Iterative Reconstruction Algorithms in Compressed Sensing

Ludovic Danjean[♯,†], Vida Ravanmehr[†], David Declercq[♯], Bane Vasić[†]

[†] Dept. of Electrical and Computer Eng.
University of Arizona
Tucson, AZ 85721, USA
Email: {vravanmehr,vasic}@ece.arizona.edu

[♯] ETIS
ENSEA / Univ. Cergy-Pontoise / CNRS
F-95000 Cergy-Pontoise, France
Email: {danjean,declercq}@ensea.fr

***Abstract*—In this paper we give an overview of current results in iterative reconstruction of sparse signals using parity check matrices of low-density parity check (LDPC) codes as measurement matrices in compressed sensing. We provide a detailed explanation of two iterative reconstruction algorithms, Interval Passing (IP) algorithm and verification algorithm. We then compare their performance using parity check matrices of quasi-cyclic low-density parity check (QC-LDPC) codes with different column-weights and rates.**

***Index Terms*—compressed sensing, iterative reconstruction algorithm, low-density parity check codes.**

I. Introduction

Compressed Sensing (CS) is a relatively new field in signal processing which concerns the recovery of a sparse high dimension signal $x \in \mathbb{R}^n$ from a small set of measurements $y = Ax \ \in \mathbb{R}^m$ where $A_{m\times n}$ is the measurement matrix and $m \ll n$ [1], [2]. Finding the sparsest signal satisfying $y = Ax$ (l_0-norm minimization) is NP-hard. To solve this problem, different algorithms including Linear Programing (LP) techniques and iterative reconstruction algorithms were proposed. LP techniques provide a l_1-norm minimization solution for reconstruction of x and show a remarkable performance. However, their cubic complexity in n makes them impractical in some applications in which fast reconstruction is needed or in which the dimension of the measurement matrix A is large. A class of iterative reconstruction algorithms including Orthogonal Matching Pursuit (OMP) [3] and Stage-wise Orthogonal Matching Pursuit (StOMP) [4] were introduced to reconstruct x. StOMP is much faster than both OMP and LP and consequently is more convenient in large-scale problems. The work by Dimakis *et al.* [5] showed that a good parity check matrix of a binary low-density parity check (LDPC) code [6] is a good zero-one measurement matrix for compressed sensing under LP. With this connection between channel coding and compressed sensing, different iterative reconstruction algorithms for compressed sensing based on message passing algorithms of LDPC codes were introduced. Sarvotham *et al.* [7] introduced an iterative algorithm called CS-LDPC based on Belief Propagation (BP) algorithm and using LDPC-like measurement matrices with entries in $\{-1, 0, 1\}$ to reconstruct *approximately* sparse signals.

Two iterative reconstruction algorithms, Approximate Message Passing (AMP) by Donoho *et al.* [8] and the verification decoding algorithm by Zhang and Pfister [9] have the flavor of the bit flipping algorithm.

A simple message passing algorithm was introduced by Chander *et al.* [10] for reconstruction of non-negative real signals x. This algorithm, which is called Interval Passing (IP) algorithm and was later modified by Krishnan *et al.* [11], has low-complexity and running time and exhibits a good performance for reconstruction of non-negative sparse signals.

In this paper, we consider the recovery of non-negative real signals with two iterative reconstruction algorithms, IP and verification decoding algorithms and provide some simulation results to compare the performance of these reconstruction algorithms. In Section II, we give a brief review on compressed sensing reconstruction algorithms. In Section III, we compare the performance of IP and verification decoding algorithms using the parity check matrices of QC-LDPC codes. Section IV provides possible directions for future work.

II. Preliminaries and Prior Work

In this section, we first give a brief introduction on compressed sensing problem and provide a review on CS reconstruction algorithms.

A. Compressed Sensing

Let $x \in \mathbb{R}^n$ be an n-dimension k-sparse signal i.e. a signal with at most k non-zero elements and let $A_{m\times n}$ be the measurement matrix. Compressed sensing concerns with the recovery of x from measurements $y = Ax \in \mathbb{R}^m$ where $m \ll n$ and $k \ll n$. The first approach to recover x from the measurements y, is to find a k-sparse signal from all $\binom{n}{k}$ possible signals which is an NP-hard problem. Instead, a linear programming technique, called Basis Pursuit, provides a minimum l_1-norm for x. Candes *et al.* [12] showed that if the measurement matrix A satisfies the Restricted Isometry Property (RIP), Basis Pursuit can recover a sufficiently sparse signals.

