Sparse or Dense – Message Passing (MP) or Approximate Message Passing (AMP) for Compressed Sensing Signal Recovery

Asad Mahmood[1], Jaewook Kang[2], Jr. *Member IEEE*, and HeungNo Lee[2] *Senior Member IEEE,*

[1]School of Computer Science, University of Witwatersrand, Johannesburg, South Africa
[2]INFONET lab, Gwangju Institute of Science and Technology (GIST), Gwangju, South-Korea

***Abstract*—Compressed Sensing (CS) is one of the hottest topics in signal processing these days and the design of efficient recovery algorithms is a key research challenge in CS. Whereas, a large number of recovery algorithms have been proposed in literature, the recently proposed *Approximate Message Passing* (AMP) [19] algorithm has gained a lot of attention because of its good performance and yet simple structure. Although *Belief Propagation* (BP) algorithms were previously considered to give good performance only in *Sparse* graphs, AMP algorithm is based on the application of BP on dense graphs. The application of BP in dense graphs asks for a re-look on the design of BP algorithms over graphs which can have a bearing on many applications including coding theory, neural networks etc. This paper aims to compare different existing variants of *Message Passing* algorithms on sparse and dense graphs for the CS recovery problem.**

***Index Terms*—Compressed Sensing, Bipartite graph, Belief Propagation, Sparse Matrices, Channel Coding**

I. Introduction

Compressed Sensing (CS) is a novel paradigm in signal acquisition and processing recently proposed by Candes and Donoho [1-3]. It has completely changed the way of how signals were classically acquired and processed. Compressed Sensing replaces the *classical* two-step signal processing philosophy of 1) signal sampling (Nyquist-rate) and 2) signal compression with a single step of 'compressed/low-rate *sampling*' (sub-Nyquist sampling) without having a performance loss, subject to the constraint that the signal to be recovered is *sufficiently sparse*. The idea of CS has already been applied to a number of applications [6-10]. If **x** is our target K-sparse signal of dimension N×1 which we want to compressively acquire. The phenomenon of CS can then be represented as the following set of linear equations.

$$y_{(m\times 1)} = A_{(m\times N)} x_{(N\times 1)} \qquad (1)$$

where **A** is an m×N dimensional matrix, called as the *Sensing matrix*, representing the linear combinations of compressed sensing and **y** is the vector of resultant samples of size m×1 commonly referred to as *measurements*.

A large number of recovery algorithms have also been proposed in literature having their origins in diverse fields and areas e.g. convex optimization [2-4], linear programming [2-4], bases decomposition [23], combinatorial methods [24], iterative shrinkage [25], graphical models [16] etc.

One such recovery algorithm recently proposed by Maleki and Donoho [19] and known as the AMP (Approximate Message Passing) algorithm has shown to achieve very good performance with large reduction in complexity in comparison with existing approaches. AMP has its roots in *Graphical Modeling* concept. *Graphical Modeling* is a popular technique and is being used extensively for modeling and decision-making in a variety of applications including Inference problems, error correction codes and learning Algorithms [14] etc. Whereas *Sparse Graphs/Matrices* have been classically used for signal inference/estimate, AMP advocates the use of *dense graphs* for the signal recovery problem in CS using the Message-Passing (MP) algorithms. This is of particular interest as previously, except for a few works [15], *Message-Passing* algorithms were generally considered to perform poorer as the sparsity of the underlying graph was reduced. A fundamental understanding of the AMP algorithm can thus lead to its application in a variety of fields. This article not only aims to provide the connection of the final form of AMP with the underlying dense graph and its associated *Message-Passing* algorithm but the comparison with associated approaches for *Sparse graphs* will also be explored and the merits and de-merits of the different approaches will be outlined. Section II will present the algorithms for signal recovery in dense graphs, section III will provide the different algorithms for sparse graphs, section IV will provide the simulation results while conclusion will be given in section V.

