Sparse Binary Matrices of LDPC codes for Compressed Sensing

Weizhi Lu, Kidiyo Kpalma and Joseph Ronsin

Université Européenne de Bretagne, France
INSA de Rennes, IETR, UMR 6164, F-35708, RENNES
{*weizhi.lu, kidiyo.kpalma, joseph.ronsin*}*@insa-rennes.fr*

Abstract: Compressed sensing shows that one undetermined measurement matrix can losslessly compress sparse signals if this matrix satisfies Restricted Isometry Property (RIP). However, in practice there are still no explicit approaches to construct such matrices. Gaussian matrices and Fourier matrices are first proved satisfying RIP with high probabilities. Recently, sparse random binary matrices with lower computation load also expose comparable performance with Gaussian matrices. But they are all constructed randomly, and unstable in orthogonality. In this paper, inspired by these observations, we propose to construct structured sparse binary matrices which are stable in orthogonality. The solution lies in the algorithms that construct parity-check matrices of low-density parity-check (LDPC) codes. Experiments verify that proposed matrices significantly outperform aforementioned three types of matrices. And significantly, for this type of matrices with a given size, the optimal matrix for compressed sensing can be approximated and constructed according to some rules.

1. Introduction

Compressed sensing [1]-[4] shows that a sparse vector $\mathbf{x}$ with k nonzero entries can be losslessly compressed with an undermined measurement matrix $\mathbf{A} \in \mathbf{R}^{m \times n}$, where $k \ll n$ and $m \ll n$, supposing that matrix $\mathbf{A}$ satisfies RIP [1]. This result attracts substantial interest for its dominance in the field of high dimension reduction. Generally, there are two interesting study points for compressed sensing. One is to recover or approximate sparse vector $\mathbf{x}$, and the other one is to construct undetermined measurement matrix $\mathbf{A}$ satisfying RIP. For the former, recently it has been widely studied as a regression problem with l_1 norm penalty

$$\min\|\mathbf{x}\|_1 \quad \text{s.t. } \|\mathbf{y}\text{-}\mathbf{Ax}\|_2 < \varepsilon \tag{1}$$

which can be solved by linear programming. Moreover, other efficient algorithms [13]-[15] are also proposed successively for lower computation load. But for the latter, the explicit construction of undetermined matrices satisfying RIP, remains elusive [5, 6]. As we know, Fourier matrices [2] and Gaussian matrices [4] are first proved satisfying RIP with high probabilities. However, Gaussian matrices include significant computation load due to their randomness and density. Fourier matrices are competitive on computation but suffer from weaker orthogonality than Gaussian matrices. Furthermore, for above two types of matrices, there is no explicit way to derive their optimal orthogonality.

Recently, in terms of a weaker RIP [6, 7], sparse random matrices [7] based on expander codes [8] display comparable performance with Gaussian matrices. This is a

valuable result due to its significant reduction in computation load without performance loss. But random construction still means that the orthogonality of such matrices is
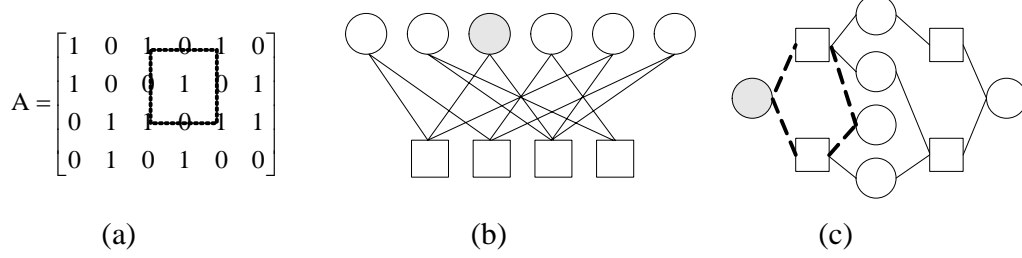
$$A = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

| (a) | (b) | (c) |

Figure 1: Parity-check matrix of LDPC codes is represented by Tanner graph

uncontrolled and weak. Further, it can't optimize the sparsity of one matrix with a given size. In other words, there is no uniform method to determine the smallest number of nonzero entries for matrices with different sizes. But this is critical in practice. Therefore, inspired by this observation, we turn to one type of structured sparse matrices, parity-check matrix of LDPC codes [9], whose orthogonality can be controlled and measured to some extent during their construction.