A matrix A is said to satisfy the RIP, if for every $\omega \in \mathbb{R}^n$ such that $\parallel \omega \parallel_0 \leq k$,

$$(1-c) \parallel \omega \parallel_2^2 \;\; \leq \;\; \parallel A\omega \parallel_2^2 \;\; \leq \;\; (1+c) \parallel \omega \parallel_2^2$$

for some $c \geq 0$, where $\parallel . \parallel_0$ and $\parallel . \parallel_2$ are l_0-norm and l_2-norm, respectively.

Dimakis *et al.* [13] provided a necessary and sufficient condition for a good CS measurement matrix, called Null Space Property (NSP), which is defined as follows. Let V be a set of columns of the measurement matrix A. A satisfies NSP if for a $k \in \mathbb{Z}_{\geq 0}$, $c \in \mathbb{R}_{\geq 0}$,

$$c\parallel z_I \parallel_1 \leq \parallel z_{I'} \parallel_1 \;\; \forall\, z \in \text{NSpace}(A),\; \forall I \subseteq V,\; |I| \leq k$$

where $\parallel . \parallel_1$ is l_1-norm, $I' = V \backslash I$ and $\text{NSpace}(A) = \{z \in \mathbb{R}^n : Az = 0\}$.

In [14], we provide an overview on iterative CS reconstruction algorithms. Here, we give a summary on some iterative CS reconstruction algorithms including AMP, IP, verification and CS-LDPC algorithms with details on IP and verification decoding algorithms which are used in our simulations. Iterative reconstruction algorithms are usually described using the graphical representation of the measurement matrices, called Tanner graph [15]. A Tanner graph is a bipartite graph with two disjoint sets of nodes: variable nodes and check nodes. Each variable (resp. check) node represents a column (resp. row) of A. A check node i is connected to a variable node j iff $A_{ij} \neq 0$.

Example 1: The Tanner graph corresponding to the following measurement matrix is shown in Fig. 1.

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

In the figure, the variable nodes $\{v_1, v_2, ..., v_8\}$ are represented by circles, and the check nodes $\{c_1, c_2, c_3, c_4\}$ are represented by squares.

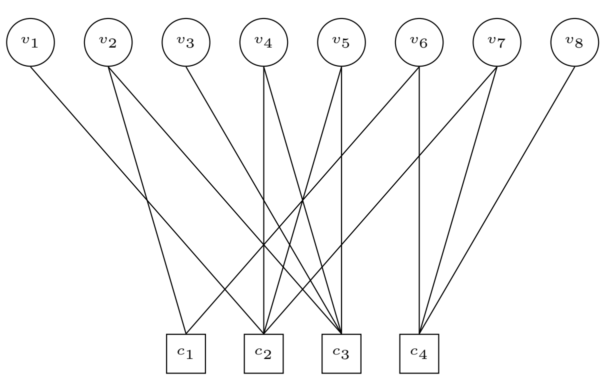

Fig. 1. The Tanner graph corresponding to the measurement matrix A in Example 1.

B. Approximate Message Passing (AMP)

An iterative thresholding algorithm called AMP was introduced Donoho *et al.* [8]. In this algorithm $x^{(l)}$ and $z^{(l)}$ are the estimates of the solution x and the residue at l^{th} iteration, respectively. Starting from $x^{(0)} = 0$ and $z^{(0)} = y$, the messages are updated as follows:

$$\begin{aligned} x^{(l+1)} &= \eta_l(A^T z^{(l)} + x^{(l)}) \\ z^{(l)} &= y - Ax^{(l)} + \frac{1}{\delta} z^{(l-1)} \langle \eta'_{l-1}(A^T z^{(l-1)} + x^{(l-1)}) \rangle, \end{aligned}$$

where η_l is a sequence of threshold functions (applied componentwise), usually given by

$$\eta(x; \lambda) = \text{sgn}(x)(|x| - \lambda)_+$$

where the subscript $(u)_+ = u\mathbb{I}(u \geq (0))$, and $\mathbb{I}(.)$ is equal to one if the Boolean expression in its argument is true and zero elsewhere. $\eta'(u) = \partial\eta(u)/\partial u$ and $\langle u \rangle$ computes the mean of $u = [u_1, u_2, ..., u_n]$, i.e. $\langle u \rangle = (1/n)\sum_{1 \leq i \leq n} u_i$, and A^T is the transpose of A [8], [14].

