Design of Projection Matrix for Compressive Sensing by Nonsmooth Optimization

W.-S. Lu
Dept. of Electrical & Computer Engineering
University of Victoria
Victoria, Canada

T. Hinamoto
Graduate School of Engineering
Hiroshima University
Hiroshima, Japan

Outline

- Effective Dictionary, Mutual Coherence (MC), and Average MC
- Problem Formulation
- A Subgradient Projection Algorithm
- Experimental Studies

1. Effective Dictionary, Mutual Coherence, and Average Mutual Coherence

A. Effective Dictionary

Let $\boldsymbol{s} \in R^{n\times 1}$ be a signal of interest and consider compressive sampling of $\boldsymbol{s}$ by linear projection $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{s}$ with $\boldsymbol{P} \in R^{m\times n}$ and $m \ll n$. Let $\boldsymbol{D} \in R^{n\times L}$ with $L \geq n$ be a dictionary that sparsifies $\boldsymbol{s}$: $\boldsymbol{s} = \boldsymbol{D\theta}$ where $\boldsymbol{\theta}$ is sparse or near sparse. By convention the 2-norm of each column of $\boldsymbol{D}$ is normalized to unity. The compressed measurement can be expressed as $\boldsymbol{y} = \boldsymbol{PD\theta}$ where the product matrix $\mathcal{D} = \boldsymbol{PD}$ is called *effective dictionary*.

B. Mutual Coherence and Averaged Mutual Coherence

In CS, a popular approach to reconstruct signal $\boldsymbol{s}$ based on measurement $\boldsymbol{y}$ is to solve the convex l_1-minimization problem

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{\theta}\|_1 \\ &\text{subject to:} \;\; \mathcal{D}\boldsymbol{\theta} = \boldsymbol{y} \end{aligned} \tag{1}$$

It turns out that the performance of a CS system is closely related to the *mutual coherence* between projection $\boldsymbol{P}$ and dictionary $\boldsymbol{D}$, which is defined by

$$\mu = \max_{\substack{1 \le i, j \le L \\ i \ne j}} \frac{\boldsymbol{d}_i^T \boldsymbol{d}_j}{\|\boldsymbol{d}_i\| \cdot \|\boldsymbol{d}_j\|} \tag{2}$$

where $\boldsymbol{d}_i$ is the ith column of the effective dictionary $\mathcal{D}$. It was argued [2] that an average measure of coherence describes true behavior of a CS system. In [2], the t - averaged mutual coherence for a given $t > 0$ is defined as

$$\mu_{\mathbf{t}} = \frac{1}{|I_t|} \sum_{(i,j)\in I_t} |g_{i,j}| \tag{3}$$

where $g_{i,j} = \boldsymbol{d}_i^T \boldsymbol{d}_j / (\|\boldsymbol{d}_i\| \cdot \|\boldsymbol{d}_j\|)$, $I_t = \{(i,j) : |g_{i,j}| \ge t\}$, and $|I_t|$ denotes the cardinality of index set I_t.

II. Problem Formulation

Unlike the methods in [2] where the t-averaged mutual coherence is minimized and in [4] where an equiangular tight frame is approximated, we deal with the design of projection matrix by minimizing the mutual coherence μ in (2). Thus the design problem is formulated as

$$\underset{\boldsymbol{P}}{\text{minimize}} \ \max_{\substack{1\le i,j\le L \\ i\ne j}} \frac{|\boldsymbol{d}_i^T \boldsymbol{d}_j|}{\|\boldsymbol{d}_i\| \cdot \|\boldsymbol{d}_j\|} \tag{5}$$

where the projection matrix $\boldsymbol{P}$ is related to $\{\boldsymbol{d}_i, 1 \le i \le L\}$ by

$$\boldsymbol{PD} = \mathcal{D} = \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 & \cdots & \boldsymbol{d}_L \end{bmatrix} \tag{6}$$