As the following section details, the parity-check matrices of LDPC codes are also one type of sparse binary matrices, but they suffer from much stricter constraints on orthogonality due to the special construction conditions that LDPC codes require. Here this type of matrices is constructed for compressed sensing. For a simple statement, they are called LDPC matrices in the following parts. Compared with aforementioned three types of matrices, LDPC matrices own three dominances. One is that their orthogonality is much stronger; and the second is that their orthogonality can be controlled and analyzed by relative construction algorithms; last but not least, their structured sparse characters, such as cyclic or quasi-cyclic structures [11], are very competitive for hardware implementations. Final experiments also verify our expectations. LDPC matrices show much better performance than other matrices in compressed sensing. Furthermore, optimal LDPC matrices for compressed sensing can also be obtained by analyzing and controlling the short cycles in Tanner graph [12] during matrix construction.

The rest of this paper is organized as follows. First, LDPC matrices are introduced and analyzed in terms of orthogonality in section 2. Then experiments are designed to prove our inferences in section 3. Finally, conclusions are given in section 4.

2. LDPC matrices

Here we first introduce the structure characteristics of LDPC matrices defined by LDPC codes, as well as their construction principles in section 2.1. Then we analyze the othogonality of LDPC matrices for compressed sensing in section 2.2.

2.1 LDPC matrices defined by LDPC codes

LDPC codes are one kind of linear block codes, which are defined in the null space of sparse binary parity-check matrix (LDPC matrices). To evaluate their structure and performance, parity-check matrix is often represented and analyzed by Tanner graph . For example, Figure 1b shows a Tanner graph associated with parity-check matrix **A** in

Figure 1a. In this Tanner graph, the upper five circles are often called variable nodes, sequentially corresponding to the five columns of matrix **A**. And the lower four squares are called check nodes, corresponding to the four rows of **A**. The edges between two classes of nodes are rendered by the nonzero entries in **A**. Then we can further derive a tree from any variable node by nonrepeatedly traversing adjacent nodes. As shown in Figure 1c, the gray circle denotes the root of a tree. In this tree, there are three cycles passing through the root variable node, and they respectively includes 4, 6 and 8 edges. Theoretically, the trees with bigger cycles are preferred because it means that the root variable node keeps higher independency with other variable nodes. In other words, the columns of matrix mutually hold lower correlation in structure, which is also the intrinsic requirements of RIP.

Specially, for LDPC codes, the cycle with 4 edges, e.g. the cycle with dotted lines in Figure 1c, is usually avoided for iterative decoding. It discloses that nonzero entries can't form a square structure in matrix **A**, as the square with dotted lines in Figure 1a. In other words, there are at most one same place assigned with nonzero entries for arbitrary two columns. So LDPC matrices hold stronger constraint on orthogonality in structure than other matrices. To present this constraint on cycles, we define *local girth* , shortly named g, as the number of edges in the shortest cycles passing through the root variable node on one tree. Each variable node holds one g in Tanner Graph, which might be equal to 4, 6, 8 and etc.. Relative construction algorithms for LDPC matrices are developed to make all variable nodes with $g>4$. Further we can also evaluate the orthogonality of a matrix by counting the distribution of variable nodes with different g. The matrix with variable nodes holding bigger g usually own stronger orthogonality in structure.

However, as the number of nonzero entries increases in matrix, all cycles passing through the root variable node incline to be shorter, and finally $g=4$ is inevitable. So if we define degree d as the average number of nonzero entries in each column of matrix, during the construction of matrix given degree d, the principle of construction parity-check matrix of LDPC codes is to achieve as much big g as possible. Since bigger g denotes higher orthogonality in structure for a matrix with a given degree d, LDPC matrices construction algorithms are also available to construct measurement matrices for compressed sensing.

2.2 LDPC matrices for compressed sensing

As we know, LDPC matrices with same size are discriminated by different degree d. In terms of orthogonality and sparsity, we search an approximately orthogonal LDPC matrix with minimal d, which is optimal for compressed sensing. However, in practice the value of optimal d is unknown, and varies for different matrix sizes. Obviously, given matrix size, it's unacceptable to search the optimal d by enumerating all possible values of d. Therefore, in this paper, considering that the orthogonality of LDPC matrix is related with the distribution of variable nodes with different g, we attempt to search and denote the optimal matrix by the distribution of g instead of d, which is also easily obeyed for matrix construction algorithms.

