Binary Graphs and Message Passing Strategies for Compressed Sensing in the Noiseless Setting

Francisco Ramirez-Javega, Meritxell Lamarca, Javier Villares
Signal Processing and Communications Group,
Department of Signal Theory and Communications (TSC)
Universitat Politecnica de Catalunya (UPC)
Email: javega@tsc.upc.edu, meritxell.lamarca@upc.edu, javier.villares@upc.edu

***Abstract*—We propose a scheme for Compressed Sensing in the noiseless setting that reconstructs the original signal operating on a binary graph where the samples are obtained sequentially. The proposed scheme has an affordable computational complexity and a large performance enhancement with respect to similar schemes in the literature, thanks to the proposed measurement matrix structure and enhanced decoding based on a message passing algorithm.**

I. Introduction

The Compressed Sensing (CS) problem considers the estimation of an unknown and sparse signal vector $\mathbf{x} \in \mathbb{R}^n$ from a vector of linear observations $\mathbf{y} \in \mathbb{R}^m$, $m < n$, $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a fixed randomly generated matrix known as measurement matrix and only a small number k (the sparsity index), $k << n$, of elements of $\mathbf{x}$ are non-zero. The set containing the positions of these elements is known as the *support set*, defined as $\mathcal{S} \triangleq \{i \in 1, ..., n : x_i \neq 0\}$, with cardinality $|\mathcal{S}| = k$. The solution to this system of equations is known to be given by the vector $\hat{\mathbf{x}}_0$ that minimizes $\|\mathbf{x}\|_0$ (l_0 norm) subject to $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}_0$, which is a non-convex optimization problem.

The authors in [1], [2] established that the vector $\hat{\mathbf{x}}_1$ with minimum l_1 norm subject to $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}_1$ coincides with $\hat{\mathbf{x}}_0$ whenever the measurement matrix satisfies the RIP condition [1]. Some examples of RIP matrices are given, with high probability, by matrices whose entries are i.i.d. Radamacher or Gaussian [3] [4]. These facts enabled the development of several l_1-based reconstruction algorithms to solve the set of equations (see [5], [6] and references therein) and the characterization of the intrinsic limits, in terms of samples (m), of l_1-based algorithms [7].

Recently, within this framework, the authors in [8] showed that $m \geq k$ samples suffices to reconstruct a real-valued signal with k non-zero components in the almost lossless compression setting in the n asymptotic regime. This result indicated that l_1-based algorithms are non-optimal, since they require a number of samples larger than k to succesfully recover the signal [7]. Simultaneously, in the literature some algorithms based on message propagation over graph representations of measurement matrices were shown to outperform l_1-based theoretical limits, e.g. the scheme proposed in [9].

In this paper a new graph-based scheme is introduced that also outperforms the l_1 theoretical limits and has very low computational complexity, thanks to the use of binary sparse matrices. The proposed scheme stems from the original design of sudocodes [10], which has the distinctive features of taking infinite reliability decisions (i.e. no errors are made), requiring binary sparse measurement matrices and its worst case computational complexity at the decoder side is $O(k \log(k) \log(n))$.

The scheme in this paper combines the introduction of a block-wise constructed binary measurement matrix structure with the proposal of a check node degree profile design criterion and an extension of the existing verification-based decoding algorithms [11], [12]. This enhancement of the verification algorithm in [11] is particularly relevant for very sparse signals, since in this case length-four cycles cannot be avoided in the graph. The use of binary measurement matrices is a key point to reduce the computational complexity at the decoder side, as in this case the decoding process only performs additions and comparisons, enabling the use of both a sudocode-based or verification-based decoder, which has an $O(n)$ computational complexity at the decoder side. The proposed algorithm outperforms the results in [10] and [11] in terms of number of measurements while having a similar decoding complexity than [11].

This paper is structured as follows. In section II we present the proposed scheme. We introduce a systematic method to construct binary matrices in section III. We propose in section IV an extension of the decoding algorithm proposed in [11] aimed at mitigating the effect of cycles of length 4 in dense binary graphs for the noiseless setting. In section V we give a closed-form expression to obtain the appropriate check node degree profile. Finally, in section VI simulation results illustrate the performance enhancement obtained by the proposed compressed sensing system.