with a given dictionary $\boldsymbol{D}$. A natural way to address the problem in (5) is to treat the $\boldsymbol{d}_i$'s as unknowns, then find $\boldsymbol{P}$ via (6) (typically using a least squares technique). Thus we consider the minimax problem

$$\underset{\boldsymbol{d}_i, 1 \le i \le L}{\text{minimize}} \max_{\substack{1 \le i, j \le L \\ i \ne j}} \frac{|\boldsymbol{d}_i^T \boldsymbol{d}_j|}{\|\boldsymbol{d}_i\| \cdot \|\boldsymbol{d}_j\|} \tag{7}$$

Evidently, (7) is a nonconvex problem with a total of mL unknowns. For a small problem size e.g. $m = 30$, $n = 80$, and $L = 120$, the number of unknowns $mL = 3600$ is already fairly large. Furthermore, operation "max" and the absolute values involved within "max" imply a highly nonsmooth objective function in (7).

III. A SUBGRADIENT PROJECTION ALGORITHM

A. Problem Reformulation

Let $\boldsymbol{x}_i$ be the normalized vector $\boldsymbol{d}_i$, i.e., $\boldsymbol{x}_i = \boldsymbol{d}_i / \|\boldsymbol{d}_i\|$. In terms of $\boldsymbol{x}_i$, (7) becomes

$$\begin{aligned} &\underset{\boldsymbol{x}_i, 1\le i\le L}{\text{minimize}} \; \max_{\substack{1\le i,j\le L \\ i\ne j}} \; |\boldsymbol{x}_i^T \boldsymbol{x}_j| \\ &\text{subject to: } \boldsymbol{x}_i^T \boldsymbol{x}_i = 1 \text{ for } 1 \le i \le L \end{aligned} \tag{8a-b}$$

We replace $|\boldsymbol{x}_i^T \boldsymbol{x}_j|$ by $(\boldsymbol{x}_i^T \boldsymbol{x}_j)^2$ in (8a) to get an equivalent problem with an objective function that is smoother than the magnitude of inner product:

$$\begin{aligned} &\underset{\boldsymbol{x}_i, 1\le i\le L}{\text{minimize}} \; \max_{\substack{1\le i,j\le L \\ i\ne j}} \; (\boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \\ &\text{subject to: } \boldsymbol{x}_i^T \boldsymbol{x}_i = 1 \text{ for } 1 \le i \le L \end{aligned} \tag{9a-b}$$

The objective function in (9a), however, remains non-differentiable because of the "max" operation. Furthermore, (9) is a highly nonconvex problem because the objective function is nonconvex and the feasible region defined in (9b) is also nonconvex.

Define

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_L \end{bmatrix} \tag{10}$$

and let $\mathcal{D}_0 = \boldsymbol{P}_0 \boldsymbol{D}$ be an initial effective dictionary matrix with $\boldsymbol{D}$ a given dictionary and $\boldsymbol{P}_0$ an initial random projection matrix. The columns of $\mathcal{D}_0$ are normalized to have unity 2-norm and an initial point of (9), $\tilde{\boldsymbol{x}}_0$, is then constructed by stacking the columns of $\mathcal{D}_0$.

Now assume that we are in the kth iteration to update point $\tilde{\boldsymbol{x}}_k$ to $\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{x}}_k + \boldsymbol{\delta}$ where $\boldsymbol{\delta} = \left[\boldsymbol{\delta}_1 \quad \boldsymbol{\delta}_2 \quad \cdots \quad \boldsymbol{\delta}_L\right]^T$ is supposed to improve the current iterate $\tilde{\boldsymbol{x}}_k$ in the sense of reducing the objective function (i.e., mutual coherence) in (9a). To construct a convex model, it is critical that $\boldsymbol{\delta}$ is maintained small in magnitude so that each term in (9a) is well approximated by

$$\left[\left(\boldsymbol{x}_i + \boldsymbol{\delta}_i\right)^T \left(\boldsymbol{x}_j + \boldsymbol{\delta}_j\right)\right]^2 \approx \left[\boldsymbol{\delta}_i^T \boldsymbol{x}_j + \boldsymbol{\delta}_j^T \boldsymbol{x}_j + \boldsymbol{x}_i^T \boldsymbol{x}_j\right]^2$$