II. Signal recovery via Dense Graphs

A. Sum-Product MP Algorithm for CS Signal Recovery

Classically, when MAP estimate is desired of the unknown signal, the *Sum-Product (SP)* [12] variant of the MP algorithm is used. As the AMP algorithm is derived from the classical MP algorithm, the first step is to formulate a *Sum-Product*

978-1-4799-1501-9/13/$31.00 ©2013 IEEE

algorithm that *aims* to find the MAP estimate of the unknown **x** in the compressed sensing framework **y**=A**x** as given in eq. 1. This requires maximizing the *A-Posterior-Probability* (*APP)* p(**x**|**y**) which is equivalent to maximizing the joint distribution p(**x**,**y**) by the application of Bayes rule, given in the following equation.

$$\hat{x}_i^{MAP} = \arg\max_{x_i} p(x_i \mid \mathbf{y}) = \arg\max_{x_i} \frac{p(x_i, \mathbf{y})}{p(\mathbf{y})} \sim \arg\max_{x_i} p(x_i, \mathbf{y}) \quad (2)$$

We will equivalently represent p(**x**,**y**) as $p_{\mathbf{y}}(\mathbf{x})$ to indicate the joint distribution of all the unknown variables **x** parameterized by the *known* vector **y.** Thus the problem boils down to maximizing the joint distribution of the unknown vector **x.** The concept of *'postulated prior distribution'* is used here to *postulate* a *desired prior distribution* on the unknown vector **x** which *can* be different than the '*actual prior distribution'* of the signal **x** but nonetheless is used to promote sparsity in solution.

The performance of the resultant SP algorithm heavily relies on the *goodness* of the postulated prior distribution. One such *postulated prior distribution* which has several nice characteristics such as it models well the sparsity in the signal, its distribution function factors perfectly with respect to a factor-graph [13] etc. is given below.

$$p_{\mathbf{y}}(\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{N}\exp(-\beta|x_i|)\prod_{j=1}^{M}\delta(y_j = A_j\mathbf{x}) \quad (3)$$

Where Z is a normalization constant to make the probability sum equal to unity and A_j represents the jth row of matrix A which corresponds to the measurement y_j. There are two distinct components of this distribution; 1) the negative exponential component parameterized with the parameter β. It can be observed that this is a sparsity promoting/enforcing component whose combined[1] value will get smaller as the signal becomes less sparse i.e. more non-zero values. 2) the dirac delta distribution δ(.) which is only non-zero when the solution satisfies the constraint **y**=A**x.** Thus the product in eq. 3a will assign *weights* to the solutions of the linear system y=Ax that will decay exponentially with the l1 norm of the solutions.

The marginals of the distribution given in eq. 3 will give the APP distributions of individual x_i which can be individually maximized to find a symbol-by-symbol MAP estimate of the unknown sparse-signal **x**. SP algorithm is a low-complexity alternative to find the marginals of such functions which factorizes over a factor-graph. First we represent the CS framework **y**=A**x** via a factor-graph/bi-partite graph, like the one shown in figure below, with **y**, **x** and A being represented by the *measurement-nodes*, *variable-nodes* and the *edges* respectively.

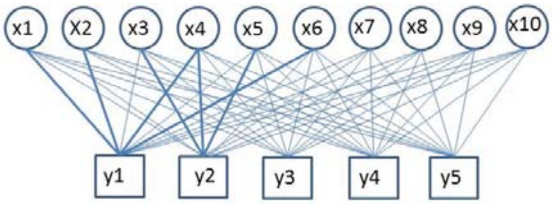

Fig1: Bi-partite graph corresponding to a *dense* adjacency matrix

We can observe the joint distribution function given in eq. 3 is a product of N+M *factors* and thus factorizes perfectly with respect to such a factor/bi-partite graph. The SP algorithm for calculating the marginals of the function given in eq. 3 on the associated bi-partite graph is given in table 1.

The presence of an 'integral' in eq. 5 instead of a 'summation' is because the unknown values x_i are generally considered to be continuous in the CS framework.