II. System Description

We assume that the signal vector $\mathbf{x}$ is strictly k-sparse, $0 < k \leq n$, i.e. all the realizations have exactly k non-null components and the components that are not indexed in $\mathcal{S}$ are strictly zero. The coefficients indexed in $\mathcal{S}$ are randomly drawn according to a generic probability density function. We

[0] This work has been partially funded by the Spanish Government under the projects: TEC2007-68094-C02-02, TEC2010-21245-C02-01, CSD2008-00010, CEN-20101019, and the Catalan Government under Grant 2009SGR1236.

978-1-4673-2579-0/12/$31.00 ©2012 IEEE

assume through all the work that the measures $\mathbf{y} = \mathbf{A}\mathbf{x}$ are not corrupted by noise. $\mathbf{A}$ is a binary matrix where the number of ones depends on the sparsity ratio k/n and is sparse becoming denser for very low values of k/n. Matrix $\mathbf{A}$ is fixed for all signal realizations. It is randomly generated according to the procedure described in section III and it is characterized by the fraction of rows/columns with d ones, denoted respectively as $\mathrm{R(d)}$ and $\mathrm{L(d)}$, the check and variable node degree profiles of the associated graph.

The proposed system employs a sequential non-adaptive sampling procedure, i.e., the measurements are drawn one by one following the order of the rows of matrix $\mathbf{A}$ and after each new measurement is drawn a new attempt is made to recover the sequence $\mathbf{x}$. This procedure is followed until all the components in $\mathbf{x}$ are identified. Note that the number of measurements required to reconstruct vector $\mathbf{x}$ may be different for each signal realization.

The binary and sparsity features of $\mathbf{A}$ are exploited to retrieve the signal vector $\mathbf{x}$ with a low-complexity decoding algorithm based on the description of the system by a graph. The decoding attempts are done using the algorithm described in section IV-B.

III. Construction of the Measurement Matrix

The measurement matrix $\mathbf{A}$ can be represented by a graph having three sets of nodes: signal nodes (elements of vector $\mathbf{x}$), measurement nodes (elements of vector $\mathbf{y}$) and check nodes connected to d signal and one measurement nodes as indicated by the binary measurement matrix $\mathbf{A}$. Hereafter we refer to the signal nodes as variable nodes.

The graph is fundamentally characterized by the degree distribution of the check nodes $\mathrm{R}(d)$ and that of the variable nodes $\mathrm{L}(d)$. We design the distribution $\mathrm{L}(d)$ to be regular, i.e. all the variable nodes are connected to the same number/degree N of measurement nodes; the distribution $\mathrm{R}(d)$ is optimized in section V, where a profile $\mathrm{R}(d)$ having at most two consecutive degrees is adopted.

Rather than generating matrix $\mathbf{A}$ as a random realization of the ensemble of matrices having degrees profiles $\mathrm{R}(d)$ and $\mathrm{L}(d)$, this paper proposes to introduce some structure in the construction of matrix $\mathbf{A}$. The purpose is to guarantee that all components of vector $\mathbf{x}$ contribute to a measurement before any of them is sensed again. This structure is beneficial in the sequential sampling procedure: it ensures that all the components of vector $\mathbf{x}$ are sensed equally early, avoiding the possibility that one non-zero sample may not be sensed until late and, therefore, the sequential sampling may require a large number of samples to stop.

Hence, we propose to construct a matrix $\mathbf{A}$ by sub-blocks. Let us assume, for the sake of simplicity that all rows have the same degree d. Then, $N = md/n$ sub-matrices $\mathbf{A}_1 ... \mathbf{A}_N$ of dimension $n/d \times n$ are randomly drawn from the ensemble of all matrices of row degree d and column degree 1. Afterwards, these sub-matrices are stacked to build matrix $\mathbf{A}$ as $\mathbf{A} = [\mathbf{A}_1^{\mathrm{T}} ... \mathbf{A}_N^{\mathrm{T}}]^{\mathrm{T}}$.