This leads to a *convex* problem with respect to $\boldsymbol{\delta}$:

$$\begin{aligned} &\underset{\boldsymbol{\delta}}{\text{minimize}} \max_{\substack{1\le i,j\le L \\ i\ne j}} \ (\boldsymbol{\delta}_i^T \boldsymbol{x}_j + \boldsymbol{\delta}_j^T \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \\ &\text{subject to:} \ \ \|\boldsymbol{\delta}_i\|_2 \le \beta \quad \text{for } 1 \le i \le L \end{aligned} \qquad \text{(11a-b)}$$

where $\beta > 0$ is a small constant that controls the size of the feasible region. Once (11) is solved, the optimal $\boldsymbol{\delta}^*$ is used to update $\tilde{\boldsymbol{x}}_k$ to $\tilde{\boldsymbol{x}}_k + \boldsymbol{\delta}^*$, then each length-$m$ block of $\tilde{\boldsymbol{x}}_k + \boldsymbol{\delta}^*$ is normalized to have unit 2-norm so as to satisfy the constraints in (9b). It is this normalized $\tilde{\boldsymbol{x}}_k + \boldsymbol{\delta}^*$ that becomes the next iterate $\tilde{\boldsymbol{x}}_{k+1}$.

If we define

$$f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k) = \max_{\substack{1 \le i, j \le L \\ i \ne j}} (\boldsymbol{\delta}_i^T \boldsymbol{x}_j + \boldsymbol{\delta}_j^T \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \tag{12}$$

which is a *convex* model of the coherence surrounding $\tilde{\boldsymbol{x}}_k$, problem (11) becomes

$$\begin{aligned} &\underset{\delta}{\text{minimize}} \quad f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k) \\ &\text{subject to: } \ \|\boldsymbol{\delta}_i\|_2 \le \beta \quad \text{for } 1 \le i \le L \end{aligned} \tag{13a-b}$$

B. Solving (13) *by Subgradient Projection*

Function $f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k)$ involves a “max” operation, hence it does not have a gradient. However $f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k)$ is convex and possesses subdifferential $\partial f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k)$ which is defined as the set of vectors satisfying

$$f(\boldsymbol{\delta}_1, \tilde{\boldsymbol{x}}_k) \geq f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k) + \partial f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k)^T (\boldsymbol{\delta}_1 - \boldsymbol{\delta}) \tag{14}$$

for any $\boldsymbol{\delta}$ and $\boldsymbol{\delta}_1$. Using (14), it is easy to verify that a subgradient of $f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k)$ is given by

$$\partial f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k) = 2 p_{i^*, j^*} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{x}_{j^*} (i^* \text{th block}) \\ \mathbf{0} \\ \boldsymbol{x}_{i*} (j^* \text{th block}) \\ \mathbf{0} \end{bmatrix} \tag{15}$$

where (i^*, j^*) denotes the index pair at which the maximum of $(\boldsymbol{\delta}_i^T \boldsymbol{x}_j + \boldsymbol{\delta}_j^T \boldsymbol{x}_j + \boldsymbol{x}_i^T \boldsymbol{x}_j)^2$ is achieved and

$$p_{i^*,j^*} = \boldsymbol{\delta}_{i^*}^T \boldsymbol{x}_{j^*} + \boldsymbol{\delta}_{j^*}^T \boldsymbol{x}_{i^*} + \boldsymbol{x}_{i^*}^T \boldsymbol{x}_{j^*} \tag{16}$$

