Sublinear Compressive Sensing Reconstruction via Belief Propagation Decoding

Hoa V. Pham, Wei Dai, Olgica Milenkovic
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

***Abstract*—We propose a new compressive sensing scheme, based on codes of graphs, that allows for joint design of sensing matrices and low complexity reconstruction algorithms. The compressive sensing matrices can be shown to offer asymptotically optimal performance when used in combination with OMP methods. For more elaborate greedy reconstruction schemes, we propose a new family of list decoding and multiple-basis belief propagation algorithms. Our simulation results indicate that the proposed CS scheme offers good complexity-performance trade-offs for several classes of sparse signals.**

I. Introduction

Compressive sensing is a sampling technique usually applied to so-called compressible and/or K-sparse signals, i.e., signals that can be represented by $K \ll N$ significant coefficients over an N-dimensional basis. Sampling of a K-sparse, discrete-time signal $\mathbf{x}$ of dimension N is accomplished by computing a measurement vector **y** that consists of $m \ll N$ linear projections, i.e.,

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}.$$

Here, $\mathbf{\Phi}$ represents an $m \times N$ matrix, usually over the field of real numbers [1].

Although the reconstruction of the signal $\mathbf{x} \in \mathbb{R}^N$ from the possibly noisy random projections is an ill-posed problem, the prior knowledge of signal sparsity allows for recovering $\mathbf{x}$ using $m \ll N$ observations only. One of the outstanding results in CS theory is that the signal $\mathbf{x}$ can be reconstructed using optimization strategies aimed at finding the sparsest signal that matches the m projections. In other words, the reconstruction problem can be cast as an ℓ_0 minimization problem. It can be shown that to reconstruct a K-sparse signal $\mathbf{x}$, ℓ_0 minimization requires only $m = 2K$ random projections when the signal and the measurements are noise-free. Unfortunately, the ℓ_0 optimization problem is NP-hard. This issue has led to a large body of work in CS theory centered around the design of measurement and reconstruction algorithms with tractable reconstruction complexity.

The work by Donoho and Candès et. al. [1], [2] demonstrated that CS reconstruction is a polynomial time problem – albeit under the constraint that more than $2K$ measurements are used. The key observation behind these findings is that it is not necessary to resort to ℓ_0 optimization to recover $\mathbf{x}$ from the under-determined inverse problem; a much easier ℓ_1 optimization, based on Linear Programming (LP) techniques, yields an equivalent solution, as long as the sampling matrix $\mathbf{\Phi}$ satisfies the so called *restricted isometry property* (RIP) with a constant RIP parameter.

While LP techniques play an important role in designing computationally tractable CS decoders, their complexity is still highly impractical for many applications. In such cases, the need for faster reconstruction algorithms - preferably operating in linear time - is of critical importance, even if one has to increase the number of measurements. Several classes of low-complexity reconstruction techniques were recently put forward as alternatives to linear programming (LP) recovery, including group testing methods, pursuit strategies, as well as coding theoretic methods [3].

In this work, we focus our attention on reconstruction techniques with very low computational complexity, for which the basic steps are borrowed from the theory of iterative decoding on graphical models. We show how a simple modification of the well known loopy belief propagation (BP) decoding technique can produce a small-sized list of potential indices of the support of **x** that, with high probability, contains the exact support set of **x**. The BP algorithm operates on the columns of the matrix $\mathbf{\Phi}$ with dimension $m \times N$, and consequently, for $m \sim K \log (N/K)$ (which is a standard setup in compressive sensing theory), its reconstruction complexity is $O(m)$. This can be achieved only through a careful choice of the sensing matrix $\mathbf{\Phi}$ which allows both for extremely efficient computation of correlations between the sensing vector **y** and the columns of $\mathbf{\Phi}$, and which contains many near-orthogonal columns.