Considering RIP is impractical to measure the orthogonality of matrix, here we calculate the distribution of the correlation values between arbitrary two columns of one matrix. The distributions of correlation values in LDPC matrices and sparse random

matrices are shown in Figures 2 and 3, respectively, where the abscissa denotes the correlation values that arbitrary two columns might take in one matrix, and the ordinate denotes the probability that correlation values might be taken. It's clear that
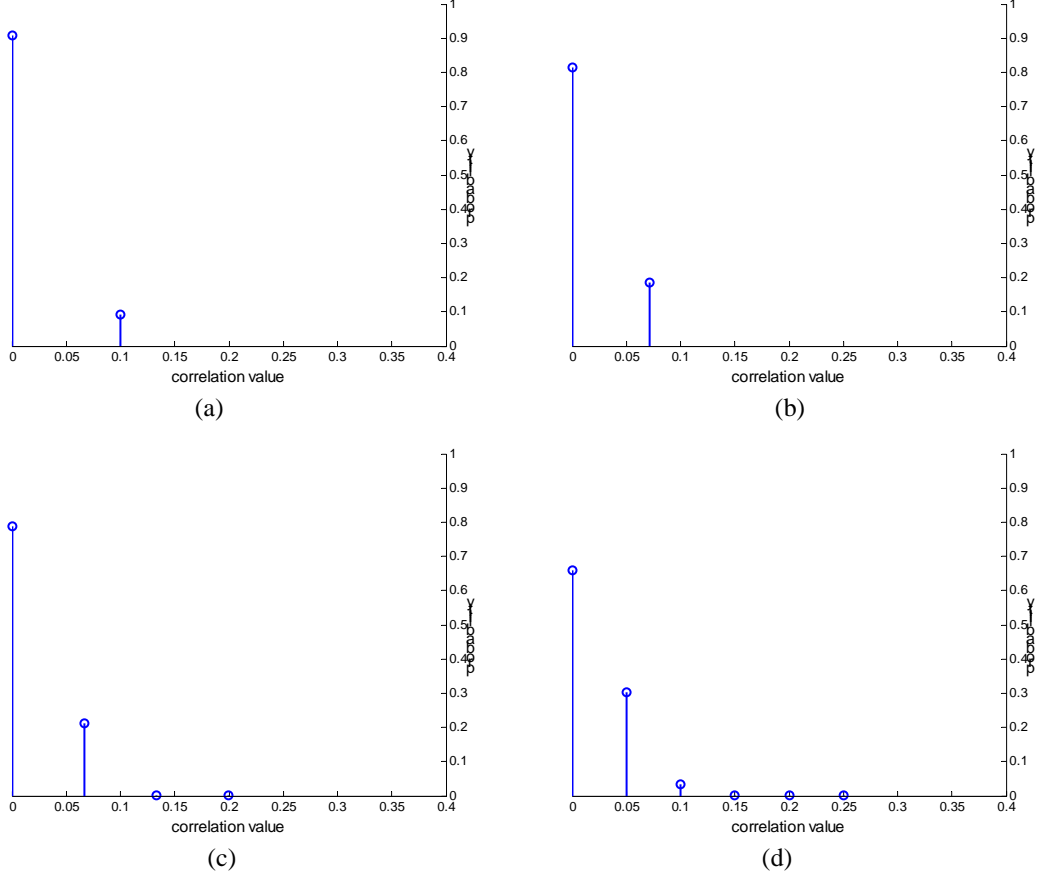

Figure 2: The distribution of correlation values between columns of LDPC matrices (1024,2048) with $d=10$ in (a), $d=14$ in (b), $d=15$ in (c) and $d=20$ in (d).
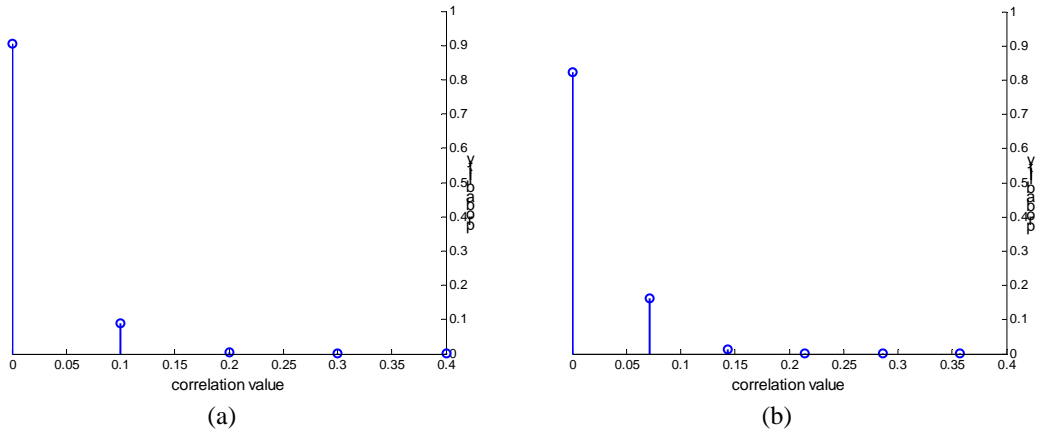