Note that the subgradient in (15) can be evaluated efficiently because it has only two nonzero length-m blocks to fill. With (15), the subgradient projection method [6][7] can be applied to iteratively solve problem (13) as

$$\boldsymbol{\delta}_{l+1} = \prod\nolimits_{\beta}[\boldsymbol{\delta}_l - \alpha_l \partial f(\boldsymbol{\delta}_l, \tilde{\boldsymbol{x}}_k)] \tag{17}$$

for $l = 1, 2, \ldots$ where $\alpha_l > 0$ is a step size and $\prod_{\beta}(\boldsymbol{v})$ is a projection operator that applies to each length-m block in vector $\boldsymbol{v}$ so that if the 2-norm of the vector block does not exceed β, then the operator leaves it unaltered, otherwise the vector block is multiplied by a scaling factor $0 < \gamma < 1$ such that the 2-norm of the scaled block equals to β. It can be shown [6][7] that if the step size α_l in (17) is chosen such that $\sum_l \alpha_l = \infty$ and $\sum_l \alpha_l^2 < \infty$, then δ_l in (17) converges to a global solution of (13) as $l \to \infty$. For example, sequence α_l = $1/(l+1)$ shall work. In practice, iteration (17) is carried out for a finite number (M) of times, and the step size is often chosen as a *constant* e.g. $\alpha_l = c/\sqrt{M}$ with an appropriate constant c.

C. An Algorithm for Solving (9)

Based on the above development, an algorithm for solving (9) (hence (7)) can be outlined as follows.

Input: A sparsifying dictionary $\boldsymbol{D} \in R^{n\times L}$, an initial random projection $\boldsymbol{P}_0 \in R^{m\times n}$, number of outer iterations K, and number of inner iterations M.

<u>Outer iteration</u>

for $k = 0, 1, \ldots, K-1$

Inner iteration

set $\delta_0 = \mathbf{0}$

for l = $0, 1, 2, \ldots M-1$

use (17) to obtain δ_{l+1}

end

$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_k + \boldsymbol{\delta}_M$

use (10) to get individual $\boldsymbol{x}_i$'s

construct

$$\mathcal{D} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_L \end{bmatrix} \tag{18}$$

Find $\boldsymbol{P}_{k+1}$ by solving

$$\underset{\boldsymbol{P}}{\text{minimize}} \quad \| \boldsymbol{P}\boldsymbol{D} - \mathcal{D} \|_F \tag{19}$$

Compute $\mathcal{D}_{k+1} = \boldsymbol{P}_{k+1}\boldsymbol{D} \equiv \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 & \cdots & \boldsymbol{d}_L \end{bmatrix}$

Normalize columns of $\mathcal{D}_{k+1}$ to get $\boldsymbol{x}_i = \boldsymbol{d}_i / \| \boldsymbol{d}_i \|$

Construct iterate $\tilde{\boldsymbol{x}}_{k+1}$ using (10).

end

Concerning the problem in (19), we consider two cases. The first case is when $L = n$ and $\boldsymbol{D}$ is an orthonormal basis. In this case the solution of (19) is simply $\boldsymbol{P}_{k+1} = \mathcal{D}\boldsymbol{D}^T$, and the effective dictionary is $\mathcal{D}_{k+1} = \mathcal{D}$ which is obtained from (18). The second case is when $L > n$, thus $\boldsymbol{D}$ is an overcomplete dictionary. In this case we assume $\boldsymbol{D}$ has full row-rank, i.e. rank$(\boldsymbol{D}) = n$. It can readily be shown that the solution of (19) is given by

$$\boldsymbol{P}_{k+1} = \mathcal{D}\boldsymbol{D}^{\dagger} = \mathcal{D}\boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1} \tag{20}$$