C. CS-LDPC

A message passing CS algorithm based on LDPC-like matrices was proposed by Sarvotham *et al.* to reconstruct approximately sparse signals [7]. The elements of the measurement matrix A are in $\{-1, 0, 1\}$. The approximately is modeled using two-state Gaussian mixture distribution and the reconstruction algorithm is based on Belief Propagation (BP) (for more details the reader is referred to [7]). The complexity of this algorithm is $O(n \log n)$ and the minimum number of measurements is $O(k \log n)$.

D. Interval Passing (IP)

Interval Passing (IP) algorithm [10] is a simple message passing algorithm for reconstructing non-negative signals using binary measurement matrices. The complexity of algorithm is $O(n(\log(\frac{n}{k}))^2 \log(k))$ and the number of measurements is $m = O(k \log(\frac{n}{k}))$. Krishnan *et al.* [11] modified the IP and showed that the algorithm fails if the non-zero values of the signal x contain a stopping set [11]. In this algorithm, the messages passing through edges are intervals $[\mu, M]$.

Let $V = \{v_1, v_2, ..., v_n\}$ and $C = \{c_1, c_2, ..., c_m\}$ be the sets of variable nodes and check nodes, respectively. At each iteration l, messages from the variable v_i to the check node c_j are updated as follows [11], [14]:

$$\mu_{v_i \to c_j}^{(l)} = \max_{c'_j \in \mathcal{N}(v_i)} \left(\mu_{c'_j \to v_i}^{(l-1)} \right) \quad (1)$$

$$M_{v_i \to c_j}^{(l)} = \min_{c'_j \in \mathcal{N}(v_i)} \left(M_{c'_j \to v_i}^{(l-1)} \right) \quad (2)$$

and the messages from the check node c_j to the variable node v_i are updated as:

$$\mu_{c_j \to v_i}^{(l)} = \max\left\{ 0, y_j - \sum_{v'_i \in \mathcal{N}(c_j) \backslash \{v_i\}} M_{v'_i \to c_j}^{(l)} \right\} \quad (3)$$

$$M_{c_j \to v_i}^{(l)} = y_j - \sum_{v'_i \in \mathcal{N}(c_j) \backslash \{v_i\}} \mu_{v'_i \to c_j}^{(l)} \quad (4)$$

where $\mathcal{N}(v_i)$ (resp. $\mathcal{N}(c_j)$) is the set of check (resp. variable) nodes which are the neighbors of v_i (resp. c_j) in the Tanner graph. Updating messages from a variable (check) node to a check (variable) node are shown in Fig. 2 and 3, respectively.

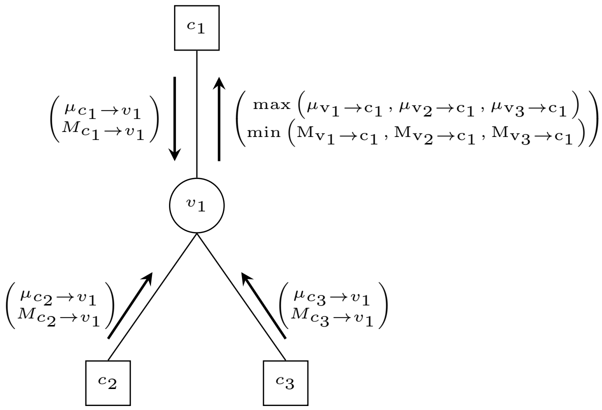

Fig. 2. IP algorithm: Updating messages from the variable node v_1 to the check node c_1.