The paper is organized as follows. Section II provides a brief introduction to the main results in compressive sensing theory. Section III introduces our problem statement. Section IV includes the main results of this work: the description of a structured design approach for the compressive sensing matrix $\mathbf{\Phi}$, amenable to $O(K \log N)$ complexity decoding of very sparse vectors; and, a new "biased list decoding" framework for the BP algorithm. Section V includes the description of multiple-basis belief propagation algorithm for CS reconstruction. The driving applications for all the sensing scheme considered in the paper are for digital fingerprinting.

II. Compressive Sensing and the Restricted Isometry Property

Let $\mathbf{x}$ be a N-dimensional real-valued signal with at most K non-zero components, called a K-sparse signal. Let supp$(\mathbf{x})$ denote the set of indices of the non-zero coordinates of the vector $\mathbf{x} = (x_1, \ldots, x_N)$, and let $|\mathrm{supp}(\mathbf{x})| = \|\cdot\|_0$ denote the support size of $\mathbf{x}$, or equivalently, its ℓ_0 norm[1]. Assume next that $\mathbf{x} \in \mathbb{R}^N$ is an unknown signal with $|\mathrm{supp}(\mathbf{x})| \le K$, and let $\mathbf{y} \in \mathbb{R}^m$ equal $\mathbf{\Phi x}$, where $\mathbf{\Phi} \in \mathbb{R}^{m \times N}$ is henceforth referred to as the *sampling matrix*.

We are concerned with the problem of low-complexity recovery of the unknown signal $\mathbf{x}$ from the measurement $\mathbf{y}$. A natural formulation of the recovery problem is within an ℓ_0 norm minimization framework, which seeks a solution to the problem

$$\min \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{\Phi x}.$$

Unfortunately, the above ℓ_0 minimization problem is NP-hard, and hence cannot be used for practical applications [2].

One way to avoid using this computationally intractable formulation is to consider a ℓ_1-regularized optimization problem,

$$\min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = \mathbf{\Phi x},$$

where

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$$

denotes the ℓ_1 norm of the vector $\mathbf{x}$.

The reconstruction accuracy of the ℓ_1-LP method is described by the *restricted isometry property* (RIP), formally defined below.

Definition 1: A matrix $\mathbf{\Phi} \in \mathbb{R}^{m \times N}$ is said to satisfy the Restricted Isometry Property (RIP) with parameters (K, δ) for $K \le m$, $0 \le \delta \le 1$, if for all index sets $I \subset \{1, \cdots, N\}$ such that $|I| \le K$ and for all $\mathbf{q} \in \mathbb{R}^{|I|}$, one has

$$(1-\delta)\|\mathbf{q}\|_2^2 \le \|\mathbf{\Phi}_I \mathbf{q}\|_2^2 \le (1+\delta)\|\mathbf{q}\|_2^2, \tag{1}$$

where $\mathbf{\Phi}_I$ consists of the columns of $\mathbf{\Phi}$ with indices $i \in I$. We define δ_K, the RIP constant, as the infimum of all parameters δ for which the RIP holds, i.e.

$$\delta_K := \inf \Big\{\delta : (1-\delta)\|\mathbf{q}\|_2^2 \le \|\mathbf{\Phi}_I \mathbf{q}\|_2^2 \le (1+\delta)\|\mathbf{q}\|_2^2, \\ \forall |I| \le K, \ \forall \mathbf{q} \in \mathbb{R}^{|I|}\Big\}.$$

The best known sufficient condition for exact recovery is $\delta_{2K} < \sqrt{2} - 1$ [2], [4].

As one can notice from the discussion above, the major challenge associated with sparse signal reconstruction is to identify in which subspace, generated by not more than K columns of the matrix $\mathbf{\Phi}$, the measured signal lies in. Once the correct subspace is determined, the non-zero signal coefficients are calculated by applying the pseudoinversion process.