Figure 3: The distribution of correlation values between columns of sparse random binary matrices (1024, 2048) with $d=10$ in (a) and $d=14$ in (b).

approximately orthogonal matrix is preferred to own smaller correlation values between arbitrary two columns. Interestingly, for LDPC matrix with all variable nodes holding g >4, the correlations between arbitrary two columns take only two values, 0 or $1/d$. Note that all matrices are normalized for compressed sensing in this paper. E.g., for LDPC matrix with $d=10$ in Figure 2a, the correlation values only take 0 and $1/10$; for $d=14$ in Figure 2b, they only take 0 and $1/14$. Clearly, nonzero correlation values decrease as d increases, which is preferred for orthogonality. In contrast, as Figure 3 shows, correlation values between arbitrary columns in sparse random matrices usually take some values bigger than $1/d$. However, the dominance of LDPC matrices will be seriously deteriorated as the variable nodes with $g=4$ appear due to the increasing d. As shown in Figures 2c and 2d, the LDPC matrix with $d=15$ and $d=20$ both give some correlation values bigger than $1/d$. So as d increases, LDPC matrices can't get stronger orthogonality all the time. But on the condition of $g>4$, the nonzero correlation values decrease as d increases. So it can be reasonably predicted that there should be a turning point for d in orthogonality. In other words, there might be an optimal d for LDPC matrices with a given size.

Since bigger d means smaller correlation value $1/d$, whether we can assert that the optimal d should be the biggest d that holding all variable nodes with $g>4$? Unfortunately, as d increases, the correlation values will take $1/d$ rather than 0 with higher probability, though nonzero correlation value $1/d$ between two columns decreases, as shown in Figure 2. This is a conflict process for maximizing orthogonality. Therefore experiments are designed to further verify our conjecture.

3. Experiments

First, we introduce the methodology of experiments, and construct relative matrices in section 3.1. Then in section 3.2, we analyze the results and give the rules that search and construct optimal LDPC matrices for compressed sensing. Finally, we discuss the whole experiments in section 3.3.

3.1 Methodology

Experiments are designed for two goals. First, the performance of LDPC matrices for compressed sensing should be compared with other types of matrices. Considering Fourier matrices are worse than Gaussian matrices, and sparse random matrices are comparable with Gaussian matrices in some cases [7], our experiments only take Gaussian matrices and sparse random matrices in comparisons. Second, to verify whether there exists an optimal degree d for LDPC matrices given size, the performance of LDPC matrices with different degree d should be evaluated.

To obtain convincing results, LDPC matrices with kinds of compression ratios and dimension levels are respectively constructed by PEG algorithms [10].Tables 1-4 show 4 types of LDPC matrices, respectively, with sizes (100, 300) , (100, 500), (1024,2048) and (1024, 4096), and compression ratios 1/3, 1/5, 1/2 and 1/4. Moreover, for each type of LDPC matrices with a given size, a number of LDPC matrices with different degrees are constructed. E.g., in Table 3, 14 matrices with different d are constructed, where 10~14 corresponds to integer interval [10 14]. Note that each Table shows the biggest d that PEG algorithms can obtain for matrices with a given size. In addition, to discriminate their structures, the distribution of variable nodes with different values g are counted for each matrix. E.g., LDPC matrix (1024, 2048) with $d=15$, includes 1127 variables with

g=4, and 921variable nodes with g=6. As d increases, the number of variable nodes with g=4 also grows. Considering the matrix with biggest d and g>4 might be the optimal

Table 1: The distribution of variable nodes with different g in each LDPC matrix (100, 300) with degree d

d		3	4~5	**5.5**	**6**	7~10
g	4	0	0	150	284	300
	6	280	300	150	16	0
	8	20	0	0	0	0