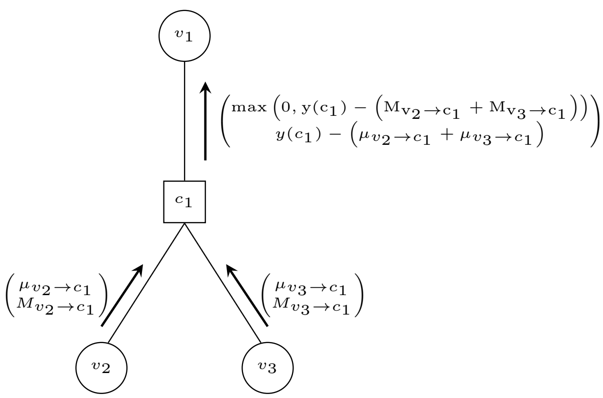

Fig. 3. IP algorithm: Updating messages from the check node c_1 to the variable node v_1.

Formally, the IP algorithm is given in Algorithm 1 where L represents the maximum number of decoding iterations. The decoding stops either if the maximum of iteration is reached, or if the lower bound and the upper bound of the interval from variable nodes to check nodes has converged to a common value for every variable node.

Algorithm 1: Interval Passing Algorithm

Input : y and A such that $y = Ax$, L.
Output : $\hat{x}$ the estimate of x.
Initialization: $\forall c_j \in C, \forall v_i \in \mathcal{N}(c_j)$, $\mu^{(0)}_{c_j \to v_i} = 0$ and $M^{(0)}_{c_j \to v_i} = y(c_j)$;
for $l = 1$ *to* L **do**
1 **foreach** $v_i \in V$ **do**
  foreach $c_j \in \mathcal{N}(v_i)$ **do**
    $\mu^{(l)}_{v_i \to c_j} = \max_{c'_j \in \mathcal{N}(v_i)} (\mu^{(l-1)}_{c'_j \to v_i})$;
    $M^{(l)}_{v_i \to c_j} = \min_{c'_j \in \mathcal{N}(v_i)} (M^{(l-1)}_{c'_j \to v_i})$;
2 **foreach** $c_j \in C$ **do**
  foreach $v_i \in \mathcal{N}(c_j)$ **do**
    $\mu^{(l)}_{c_j \to v_i} =$
    $\max\left\{0, y(c_j) - \sum_{v'_i \in \mathcal{N}(c_j) \setminus \{v_i\}} M^{(l)}_{v'_i \to c_j}\right\}$;
    $M^{(l)}_{c_j \to v_i} = y(c_j) - \sum_{v'_i \in \mathcal{N}(c_j) \setminus \{v_i\}} \mu^{(l)}_{v'_i \to c_j}$;
3 **for** $v_i \in V$ **do**
  if ($l > 1$ & $\mu^{(l)}_{v_i \to \mathcal{N}(v_i)} = M^{(l)}_{\mathcal{N}(v_i) \to v_i}$) $\|$ $l = L$ **then**
    $\hat{x}(v_i) = \mu^{(l)}_{v_i \to \mathcal{N}(v_i)}$;
    Output $\hat{x}$

E. Verification Algorithm

Zhang and Pfister's verification algorithm [9] is an iterative algorithm to reconstruct strictly sparse signals. The algorithm can be considered as a bit flipping algorithm as introduced in [16], since the messages correspond to the vertices (variable and check nodes in the factor graph) and not to the edges. This algorithm is summarized in the following steps and is described in the Algorithm 2. In this algorithm each variable node can have two states; one *unverified state* [*] (no value has yet been estimated), and one *verified state* (the variable node has been estimated).

Step 1. Variable nodes which are the neighbors of zero-value check nodes are verified as 0.

Step 2. Variable nodes connected to check nodes with degree one (only one edge connected) are verified as the value of the check node.

Step 3. A single variable node connected to two check nodes with the same measurement value is verified to the common value of the check nodes.

Step 4. Subtract the values of the verified variable nodes from the neighboring check nodes and then remove all verified variable nodes and edges connected to them.