Table 1 – SP MP Algorithm

At Each Iteration:

1. **At Each Variable-Node:**
   Variable (x_i)-to-Measurement (y_j) Node message[2] =
   $$m_{x_i \to y_j} = \exp(-\beta|x_i|)\cdot\prod_{j' \in \{n(x_i)\backslash y_j\}} m_{y_{j'} \to x_i} \quad (4)$$
2. **At Each Measurement-Node:**
   Measurement (y_j) –to-Variable (x_i) Node message
   $$\hat{m}_{y_j \to x_i} = \int_{\mathbf{x}\backslash x_i} \delta(y_j = A_j\mathbf{x}) \prod_{x_{i'} \in \{n(y_j)\backslash x_i\}} m_{x_{i'} \to y_j} \quad (5)$$

B. Approximate Parameter-Passing Algorithm for CS Signal Recovery

The SP MP algorithm given in table 1 is of no practical use because it involves passing over a large no. of messages in each iteration (dense graph), where the messages are density functions over the real line which makes it computationally prohibitive. Fortunately, however, the situation gets simplified when certain approximations which are valid *only* in *large*

[1] The product of all N factors

[2] Although in many application of SP algorithm e.g. in LDPC decoding, the variable node to measurement node message is only a product of all incoming *extrinsic* messages, but here they are also multiplied with the corresponding variable *local function* i.e. the negative exponential component because the objective function/distribution is factorized in this manner

system limit[3] simplify the MP algorithm without compromising in performance. Thus we can *simplify*/*approximate* the messages $m_{x_i \to y_j}$ and $\hat{m}_{y_j \to x_i}$ in the SP MP algorithm by *assuming* they belong to a *particular structure/distribution*. This makes it possible to *track* the behavior of these messages as instead of keeping track of all the values of all the probability densities (each message will be a density function in this case as explained previously), we will only be required to keep *track* of the *parameters* of these *representative* probability density functions e.g. mean and variance for a Gaussian distribution.

For large system limit, it was shown that the measurement-to-variable node messages $\hat{m}_{y_j \to x_i}$ can be approximated by Gaussian distributions while the variable-to-measurement node messages $m_{x_i \to y_j}$ can be approximated by a distribution which is a product of Gaussian and Laplace distributions. The proof of these approximations is beyond the scope of this paper and we refer the interested reader to [22]. The resultant algorithm can be termed as the *Parameter-Passing (PP)* MP algorithm, and is given in the table 2 below

Table 2 – PP MP Algorithm

At Each Iteration:

1. **At Each Variable-Node:**
Variable (x_i)-to-Measurement (y_j) Node message =

$$\mu_{x_i \to y_j} \overset{\lim_{\beta \to \infty}}{\square} \sum_{j' \neq j} {}_{,}{}'_{i} \hat{\mu}_{y_{j'} \to x_i}, \hat{\tau}^t) \qquad (6)$$

2. **At Each Measurement-Node:**
Measurement (y_j) –to-Variable (x_i) Node message

$$\hat{\mu}_{y_j \to x_i} \overset{\lim_{\beta \to \infty}}{\square} {}_{,} \sum_{i' \neq i} A_{ji'} \mu_{x_{i'} \to y_j} \qquad (7)$$

3. **Variance Update once per iteration (Same for all the nodes)**

$$\hat{\tau}^{t+1} \overset{\lim_{\beta \to \infty}}{\square} \left(\frac{\hat{\tau}^t}{N\delta} \right) \sum_i \eta' \left(\sum_j A_{ji.} \hat{\mu}_{y_j \to x_i}, \hat{\tau}^t \right) \qquad (8)$$

