Design of Projection Matrix for Compressive Sensing by Nonsmooth Optimization

Wu-Sheng Lu
Dept. of Electrical and Computer Engineering
University of Victoria
Victoria, BC, Canada V8W 2Y2
Email: wslu@ece.uvic.ca

Takao Hinamoto
Graduate School of Engineering
Hiroshima University
Higashi-Hiroshima, 739-8527, Japan
Email: hina@crest.ocn.ne.jp

***Abstract*—Sparsity and incoherence are the two key ingredients in compressive sensing (CS). Given a sparsifying dictionary $\boldsymbol{D}$, the projection matrix $\boldsymbol{P}$ must be as incoherent with $\boldsymbol{D}$ as possible for the CS system to be efficient. Thus the design of projection matrix is naturally a problem of minimizing the coherence between $\boldsymbol{P}$ and $\boldsymbol{D}$. Unfortunately, this turns out to be a nonconvex, nonsmooth, large-scale problem even for a CS system of moderate size. In this paper, the above-mentioned problem is investigated in a formulation where the problem is converted into a sequence of nonsmooth but convex subproblems. A subgradient projection algorithm is proposed to solve the nonsmooth subproblems that converges to a projection matrix with improved performance. The performance of the proposed algorithm is evaluated by simulations and comparisons with several existing techniques.**

I. Introduction

Sparsity and incoherence are the two key ingredients in the theory and practice of compressive sensing [1]. Given a dictionary $\boldsymbol{D}$ that sparsifies a class of signals of interest, the projection matrix $\boldsymbol{P}$ of a CS system must be as incoherent as possible with $\boldsymbol{D}$. Random projection matrices are largely incoherent with practically any dictionary $\boldsymbol{D}$. Indeed, if one constructs an orthonormal bases $\boldsymbol{\Phi}$ by orthonormalizing n vectors sampled independently and uniformly on the unit sphere, then with high probability the coherence between $\boldsymbol{\Phi}$ and $\boldsymbol{D}$ is about $\sqrt{2\log n}$ [1]. A satisfactory projection matrix $\boldsymbol{P}$ can then be constructed by selecting m rows from matrix $\boldsymbol{\Phi}$ at random. A question that naturally arises is whether such a projection matrix can be improved to enhance the performance of the CS system. Several authors have investigated the problem in the past several years [2]–[4]. In [2], an averaged measure μ_t of mutual coherence of the effective dictionary (which is defined as $\boldsymbol{PD}$) is introduced and an algorithm to minimize μ_t is proposed. In [3], the projection matrix $\boldsymbol{P}$ is constructed such that the Gram matrix of $\boldsymbol{PD}$ approximates the identify matrix in Frobenius norm. In [4], $\boldsymbol{P}$ is generated such that $\boldsymbol{PD}$ approximates an equiangular tight frame [5].

In this paper, the minimization of mutual coherence is investigated in a formulation of sequential nonsmooth convex programming. A subgradient projection algorithm is proposed to solve the minimax problem involved. The performance of the proposed algorithm is evaluated by experimental studies which include comparisons with several existing techniques.

II. Preliminaries

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 $\boldsymbol{s}$: $\boldsymbol{s} = \boldsymbol{D}\boldsymbol{\theta}$ 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 $\boldsymbol{y} = \boldsymbol{PD}\boldsymbol{\theta}$ 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

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{\theta}\|_1 && \text{(1a)} \\ \text{subject to} \quad & \mathcal{D}\boldsymbol{\theta} = \boldsymbol{y} && \text{(1b)} \end{aligned}$$

It turns out 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

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

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 $t > 0$ is defined as

$$\mu_t = \frac{1}{|\mathcal{I}_t|} \sum_{(i,j)\in\mathcal{I}_t} |g_{ij}| \tag{3}$$

where $g_{ij} = \boldsymbol{d}_i^T \boldsymbol{d}_j / (\|\boldsymbol{d}_i\| \cdot \|\boldsymbol{d}_j\|)$, $\mathcal{I}_t$ is the index set $\mathcal{I}_t = \{(i,j) : |g_{ij}| \geq t\}$, and $|\mathcal{I}_t|$ denotes the cardinality of $\mathcal{I}_t$.