IV. EXPERIMENTAL STUDIES

A. Coherence Minimization

The purpose of this part of experimental studies is to demonstrate the proposed algorithm offers a good local solution for problem (7). To this end, the proposed algorithm as well as the algorithms in [2] and [4] were applied to a CS system of size $m = 30$, $n = 200$, and $L = 400$, where the dictionary $\boldsymbol{D} \in R^{200 \times 400}$ is a random matrix drawn from i.i.d. zero-mean, unit-variance Gaussian distributions. In our simulations, each algorithm run 2000 iterations (this means to set $K = 2000$, the number of subgradient projections was set to $M = 100$). For the algorithm in [2], parameters γ and t were set to 0.95 and 0.2, respectively; for the algorithm in [4], $\bar{\mu}$ was set to 0.175; and for the proposed algorithm parameters β and c were set to 0.025 and 0.1, respectively. The profiles of mutual coherence μ versus iterations for the three algorithms are shown in Fig. 1 where the curves generated by the algorithms of [2] and [4] are marked as “Elad” and “Yu-Li-Chang”, respectively. The value of μ associated with the initial projection matrix was 0.7440. From Fig. 1, it is evident that the proposed algorithm does minimizing the mutual coherence.

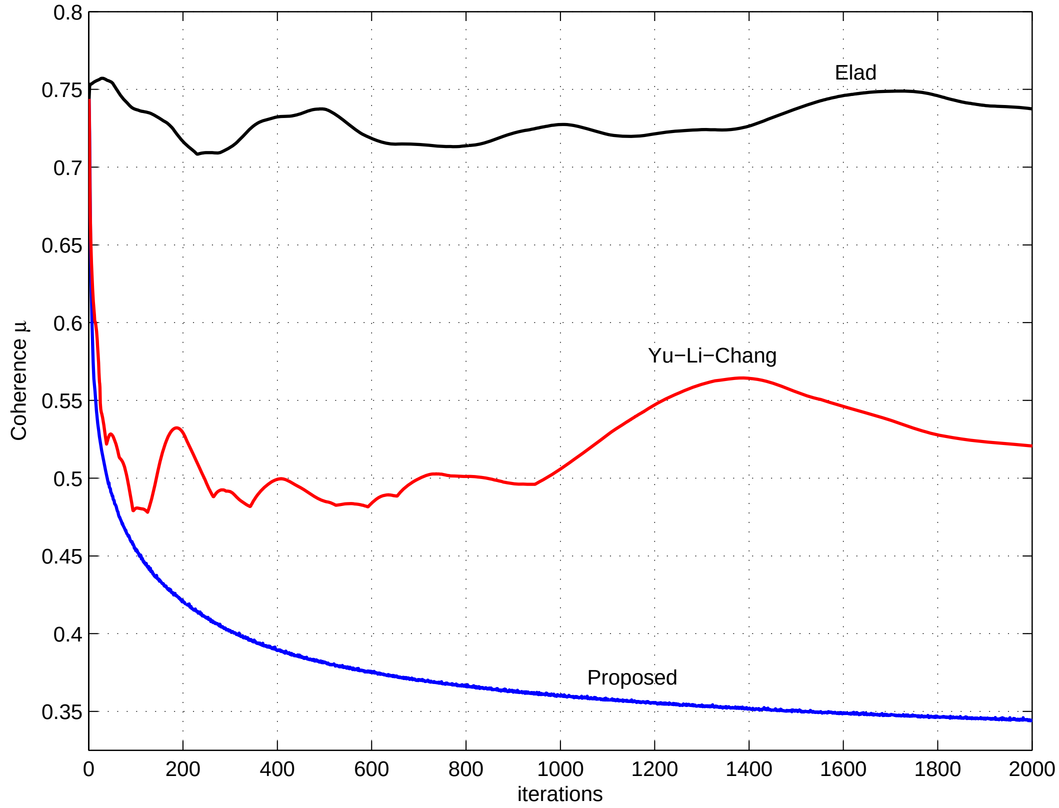

Fig. 1 Mutual coherence μ versus iterations for [2], [4], and the proposed algorithm.