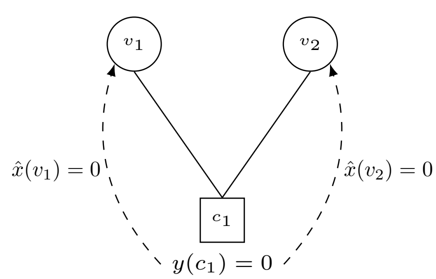

Fig. 4. Verification decoding algorithm: First Step

Steps 1 to 4 are repeated until reconstruction succeeds or no more progress can be applied. The Fig. 4 and 5 sketch

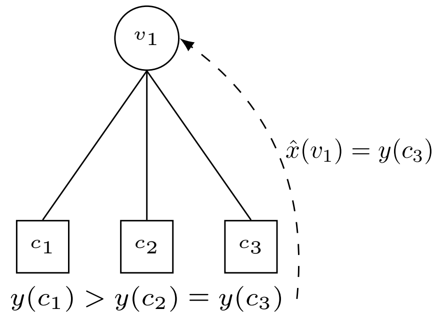

Fig. 5. Verification decoding algorithm: Third Step

the Steps 1 and 3. Note that the verification decoding was originally introduced for the q-ary Symmetric Channel [17] . The justification of Step 3 is based on the observation that, over large alphabets, the probability that two independent random numbers are equal (or equivalently two independent measurements) is quite small. This leads to consider that any two common measured values (during decoding) are generated by the same set of non-zero data. This observation holds for large alphabets and then for the real numbers too. Note that in the Algorithm 2, we added one condition to the Step 3 (Fig. 5) to avoid that at one iteration, the value on one check node becomes negative. In the Algorithm 2, L is maximum number of iterations for the decoding, and δ is a precision threshold to check the correct recovery of our signal (practically $\delta \sim 10^{-3}$).

Algorithm 2: Verification Decoding

Input : y and A such that $y = Ax$, L, δ.
Output: $\hat{x}$ the estimate of x.
$l = 0$;
$\hat{x} = [*, \dots, *]$;
while $\mathbb{E}\left[|\hat{x} - x|^2\right] > \delta$ *or* $l < L$ **do**

1 **foreach** c_j *such that* $y(c_j) = 0$ **do**
    $\hat{x}\left(\mathcal{N}(c_j)\right) = 0$;

2 **foreach** c_j *such that* $CheckDegree(c_j) = 1$ **do**
    $\hat{x}\left(\mathcal{N}(c_j)\right) = y(c_j)$;
    $y\left(\mathcal{N}(v_i)\right) = y\left(\mathcal{N}(v_i)\right) - y(c_j)$ with $v_i = \mathcal{N}(c_j)$;

3 **foreach** (c_j, c_k) *such that* $|\mathcal{N}(c_j) \cap \mathcal{N}(c_k)| = 1$ *and* $y(c_j) = y(c_k)$ **do**
    $v_i = \mathcal{N}(c_j) \cap \mathcal{N}(c_k)$;
    if $y\left(\mathcal{N}(v_i)\right) - y(c_j) \geq 0$ **then**
        $\hat{x}(v_i) = y(c_j)$;
        $y\left(\mathcal{N}(v_i)\right) = y(\mathcal{N}(v_i)) - y(c_j)$;

4 **foreach** v_i *such that* $\hat{x}(v_i)$ *is verified* **do**
    $A\left(\mathcal{N}(v_i), v_i\right) = 0$;

  $l = l + 1$;

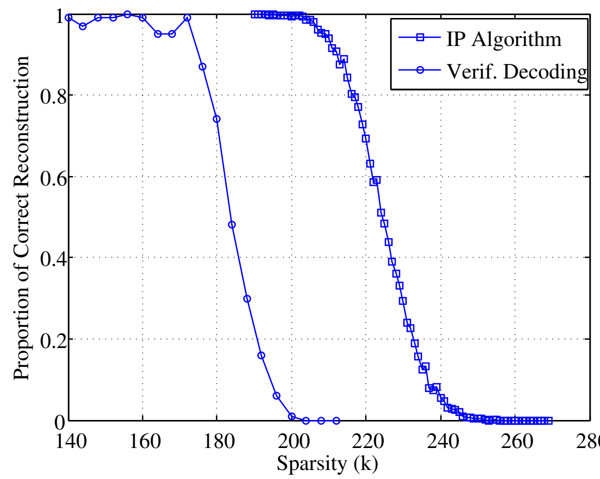

Fig. 6. Reconstruction of a k-sparse signal with a QC-LDPC measurement matrix of length $n = 600$, $d_v = 4$, $R = 0.2$

