Windowed Encoding of Spatially Coupled LDGM Codes for Lossy Source Compression

Ahmad Golmohammadi*, David G. M. Mitchell*, Jörg Kliewer†, and Daniel J. Costello, Jr.‡
*Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA, {golmoham, dgmm}@nmsu.edu
†Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA, jkliewer@njit.edu
‡Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, USA, costello.2@nd.edu

***Abstract*—Recently, it has been shown that a class of spatially coupled low-density generator-matrix (SC-LDGM) code ensembles displays distortion saturation for the lossy binary symmetric source coding problem with the belief propagation guided decimation (BPGD) algorithm, i.e., the BPGD distortion approaches the optimal expected distortion of the underlying ensemble asymptotically in code length. Here, we investigate the distortion performance of a practical class of protograph-based SC-LDGM code ensembles and demonstrate distortion saturation numerically. Moreover, we propose an efficient windowed encoding (WE) algorithm that takes advantage of the convolutional structure of the SC-LDGM codes. By using the WE algorithm, a distortion very close to the rate-distortion limit can be achieved for a fixed compression rate with low-to-moderate encoding latency.**

I. Introduction

Compared to lossless compression, where the source must be reconstructed identically from its compressed version, the *lossy* source compression problem requires the reconstructed data to be correct only up to some specified distortion measure. Applying linear codes for lossy source compression of discrete sources is a classical idea and various bounds on distortion performance have been proposed for different types of linear codes, e.g., trellis codes [1] and low-density parity-check (LDPC)-like codes [2]. Low-density generator-matrix block codes (LDGM-BCs), dual to LDPC block codes (LDPC-BCs), were shown to achieve the rate-distortion lower bound under optimal encoding as the average node degrees increase [3], and lower bounds on the distortion have been derived both for random ensembles [4] and for specific code constructions [5].

Like LDPC-BCs, LDGM-BCs can be defined on a sparse graph and are amenable to low complexity message passing algorithms. Unfortunately, this approach typically fails because a lossy source coding problem has multiple optimal (or near-optimal) solutions and one cannot find the relevant fixed point without reducing the solution space. Heuristic decimation algorithms based on belief propagation (BP) [3], [6] and survey propagation [3], [7] have been proposed for lossy data compression for both linear and non-linear code variants and have been successfully applied to k-satisfiability (k-SAT) constraint problems [6]. Also, good LDGM-BCs have been designed for the binary erasure source [8]; however, to the best of our knowledge, there have been no constructive LDGM-BC designs proposed with performance guarantees for general binary discrete sources. Indeed, the best known code designs are heuristic in nature and typically use dual codes to LDPC-BCs that were optimized for the binary symmetric channel (see, e.g., [3], [9], [10]).

Spatially coupled LDGM (SC-LDGM) codes can be obtained by coupling together (connecting) a series of L LDGM-BC graphs to make a larger connected graph with a convolutional-like structure. In [11], [12], a belief propagation guided decimation (BPGD) algorithm was applied to a class of SC-LDGM codes with regular check node degrees and Poisson distributed variable node degrees for lossy source compression of the binary symmetric source. It was demonstrated there that the SC-LDGM code ensembles achieve *distortion saturation*, in the sense that the distortion of the SC-LDGM code ensemble approaches the optimal distortion for the underlying LDGM-BC ensemble as the coupling length L and code length tend to infinity, which, in turn, approaches the rate distortion (RD) limit as the node degrees increase. As a result, SC-LDGM codes have great promise for the lossy source coding problem, but the large code lengths required for distortion saturation with the standard BPGD algorithm make them unattractive in practice.

In this paper, we present a practically interesting (J, K)-regular SC-LDGM code construction based on protographs, which are amenable to efficient implementation [13]. We first demonstrate distortion saturation numerically for these protograph-based code ensembles. To combat the need for very long code lengths, we propose a novel low-latency windowed encoding (WE) scheme and demonstrate distortion performance close to the RD limit with moderate latency for the binary symmetric source. To the best of our knowledge, regular SC-LDGM codes are the first regular LDGM constructions that perform close to the RD limit for low complexity encoding and moderate code length.