An alternative to ℓ_1 methods is the family of greedy algorithms, including Orthogonal Matching Pursuit (OMP), the Regularized OMP (ROMP), Stagewise OMP (StOMP), Subsapce Pursuit (SP) and the Compressive Sampling Matching Pursuit (CoSaMP) algorithms. The basic idea behind these methods is to find the support of the unknown signal sequentially. At each iteration of the algorithms, one or several coordinates of the vector $\mathbf{x}$ are selected for testing, based on the correlation values between the columns of $\mathbf{\Phi}$ and the regularized measurement vector. If deemed sufficiently reliable, the candidate column indices are used as current estimates of the support set of $\mathbf{x}$. The pursuit algorithms iterate this procedure until all the coordinates in the correct support set are included in the estimated support set, or until reconstruction failure is declared. The computational complexity of OMP strategies depends on the number of iterations needed for exact reconstruction: standard OMP always runs through K iterations, and therefore its reconstruction complexity is roughly $O(KmN)$. This complexity is significantly smaller than that of LP methods. However, the pursuit algorithms do not have provable reconstruction quality at the level of LP methods.

The important, and still unresolved question in CS theory pertains to the existence of low complexity algorithms for compressive sensing reconstruction and the existence of low-dimensional sensing matrices that can support such a decoding scheme. Restricting the choices for the sensing matrix necessarily introduces a performance loss so that one would like to investigate the inherent trade-off between the performance and complexity of such reconstruction algorithms. Preliminary results pertaining to these questions are presented in the following section.

III. Problem Statement

Throughout the remainder of the paper we will be concerned with the class of low density parity-check (LDPC) codes.

Consider a $[n, k, d_{\min}]$ LDPC code $\mathcal{C}$. We construct a $(m = n) \times (N = 2^k)$ sampling matrix $\mathbf{\Phi}$ in the following manner. First, we convert all codewords of $\mathcal{C}$ into their Binary Phase Shift Keying (BPSK) images, and then normalize each image by $1/\sqrt{n}$. These normalized codewords are used as columns of the matrix $\mathbf{\Phi}$. The columns of $\mathbf{\Phi}$ and their corresponding codewords will henceforth be denoted by $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_N$ and $\mathbf{c}_1, \ldots, \mathbf{c}_N$, respectively. Let $\mathbf{x} \in \mathbb{R}^N$ be a K-sparse signal, and let $\mathbf{y} \in \mathbb{R}^m$ be the CS observation vector. The maximum correlation between distinct columns of the matrix is denoted by

$\mu \triangleq \max_{i \ne j} |\langle \boldsymbol{\varphi}_i, \ \boldsymbol{\varphi}_j \rangle|$, where $\langle \boldsymbol{\varphi}_i, \ \boldsymbol{\varphi}_j \rangle = \sum_{l=1}^{m} \varphi_{i,l}\varphi_{j,l}$.

The parameter μ is called the incoherence parameter of the matrix $\mathbf{\Phi}$. Most performance guarantees of OMP algorithms are expressed in terms of this parameter. The standard OMP algorithm guarantees exact recovery of the vector $\mathbf{x}$ as long as $\mu \le 1/2K$ [5]. It is straightforward to express the incoherence parameter of the code-based matrix $\mathbf{\Phi}$ in terms the distance between codewords, as shown below:

$$\begin{aligned} \langle \boldsymbol{\varphi}_i, \ \boldsymbol{\varphi}_j \rangle &= \sum_{l=1}^{n} \varphi_{i,l} \ \varphi_{j,l} = \sum_{\substack{1 \le l \le n \\ \varphi_{i,l} = \varphi_{j,l}}} \frac{1}{n} + \sum_{\substack{1 \le l \le n \\ \varphi_{i,l} = \varphi_{j,l}}} \left(-\frac{1}{n}\right) \\ &= 1 - \frac{2 d_{\mathrm{H}}(c_i, c_j)}{n}. \end{aligned}$$

Here, $d_{\mathrm{H}}(c_i, c_j)$ denotes the Hamming distance between the codewords c_i and c_j. Note that if for all $i \ne j$,

$$\frac{d_H(c_i, c_j)}{n} > \frac{1}{2} - \frac{1}{4K}, \ \text{ then } \ \frac{n - 2d_H(c_i, c_j)}{n} < \frac{1}{2K},$$

