Optimized Short-Length Rate-Adaptive LDPC Codes for Slepian-Wolf Source Coding

Fangping Ye, Zeina Mheich, Elsa Dupraz, Karine Amis
IMT Atlantique, Lab-STICC, UBL, 29238 Brest, France

Abstract—**This paper proposes a new construction of rate-adaptive coding schemes based on Low Density Parity Check (LDPC) codes for Slepian Wolf source coding. Unlike standard rate-adaptive source coding schemes, the code construction we propose is based on finite-length code design tools that permit to greatly improve the decoding performance at short to moderate length. In particular, our method permits to limit the number of short cycles in the codes at all rates of interest, and to avoid eliminating some source bits from the code constraints. The proposed method shows a performance improvement of up to an order of magnitude at almost all the considered rates compared to the standard LDPCA construction.**

I. Introduction

Low-density parity-check (LDPC) codes were invented by Gallager in 1962 [1] and they are known to almost approach the Shannon capacity in channel coding. LDPC codes can also be used in Slepian-Wolf (SW) source coding that is the problem of source coding with side information available at the decoder [2]. Several recent source coding problems can be represented as SW source coding, like Distributed Source Coding (DSC) in sensor networks [3], Free Viewpoint Television (FTV) [4] or Massive Random Access (MRA) [5]. For instance, in FTV, the source X represents the current view to be transmitted, while the source Y represents the previously transmitted view.

In the SW scheme, the coding rate depends on the statistical model between the source and the side information. But in the aforementioned applications, this statistical model can change from one transmission to another. Using an LDPC code with a fixed rate may cause either a rate loss or a decoding failure. This is why we need to consider rate-adaptive LDPC codes. There are two traditional LDPC-based rate-adaptive schemes for SW source coding, that are Rateless codes [6] and LDPC Accumulate (LDPCA) codes [7].

The Rateless scheme [6] starts by constructing a low-rate code LDPC code. In order to obtain higher rates, it sends all the parity bits and a part of the source bits. The construction of good low-rate LDPC codes is however a difficult problem and is thus the main issue of this scheme [8]. On the opposite, the LDPCA scheme [7] uses a high-rate LDPC code as initial code. In order to obtain lower rates, it computes accumulated parity bits and sends only a part of the accumulated parity bits. However, the LDPCA construction does not allow a fine control of the code structures at lower rates. In particular, when we use LDPCA, an important amount of short cycles may be created and some constraints for source bits may be eliminated. Both features degrade the decoding performance of LDPCA if not avoided.

The combination of Rateless and LDPCA methods is a way to improve the construction of rate-adaptive LDPC codes. In [9], [10], it is proposed to start with an initial code of rate 1/2 referred to as the mother code and then to apply either the Rateless scheme to get higher rates or the LDPCA scheme to get lower rates. Such a construction avoids the shortage of the Rateless scheme, but does not solve the problems raised by the LDPCA construction. The solutions that were proposed to improve the LDPCA construction only optimize the code construction from design tools that assume asymptotic codeword length [11], [12]. However, in applications such as DSC in sensor networks or MRA, the codeword length can be short (from 100 to 1000 bits). Asymptotic design tools do not permit to address the cycle issue that highly degrades the code performance at these lengths.

In this paper, we propose a new method to replace the LDPCA part in the rate-adaptive code construction proposed in [9]. The method we propose is based on a rate-adaptive structure that was initially proposed in [10] for non-binary LDPC codes. In [10], the code construction is realized from an exhaustive search which is not convenient when the codeword length increases. For instance, a non-binary LDPC code of size 12×24 in GF(256) corresponds to a binary code of size 96×192, for which the exhaustive search becomes unfeasible. In this paper, we propose a more efficient systematic construction that is based on finite-length code design tools and permits a fine control of the code structures at rates lower than $R = 1/2$. In particular, it allows a great reduction of the number of short cycles at these rates and it also avoids constraint elimination for all the source bits. In the end, the proposed method shows better performance of up to an order of magnitude compared to LDPCA at almost all the considered rates.

The paper is organized as follows. Section II describes the existing rate-adaptive LDPC code constructions for SW source coding. Section III presents the rate-adaptive construction of [10]. Section IV introduces our new method for the construction of rate-adaptive LDPC codes. Section V shows the simulation results.

II. LDPC codes for Slepian-Wolf Source coding

This section first explains how LDPC codes can be used for SW source coding. It then describes the standard rate-adaptive code constructions and their limitations.

A. LDPC Codes

Fig. 1. Slepian-Wolf source coding

In the SW source coding scheme depicted in Figure 1, the source Y is used as side information at the decoder in order to reconstruct the source X. In this paper, we consider binary sources X and Y. The source X (respectively Y) generates independent and identically distributed (i.i.d.) symbols $X_1, \cdots, X_n$ (respectively $Y_1, \cdots, Y_n$). The probability mass function of the source X is denoted by $P(X = i), i = \{0, 1\}$. The conditional probability mass function or correlation channel between X and Y is denoted by $P(Y = j \mid X = i), i, j \in \{0, 1\}$. LDPC codes allow to achieve a coding rate close to the theoretical limit $H(X \mid Y)$ of the lossless SW coding scheme [2] (the theoretical rate without side information is $H(X) \geq H(X|Y)$).

Here, since we assume binary sources X and Y, we consider binary LDPC codes. Let $\underline{x}^n = (x_1, x_2, \cdots x_n)^T$ stand for a source vector of length n to be transmitted to the decoder. We denote by H a LDPC parity check matrix with dimension $m \times n$ $(m < n)$. This gives a coding rate $R = m/n$. The matrix H is sparse and its non-zero components are all equal to 1. The syndrome $\underline{s}^m = (s_1, s_2, \cdots s_m)^T$ that is transmitted to the decoder is calculated from $\underline{x}^n$ and H as

$$\underline{s}^m = H \cdot \underline{x}^n \tag{1}$$

The decoder produces an estimate $\hat{\underline{x}}^n$ of $\underline{x}^n$ by applying the Belief Propagation algorithm (BP) to the received syndrom $\underline{s}^m$ and the side information vector $\underline{y}^n$ [2]. The decoding performance highly depends on the choice of the parity check matrix H.