II. SC-LDGM Code Ensembles for Lossy Compression

A. Lossy source compression

In this paper, we consider compressing the *symmetric Bernoulli source*, where the source sequence $\mathbf{s} = (s_1, s_2, ..., s_n) \in F_2^n$ consists of independent and identically distributed (i.i.d.) random variables with $\mathbb{P}(s_i = 1) = 1/2$. We wish to represent a given source sequence by some codeword $\mathbf{z} \in F_2^m$ from a given code C containing $2^m = 2^{nR}$

This work was supported in part by NSF grants CCF-1439465, CCF-1440001, and CCF11-61754.

978-1-5090-1805-5/16/$31.00 ©2016 IEEE

codewords, where $m \ll n$ and $R = \frac{m}{n}$ is the *compression rate*. The codeword $\mathbf{z}$ is used to reconstruct the source sequence as $\hat{\mathbf{s}}$, where the mapping $\mathbf{z} \rightarrow \hat{\mathbf{s}}(\mathbf{z})$ depends on the code C. The quality of reconstruction is measured by a *distortion metric* $d : \mathbf{s} \times \hat{\mathbf{s}} \rightarrow \mathbf{R}^+$ and the resulting source encoding problem is to find the codeword with minimal distortion, i.e., $\hat{\mathbf{z}} = \arg \min_{\mathbf{z}} d(\mathbf{s}, \hat{\mathbf{s}})$. For a symmetric Bernoulli source, the typical measure of distortion is the *Hamming metric* $d_H(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{n}\sum_{i=1}^{n} |s_i - \hat{s}_i|$, and thus the average quality of the reconstruction is measured as $D = \mathbb{E}_{\mathbf{s}}[d_H(\mathbf{s}, \hat{\mathbf{s}})]$. For any encoding scheme, the average distortion is bounded below by D_{Sh}, defined implicitly by the rate-distortion function $h(D_{Sh}) = 1 - R, D_{Sh} \in [0, 0.5]$, where $h(\cdot)$ is the binary entropy function. The goal is to find an encoding scheme that achieves the rate-distortion bound for a given R.

B. Protograph-based SC-LDGM codes

An LDGM-BC or code ensemble can be represented by means of a *protograph* [14]. For an LDGM-BC ensemble, a protograph is a small bipartite graph that connects a set of n source (information) and m code (compressed) nodes to a set of n generator nodes by a set of edges, and it can be represented by a generator or *base* biadjacency matrix $\mathbf{B}$, where $B_{x,y}$ is taken to be the number of edges connecting generator node g_y to code node z_x. The generator matrix $\mathbf{G}$ of a protograph-based LDGM-BC can be created by replacing each non-zero entry in $\mathbf{B}$ by a sum of $B_{x,y}$ non-overlapping permutation matrices of size $M \times M$ and each zero entry by the $M \times M$ all-zero matrix. Fig. 1 depicts the protograph representation of a $(3, 6)$-regular LDGM-BC ensemble with the all-ones base matrix $\mathbf{B}$ of size 3×6. The white and black circles represent source (information) and code (compressed) symbols, respectively. After compression, the reconstructed source symbols $\hat{s}_i$ are obtained by a modulo 2 summation at the generator nodes.

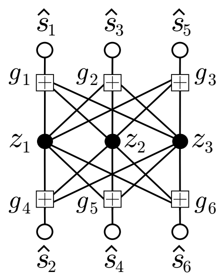


Fig. 1: Protograph representing a $(3, 6)$-regular LDGM-BC ensemble.