The design of sub-matrices $\mathbf{A}_1 ... \mathbf{A}_N$ is done such that cycles of length four are avoided whenever possible. However, unfortunately, when the signal to be sensed is very sparse these cycles are unavoidable if one wants to keep the number of measurements low (and close to the theoretical limit) and the vector length n is finite. Indeed, if the number of measurements is in the order of k then the check node degree must be at least n/k in order to guarantee that all signal coefficients are sensed once. Hence the check node degree grows as the signal becomes sparse, matrix $\mathbf{A}$ becomes denser and length-four cycles become unavoidable for finite length n. This behavior explains the relevance of the enhancement of the decoding algorithm proposed in the next section.

IV. Reconstruction Algorithm

In this section we present the proposed reconstruction (decoding) algorithm, which is an enhancement of the verification algorithm proposed in [11]. This algorithm performs message passing over the bipartite graph representation of the measurement matrix $\mathbf{A}$.

The following notation is used hereafter: we denote as ϕ and $\mathcal{I}$ a row vector of $\mathbf{A}$ and the set of non-null components of ϕ, respectively. Hence, the cardinality of $\mathcal{I}$ is $|\mathcal{I}| = d$, the check node degree.

A. Verification Algorithm (VA)

First, we present the application of the verification algorithm (VA) in [11] to reconstruct the vector $\mathbf{x}$ in the set-up of section II. This algorithm was identified by [11] to be a different implementation of the sudocode decoding algorithm [10], which has $O(k \log(k) \log(n))$ computational complexity as opposed to [11] whose computational complexity is $O(n)$. It must be noted that the algorithm presented in section IV-B can be implemented as in [10] or in [11].

When applied in the CS framework, VA exploits that (*i*) there is no noise, (*ii*) the probability density function of the components of vector $\mathbf{x}$ is a continuous function with a mass concentration at zero, $p(x = 0) = 1 - k/n$, and (*iii*) the graph has no cycles of length four. Due to these facts the following statements hold:

- **S1**. If a measurement y is zero, all the variable nodes indexed by $\mathcal{I}$ are zero.
- **S2**. When the graph does not exhibit cycles of length four and two measurements y_j and y_v are identical ($y_j = y_v$), that pair of measurements share a single variable node, $|\mathcal{I}_j \cap \mathcal{I}_v| = 1$. Therefore, it follows that the common component in the intersection set must be equal to y_j and the remaining variable nodes in $\mathcal{I}_j$ and $\mathcal{I}_v$ must be equal to zero.

Keeping the above statements in mind, the algorithm works as follows. The check (variable) nodes exchange with the variable (check) nodes messages having the following form: $\{$*state,x*$\}$ where *state*='v' indicates that we know for sure that the value of the associated variable node is *x* (i.e. the variable node is *verified*) whereas *state*='?' informs that *x* is solely an estimate of the variable node value (i.e. the variable node is still *not verified*). A check node is said to *verify* a neighboring variable node whenever the first can infer with infinite reliability the value of the latter using statements S1 and S2. Then, the verified variable node propagates its state to the other check nodes in its neighborhood so that these check nodes can remove the contribution of the verified variable node from their respective measurement and try to infer, in the next round, the values of the remaining variable nodes connected to them.

The iterative process starts with the variable nodes sending a *non-verified* message $\{$'?',$-\}$, with dummy value $-$, to all the attached check nodes. On its turn, the measurement nodes simply send to their respective check nodes a message containing the measurement value, $\{$'v', $y\}$, and do not update their message during the decoding process.

Then, the check nodes are activated and send a message computed as indicated by *Algorithm 1*, which is now explained briefly, to the variable nodes in their neighborhood. First of all, each check node removes the contributions of the incoming *verified* messages from its measurement y to obtain y'. Next, it updates all the edges connected to variable nodes that remain not verified (indexed in the set $\mathcal{I}_?$), according to rules C1, C2 and C3. Rule C1 exploits statement S1. Rule C2 deals with the trivial case in which there is a single non-verified variable node connected to the check node. Rule C3 deals with the remaining cases: whenever the check node cannot infer the value of a variable node it propagates the *non-verified* message with the updated measurement, ('?',y').