C. Equiangular Tight Frames

A set of column vectors $\{\boldsymbol{\phi}_i \in R^{m\times 1},\ 1 \leq i \leq L\}$ is said to be a frame if there exist constants $B \geq A > 0$ such that $A\|\mathbf{f}\|_2^2 \leq \sum_{i=1}^{L} |\langle \mathbf{f}, \boldsymbol{\phi}_i\rangle|^2 \leq B\|\mathbf{f}\|_2^2$ holds for any $\mathbf{f} \in R^m$. The frame is said to be A-tight if $A = B$. A unit-norm tight frame $\{\boldsymbol{\phi}_i, 1 \leq i \leq L\}$ is said to be an equiangular tight frame if the absolute inner products of all pairs of distinct columns

978-1-4799-3432-4/14/$31.00 ©2014 IEEE

of the frame are equal. This value of absolute inner product is known to be [5]

$$\bar{\mu} = \left[\frac{L-m}{m(L-1)}\right]^{1/2} \tag{4}$$

III. 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 optimal projection matrix by minimizing the mutual coherence μ in (2). Thus the design problem is formulated as

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

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

$$\boldsymbol{P}\boldsymbol{D} = \mathcal{D} = [\boldsymbol{d}_1 \;\; \boldsymbol{d}_2 \;\; \cdots \boldsymbol{d}_L] \tag{6}$$

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, see below). Under this circumstance, we consider the minimax problem

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

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).

IV. A subgradient projection algorithm to solve (7)

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

$$\underset{\boldsymbol{x}_i, 1\le i\le L}{\text{minimize}} \max_{\substack{1\le i,j\le L\\ i\neq j}} |\boldsymbol{x}_i^T \boldsymbol{x}_j| \tag{8a}$$

$$\text{subject to: } \boldsymbol{x}_i^T \boldsymbol{x}_i = 1 \quad \text{for } 1 \le i \le L \tag{8b}$$

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:

$$\underset{\boldsymbol{x}_i, 1\le i\le L}{\text{minimize}} \max_{\substack{1\le i,j\le L\\ i\neq j}} (\boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \tag{9a}$$

$$\text{subject to: } \boldsymbol{x}_i^T \boldsymbol{x}_i = 1 \quad \text{for } 1 \le i \le L \tag{9b}$$

The objective function in (9a), however, remains nondifferentiable because of the "max" operation. Furthermore, (9) is a highly nonconvex problem because the objective function is nonconvex and the feasible region defined by (9b) is also nonconvex. Below we describe a technique that solves (9) by locally convexifying (9) in conjunction with subgradient projections in a sequential manner. In what follows, we use $\tilde{\boldsymbol{x}}$ to denote a vector of dimension mL that collects the L unknown vectors $\{\boldsymbol{x}_i\}$ in the form of

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

It is well known that random projections as a means of compressive sampling work well [1]. From an optimization perspective, this implies the availability of a good initial point. Let $\boldsymbol{P}_0$ be a random projection matrix. Given a dictionary $\boldsymbol{D}$ and projection $\boldsymbol{P}_0$, an initial effective dictionary $\mathcal{D}_0$ is constructed as $\mathcal{D}_0 = \boldsymbol{P}_0\boldsymbol{D}$. The columns of $\mathcal{D}_0$ is then normalized to have unit 2-norm, and an initial point for problem (9), $\tilde{\boldsymbol{x}}_0$, is generated by stacking the columns of the normalized effective dictionary.

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

$$\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{\delta}_1 \\ \vdots \\ \boldsymbol{\delta}_L \end{bmatrix}$$

is responsible for improving the current iterate $\tilde{\boldsymbol{x}}_k$ in the sense of reducing the objective function (i.e., mutual coherence) in (9a). To produce a convex model, it is critical that $\boldsymbol{\delta}$ is maintained small in magnitude so that each term in (9a) is well approximated by

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

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

$$\min_{\boldsymbol{\delta}} \max_{\substack{1\le i,j\le L\\ i\neq j}} (\boldsymbol{\delta}_i^T \boldsymbol{x}_j + \boldsymbol{\delta}_j^T \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{x}_j)^2 \tag{11a}$$

$$\text{subject to: } \|\boldsymbol{\delta}_i\|_2 \le \beta \qquad \text{for } 1 \le i \le L \tag{11b}$$

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 $\tilde{\boldsymbol{x}}_{k+1}$.

At $\tilde{\boldsymbol{x}}_k$, we define

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

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

$$\underset{\boldsymbol{\delta}}{\text{minimize}}\, f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k) \tag{13a}$$

$$\text{subject to: } \|\boldsymbol{\delta}_i\|_2 \le \beta \qquad \text{for } 1 \le i \le L \tag{13b}$$

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 (called subgradients, also denoted by $\partial f(\boldsymbol{\delta}, \tilde{\boldsymbol{x}}_k)$ here) satisfying

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

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

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

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

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

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 $\boldsymbol{\delta}_l$ in (17) converges to a global solution of (13) as $l \to \infty$. For example, in theory $\alpha_l = 1/(l+1)$ shall work. In practice, iteration (17) is carried out by 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 development made in Sections IV.A and IV.B, 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.