The value of μ after 2000 iterations was found to be 0.3439. The profiles associated with the methods of [2] and [4] do not appear to converge. The minimum values of μ achieved by [2] was 0.7083 and by [4] was 0.4781. This non convergence is not surprising because the algorithms in [2] and [4] are not designed for minimizing μ, rather they are developed for minimizing t-averaged μ and approximating the equiangular tight frame, respectively.

Let $\boldsymbol{G} = \mathcal{D}^T \mathcal{D}$ be the Gram matrix of the effective dictionary $\mathcal{D}$ whose columns are normalized to the unit 2-norm. The histograms of the absolute off-diagonal elements of $\boldsymbol{G}$ (only those above the diagonal are counted as $\boldsymbol{G}$ is symmetrical) for the three algorithms are evaluated and averaged over 100 CS systems of the same size $m = 30, n = 200$, and $L = 400$. Each algorithm run 1000 iterations with all parameters involved set to the same values as before. The results are depicted in Fig. 2.

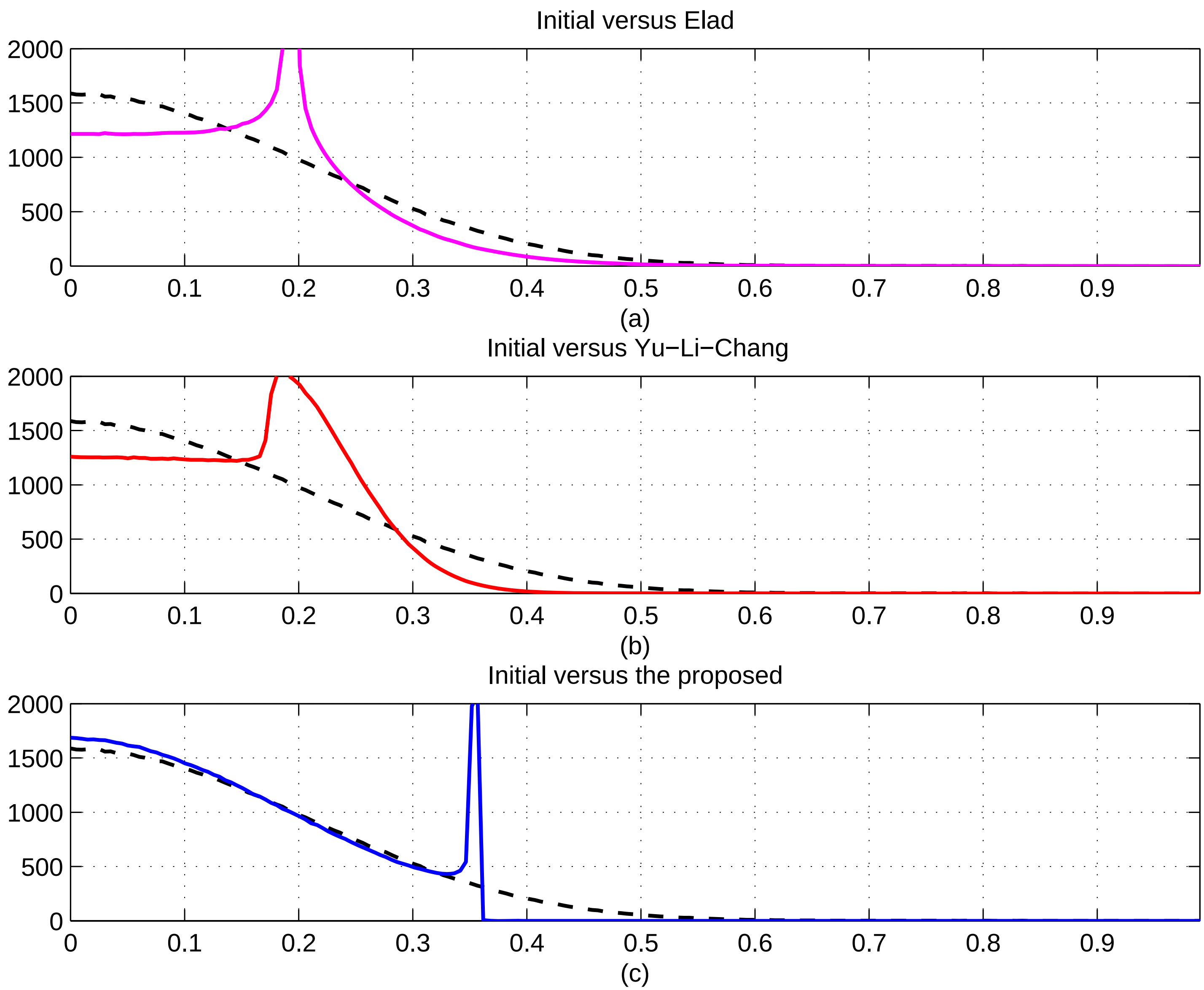

Fig. 2 Histogram of absolute off-diagonal elements of $\boldsymbol{G}$ produced by (a) algorithm [2], (b) algorithm [4], and (c) the proposed algorithm versus those of the initial $\boldsymbol{G}$.

It is observed that

(a) the algorithm in [2] was able to reduce the number of absolute inner products that are greater than 0.24. However, the algorithm also reduces the number of absolute inner products that are less than 0.135 that is evidently undesirable;