[1] We interchangeably use both notations in the paper

and if

$$\frac{d_H(c_i,c_j)}{n} < \frac{1}{2} + \frac{1}{4K}, \text{ then, } \frac{n - 2d_H(c_i,c_j)}{n} > -\frac{1}{2K}.$$

Consequently, one has

$$\frac{d_H(c_i,c_j)}{n} \in \left(\frac{1}{2} - \frac{1}{4K}, \frac{1}{2} + \frac{1}{4K}\right) \Rightarrow |\langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle| < \frac{1}{2K}.$$

Hence, to guarantee exact recovery with the OMP algorithm for all K-sparse signals, we need to identify LDPC codes with

$$\frac{1}{2} - \frac{1}{4K} < \frac{d_H(c_i,c_j)}{n} < \frac{1}{2} + \frac{1}{4K}, \forall i \neq j.$$

That such codes indeed exist is shown in the proposition below. In the proof, we use the following ensemble of codes [6].

Ensemble E: The parity-check matrix $\mathbf{H}$ of the code is chosen with uniform probability from the ensemble of $(n-k) \times n$ (0,1)-matrices with row sums equal to d_c.

Proposition 1: Consider an LDPC code from the Ensemble E with $d_c \geq 3$. Let $K \in \mathbb{Z}^+$ and let $K, n, N = 2^k$ go to infinity with $n = O(K^2 \log N)$. Then, with high probability,

$$\frac{1}{2} - \frac{1}{4K} < \frac{d_H(c_i,c_j)}{n} < \frac{1}{2} + \frac{1}{4K}, \ \forall i \neq j.$$

To prove Proposition 1, we introduce the following terminology.

Let $\{\mathcal{C}_n\}$ be an ensemble of codes of length n defined by parity-check matrices of size $(n-k) \times n$. For a code $\mathcal{C} \in \mathcal{C}_n$, let

$$B_i(\mathcal{C}) = |\{\mathbf{c} \in \mathcal{C} : \text{wt}(\mathbf{c}) = i\}|, \quad i = 0, 1, \ldots, n,$$

where $\text{wt}(\cdot)$ denotes the Hamming weight. The average ensemble distance distribution is

$$\bar{B}(\mathcal{C}_n) = (\bar{B}_0(\mathcal{C}_n), \bar{B}_1(\mathcal{C}_n), \ldots, \bar{B}_n(\mathcal{C}_n)),$$

where

$$\bar{B}_i(\mathcal{C}_n) := \bar{B}_i = \frac{1}{|\mathcal{C}_n|} \sum_{\mathcal{C} \in \mathcal{C}_n} B_i(\mathcal{C}).$$

We use the following theorem from [6].

Theorem 1: Let $\alpha := (n-k)/k = 1 - R$. For $\theta \in (0,1)$, the average distance distributions are of the form

$$b_\theta := \lim_{n \to \infty} \frac{1}{n} \ln \bar{B}_{\theta n} = H(\theta) + p_\theta^\alpha,$$

where $H(\theta)$ denotes Shannon's entropy function. For ensemble E, one has

$$p_\theta^\alpha = \alpha \ln \left(\frac{1 + (1-2\theta)^{d_c}}{2} \right).$$

Proof of Proposition 1: We use the fact that if $b_\theta = \frac{1}{n} \ln \bar{B}_{n\theta} = c_0$, for some constant $c_0 < 0$, then

$$\frac{1}{n} \ln \Pr\{\mathcal{C} : B_{n\theta}(\mathcal{C}) \neq 0\} \leq \frac{1}{n} \ln \bar{B}_{n\theta} \leq c_0,$$

which means that with high probability, a code randomly chosen from the ensemble E has $d > \theta n$.