SC-LDGM codes are constructed by *coupling* together a series of L disjoint, or uncoupled, LDGM-BC graphs into a single coupled chain. Starting from a $b_c \times b_r$ block base matrix $\mathbf{B}$, an "edge-spreading" construction can be used to form the base matrix of an SC-LDGM code ensemble with *coupling length* L as

$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ \vdots & \mathbf{B}_1 & \ddots & \\ \mathbf{B}_w & \vdots & \ddots & \mathbf{B}_0 \\ & \mathbf{B}_w & & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_w \end{bmatrix}_{(L+w)b_c \times Lb_r}, \quad (1)$$

where $\mathbf{B}_0 + \mathbf{B}_1 + \cdots + \mathbf{B}_w = \mathbf{B}$, w denotes the *coupling width*, and the $b_c \times b_r$ matrices $\mathbf{B}_i$ are referred to as *component base matrices*, $i = 0, 1, \ldots, w$. An ensemble of SC-LDGM codes can then be formed from $\mathbf{B}_{[0,L-1]}$ using the protograph construction method described above. The *design compression rate* of the ensemble of SC-LDPC codes is

$$R_L = \left(1 + \frac{w}{L}\right)\frac{b_c}{b_r}, \quad (2)$$

where we note that R_L is monotonically decreasing and approaches b_c/b_r as $L \rightarrow \infty$.

In this paper we consider (J, K)-regular SC-LDGM code ensembles where $\mathbf{B} = [J, J]$ and the component base matrices are chosen as $\mathbf{B}_0 = \mathbf{B}_1 = \cdots = \mathbf{B}_w = [1, 1]$ for $w = J - 1$. In order to improve distortion performance for BP algorithms, analogous to [11] we found it necessary to reduce the generator node degrees at the start of the chain (this will be explained in Section III). This was achieved by deleting the first $J - 2$ rows of (1). The resulting ensembles are denoted $\mathrm{SC}(J, K)$ and have modified design compression rate

$$R_L = \left(1 + \frac{w - J + 2}{L}\right)\frac{b_c}{b_r} \xrightarrow[L\rightarrow\infty]{} \frac{b_c}{b_r}. \quad (3)$$

Fig. 2 depicts the protograph representation of the $\mathrm{SC}(3, 6)$ SC-LDGM code ensemble.

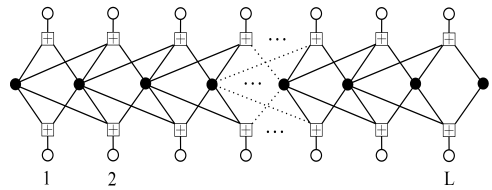


Fig. 2: Protograph of the $(3, 6)$-regular SC-LDGM ensemble $\mathrm{SC}(3, 6)$.

III. Source Reconstruction Using SC-LDGM Codes

A. Decimation Algorithm

Although the sum product algorithm (SPA) can achieve excellent performance for channel coding problems, directly applying it to the lossy source coding problem does not give satisfactory results because the approximate marginal calculated by the SPA does not provide reliable information about optimal encoding [3]. Modified BP algorithms, which include a *decimation* step and are known as the hard-decimation [3] and soft-decimation [9] algorithms, have been proposed. According to [3], hard-decimation involves two phases: 1) message passing, which approximates the marginal distributions, and 2) decimation, where some code nodes are set to their preferred values and the graph is reduced by removing decimated code nodes. The procedure runs until the algorithm assigns a value to all code nodes. In the soft-decimation algorithm, the decimation process is performed by modifying the BP equations at each iteration to assign appropriate *bias* values and, as such, there is no need to reduce the graph. The BP update equations at each iteration t for soft decimation are given as follows [9]:

Code to generator node messages:

$$R_i^{(t+1)} = \sum_{a \in G(i)} R_{a \rightarrow i}^{(t)}, \; R_{i \rightarrow a}^{(t+1)} = \sum_{b \in G(i) \setminus a} R_{b \rightarrow i}^{(t)} \quad (4)$$

Generator to code node messages:

$$R_{a\to i}^{(t+1)} = \frac{1}{\mu} R_{i\to a}^{(t)} + 2(-1)^{s_a+1} \tanh^{-1}\left(\beta \prod_{j\in Z(a)\setminus i} B_{j\to a}^{(t)}\right) \quad (5)$$