(b) the algorithm in [4] was able to push the histogram profile of the initial Gram matrix further relative to that achieved by the algorithm in [2]. However it reduces the number of absolute inner products less than 0.14;

(c) the proposed algorithm exhibits a clear-cut profile with a much smaller upper bound $\mu = 0.3618$. In addition, the algorithm maintains the histogram of the initial $\boldsymbol{G}$ for small absolute inner products. In effect, the number of absolute inner products that are less than 0.16 was even slightly increased.

B. Reconstruction Performance

The performance of the optimized projection matrix was evaluated by applying it to the l_1-minimization (known as basis pursuit (BP)) method for signal reconstruction. The CS system is of size (m, n, L) with $n = L = 128$ and m varying from 31 to 41. The dictionary is a random matrix of size 128×128 with all columns normalized. For each $m \in [31,\ 41]$, a total of 105 sparse signals with sparsity 6 were used and a reconstruction instance was deemed erroneous if the 2-norm reconstruction error exceeded 10^{-4}. The relative number of errors versus the number of measurements m is shown in Fig. 3. For comparison, the methods of [2] and [4] were also evaluated with the same system setting. Since the performance of these two methods for BP are very close to each other (see Sec. 4 of [4]), Fig. 3 only compares the proposed algorithm with the algorithm in [2]. Performance improvement by the proposed algorithm is observed for most values of m.