We want to prove that with high probability, $b_\theta < 0$ for any $\theta \notin \left(\frac{1}{2} - \frac{1}{4K}, \frac{1}{2} + \frac{1}{4K}\right)$.

Let $\theta^- = \frac{1}{2} - \frac{1}{4K}$ and choose $\epsilon > 0$ such that $\theta = \frac{1-\epsilon}{2}$. Then

$$\begin{aligned} H(\theta) &= \log 2 - \frac{1-\epsilon}{2} \log(1-\epsilon) - \frac{1+\epsilon}{2} \log(1+\epsilon) \\ &= \log 2 - \frac{1-\epsilon}{2} \left(-\epsilon + \frac{\epsilon^2}{2} + o(\epsilon^2) \right) \\ &\quad - \frac{1+\epsilon}{2} \left(\epsilon + \frac{\epsilon^2}{2} + o(\epsilon^2) \right) \\ &= \log 2 - \frac{\epsilon^2}{2} + o(\epsilon^2). \end{aligned} \tag{2}$$

With $\theta^- = \frac{1}{2} - \frac{1}{4K}$ or $\epsilon^- = \frac{1}{2K}$, (2) becomes $H(\theta^-) = \log 2 - \frac{1}{8K^2} + o\left(\frac{1}{K^2}\right)$. Note that $H(\theta) \leq H(\theta^-), \forall \theta < \theta^-$. Then, for $\alpha = 1 - R$,

$$\begin{aligned} b_\theta &\leq b_{\theta^-} \\ &= H(\theta^-) + p_{\theta^-}^\alpha \\ &= \log 2 - \frac{1}{8K^2} + o\left(\frac{1}{K^2}\right) - (1-R)\log 2 \\ &\quad + (1-R) \log \left(1 + \left(\frac{1}{2K}\right)^{d_c} \right) \\ &= R \log 2 - \frac{1}{8K^2} + o\left(\frac{1}{K^2}\right) + o\left(\frac{1}{K^{d_c}}\right). \end{aligned}$$

Similarly, one can prove that $\forall \theta > \theta^+ = \frac{1}{2} + \frac{1}{4K}$,

$$b_\theta \leq b_{\theta^+} = R \log 2 - \frac{1}{8K^2} + o\left(\frac{1}{K^2}\right) + o\left(\frac{1}{K^{d_c}}\right).$$

Therefore, if $R < \frac{1}{K^2} \frac{1}{8 \log 2}$, or equivalently, if $n > 8 \log(2) K^2 \log N$, then $\forall \theta \notin \left(\frac{1}{2} - \frac{1}{4K}, \frac{1}{2} + \frac{1}{4K}\right)$, $b_\theta < 0$, when K, n, N go to infinity. This proves the claimed result. ■

We can now bound the RIP parameter of the code-based matrix $\mathbf{\Phi}$. Applying the Gershgorin circle theorem [7] on the matrix $\mathbf{A} = \mathbf{\Phi}_I^* \mathbf{\Phi}_I$ shows that all the eigenvalues of $\mathbf{A}$ lie in a disc $D(1, r)$ centered at one, with radius r, where

$$r = \max_i \sum_{j \neq i} |\langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle| < K\mu.$$

Therefore, every eigenvalue λ of $\mathbf{A}$ satisfies $1 - K\mu < \lambda < 1 + K\mu$. Hence, if $\mu \leq 1/2K$, it is easy to see that the RIP parameter satisfies $\delta_K \leq 1/2$.

As already described, the main component of greedy reconstruction algorithms is the step of identifying one (or multiple) columns of $\mathbf{\Phi}$ that exhibit maximum correlation with the vector $\mathbf{y}$. Usually, such a step is implemented via exhaustive search, resulting in an algorithmic complexity of the order of N. The crux of the CS technique based on LDPC sensing matrices is that columns of maximum correlation correspond to the most likely codewords, and can hence be efficiently identified using iterative methods such as BP. It is important to point out that BP decoders in this context operate in a vastly different regime than standard decoders: they must be able to handle high *interference noise* and potentially identify *lists* of most likely codewords. These, and other issues associated with CS BP are discussed in the next sections.