Where $\mu_{x_i \to y_j}$ and $\hat{\mu}_{y_j \to x_i}$ represent the *mean* of the variable-to-measurement node and measurement-to-variable node messages/densities respectively, as given in eq. 4 and 5, and $\hat{\tau}^t$ is a parameter which *represents* the variance of the PP MP algorithm messages in iteration *t*. The calculation of these *parameters/messages* have been simplified via use of the *soft-thresholding* function $\eta(x,\theta)$ which is defined for a variable *x* w.r.t a threshold θ as follows

$$\eta(x;\theta) = \begin{cases} x - \theta & \text{if } x > \theta \\ 0 & \text{if } -\theta \le x \le \theta \\ x + \theta & \text{if } x < -\theta \end{cases} \qquad (9)$$

And $\eta'(x,\theta)$ is defined as the derivative of the soft thresholding function

$$\eta'(x;\theta) = \begin{cases} 1 & \text{if } -\theta > x > \theta \\ 0 & \text{if } -\theta \le x \le \theta \end{cases} \qquad (10)$$

C. Approximate Message-Passing (AMP) Algorithm for CS Signal Recovery

Although the *Parameter-Passing* algorithm given in table 2 has less complexity than the classical SP MP algorithm of table 1, it still involves updating 2MN messages per iteration which can result in huge complexity for large N. Fortunately, via application of first order Taylor approximations [22], the *PP MP* algorithm can be reduced to the *Approximate Message Passing* (AMP) algorithm, as given below

Table 3 – AMP Algorithm

At Each Iteration:

1. **At Each Variable-Node:**

$$\mu_i = \eta(\left(\sum_{j'=1}^{M} A_{j'i} \hat{\mu}_{j'} + \mu_i \right), \hat{\tau}^t)$$

In vector notation

$$\boldsymbol{\mu}^{t+1} = \eta(\left(\mathbf{A}^T \hat{\boldsymbol{\mu}}^t + \boldsymbol{\mu}^t \right), \hat{\tau}^t) \qquad (11)$$

2. **At Each Measurement-Node:**

$$\hat{\mu}_j = y_j - \sum_{i'=1}^{N} A_{ji'} \mu_{i'} - \frac{\hat{\mu}_j}{\delta} \left\langle \eta' \left(\left(\sum_{j'=1}^{M} A_{j'i} \hat{\mu}_{j'} + \sum_{j'=1}^{M} A_{j'i} \delta \hat{\mu}_{j' \to i} \right), \hat{\tau}^t \right) \right\rangle$$

In vector notation

$$\hat{\boldsymbol{\mu}}^t = \mathbf{y} - \mathbf{A}\boldsymbol{\mu}^t - \underbrace{\frac{\hat{\boldsymbol{\mu}}^{t-1}}{\delta} \left\langle \eta' \left(\left(\mathbf{A}^T \hat{\boldsymbol{\mu}}^{t-1} + \boldsymbol{\mu}^{t-1} \right), \hat{\tau}^t \right) \right\rangle}_{\text{Onsager Term}} \qquad (12)$$

3. **Variance Update once per iteration (Same for all the nodes)**

$$\hat{\tau}^{t+1} = \left(\frac{\hat{\tau}^t}{\delta} \right) \left\langle \eta' \left(\left(\sum_{j'=1}^{M} A_{j'i} \hat{\mu}_{j'} + \mu_i \right), \hat{\tau}^t \right) \right\rangle \qquad (13)$$

Where δ is defined as *M*/*N* and $\langle . \rangle$ is the averaging operator[4]. The *interesting* thing to observe in the above AMP algorithm is that the *same* message is sent from a given

[3] Large system limit assumes large values for M and N and is a valid assumption for CS framework as many applications involve thousands and possibly millions of variables

[4] $\langle . \rangle$ for a vector **x** of length N is defined as $\langle \mathbf{x} \rangle = \frac{\sum_{i=1}^{N} x_i}{N}$

variable/measurement node to its neighbors and hence this results in reducing the total no. of messages from 2MN to M+N thereby greatly reducing the complexity. The *Onsager Term* is what makes the performance of AMP algorithms superior to other [21] iterative thresholding algorithms and have been named so because its links with similar sort of terms in *statistical mechanics*[19]

III. Signal recovery via sparse Graphs

The nature of the MP algorithms proposed for CS Signal recovery via use of Sparse graphs is very *similar in nature* to the different forms of Message-Passing and Parameter-Passing algorithms we saw in the last section for dense matrices/graphs. The reason for this is that the goal in this/sparse case remains the same i.e. to iteratively update and finally maximize the posterior density $p(x_i \mid \mathbf{y})$ by exchanging probability messages across the *edges* of the bi-partite graph given by measurement/adjacency matrix **A**.