Code node bias update:

$$B_i^{(t)} = -\tanh\left(\frac{R_i^{(t)}}{2}\right),\; B_{i\to a}^{(t)} = -\tanh\left(\frac{R_{i\to a}^{(t)}}{2}\right) \quad (6)$$

where $R_{i\to a}^{(t)}$, $R_{a\to i}^{(t)}$, and $B_{i\to a}^{(t)}$ denote the message sent from code node i to generator node a, the message sent from generator node a to code node i, and the bias associated with $R_{i\to a}^{(t)}$ at iteration t, respectively, and β and μ are parameters. Then $R_i^{(t)}$ and $B_i^{(t)}$ denote the likelihood ratio of code node i and the bias associated with $R_i^{(t)}$, repectively. Also, for an LDGM graph Γ with code nodes Z and generator nodes G, $Z(a)$ denotes the set of all code node indices connected to generator node a and $G(i)$ denotes the set of all generator node indices connected to code node i, $\forall i \in Z$ and $\forall a \in G$. The decimation process is embedded in (5) by the $\frac{1}{\mu} R_{i\to a}^{(t)}$ term, which scales the belief of the code nodes at each iteration [9]. Here, the parameter $\mu > 0$ determines the accuracy of the approximation. Since β and μ are free parameters, they can be combined as $\beta = (1-\xi)/(1+\xi)$ and $\mu = 1/\xi$ for simplicity. This choice was shown to yield good numerical results in [9].

The BP algorithm used in this paper is a mixture of both hard and soft decimation. The algorithm uses the soft decimation equations given above, but after each iteration it searches for a code node with maximum bias value. This code node is then decimated (fixed) and the current graph reduced following the hard decimation rule. This modification is applied to force decimation to occur in designated areas of the code graph in order to take advantage of the convolutional structure of SC-LDGM codes. The algorithm is described in detail as Algorithm A, where t indicates the iteration number, $\Gamma^{(t)}$ is the LDGM code graph at iteration t, i is the location of code node i, and z_i represents the binary value assigned to code node i. The initial code to generator node messages, $R_{i\to a}^{(0)}$, are set to $+0.1$ and reset at iteration 1.

Algorithm A: Soft-Hard Decimation

1) At iteration $t = 0$, initialize graph instance $\Gamma^{(t=0)}$
2) Update equations (4), (5), and (6)
3) Find maximum bias $B^{(t)} = \max_i\{B_i^{(t)} |\ i \text{ not fixed}\}$
4) **If** $B^{(t)} < 0$ then
   $z_i \leftarrow '1'$
   else
   $z_i \leftarrow '0'$
5) Decimate graph as
   a) $\forall a \in G(i)$, $s_a \leftarrow s_a \oplus z_i$ (update source symbols)
   b) Reduce the graph as $\Gamma^{(t+1)} = \Gamma^{(t)} \setminus \{i\}$ (remove code node i and all its edges)
6) **If** there exist any unassigned code symbols i
   go to 2)
   else
   exit algorithm and return code symbols

B. Experimental results

In this section, the results of various experiments of a C++-based implementation of Algorithm A are reported for different SC-LDGM code parameters. Codes were obtained by randomly lifting the protograph and all results are obtained by averaging over 1000 trials. For each simulation, ξ was determined experimentally to three decimal places to give the lowest distortion value.