Table 2: The distribution of variable nodes with different g in each LDPC matrix (100, 500) with degree d

d		3~4	**4.5**	**4.65**	**5**	6~10
g	4	0	98	244	431	500
	6	500	402	256	69	0

Table 3: The distribution of variable nodes with different g in each LDPC matrix (1024, 2048) with degree d

d		10~14	14.2	14.3	14.5	**15**	16	17~20
g	4	0	0	15	173	1127	2043	2048
	6	2048	2048	2033	1875	921	5	0

Table 4: The distribution of variable nodes with different g in each LDPC matrix (1024, 4096) with degree d

d		6~11	11.3	11.4	11.7	**12**	13	13.5
g	4	0	0	48	339	1929	4088	4096
	6	4096	4096	4048	3757	2167	8	0

matrix for compressed sensing, we obtain the extreme case with decimal d by matrix construction. Here decimal d is the average of some integers. E.g, the matrix with d=14.3 in Table 3 denotes that 70% columns take d=14 and 30% columns take d= 15.

Likewise, Gaussian matrices and sparse random matrices with equivalent size are also constructed. In order to reflect their performance comprehensively, they are randomly generated in each simulation. It should be noted that, for the sparse random matrices, there also exists the problem of selecting optimal d, which is not addressed in [7]. So in our experiments, we first evaluate their performance with d at integer interval [4 20]. E.g., for the sparse random matrix with size (1024, 4096), 17 matrices with d at integer interval [4 20] are first evaluated for compressed sensing. And then for a simple comparison, we only mention their best performance as baselines in Figures 4-7.

The performances of matrices are measured by the mean square error (MSE) between recovered signal and input sparse signal. To verify robustness of optimal LDPC matrices, LDPC matrices are tested by sparse signals with diverse sparsity levels and numerical levels. For sparsity levels, considering the precisions of error levels in a finite number of times by simulation, k=300 and 400 are selected for LDPC matrices (1024, 2048), and k = 200 and 300 for LDPC matrices (1024, 4096). Six integers [20 30 40 50 60] are
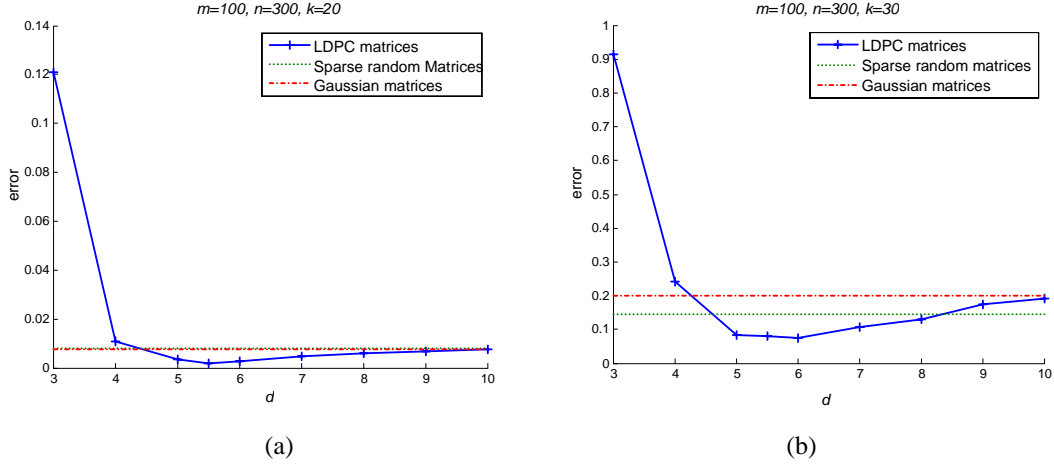
Figure 4: The MSE of LDPC matrices (100, 300) with degree *d* for sparse signals with sparsity *k*=20 in (a) and *k*=30 in (b). As references, the performances of Gaussian matrices and sparse random matrices are displayed as two baselines.