In the classical [2] *l1*-magic approach for CS signal recovery, the shape of the l_1-ball plays a role to pursue the sparse solution. In the MP approaches, the *postulated* prior density $p_{\mathrm{y}}(\mathbf{x})$ is chosen so as to enforce the *sparsity* in recovered vector. Hence, proper choice of prior density significantly affects the sparse recovery, as was the case for dense matrices as well. Thus, several types of prior densities have been experimented in different MP-based algorithms for signal recovery in CS, as will be mentioned below.

The first application of MP algorithms for the CS signal recovery came in the form of *Density Passing* (DP-Sparse) algorithms over Sparse graphs, initially proposed by Baron et al [16]. The idea was simple i.e. to pass the *samples* of the *density* of the *unknown variables* $\mathbf{x_i}$ as messages along the edges of the associated bi-partite graph and then to *update* the densities via application of the SP MP update rules. It is important to mention here that although the *DP-Sparse* algorithm is very similar in nature to the famous *Density-Evolution (DE)* approach, the *DE* approach was classically used for determining the *performance* of the LDPC codes, while the *DP-Sparse* algorithm is used as a *practical signal-recovery algorithm* in this case of CS Signal recovery.

IV. Simulation results and analysis

We compare performance of the algorithm associated with *sparse graphs* i.e. the *DP-Sparse* CS- BSD algorithm with another recently proposed sparse-graph based algorithm [17]. It is derived by combining SP MP algorithm and Expectation Maximization (EM) algorithm, and thus we denote it by *EM-Sparse*. Like *PP-MP*, in *EM*-Sparse only parameters of the concerned distributions are passed as messages in the MP framework which results is less complexity. The performances of the algorithms associated with *dense graphs*, i.e. the *PP-MP* algorithm and the *AMP* algorithm are also evaluated in terms of their MSE convergence rate with respect to iterations. The parameters were kept uniform for all the algorithms i.e. N=1000, M=500, K= 50,100,150 and 200 and all the performances of all the algorithms were evaluated in the noiseless setting.

Table 4 – *DP-Sparse* Algorithm

At Each Iteration:

1. **At Each Variable-Node:**

Variable (x_i)-to-Measurement (y_j) Node message =

$$m_{x_i \to y_j} = \prod_{j' \in \{n(x_i) \setminus y_j\}} m_{y_{j'} \to x_i}(s)\Big|_{s \in \text{Uniform Samples}} \qquad (14)$$

2. **At Each Measurement-Node:**

Measurement (y_j) –to-Variable (x_i) Node message

$$\begin{aligned} \hat{m}_{y_j \to x_i} &= \int_{\mathbf{x} \setminus x_i} \delta(y_j = A_j \mathbf{x}) \prod_{x_{i'} \in \{n(y_j) \setminus x_i\}} m_{x_{i'} \to y_j} \\ &= \sum_{\mathbf{x} \setminus x_i} \delta(y_j = A_j \mathbf{x}) \prod_{x_{i'} \in \{n(y_j) \setminus x_i\}} m_{x_{i'} \to y_j}(s)\Big|_{s \in \text{uniform samples}} \\ &= \delta(y_j = A_j \mathbf{x}) \otimes \left(\bigotimes_{i' : a_{ji'} \neq 0, i' \neq i} \mathrm{F}\left(m_{x_{i'} \to y_j}(s)\Big|_{s \in \text{uniform samples}} \right) \right) \end{aligned} \qquad (15)$$