1) Effect of increasing coding length L: We first consider SC-LDGM codes with fixed lifting factor $M = 512$ and L ranging from $L = 4$ to $L = 100$. Fig. 3 presents the average deviation from the RD limit obtained with respect to codeword length for SC$(3,6)$, SC$(4,8)$, and SC$(5,10)$ codes. It can be seen from Fig. 3 that the average distortion deviation obtained by increasing L decreases at an exponential rate and saturates to a value close to the RD limit; moreover, for fixed M, the gap decreases with increasing density J.[1]

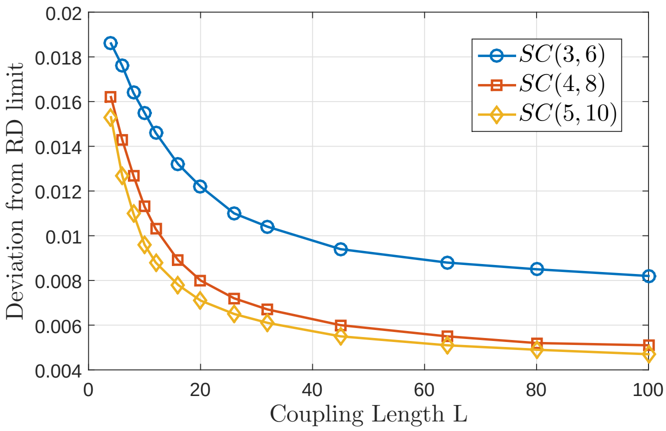


Fig. 3: Average distortion values obtained for SC$(3,6)$, SC$(4,8)$, and SC$(5,10)$ codes for various code lengths with fixed $M = 512$ and increasing L.

2) Effect of increasing M: Table I shows the effect of increasing the lifting factor M. Here we consider SC$(4,8)$ codes with fixed $L = 16$ and corresponding rate $R = 0.5313$ with RD limit $D_{Sh} = 0.09992$. We observe that, as expected, increasing M reduces the average distortion and we obtain saturating values approaching the RD limit. Since SC-LDGM codes experience saturation in their distortion values both for increasing L and M, combining these results (by letting $M \to \infty$ and $L \to \infty$), implies that SC(J,K) codes numerically approach the optimal distortion values of the underlying (J,K)-regular SC-LDGM codes. For example, for the graph of an SC$(4,8)$ with $L = 100$ and $M = 512$, we obtained an average distortion of $D = 0.113463$, where the minimal distortion given in [12] is $D_{opt} = 0.1111$.

3) Reducing complexity: Fig. 4 shows the distortion evolution over each section of the graph of an SC$(3,6)$ code with $L = 45$, $M = 512$ and 23552 code nodes over various iterations t of Algorithm A. Note that the behavior is similar to the wave-like decoding of SC-LDPC codes for channel coding [15], where the nodes at the left end of the graph generate

[1]Note that the compression rate varies according to (2). Each point was compared to the RD limit for the given rate.

TABLE I: Effect of the lifting factor M on SC(4, 8) codes with fixed $L = 16$ and compression rate $R = 0.5313$.

Lifting factor (M)	Distortion	Deviation from RD limit
128	0.112675	0.0128
192	0.110917	0.0110
256	0.110074	0.0102
320	0.109571	0.0097
640	0.108645	0.0087
832	0.108478	0.0086
1024	0.108396	0.0085
1760	0.108225	0.0083
2048	0.108168	0.0082

strong bias values and reliable information and decimation propagates through the graph from left to right with increasing iterations. Initially, the distortion is around 0.5, and then after a few iterations the degree two generator nodes (at the left side of the graph) facilitate convergence of the attached code nodes, thereby generating large bias values and starting an "encoding wave". Note that if we decimate a single code node per iteration, the algorithm requires 23552 iterations to terminate, but Fig. 4 shows that after decimating only 1500 code nodes, the bias values have already saturated and the algorithm can be stopped. Note that the graph is not symmetric (see Fig. 2), hence the encoding wave moves from left to right.[2] The propagation of reliable bias values from left to right motivates the windowed encoding scheme considered in Section IV.

Remark: The complexity can be further decreased by decimating multiple code nodes per iteration. This could be done in a fixed way, i.e., by decimating x nodes per iteration, or by using threshold values, i.e., by decimating nodes only when a sufficiently large bias value is obtained. Both approaches are the subject of ongoing study.