After the check nodes update, the variable nodes are activated and proceed as indicated in *Algorithm 2*. Rule V1 deals with verified variable nodes. In this case, due to the absence of noise, each verified variable node informs to the check nodes in its neighborhood that it is verified by propagating a *verified* message along with its value. Rule V2 exploits statement S2. In this case, the variable node receives more than one *non-verified* message with equal estimate y' so it can infer its value. Hence, the variable node switches its state to verified and informs of this update to all the check nodes in its neighborhood by sending them the message $\{$'v',$y'\}$. Finally, in case of rule V3, the variable node just sends a *non-verified* message $\{$'?',$-\}$ to all the neighboring check nodes.

Algorithm 1 Check node update rules

```
for  all check nodes and their measurement $y$
     for  all the verified messages from the variable nodes in $\mathcal{I}$
          Subtract their values from $y$ and remove their indexes from $\mathcal{I}$ to compute $y'$ and $\mathcal{I}_?$.
     endfor
     for  all variable nodes in $\mathcal{I}_?$ apply rules C1-C3:
          C1  If $y' = 0$, propagate to all the variable nodes indexed by $\mathcal{I}_?$: $\{$'v', $0\}$.
          C2  If $|\mathcal{I}_?| = 1$, propagate to the variable nodes in $\mathcal{I}_?$: $\{$'v', $y'\}$.
          C3  Otherwise, propagate to all the variable nodes indexed by $\mathcal{I}_?$: $\{$'?', $y'\}$.
     endfor
endfor
```

Algorithm 2 Variable node update rules

```
for  each non-verified variable node apply rules V1-V3:
     V1  If the variable node receives a verified message, then propagate this message through the remaining edges $\{$'v', $y'\}$.
     V2  If the variable node receives at least two non-verified messages with the same estimate $y'$, then propagate $\{$'v', $y'\}$ to all the nodes connected to them.
     V3  Else, propagate $\{$'?', $-\}$ through its edges.
endfor
```

B. Enhanced Verification Algorithm (EVA)

This paper proposes a modification of the verification algorithm in section IV-A that improves its performance for very sparse sources while preserving its $O(n)$ computational complexity and requiring only local operations. This enhancement is similar in spirit to the one proposed in [13] for decoding of LDPC codes in BEC channels. More specifically, the improvement consists in extending rule V2 in *Algorithm 2* to deal with graphs having cycles of length four, i.e., $|\mathcal{I}_j \cap \mathcal{I}_v| \geq 1$. In the presence of such cycles, if two measurements y_j and y_v take the same value ($y_j = y_v$), we know that the variable nodes that are not in the intersection set $\mathcal{I}_j \cap \mathcal{I}_v$ are equal to zero but we cannot determine the individual value of each variable node that are in the intersection; in fact, we can only affirm that variable nodes that are in the intersection must sum up to y_j. In that case, it is important that the variable nodes in the intersection inform the check nodes of this *coincidence* so that the variable nodes that do not belong to the intersection set can be verified as zero.

To implement the above mechanism, we have modified the original message passing algorithm including a new message named *coincidence*. This message is generated at the variable nodes that detect coincident measurements and is sent to the check nodes detected to be in the *coincidence* state. Thanks to this, the check nodes know that (*i*) there is at least another check node with a measurement equal to its measurement and the cardinality of the intersection (the number of received coincident messages) and (*ii*) one or more of the nodes in coincident state sum up to the measurement. In other words, if a check node receives only a *coincidence* message it means that there is no cycle of length four so it can verify the variable node that sent the *coincidence* message with the value of the measurement and the remaining nodes with zero (applying the original rule V2). Otherwise, if the check node receives more than one *coincidence* message, it sends a *non-verified* message to the variable nodes that sent *coincidence* messages and propagates verification messages $\{$'v',$x = 0\}$ to the remaining ones. Briefly, these changes are summarized as follows:

- *New variable node rule* **V2**: If a variable node receives at least two *non-verified* messages with the same estimate, it sends a *coincidence* message to these check nodes, and transmits {'?',−} to the remaining nodes in its neighborhood.

- *Check node rule* **C4**: If a check node receives at least a *coincidence* message, it verifies as zero the variable nodes that have not sent *coincidence* messages. Additionally, if the check node receives only a *coincidence* message, it verifies the corresponding variable node using its own measurement. Otherwise, it propagates {'?',y'} to the variable nodes that sent *coincidence* messages.

