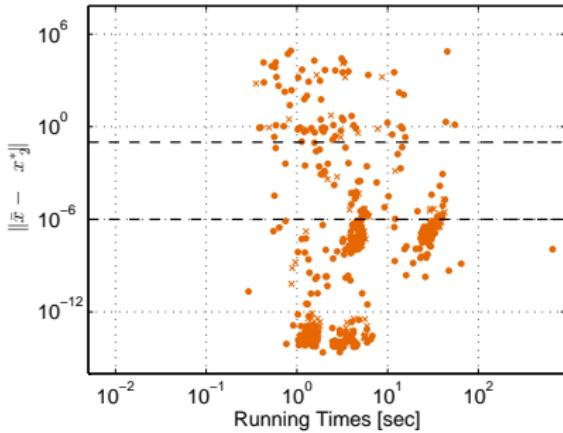
not fast, inaccurate ?!



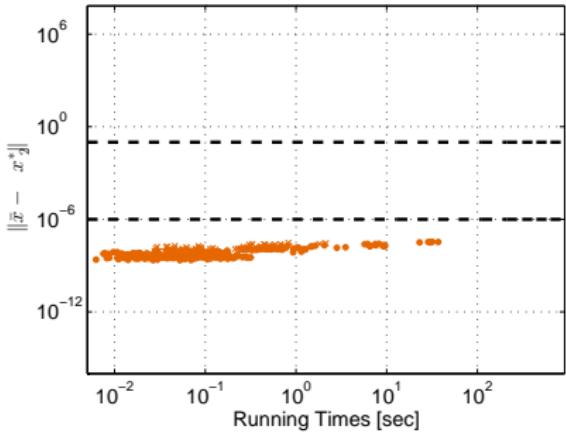
# Rehabilitation of $\ell_1$ -Homotopy

The Homotopy method **provably** solves (L1) via a sequence of problems  $\min \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2 + \lambda \|\mathbf{x}\|_1$  for parameters  $\lambda \geq 0$  decreasing to **zero**. Also: Provably fast for suff. sparse solutions.

final  $\lambda = 0$



final  $\lambda = 10^{-9}$



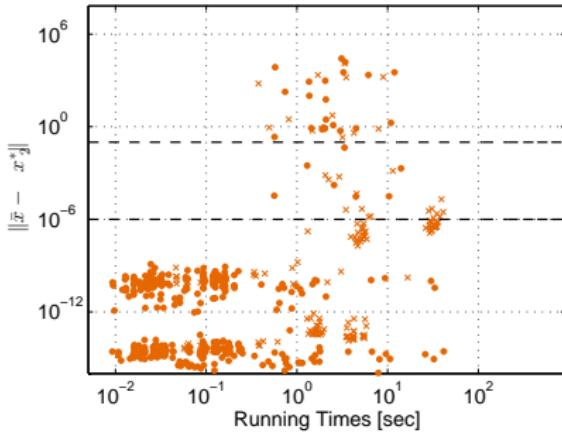
# Rehabilitation of $\ell_1$ -Homotopy



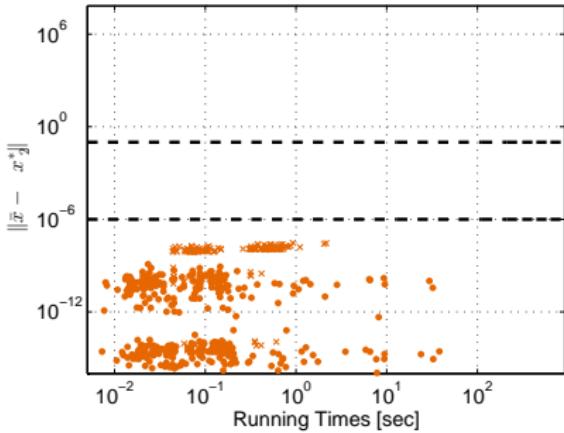
The Homotopy method **provably** solves (L1) via a sequence of problems

$\min \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2 + \lambda \|\mathbf{x}\|_1$  for parameters  $\lambda \geq 0$  decreasing to **zero**. Also: Provably fast for suff. sparse solutions.

final  $\lambda = 0$ , w/ HOC



final  $\lambda = 10^{-9}$ , w/ HOC



# Possible Future Research



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- ▶ Fine-tune HOC integration into solvers?
- ▶ HOC helpful in Approximate Homotopy Path algorithm?
- ▶ Extensions to Denoising problems?
  - ▶  $\mathcal{P}_{\mathcal{F}}^\varepsilon$  for, e.g.,  $\mathcal{F} = \{ \mathbf{x} \mid \| \mathbf{A}\mathbf{x} - \mathbf{b} \|_2 \leq \delta \}$ ?
  - ▶ HOC schemes?
  - ▶ Testsets, solver comparisons (also for implicit matrices), ... ?
- ▶ “Really hard” instances?
  - ▶ e.g., based on construction of Mairal & Yu for which the (exact) Homotopy path has  $\mathcal{O}(3^n)$  kinks?
- ▶ ...

ISA theory:

*Lorenz, Pfetsch & T.: “An Infeasible-Point Subgradient Method Using Adaptive Approximate Projections”, 2011*

ISAL1 theory, HOC & numerical results:

*Lorenz, Pfetsch & T.: “Solving Basis Pursuit: Subgradient Algorithm, Heuristic Optimality Check, and Solver Comparison”, 2011/2012*

Papers, MATLAB Codes (ISAL1, HOC, L1TestPack), Testset, Slides, Posters etc.  
— available at:

[www.math.tu-bs.de/mo/spear](http://www.math.tu-bs.de/mo/spear)