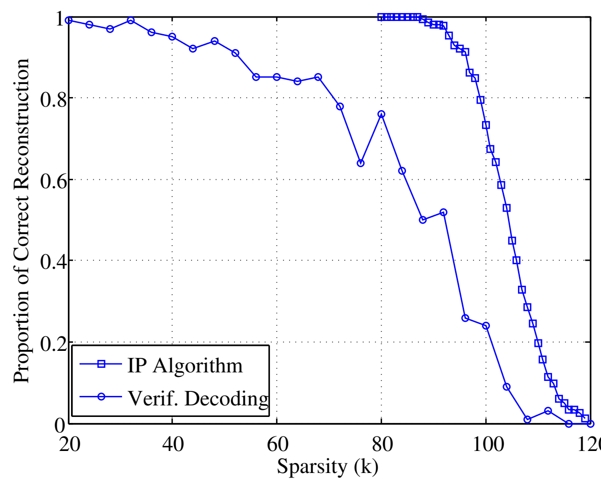

Fig. 7. Reconstruction of a k-sparse signal with a QC-LDPC measurement matrix of length $n = 600$, $d_v = 4$, $R = 0.5$

III. Simulation Results

In this section, we provide simulation results on CS reconstruction with IP and verification decoding algorithms using QC-LDPC measurement matrices of length $n = 600$ and column-weights $d_v = 4$ and $d_v = 5$ for two different rates (R) in each cases. Considering different rates is equivalent to take different number of measurements. For both algorithms we realized 50 decoding iterations, and for each sparsity we detected the number of vectors that cannot be fully recovered among 100 randomly generated sparse vectors. We present the results in terms of the Proportion of Correction Reconstruction versus the sparsity of our signal.

As can be seen in Figures 6 to 9, the performance of the IP algorithm is better than the verification decoding algorithm. However, for the measurement matrices of rate $R = 0.5$ (Figures 7 and 9), both algorithms fail in recovery of the signals at almost the same sparsity parameter although the slope of the IP algorithm is much higher (in absolute value) than the verification decoding.

For measurement matrices of rates $R = 0.2$ and $R = 0.17$, the verification decoding algorithm fails on recovery of signals with sparsity parameter much less than that of IP algorithm. Indeed, the proportion of correct reconstruction is decreasing at a higher sparsity for the IP algorithm that the verification decoding. Another point to notice is the influence of the column-weight (or equivalently the connection degree on the variable nodes) with a constant rate. As expected the performance of these reconstruction algorithms decrease as we increase the column-weight of the measurement matrix. Indeed when we increase the column-weight of the measurement matrix, this will increase the number of edges (constraints) on the measurement matrix, and then the linear dependency between samples of the signal on the measurement

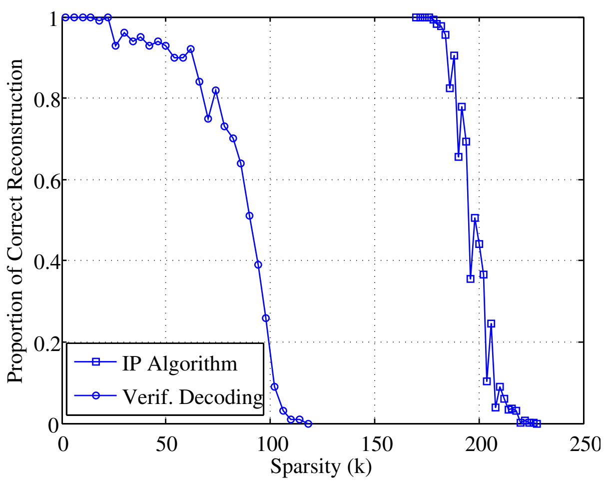

Fig. 8. Reconstruction of a k-sparse signal with a QC-LDPC measurement matrix of length $n = 600$, $d_v = 5$, $R = 0.17$