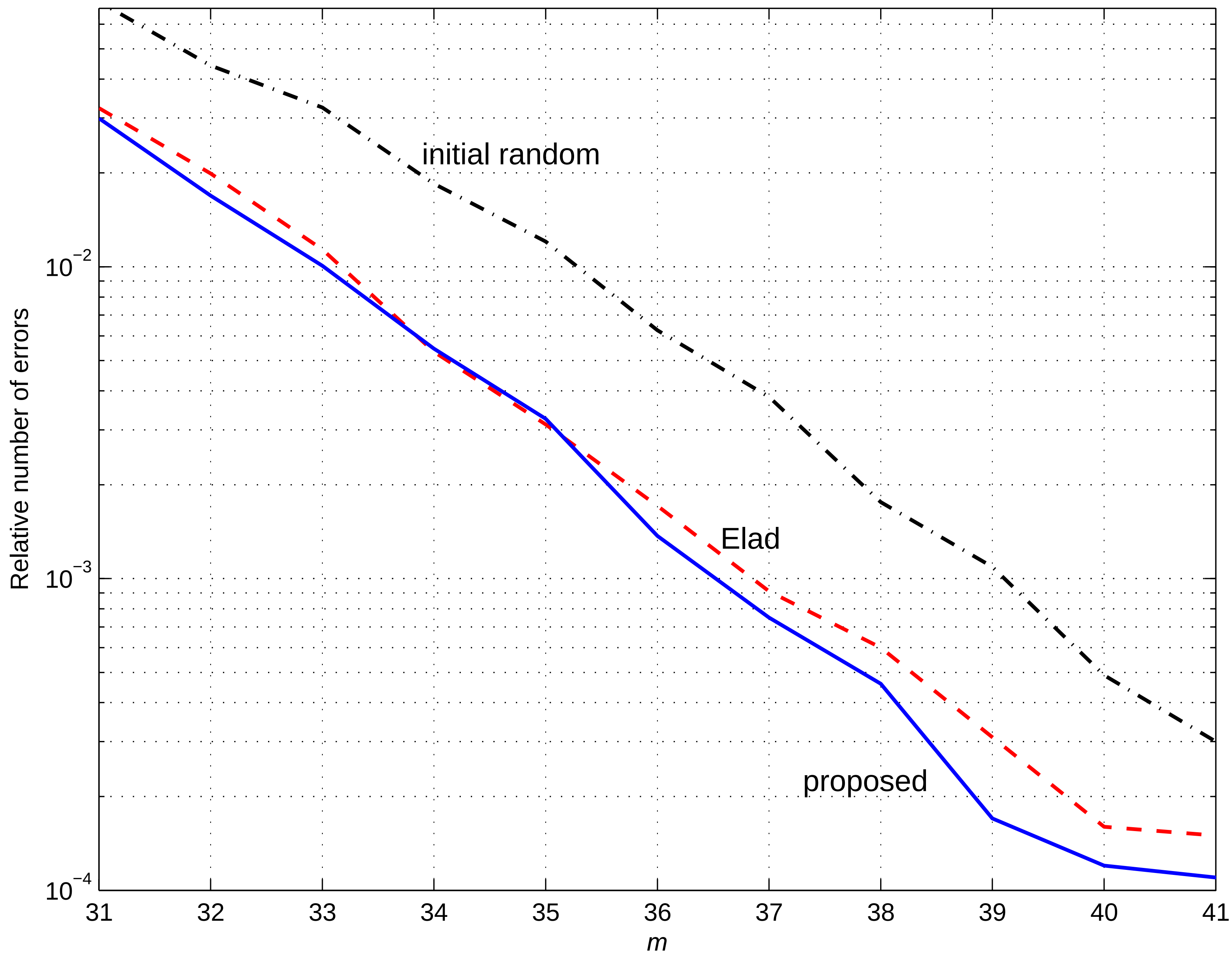

Fig. 3 Relative number of errors versus m for BP-based CS systems.

REFERENCES

[1] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[2] M. Elad, “Optimized projections for compressed sensing,” *IEEE Trans. Signal Processing*, vol. 55, no. 12, pp. 5695–5702, Dec. 2007.

[3] J. M. Duarte-Carvajalino and G. Sapiro, “Learning to sense sparse signals: simultaneous sensing matrix and sparsifying dictionary optimization,” *IEEE Trans. Image Processing*, vol. 18, no. 7, pp. 1395–1408, 2009.

[4] L. Yu, G. Li, and L. Chang, “Optimizing projection matrix for compressed sensing systems,” *Proc. 8th Int. Conf. Information, Communications and Signal Processing*, Paper WP4.6, Singapore, Dec. 2011.

[5] T. Strohmer and R. W. Heath, “Grassmannian frames with applications to coding and communications,” *Appl. Comp. Harmonic Anal.*, vol. 14, no. 3, pp. 257–275, May 2003.

[6] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, NY, 1987.

[7] Y. Nesterov, *Introductory Lectures on Convex Optimization — A Basic Course*, Kluwer Academic Publishers, Boston, MA, 2004.