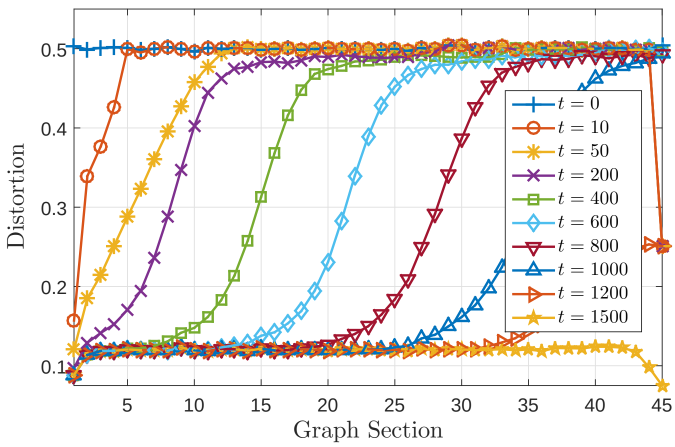


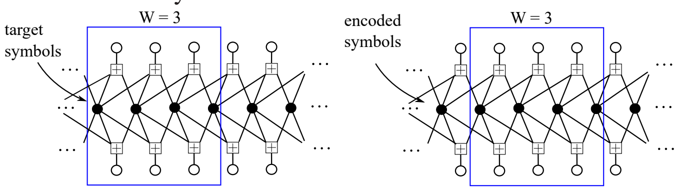
Fig. 4: Distortion evolution of an SC(3, 6) code with $L = 45$ and $M = 512$ over various iterations of Algorithm A.

IV. Windowed Encoding (WE) Scheme

A. Windowed encoding

For practical implementations of SC-LDGM codes, it is essential to reduce the encoding latency. To this end, we propose a novel sliding window encoder, where a window of size W (containing W sections of the graph) slides over the graph. At each window position, the modified BP algorithm is applied to a fraction of the code nodes and all their neighboring generator nodes in order to encode a subset of code symbols, called *target* symbols. After encoding the current target symbols (i.e., when they are all decimated), the window slides over one section and again executes the modified BP algorithm to encode the next set of target symbols, using both the current source symbols and some previously encoded target symbols. Fig. 5 shows the WE procedure on the protograph of a SC(3, 6) code with window size $W = 3$ (covering 3 graph sections, or $3M$ code symbols). Here $2M$ source symbols enter the window at each window position and M source symbols leave (are encoded). Details are given in Algorithm B. In addition to the previous notation, $W^{(t)}$ denotes the window at iteration t, $W_Z^{(t)}$ denotes the set of code nodes inside the window, $W_G^{(t)}$ denotes the set of generator nodes inside the window, and $T\left(W^{(t)}\right)$ denotes the set of target symbols inside the window. Finally, we use $W_{Z_p}^{(t)}$ and $W_{Z_f}^{(t)}$ to denote the set of past code nodes (to the left of the window) and future code nodes (to the right of the window), respectively, and $W_{G_p}^{(t)}$ and $W_{G_f}^{(t)}$ are defined similarly.

[2]We found that a symmetric graph resulted in large distortion values in the center of the chain when the two encoding waves (left to right and right to left) met and BP could not agree on the code symbol values.



(a) SC-LDGM code: time T (b) SC-LDGM code: time T + 1

Fig. 5: The WE procedure operating on the protograph of an SC(3, 6) code.

Algorithm B: Soft-Hard Decimation for windowed encoding

1) At iteration $t = 0$, initialize graph instance $\Gamma^{(t=0)}$
2) $\forall i \in W_Z^{(t)}, a \in W_G^{(t)}$ update equations (4), (5), and (6) such that
   a) there is no incoming or outgoing message $\forall a \in W_{G_f}^{(t)}$
   b) there are incoming messages $\forall i \in W_{Z_p}^{(t)}$ and $\forall a \in W_{G_p}^{(t)}$, but no outgoing messages to those nodes
3) Find maximum bias over target symbols $B^{(t)} = \max_i \{B_i^{(t)} | i \text{ not fixed}, i \in T(W^{(t)})\}$
4) **If** $B^{(t)} < 0$ then
   $z_i \leftarrow '1'$
   else
   $z_i \leftarrow '0'$
5) Decimate graph as
   a) $\forall a \in G(i)$, $s_a \leftarrow s_a \oplus z_i$ (update source symbols).
   b) Reduce the graph as $\Gamma^{(t+1)} = \Gamma^{(t)} \backslash \{i\}$ (remove code node i and all its edges)
6) **if** there exist any unassigned code symbols $i \in T(W^{(t)})$, go to 2)
   else if all source stream symbols are not encoded, shift window to the next position, go to 2)
   else exit algorithm and return code symbols