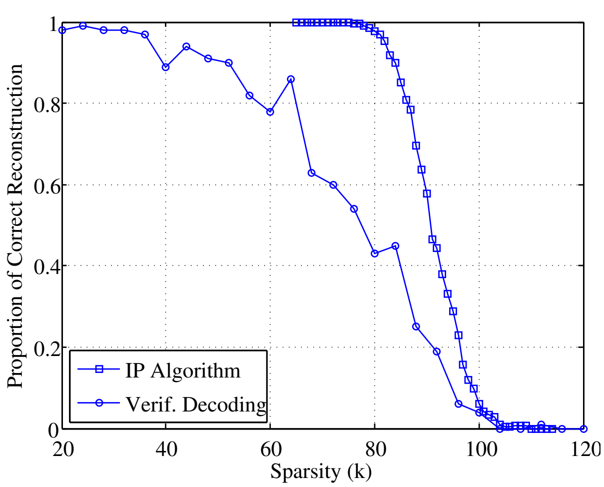

Fig. 9. Reconstruction of a k-sparse signal with a QC-LDPC measurement matrix of length $n = 600$, $d_v = 5$, $R = 0.5$

IV. Future Work

Future work will include the analysis of convergence of IP and finding the sparsest vectors for which the IP and the verification algorithms fail. Analysis similar to the idea of trapping sets [18] or instanton search algorithms [19] should give us understanding of the algorithms in order to modify them to obtain better sparse signal recovery performance.

V. Acknowledgments

This work is funded by the NSF under the grant CCF-0830245, and DARPA KeCom Program under the grant CCF-0963726.

References

[1] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
[2] R. J. Candes, E. and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[3] J. Tropp and A. Gilbert, “Signal recovery from partial information via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
[4] D. Donoho, Y. Tsaig, I. Drori, and J.-C. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit,” Mar. 2006, preprint.
[5] A. G. Dimakis and P. Vontobel, “Lp decoding meets lp decoding: a connection between channel coding and compressed sensing,” in *Proc. 47th Allerton Conference on Communications, Control, and Computing*, Monticello, Illinois, USA, Sep. 30-Oct.2 2009.
[6] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: M.I.T. Press, 1963.
[7] S. Sarvotham, D. Baron, and R. G. Baraniuk, “Compressed sensing reconstruction via belief propagation,” Tech. Rep., 2006.
[8] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing,” *Proc. of National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.
[9] F. Zhang and H. D. Pfister, “On the iterative decoding of high-rate ldpc codes with applications in compressed sensing,” *CoRR*, vol. abs/0903.2232, 2009.
[10] V. Chandar, D. Shah, and G. Wornell, “A simple message-passing algorithm for compressed sensing,” in *Proc. IEEE International Symposium on Information Theory (ISIT 2010)*, June 2010, pp. 1968 –1972.
[11] A. R. Krishnan, S. Sankararaman, and B. Vasic, “Graph-based iterative reconstruction of sparse signals for compressed sensing,” in *Proc. of TELSIKS Conference*, October 2011, Nis, Serbia.
[12] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
[13] A. G. Dimakis, R. Smarandache, and P. O. Vontobel, “Ldpc codes for compressed sensing,” *CoRR*, vol. abs/1012.0602, 2010.
[14] V. Ravanmehr, L. Danjean, D. David, and B. Vasic, “On iterative compressed sensing reconstruction of sparse non-negative vectors,” in *Proc. of ISABEL'2011*, October 2011, Barcelona, Spain.
[15] R. Tanner, “A recursive approach to low complexity codes,” *Information Theory, IEEE Transactions on*, vol. 27, no. 5, pp. 533–547, 1981.
[16] M. Sipser and D. Spielman, “Expander codes,” *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.
[17] F. Zhang and H. Pfister, “Compressed sensing and linear codes over real numbers,” in *Information Theory and Applications Workshop (ITW'08*, Feb. 2008, pp. 558 –561.
[18] S. K. Chilappagari, D. V. Nguyen, B. V. Vasic, and M. W. Marcellin, “On trapping sets and guaranteed error correction capability of ldpc codes and gldpc codes,” *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1600–1611, 2010.
[19] S. K. Chilappagari, M. Chertkov, and B. Vasic, “Worst configurations (instantons) for compressed sensing over reals: a channel coding approach,” 2010, *IEEE International Symposium on Information Theory (ISIT 2010)*. [Online]. Available: arXiv:1001.5113v2[cs.IT]