Where $|_s$ indicates the uniform samples of the continuous distribution, and $\mathrm{F}(.)$ and $\otimes$ represent the Fourier and convolution operator respectively

Amongst the *sparse graph* based algorithms, it can be observed from figure 2 and 3 that although the MSE convergence rate for the *EM-Sparse* algorithm is better than the *DP-Sparse* algorithm for the cases with high sparsity i.e. K= 50 and K=100, the *EM-Sparse* algorithm failed to converge when the sparsity is decreased to K=150 and K=200. In terms of complexity, the *DP-Sparse* algorithm has significantly higher complexity than the *EM-Sparse* algorithm due to the simple fact that the messages in the former case are the vectors containing the samples of the message densities while in the latter (*EM-Sparse*) case, the messages consist only of the parameters of a given distribution.

Amongst the *dense graph* based algorithms, it can be observed from figure 4 and 5 that the MSE convergence rate with respect to iterations for both the *PP-MP* and *AMP* algorithm is approximately same for all levels of sparsity. This is of great interest because *AMP* algorithm is an *approximation* of the *PP-MP* algorithm and even for such moderate lengths as in our case of N=1000, this *approximation* is very valid. In terms of complexity, there is a huge difference between the complexity of the *AMP* algorithm and the *PP-MP* algorithm. This can be understood from the fact that the number of messages in an iteration of the *PP-MP* algorithm are on the order of *O(2MN)* in comparison with the order of messages of *O(M+N)* for the AMP algorithm. When comparing *AMP* algorithm with the sparse-graph based approach *DP-Sparse*, it is observed to give superior performance with much reduced complexity especially for the cases with reduced sparsity.

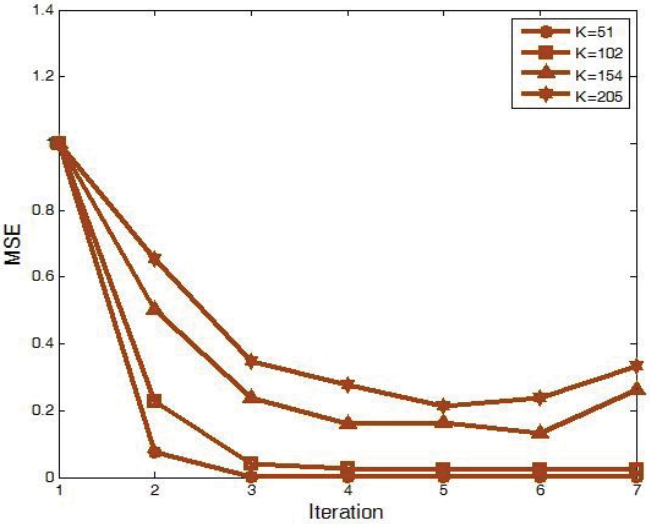

Figure 2: MSE convergence for *DP-Sparse* [16] Algorithm

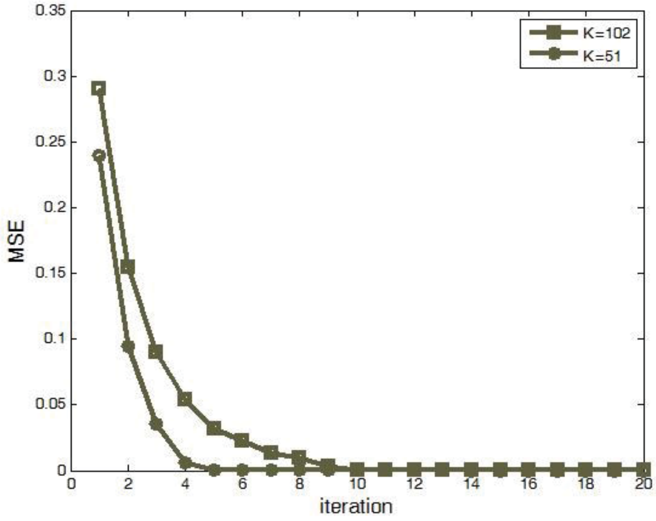

Figure 3: MSE convergence for *EM-Sparse* [17] Algorithm

AMP algorithm thus emerges as winner amongst the plethora of *Message-Passing* giving the best performance-to-complexity tradeoff for all values of signal sparsity.