Figure 5: The MSE of LDPC matrices (100, 500) with degree *d* for sparse signals with parsity *k*=30 in (a) and *k*=40 in (b). As references, the performances of Gaussian matrices and sparse random matrices are displayed as two baselines.
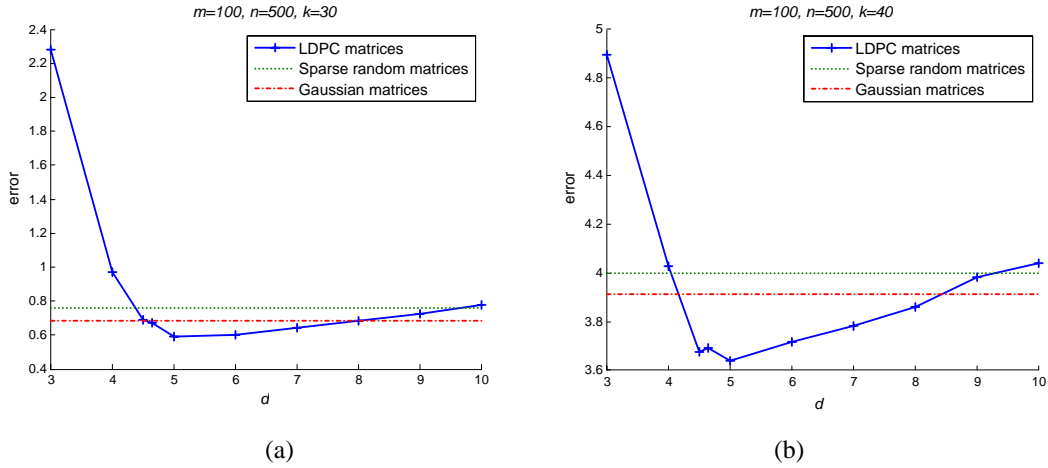
assigned to *k* for LDPC matrices (100, 300) and (100, 500). For numerical levels of nonzero entries, 'big' level integers at interval [-255 255] are applied for aforementioned two types of bigger matrices, and 'small' level values sampled from normal Gaussian distribution are adopted for two types of smaller matrices. Of course, optimal LDPC matrices should be robust for sparse signals with diverse sparsity levels and numerical levels. As for the solution algorithms, we use practical OMP algorithms [15] with known sparsity k. All error performances are averaged by 10000 times of simulations.

3.2 Results
The performances of four types of LDPC matrices versus the scope of degrees are respectively shown in Figures 4-7. In contrast, the performances of Gaussian matrices as well as selected optimal performance of random sparse matrices are also displayed as two
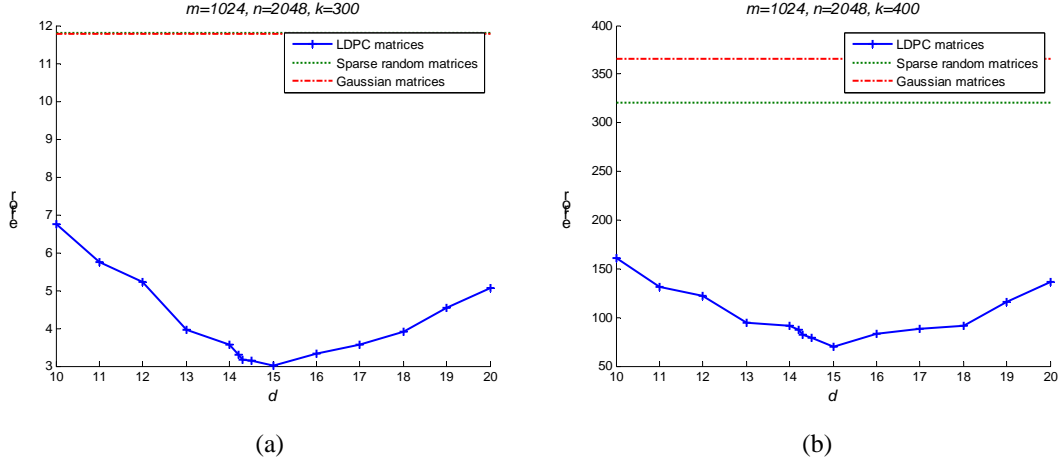
Figure 6: The MSE of LDPC matrices (1024, 2048) with degree d for sparse signals with sparsity k=300 in (a) and k=400 in (b). As references, the performances of Gaussian matrices and sparse random matrices are displayed as two baselines.