V. Design of the Check Node Degree

This section analyzes the influence of the sparsity ratio on the performance of a VA-based decoding scheme and applies the results to obtain a criterion to design $\mathrm{R}(d)$ for the verification algorithm. Even though it is out of the scope of the current work, it must be noted that the performance of the EVA can be further improved in graphs with length-four cycles by changing the criteria to design of the $\mathrm{R}(d)$.

Let us consider the case $n \to \infty$ and call p the non-vanishing probability of a signal component being 0, $p = 1 - k/n$. For simplicity we assume that all the ϕ vectors have degree d. Then, when all the ones in ϕ overlap with null components of $\mathbf{x}$, $|\mathcal{I} \cap \mathcal{S}| = 0$, the measurement $y = \phi^T \mathbf{x}$ is equal to zero, using statement S1. This happens with probability:

$$p(|\mathcal{I} \cap \mathcal{S}| = 0) = p^d \tag{1}$$

and tells us the probability of recovering d zeros of vector $\mathbf{x}$ from a single measurement. Obtaining a vector ϕ which overlaps only with one element of the support set of $\mathbf{x}$, i.e. $|\mathcal{I} \cap \mathcal{S}| = 1$, happens with probability

$$p(|\mathcal{I} \cap \mathcal{S}| = 1) = d\,p^{d-1}\,(1-p). \tag{2}$$

However, we also generate measurements with contribution of more than one non-null component of vector $\mathbf{x}$ such that $|\mathcal{I} \cap \mathcal{S}| > 1$, with a probability

$$p(|\mathcal{I} \cap \mathcal{S}| > 1) = 1 - p(|\mathcal{I} \cap \mathcal{S}| = 1) - p(|\mathcal{I} \cap \mathcal{S}| = 0). \tag{3}$$

According to equations (1)-(3), the probability of verifying null and non-null components of $\mathbf{x}$ depends critically on d and p. Hence, the verification process can be sped up by selecting a degree $\mathrm{R}(d)$ that balances properly the amount of vectors generated according to equations (1) and (2). With this purpose, we propose to employ the check node degree distribution $\mathrm{R}(d)$ that maximizes the average number of variable nodes connected to check nodes so as $|\mathcal{I} \cap \mathcal{S}| \leq 1$, as follows

$$\begin{aligned} \underset{\mathrm{R}(d)}{\text{maximize}} \quad & \sum_{\forall d} \mathrm{R}(d)\, d\, p(|\mathcal{I} \cap \mathcal{S}| \leq 1) \\ \text{subject to} \quad & \mathrm{R}(d) \geq 0, \\ & \sum_{\forall d} \mathrm{R}(d) = 1 \end{aligned} \tag{4}$$

where $p(|\mathcal{I} \cap \mathcal{S}| \leq 1) = p(|\mathcal{I} \cap \mathcal{S}| = 0) + p(|\mathcal{I} \cap \mathcal{S}| = 1)$. Note that $p(|\mathcal{I} \cap \mathcal{S}| \leq 1)$ is the fraction of check nodes with $|\mathcal{I} \cap \mathcal{S}| \leq 1$ for asymptotic m. Then it follows that $d\,p(|\mathcal{I} \cap \mathcal{S}| \leq 1)$ is the average number of edges per check node that satisfy $|\mathcal{I} \cap \mathcal{S}| \leq 1$ for a given p and d.

n,k	Sudocodes [10]	Proposed method
10000,100	m=803	m=375
10000,10	m=461	m=85
100000,10	m=931	m=92

Table I
Comparison of sudocodes with EVA.