V. Conclusion

In this paper, we have reviewed, analyzed and compared different variants of the Message-Passing algorithms for the Compressed Sensing signal recovery problem. Special emphasis has been put to the recently proposed Approximate Message Passing algorithm which goes against the commonly accepted wisdom and apply Belief Propagation / Message-Passing algorithms on *dense* graphs for Compressed Sensing signal recovery problem. The algorithms based on dense-graphs were compared with those proposed over Sparse graphs and the advantages and disadvantages of different approaches, along with the performance and complexity comparison, were analyzed in this paper.

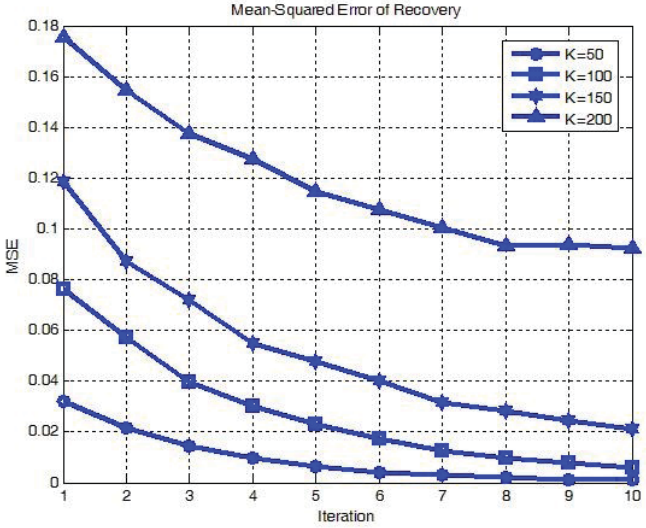

Fig 4: MSE Convergence for the *AMP* [19] algorithm for *dense graphs*

`

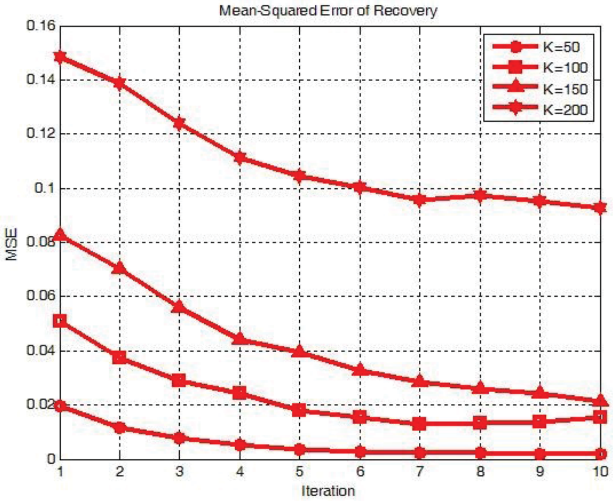

Fig 5: MSE Convergence for the *PP* algorithm [22] for *dense* graphs

VI. Acknowledgement

This work was supported by the National Research Foundation (NRF) of Korea grant funded by the Korea Government (MEST) (Do-Yak Research Program, NO. 2013-035295). A. Mahmood was also supported by CHPC (Centre for High Performance Computing) South-Africa flagship project: Computational Research Initiative in Imaging and Remote Sensing.

VII. References

[1] E. Candes, J. Romberg and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information” *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489-509, 2006

[2] D. L. Donoho, “Compressed sensing” *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, Sept. 2006.

[3] E. Candes, J. Romberg and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements” *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207-1223, 2006.

[4] E. Candes and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. Inform. Theory*, 52(12):5406-5425, 2006.