Outer iteration
for $k = 0, 1, \ldots, K-1$
  Inner iteration set $\boldsymbol{\delta}_0 = \boldsymbol{0}$
  for $l = 0, 1, 2, \ldots M-1$
    use (17) to obtain $\boldsymbol{\delta}_{l+1}$
  end
  $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_k + \boldsymbol{\delta}_M$
  use (10) to get individual $\boldsymbol{x}_i$'s

construct $\mathcal{D} = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \ldots \ \boldsymbol{x}_L]$ (18)

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

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

  Compute $\mathcal{D}_{k+1} = \boldsymbol{P}_{k+1}\boldsymbol{D} \equiv [\boldsymbol{d}_1 \ \boldsymbol{d}_2 \ \ldots \ \boldsymbol{d}_L]$
  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 $\mathcal{D}$ is an orthonormal basis. In this case the solution of (18) 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

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

V. 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. 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.

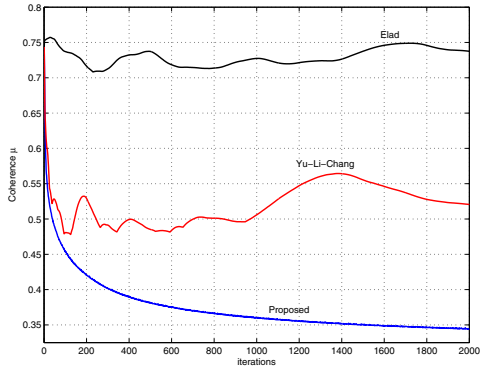

Fig. 1. Mutual coherence μ versus iterations for [2], [4], and the proposed algorithm.

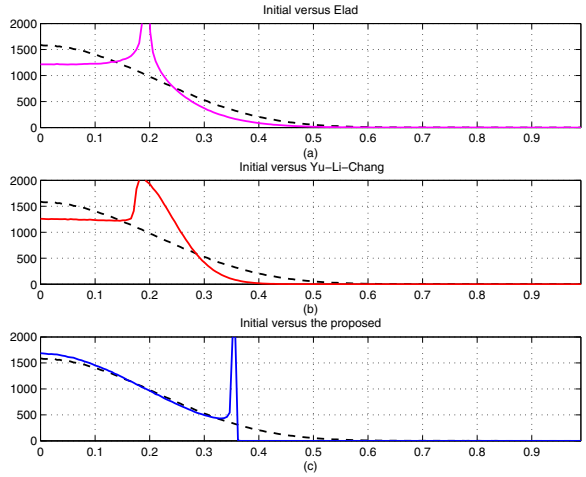

Fig. 2. Histogram of absolute off-diagonal elements of $\boldsymbol{G}$ produced by (a) algorithm [2], (b) algorithm [4], and (c) the proposed algorithm versus those of the initial $\boldsymbol{G}$.

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 10^5 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. Applications to orthogonal matching pursuit (OMP) systems were also examined with numerical results in favor of the proposed algorithm. The details are omitted here due to space limitation.

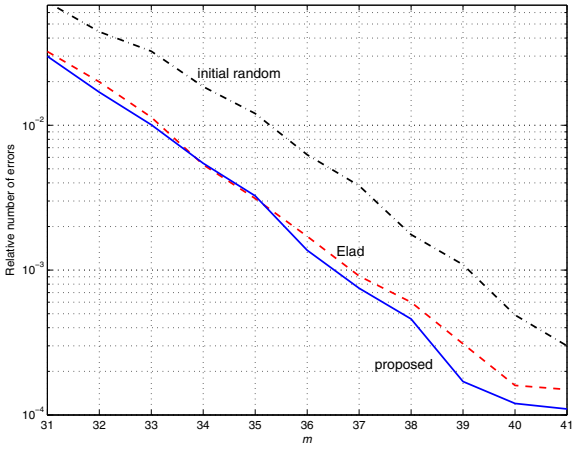

Fig. 3. Relative number of errors versus m for BP-based CS systems.

VI. Conclusions

A new algorithm for the design of optimal projection matrix for CS is proposed. The algorithm aims at directly minimizing mutual coherence of the CS system and this is achieved by a subgradient projection technique. Preliminary simulations are also reported to demonstrate the algorithm's performance in signal reconstruction.

References

[1] E. J. Candés and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazize*, 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 projectin 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.