It is straightforward to show that for a fixed p and $d \in \mathbb{R}^+$ the relaxed function $d\,p(|\mathcal{I} \cap \mathcal{S}| \leq 1)$ has a single maximum at $\bar{d}$. It can be shown that whenever we fix the number of non-null coefficients of $\mathrm{R}(d)$ to two and assuming that $n/k >> 1$, the optimum choice of the two non-null degrees are the closest ones around $\bar{d}$. In this case $\sum \mathrm{R}(d) d\,p(|\mathcal{I} \cap \mathcal{S}|\_ \leq 1)$ has a unique solution $\mathrm{R}(d)$ that can be inferred from $\bar{d}$ as follows[1]

$$\mathrm{R}(\lfloor \bar{d} \rfloor) = \lceil \bar{d} \rceil - \bar{d}, \;\; \mathrm{R}(\lceil \bar{d} \rceil) = \bar{d} - \lfloor \bar{d} \rfloor \tag{5}$$

and $\bar{d}$ can be calculated as shown in (6)

$$\begin{aligned} \bar{d} &= -t_1(1+t_2)/2 \\ t_1 &= \frac{p}{1-p} + \frac{2}{\log(p)}, \quad t_2 = \sqrt{\tfrac{1-(4p)}{\log(p)(1-p)t_1^2}} \end{aligned} \tag{6}$$

Note that (6) only requires the knowledge of the sparsity ratio of $\mathbf{x}$, $p = 1 - k/n$. It is worth mentioning that $\bar{d}$ is close to the check node degrees that were found experimentally to be optimum in [10, figure 2] and that $\bar{d}$ increases as n/k increases.

The idea behind equation (4) is that by verifying as much variable nodes per measurement as possible the phase transition[2] zone will be reached as soon as possible (in terms of samples). The EVA algorithm shows a kind of avalanche effect when it reaches its phase transition zone: once it reaches this zone the addition of a single sample may enable the verification of a large fraction of variable nodes. This happens because rule C2 in *Algorithm 1* is activated when the average number of non-verified variable nodes connected to check nodes is close to 1. When this happens the avalanche effect is triggered; before that, variable nodes can only be verified in the graph when statements S1 and S2 in section IV-A or modified S2 in section IV-B happen. Hence, by maximizing the fraction of variable nodes connected to check nodes with $|\mathcal{I} \cap \mathcal{S}| \leq 1$ we try to reduce as fast as possible the number of non-verified variable nodes while keeping low the number of generated measurements.

VI. Results

Table I compares the joint performance of the proposed matrix construction method, the proposed check node degree design and the proposed message passing algorithm with the performance of the sudocodes [10]. In this table, m represents

[1] Derivation is omitted due to space constraints

[2] In Compressed Sensing, the reconstruction algorithms show a dual behavior, i.e. algorithms do not converge to a solution until the number of samples m is large enough. The limit between the zone were algorithms converge and do not converge is known as the phase transition zone. See [7], [8] for further details.

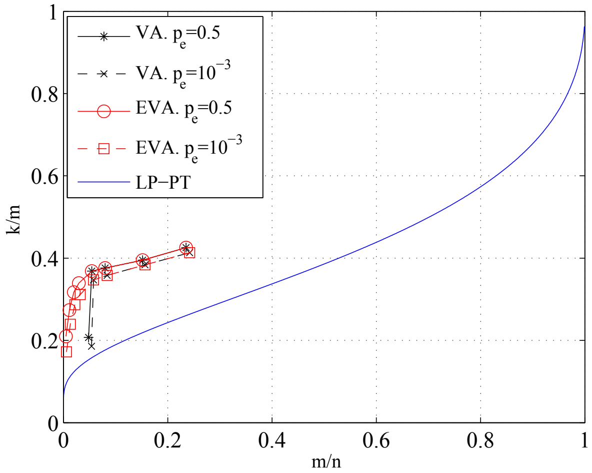

Figure 1. Phase transition diagram as a function of sampling, sparsity and block length n. Comparison of verification (VA) and enhanced verification (EVA) algorithm for n=16000 and several sparsity ratios on a phase diagram.

the maximum number of samples required for perfect reconstruction obtained after 10^5 Monte Carlo simulations. Note that the number of samples required by the proposed scheme is 2 to 10 times smaller that the one required by sudocodes. It must be noted that the proposed method also outperforms that one in [11, figure 6] where the adopted LM2-MB algorithm is the VA algorithm in presented in section IV-A. In this case for $n = 10000$ the method in [11] requires $m > 500$ measurements to reconstruct a sequence with $k = 100$; in our case, $m = 375$ suffices to reconstruct almost any sequence with $k = 100$ (see table I).