B. Windowed encoding results

In this section, we present numerical results for WE of SC(4, 8) codes as an example, but similar results were also obtained for other values of J and K. (Note that the performance of the WE algorithm is sensitive to the choice of ξ which was determined numerically for each code.) We considered a SC$(4,8)$ code with $L = 100$ and $M = 512$, which has a corresponding rate of $R = 0.5050$ and RD limit $D_{Sh} = 0.1084$. Applying Algorithm B (windowed encoding) with $W = 10$ and $\xi = 0.02$, the obtained distortion is $D = 0.1172$, while applying Algorithm A (block encoding) gives $D = 0.1135$, indicating there is only a slight loss in distortion performance for a sufficiently large W. Fig. 6 shows the distortion deviation from the RD limit for several SC$(4,8)$ LDGM code lengths n with varying L and $W = 4, 5, \ldots, 16$, but fixed $M = 512$ (encoding latency is equal to $2MW$). We observe that by increasing the code length and latency, the average distortion again saturates to a value close to the bound for the given rate (cf. (2)). Also note that increasing W (the latency) beyond a certain value improves the distortion only slightly. For example, we see from Fig. 6 that only moderate improvements in distortion are obtained by increasing W beyond 8.

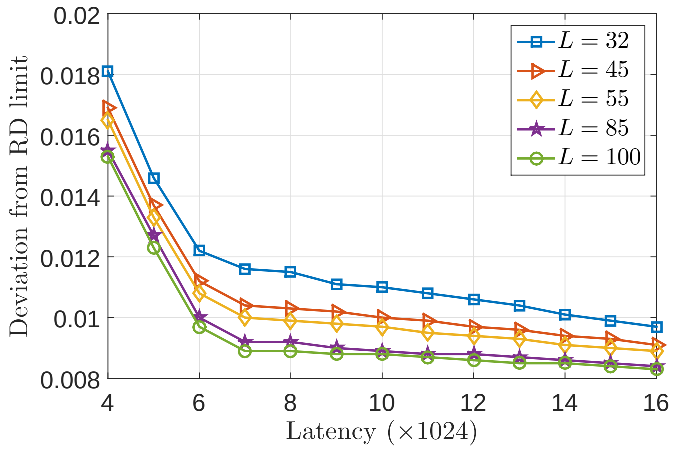


Fig. 6: Average distortion deviation from the RD limit for various SC$(4,8)$ codes with increasing window size W.

Fig. 7 plots the average distortion as a function of latency obtained by applying Algorithm B to SC$(4,8)$ codes with $L = 32$ and $W = 4, 5, \ldots, 12$ with $M = 512$, 1024, and 2048. Also plotted for comparison is the average distortion of a highly irregular LDGM-BC optimized for the soft decimation algorithm [9]. We observe that the *regular* SC-LDGM codes outperform the irregular LDGM-BC beyond a certain latency and that performance tradeoffs can be achieved for SC-LDGM codes by varying M and W. In particular, increasing W improves the performance of the encoder, while increasing M improves the performance of the code.

V. Conclusion

In this paper we introduced a new construction of (J, K) regular SC-LDGM codes based on protographs for lossy source coding. The proposed BP-based decimation algorithm was shown to perform well, giving average distortion values close to the RD limit. We then presented a novel WE algorithm that showed little performance degradation for a sufficiently large window size, allowing good source reconstruction performance to be obtained for moderate encoding latency. There are several features of the algorithm that can be improved, such as reducing the number of required iterations and designing good convolutional protographs that permit shorter window sizes. These, along with extensions to other sources and a comparison to other encoding algorithms, are the subject of ongoing work.