[5] J. Tropp, M. Wakin, M. Duarte, D. Baron and R. Baraniuk, “Random filters for compressive sampling and reconstruction”, Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse, France, May 2006.

[6] M. Lustig, D. Donoho and J.M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182 - 1195, December 2007

[7] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly and R. Baraniuk, “Single-pixel imaging via compressive sampling*”, IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83-91, March 2008

[8] J. Oliver, W.B. Lee, S.J. Park and H.N. Lee, "Improving resolution of miniature spectrometers by exploiting sparse nature of signals," *Optics Express*, vol. 20, no. 3, pp. 2613-2625, 2012.

[9] S. Hu, M. Lustig, A. P. Chen, J. Crane, A. Kerr, D.A.C. Kelley, R. Hurd, J. Kurhanewicz, S.J. Nelsona, J.M. Pauly and D.B. Vigneron, “Compressed sensing for resolution enhancement of hyperpolarized 13C flyback 3D-MRSI”, *Journal of Magnetic Resonance*, vol. 192, no. 2, pp. 258 - 264, June 2008.

[10] W. L. Chan, K. Charan, D. Takhar, K. F. Kelly, R. G. Baraniuk, and D. M. Mittleman, “A single-pixel terahertz imaging system based on compressive sensing” *Applied Physics Letters*, vol. 93, 2008.

[11] J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference” *Morgan Kaufmann* San Mateo, CA, 1988.

[12] N. Wiberg, “Codes and Decoding on General Graphs” *PhD dissertation*, Linkoping University, 1996.

[13] F.R. Kschischang, B.J. Frey and H.A. Loeliger, “Factor Graphs and the Sum-Product Algorithm” *IEEE Transactions on Information theory*, vol. 47, pp. 498-519, Feb 2001.

[14] D.J.C. Mackay, “Good error-correcting codes based on very sparse matrices*”, IEEE Trans. on Info. Theory*, vol. 45, no. 2, March 1999.

[15] T. Tanaka, "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors" *IEEE Trans. Info. Theory*, vol. 48, pp. 2888 - 2910, 2002.

[16] D. Baron, S. Sarvoham, R.G. Baraniuk, “Bayesian compressive sensing via belief propagation*”, IEEE Trans. Signal Processing*, vol. 58, no. 1,pp. 269-280, 2009.

[17] M. Akcakaya, J. Park and V. Tarokh, "A coding theory approach to noisy compressive sensing using low density frame" *IEEE Trans. Signal Processing*, vol. 59, no. 12, pp. 5369-5379, Nov. 2011.

[18] X. Tan and J. Li, "Computationally Efficient Sparse Bayesian Learning via Belief Propagation" *IEEE Trans. Signal Processing*, vol. 58, no. 4, pp. 2010-2021, Apr. 2010.

[19] D. L. Donoho, A. Maleki and A. Montanari, "Message Passing Algorithms for Compressed Sensing," Proc. Natl. Acad. Sci. 2009.

[20] J. Kang, H.N. Lee and K. Kim,” On detection-directed estimation approach for noisy compressive sensing,” 2012 [Online]. Available: arXiv:1201.3915v3 [cs.IT].

[21] D. Donoho, “Denoising by soft-thresholding” *IEEE Trans. Inform. Theory*, vol. 41, no. 3, pp. 613-627, 1995.

[22] A. Maleki, “Approximate Message Passing Algorithms”, *PhD Thesis,* Stanford Univ., 2011.

[23] D. Needell and J. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” Applied and Computational Harmonic Analysis, June 2008.

[24] G. Cormode and S. Muthukrishnan, ”Combinatorial algorithms for Compressed Sensing”. In Proc. 40th Ann. Conf. Information Sciences and Systems, Princeton, Mar. 2006.

[25] T. Blumensath and M. Davies, “Iterative hard thresholding for compressed sensing,” Applied and Computational Harmonic Analysis, vol. 27, no. 3,. pp. 265–274 2009.