Figure 7: The MSE of LDPC matrices (1024, 4096) with degree d for sparse signals with sparsity k=200 in (a) and k=300 in (b). As references, the performances of Gaussian matrices and sparse random matrices are displayed as two baselines.
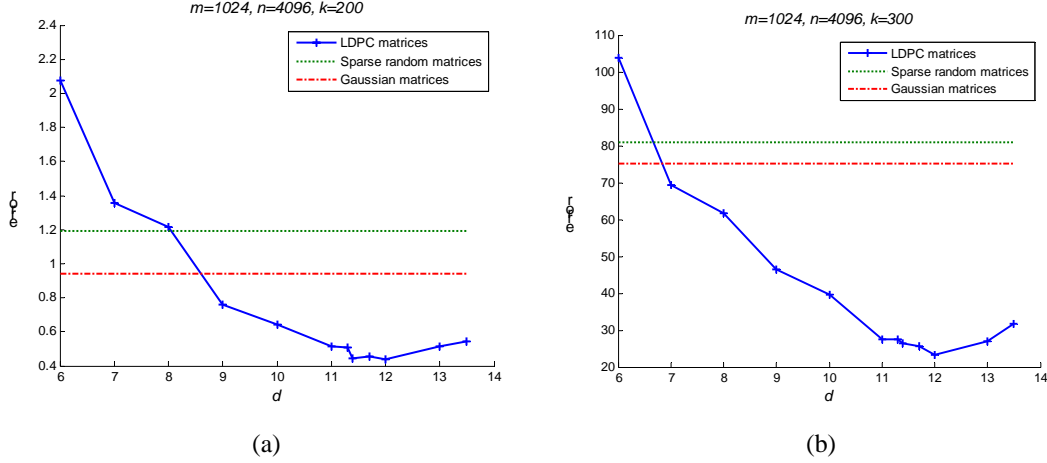
baselines in each Figure. It's clear that LDPC matrices significantly outperform other two types of matrices, verified by sparse signals with two sparsity levels and two numerical levels.

Furthermore, concavity performance lines in Figures 4-7 indicate that LDPC matrices really own the optimal d for compressed sensing. For example, LDPC matrices achieve their lowest MSE at d=15 in Figure 6, and at d=12 in Figure 7. But more simulation results in Tables 5 and 6 disclose that the optimal d for LDPC matrices with a given size is not very stable for sparse signals with variable sparsity k. E.g., for LDPC matrices (100, 500) in Table 6, the optimal d can take values 4.5, 4.65 and 5 for different sparsity k. But their differences are tiny, no more than 1. And their performances are also very close. So in practice we are allowed to select a representative degree d for a LDPC matrix given size, at little cost of performance.

Table 5: The MSE of LDPC matrices (100, 300) with degree *d* for sparse signals with sparsity *k*.

	d	3	4	5	5.5	6	7	8	9	10
	20	0.1211	0.0108	0.0034	**0.0020**	0.0028	0.0048	0.0060	0.0069	0.0077
	30	0.9139	0.2425	0.0835	0.0805	**0.0756**	0.1055	0.1294	0.1737	0.1921
k	40	3.1600	1.9678	1.3520	**1.2231**	1.2666	1.4667	1.6030	1.7965	1.8820
	50	5.4480	4.9827	4.5860	**4.4986**	4.5322	4.7121	4.8719	4.9804	5.0717
	60	7.2106	7.0510	6.9563	**6.9268**	6.9410	7.0493	7.0963	7.2074	7.2274

Table 6: The MSE of LDPC matrices (100, 500) with degree *d* for sparse signals with sparsity *k*

	d	3	4	4.5	4.65	5	6	7	8	9	10
	20	0.3726	0.0476	0.0264	**0.0202**	0.0208	0.0207	0.0206	0.0236	0.0275	0.0279
	30	2.2805	0.9685	0.6877	0.6707	**0.5905**	0.6040	0.6413	0.6848	0.7251	0.7745
k	40	4.8944	4.0271	3.6757	3.6924	**3.6380**	3.7156	3.7809	3.8586	3.9814	4.0392
	50	6.7974	6.5649	6.4684	**6.4386**	6.4428	6.5225	6.5920	6.6390	6.6933	6.7500
	60	8.2902	8.2592	**8.2100**	8.2293	8.2500	8.3069	8.4016	8.4364	8.4974	8.5051

Now of interest is to find whether the optimal *d* is the biggest *d* holding *g*>4. Turn to Tables 1-4, where the optimal values of *d* are in bold, interestingly, for four types of LDPC matrices, the optimal *d* all focused on the margin of the biggest *d* holding *g*>4. Further, optimal *d* generally doesn't surpass the bound that all variable nodes begin to hold *g*=4. Precisely, the matrices with optimal *d* all hold partial variable nodes with *g*=4 in our experiments. For example, LDPC matrices (1024, 2048) in Table 3, the optimal matrix with *d*=15 holds 1127 variable nodes with *g*=4 and 921 variable nodes *g*=6. As Tables 1-4 show, this characteristic confines optimal *d* in a small interval, which is close to the biggest *d* with *g*>4. Empirically, the difference between optimal *d* for compressed sensing and biggest *d* with *g*>4 is usually less than 1. In words, during matrix construction, we can easily determine the possible region of optimal *d* by searching the biggest *d* with *g*>4. In addition, if the sparsity of sparse signal can also be estimated before hand, we can derive one more precise *d*.