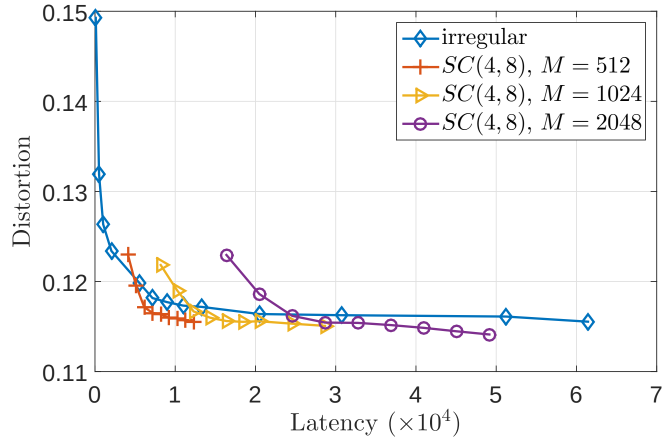


Fig. 7: Distortion versus latency for different lifting factors and $L = 32$.

References

[1] A. Viterbi and J. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 325-332, Mar. 1974.
[2] Y. Matsunaga and H. Yamamoto, "A coding theorem for lossy data compression by LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2225-2229, Sept. 2003.
[3] M. J. Wainwright, E. Maneva, and E. Martinian. "Lossy source compression using low-density generator matrix codes: Analysis and algorithms." *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1351–1368, 2010.
[4] A. Dimakis, M. Wainwright, and K. Ramchandran, "Lower bounds on the rate-distortion function of LDGM codes," in *Proc. IEEE Inf. Theory Workshop*, pp. 650-655, 2007.
[5] S. Kudekar and R. Urbanke, "Lower bounds on the rate-distortion function of individual LDGM codes," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, pp. 379-384, 2008.
[6] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborova, "Gibbs states and the set of solutions of random constraint satisfaction problems," *Proc. of the National Academy of Sciences of the United States of America*, vol. 105, no. 25, pp. 10 318-10 323, 2007.
[7] S. Ciliberti and M. Mezard, "The theoretical capacity of the parity source coder," *J. Stat. Mech.*, vol. 10, no. P10003, pp. 1-14, 2006.
[8] E. Martinian and J. S. Yedidia, "Iterative quantization using codes on graphs," in *Proc. Allerton Conference on Comm., Control, and Computing*, Monticello, IL, Oct. 2003.
[9] D. Castanheira and A. Gameiro, "Lossy source coding using belief propagation and soft-decimation over LDGM codes," in *Proc. IEEE Int. Symp. Personal Indoor and Mobile Radio Communications*, Istanbul, Turkey, Sept. 2010.
[10] T. Filler and J. Fridrich, "Binary quantization using belief propagation with decimation over factor graphs of LDGM codes," in *Proc. 45th Annual Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, Oct. 2007.
[11] V. Aref, N. Macris, M. Vuffray, "Approaching the Rate-Distortion Limit With Spatial Coupling, Belief Propagation, and Decimation," in *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3954–3979, July 2015.
[12] V. Aref, N. Macris, R. Urbanke, M. Vuffray, "Lossy source coding via spatially coupled LDGM ensembles," in *Proc. IEEE Int. Symp. Inf. Theory*, pp. 373–377, July 2012.
[13] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas in Comm.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.
[14] J. Thorpe, "Low-density parity-check LDPC codes constructed from protographs," *IPN Progress Report, Tech*. Rep. 42-154, Aug. 2003.
[15] D. G. M. Mitchell, M. Lentmaier, D. J. Costello, "Spatially Coupled LDPC Codes Constructed From Protographs," in *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866-4889, Sept. 2015.