IV. List Decoding BP and BP-OMP

In this section we focus our attention on BP decoders for $0-1$ vectors $\mathbf{x}$. The BP decoder has the purpose to identify the column of $\mathbf{\Phi}$ that maximizes the correlation with the measurement vector $\mathbf{y}$. In this setup, several observations are at place. First, when $K=2$, both columns of $\mathbf{\Phi}$ used in the superposition have the same correlation with $\mathbf{y}$. This is a particularly difficult setup for BP decoding, since the measurement vector $\mathbf{y}$ may fall in the middle of the decision regions for the two columns, but not converge to any of these two codewords. This motivates analyzing strategies for *biasing* the estimates for certain coordinates in $\mathbf{y}$ so as to move the vector away from the boundary region. How exactly biasing is performed will be described shortly. Second, whenever one tries to identify the column with largest correlation, all the remaining columns in the superposition act as interference. For large K, this interference may be very severe. Hence, the question of interest is to find an adequate model for the interference and investigate the performance of BP decoders for a very non-standard operational regime - namely, a regime significantly below the code's designed threshold.

We start with the description of the biasing scheme. For illustrational purposes, we first describe the scheme for the case $K=2$. We observe that for $K=2$, the entries of the measurement vector $\mathbf{y}$ must lie in the set $\{-2,0,2\}$. Whenever $y_i=+2$ or -2, both entries of the two columns of $\mathbf{\Phi}$ in the superposition must be +1, or -1, respectively. For zero entries of $\mathbf{y}$, one column must take the value -1 and the other must take the value +1 at that given coordinate. Therefore, one simple idea is to bias the ± 2 entries toward some large value, ideally $\pm\infty$, to ensure that the BP algorithm decodes the corresponding bits correctly. One can further bias one of the remaining zero entries of $\mathbf{y}$ either towards a large positive or negative value in order for the BP algorithm to decode it to +1 (or -1), respectively.

For the more general case, whenever $y_i=+K$ or $-K$, one can bias the corresponding entries towards some large positive or negative value, respectively. When $y_i\neq\pm K$, at least one of the columns must have the value +1 and at least one other column must have the value -1 at that coordinate. Since biasing different entries of $\mathbf{y}$ may produce different BP decoder outputs, one actually obtains a list of potential columns, which has to be expurgated from vectors that correspond to words obtained by divergent BP methods and vectors that represent repetitions. If the output list has more than K codewords, one outputs only K codewords from the list that have largest correlations with $\mathbf{y}$. Otherwise, if the list has less than K entries, one outputs the whole list. The algorithm is summarized in *Algorithm 1*.

The performance of the BP list decoder can be assessed in several different ways. One way is to declare reconstruction success if at *least one* of the columns of $\mathbf{\Phi}$ has been identified correctly. This type of measure is of special importance in digital fingerprinting applications. The second measure can be the probability of identifying *all* columns of $\mathbf{\Phi}$ within the given superposition. The larger the size of the list, the larger the probability of finding the correct codewords in the list. However, increasing the list size also means larger computational complexity, since one has to run the BP algorithm for each element in the list.

As we stated earlier in this section, whenever we want to identify a column φ_k in the superposition, the remaining columns act as interference. In our scheme, we approximate the interference, $\Delta=\sum_{j\neq k}\varphi_j$, by a zero-mean Gaussian random variable with variance $\sigma_k^2=\sum_{j\neq k}\varphi_j^2=K-1$. Furthermore, to provide an additional random bias (as opposed to the above described deterministic bias), we add a small AWGN noise to $\mathbf{y}$, with SNR $=30$ dB. The list decoding BP algorithm is summarized below.

Algorithm 1 List-based BP Algorithm

Input: K, $\mathbf{H}$, $\mathbf{y}$, biasing list size L, biasing value B.