3.3 Discussion

RIP [1] essentially requires that arbitrary set of *t* columns in one matrix approximately behaves like an orthogonal system [4], where variable *t* is less than some constant *S*. And *S* is hoped to be as big as possible for compressed sensing. But for a given matrix, the solution to the biggest *S* is a NP hard problem. So there is still no practical tools to measure the orthogonality as RIP requires. In this paper, we roughly estimate the orthogonality in terms of their correlations between two columns. But obviously it can't reflect the variation of real biggest *S*. So we need experiments to search the optimal LDPC matrices. Luckily, experiments show that the optimal LDPC matrices own some intrinsic relations with its distribution of cycles in Tanner graph, which is favorable for their construction in practice. For example, the optimal LDPC matrices constructed by PEG algorithms [10] usually contain partial variable nodes with *g*=4, and this strictly constrain degree *d* at a small interval less than 1. So they can be easily determined during the construction. In addition, optimal *d* is usually slightly bigger than the biggest *d* with *g*>4 in our experiments, and it can be regarded that the probability that correlations between columns take values bigger than 1/*d* is still tiny in this case. Moreover,

considering the structure of LDPC matrices largely depends on their construction algorithms, the bounds for optimal d might vary for different construction algorithms.

4. Conclusions

In this paper, we have explored the application of parity-check matrix of LDPC codes in compressed sensing. Experiments verify that this type of sparse binary matrix significantly outperform current other matrices. Furthermore, for LDPC matrices with a given size, the optimal matrix for compressed sensing can also be approximately constructed by constraining the short cycles in Tanner graph. In addition, in terms of computation load, their cyclic or quasi-cyclic [11] structures are more competitive for hardware implementation. In sum, LDPC matrices expose significant dominance on the application of compressed sensing.

References

[1] E. Candes and T. Tao, "Decoding by Linear Programming", IEEE Trans. on Information Theory, 51(12), pp. 4203 - 4215, Dec. 2005.

[2] E. Candes, J. Romberg and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information", IEEE Transactions on Information Theory, 52(2) pp. 489 - 509, Feb. 2006.

[3] D. Donoho, "Compressed Sensing", IEEE Trans. Inf. Theory, 52(4), pp. 1289 -1306, April 2006.

[4] E. Candes, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements", Communications on Pure and Applied Mathematics, 59(8), pp. 1207-1223, Aug. 2006.

[5] T. Tao. Open question: deterministic uup matrices. Weblog at: http://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices/

[6] V. Chandar. "A negative result concerning explicit matrices with the restricted isometry property", Tech. Rep., 2008.

[7] R. Berinde and P. Indyk, "Sparse recovery using sparse random matrices", MIT-CSAIL Technical Report, 2008.

[8] M. Sipser and D. Spielman, "Expander codes", IEEE Trans. Inf. Theory, 42(6), pp. 1710-1722, 1996.

[9] R. G. Gallager, "Low density parity check codes", IRE Trans. Inf. Theory, vol. IT-8, pp. 21–28, Jan. 1962.

[10] X.Y. Hu, E. Eleftheriou, and D. M. Arnold, "Progressive Edge-Growth Tanner Graphs", IEEE Global Telecommunications Conference 2001, vol. 2, pp. 995-1001, Nov., 2001

[11] Z. Li and B. V. K. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph", in Proc. 38th Asilomar Conf. Signals, Syst. Comput., pp. 1990-1994, 2004.

[12] R. M. Tanner, " A recursive approach to low complex codes", IEEE Trans. Inf. Theory, vol. IT-27, pp. 533-547, Sept. 1981.

[13] S.S. Chen, D.L. Donoho, M.A. Saunders, "Atomic Decomposition by Basis Pursuit", SIAM Journal on Scientific Computing, 20 (1), 33-61, 1998.

[14] B. Efron, T. Hastie, and R. Tibshirani, "Least angle regression", Annals of Statistics, 32, pp. 407–499, 2004.

[15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit", IEEE Trans. Inf. Theory, 53, pp. 4655–4666, 2007.