Figure 1 shows a phase transition diagram as a function of the number of samples m, sparsity k and block length n. The horizontal axis corresponds to the sampling ratio, $\delta = m/n$. The vertical axis represents the ratio $\rho = k/m$. The curve labeled as LP-PT is phase transition for l_1 reconstruction [7]. The curves represent the performance of the verification (VA) and enhanced verification (EVA) algorithms for different $p_e = p(\mathbf{x} \neq \hat{\mathbf{x}})$ when the proposed matrix construction is employed. The same measurement matrix was employed to run all the simulations (VA and EVA) for a fixed n/k ratio. The plots depict the performance of the verification algorithm with binary matrices (VA). These matrices were obtained with the structure presented in section III and with $\mathrm{R}(d)$ selected for $p = 1 - k/n$ as indicated by equations (6) and (5). At least 10^5 Monte Carlo simulations were run per point.

The plots obtained with VA and EVA show a dual behavior: both have the same performance as long as it is possible to generate graphs without cycles of length 4 (for $\delta < 0.05$ in this case, i.e. $n/k > 100$). Once the measurement matrix becomes dense, i.e. for large n/k ratios, the behavior of the algorithms changes: VA performance decreases dramatically whereas EVA performance falls slowly. These results can be compared with the l_1 theoretical limit for asymptotic n. Figure 1 shows that VA can outperform the l_1 in an approximate range between $\delta \in (0.05, 0.25)$ and EVA between $\delta \in (0, 0.25)$, both employing measurement matrices constructed with the method proposed in this paper. Notice that, the latter range coincides with $n/k \geq 100$ for $n = 16000$.

VII. Conclusions

This paper has proposed a graph-based CS scheme that outperforms sudocodes irrespective of the sparsity ratio and also outperforms VA in very sparse scenarios when the measurement matrix is binary. The performance enhancement for sparsity ratios where a length-4 cycle-free graph can be constructed is obtained employing the proposed matrix structure and check node degree distribution, specially aimed to exploit the features of the VA algorithm. For larger sparsity ratios, an enhanced decoding algorithm has been introduced that allows to deal with cycles of length-4. The combination of the proposed enhancements over existing methods in the literature allow to obtain a low complexity CS scheme with good performance that outperforms the l_1 theoretical limits in the noiseless setting.

Further work must be done to improve the performance of this scheme in the noiseless setting, first by finding new rules to enhance the performance of the reconstruction algorithm and finding new methods to design binary matrices to exploit both the enhancement of the EVA and these new rules.

References

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

[2] E.J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?,” *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406 –5425, December 2006.

[3] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” *Constructive Approximation*, vol. 28, pp. 253–263, 2008.

[4] Venkat Chandar, “A negative result concerning explicit matrices with the restricted isometry property,” Tech. Rep., 2008.

[5] T. Blumensath and M.E. Davies, “Stagewise weak gradient pursuits,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4333 –4346, November 2009.

[6] R. Calderbank, S. Howard, and S. Jafarpour, “Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property,” *IEEE J. of Sel. Topics in Signal Process.*, vol. 4, no. 2, pp. 358 – 374, April 2010.

[7] David L. Donoho, Arian Maleki, and Andrea Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[8] W. Yihong and S. Verdu, “Renyi information dimension: Fundamental limits of almost lossless analog compression,” *IEEE Trans. Inform. Theory*, vol. 56, no. 8, pp. 3721 –3748, August 2010.

[9] P. Schniter, “Turbo reconstruction of structured sparse signals,” in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, March 2010, pp. 1 –6.

[10] S. Sarvotham, D. Baron, and R.G. Baraniuk, “Sudocodes - fast measurement and reconstruction of sparse signals,” in *2006 IEEE International Symposium on Information Theory*, July 2006, pp. 2804 –2808.

[11] Fan 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.

[12] Fan Zhang and H.D. Pfister, “Analysis of verification-based decoding on the q -ary symmetric channel for large q,” *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6754 –6770, October 2011.

[13] P.M. Olmos, J.J. Murillo-Fuentes, and F. Perez-Cruz, “Tree-structure expectation propagation for decoding LDPC codes over binary erasure channels,” in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, June 2010, pp. 799 –803.