- Construct the list $\mathbf{RX}=(\mathbf{rx}^{(1)},\mathbf{rx}^{(2)},\dots,\mathbf{rx}^{(L)})$, where
  - Set $\mathbf{rx}^{(1)}=\mathbf{y}$
  - Set $\mathbf{rx}^{(2)}$ is the vector $\mathbf{y}$ with entries $\pm K$ being biased to $\pm B$
  - Randomly generate an index set $\mathcal{J}$ of size $(L-2)/2$, where $rx_j\neq\pm K,\forall j\in\mathcal{J}$.
  - For each $j\in\mathcal{J}$, bias the j^{th} coordinate of the vector $\mathbf{rx}^{(2)}$ to $+B$ or $-B$ to get $\mathbf{rx}^{(2l+1)}$ and $\mathbf{rx}^{(2l+2)}$, respectively, $(l=1,2,\dots,(L-1)/2)$.
- Run the BP algorithm for $\mathbf{rx}^{(i)}$ and output a binary word $\hat{\mathbf{v}}_i$ $(i=1,\dots,L)$. Delete all the words which do not satisfy $\mathbf{H}.\hat{\mathbf{v}}_i=0$ and delete all the repetitions in the list.
- Output at most K words $\hat{\mathbf{v}}_i$ whose BPSK images have largest correlations with $\mathbf{y}$.

Output: The list of potential columns of $\mathbf{\Phi}$ in the superposition.

One can combine the list decoding BP algorithm with standard OMP, as summarized below.

Algorithm 2 BP-OMP Algorithm

Input: K, $\mathbf{H}$, $\mathbf{y}$, L, B.

Initialization: $\mathbf{rx}=\mathbf{y}$.

Iteration: Run the iteration K times

- Run the list-based BP algorithm and output a list of potential columns
- Pick the codeword $\mathbf{v}$ whose BPSK image has largest correlation with $\mathbf{rx}$
- Update: $K=K-1$, $\mathbf{rx}=\mathbf{y}-\text{BPSK}(\mathbf{v})$

Output: The list of K columns of $\mathbf{\Phi}$ in the superposition.

It is straightforward to see that the computational complexity of this algorithm is $O(KLn)=O(KL\log N)$. For fixed L, the complexity equals $O(K\log N)$.

Rather than using standard OMP, one can adaptively add or remove column candidates from the estimated list of codewords. One algorithm that uses this idea, termed Subspace Pursuit (SP), can be easily modified to include the list-based BP decoding strategy as its correlation maximization step to reduce its overall complexity (for more details regarding the SP algorithm, the interested reader is refered to [8]). At each

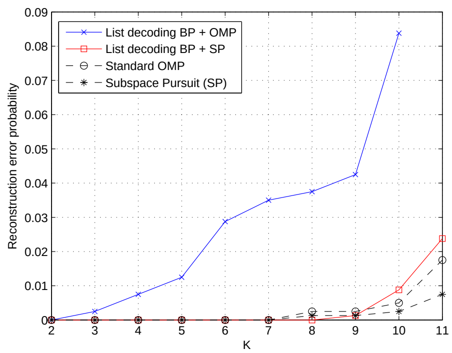

Figure 1. Simulation results list-based BP algorithm. Number of samples = 800

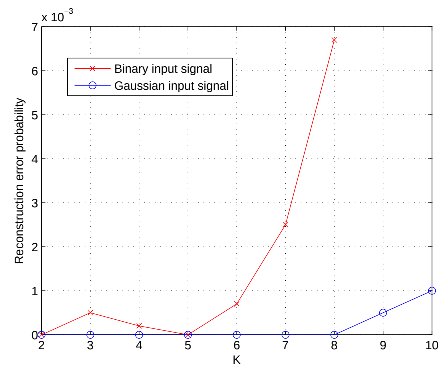

Figure 2. Reconstruction error probability for MBBP algorithm. Number of samples = 10000.

iteration of the SP algorithm, if the output list from the list-based BP decoder does not contain at least K codewords, we randomly choose columns of $\mathbf{\Phi}$ to fill up the list.

We presents some initial simulation results for the proposed algorithm. In our simulation, the parity-check matrix $\mathbf{H}$ is generated randomly with column degree three and 80% of the rows having degree three and 20% of the rows having degree four. These matrices perform comparably to row-regular matrices from ensemble E. The simulation results show that these parity check matrix performs very poorly even when K is small. However, its systematic form (a parity-check matrix of the form $\mathbf{H}_s = [\mathbf{I}|\mathbf{P}]$, where $\mathbf{I}$ denotes the identity matrix of size $(n-k)$), performs much better. This suggests that in the high interference regime, the systematic parity-check matrix works better than the non-systematic one. This may confirm the recently described results [9] that vaguely state that in very low SNR regimes, "bad" LDPC codes outperform "good" ones.

Figure 1 illustrates the performance of the BP-OMP algorithm, the BP-SP algorithm, along with that of standard OMP and SP algorithms for a systematic parity-check matrix of size 135×144 and with the list size $L = 10$. The reconstruction error probability is defined as the probability of not finding all the columns in the output list. BP-OMP algorithms are sub-optimal in terms of identifying codewords with largest correlation, and exhibit slow convergence, which does not make them amenable for this type of CS application. BP-SP techniques, as expected, perform very well.

V. Multiple Basis Belief Propagation (MBBP) Algorithm

The biasing methods of section IV can not be used for vectors $\mathbf{x}$ that are non-binary. To mitigate this problem, we propose a new scheme, termed a multiple-basis belief propagation (MBBP) algorithm, which uses several parity-check matrices in parallel to identify columns of $\mathbf{\Phi}$ that have largest correlation with $\mathbf{y}$. The idea behind the CS MBBP method is that parity-check matrices with different degree distributions may perform differently for the same interference.

In our simulation, we choose to model the interference as a zero-mean Gaussian variable with the variance $\sigma_k^2 = \max(|\mathbf{y}|)(K-1)/K$. The steps of the CS MBBP algorithm are summarized below.

Algorithm 3 MBBP Algorithm

Input: K, $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_p$, $\mathbf{y}$.

- Compute the interference $N = \max(|\mathbf{y}|)\frac{K-1}{K}$.
- For $i = 1$ to p: Run the standard BP algorithm for received signal $\mathbf{y}$ using matrix $\mathbf{H}_i$, output a word $\mathbf{v}_i$ if the algorithm converges.
- Delete all the codewords which do not satisfy $\mathbf{H}.\hat{\mathbf{v}}_i = 0$.
- Output the codeword whose BPSK image has the highest correlation with the received vector.

Output: One potential column of $\mathbf{\Phi}$ in the superposition.

Figure 2 show the simulation results of the MBBP algorithm for both binary and Gaussian signals. In this simulation, we used 4 parity-check matrices ($p = 4$): one of them is a randomly generated matrix $\mathbf{H}$ of size 150×160, with column degree three and row degree three and four, while the other three matrices are systematic forms of $\mathbf{H}$, obtained from Gaussian elimination starting at different columns. The reconstruction error probability is defined as the probability of not finding any correct columns in the output list.

References

[1] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. Candès and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[3] D. Baron, S. Sarvotham, and R. Baraniuk, "Bayesian compressive sensing via belief propagation." *Preprint*, 2008.

[4] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, 2008.

[5] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[6] S. Litsyn and V. Shevelev, "On ensembles of low-density parity-check codes: asymptotic distance distributions," *IEEE Trans. Information Theory*, vol. 48, no. 4, 2002.

[7] R. S. Varga, *Geršgorin and His Circles*. Berlin: Springer-Verlag, 2004.

[8] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing: Closing the gap between performance and complexity," *Preprint*, 2008.

[9] R. Calderbank, S. Howard, and S. Jafarpour, "Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property." *Preprint*, 2008.