The parity check matrix H can alternatively be represented by a Tanner Graph. The Tanner Graph connects the n Variable Nodes (VN) $x_1, \cdots, x_n$ with the m Check Nodes (CN) $s_1, \cdots, s_m$. There is a connection between a VN x_i and a CN s_j if there is a 1 at the corresponding matrix position $H_{i,j}$. The Tanner Graph representation of H will help us construct an LDPC code with good performance, as we now describe.

B. Construction of the Parity Check Matrix H

LDPC codes can be constructed either from the code degree distribution [13] or from a protograph [14]. Density Evolution [15] permits to evaluate under asymptotic conditions the theoretical threshold of a given degree distribution or a given protograph. The theoretical threshold is the maximum correlation channel parameter that gives a decoding error probability $P_e = 0$. It can thus be used as an optimization criterion. For a given coding rate $R = m/n$, Differential Evolution [16] can be used in order to find the degree distribution or the protograph with the best threshold.

Once the protograph or the degree distribution is fixed, the Progressive Edge Growth algorithm (PEG) [17] can be used to construct the parity check matrix H. The PEG algorithm permits to lower the number of short cycles that could severely degrade the decoding performance of the matrix H.

C. Rate-Adaptive LDPC Codes

In the above code construction, the coding rate R is fixed once for all, which implicitly assumes that the statistical model between X and Y does not vary. But if this statistical model evolves from one transmission to another, sending the data at rate R will cause either a rate loss or a decoding failure. In order to avoid this situation, we can consider rate-adaptive LDPC codes such as Rateless or LDPCA codes.

The Rateless scheme starts by constructing a low-rate LDPC code. If a higher rate is needed, all the syndrom bits $\underline{s}^m$ and a part of the sources bits $\underline{x}^n$ will be sent. However, it is difficult to construct good low-rate LDPC codes [8]. Therefore, it is not desirable to apply the Rateless construction from very low rates.

On the opposite, the LDPCA scheme uses a high-rate LDPC code as initial code. It then computes new accumulated symbols $\underline{a}^m = [a_1, a_2, \cdots, a_m]^T$ from $\underline{s}^m$ as

$$a_1 = s_1,$$
$$a_i = a_{i-1} + s_i, \quad \forall i = \{2, \cdots, m\}$$

If a lower rate is demanded, only a part of the symbols $(a_1, a_2, \cdots, a_m)$ will be sent. The problem of the LDPCA construction is that the accumulator structure is fixed and does not allow to select the combinations of syndrom symbols s_i. The accumulator structure may in particular induce short cycles in the lowest rates and eliminate some source bits from the CN constraints.

Due to the drawbacks of Rateless and LDPCA schemes, an intermediate solution was proposed in [9]. It constructs an initial code of rate $R = 1/2$. It then applies either the LDPCA method to obtain rates lower than $1/2$ or the Rateless method for rates higher than $1/2$. In this way, the shortage of the Rateless method can be avoided. But the LDPCA structure still suffers from the same cycle issue and from the source bit elimination problem, and it needs to be improved.

III. Rate-adaptive Construction

The rate-adaptive construction we propose to replace LDPCA is based on a structure initially introduced in [10] for the construction of non-binary rate-adaptive LDPC codes.

A. Code Construction

In the method proposed in [10], the mother code consists of a matrix H_1 of size $m_1 \times n$ with coding rate $R_1 = m_1/n = 1/2$. The Tanner graph $\mathcal{T}_1$ associated to H_1 connects the n

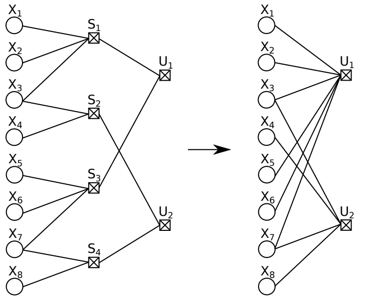

Fig. 2. The left part of the figure shows the combination of $\mathcal{T}_1$ with $\mathcal{T}_{1\to 2}$. The right part of the figure shows the resulting $\mathcal{T}_2$. Here, the matrix $H_{1\to 2}$ is full rank, and one may choose between $\mathcal{S}' = \{s_1, s_2\}$, $\mathcal{S}' = \{s_3, s_4\}$, $\mathcal{S}' = \{s_1, s_4\}$, or $\mathcal{S}' = \{s_2, s_3\}$.

VNs $\mathcal{X} = \{x_1, \cdots, x_n\}$ to m_1 CNs $\mathcal{S} = \{s_1, \cdots, s_{m_1}\}$. For the construction of H_1, we apply a standard code design as described in Section II.

From the mother matrix H_1, we want to construct a daughter matrix H_2 of size $m_2 \times n$, with $m_2 < m_1$, and rate $R_2 = m_2/n < 1/2$. The Tanner graph $\mathcal{T}_2$ associated to H_2 will connect the n VNs of $\mathcal{X}$ to m_2 CNs $\mathcal{U} = \{u_1, \cdots, u_{m_2}\}$. In order to obtain the daughter matrix H_2 from the mother matrix H_1, we construct an intermediate matrix $H_{1\to 2}$ of size $m_2 \times m_1$. The Tanner graph $\mathcal{T}_{1\to 2}$ of $H_{1\to 2}$ connects the m_1 CNs $\mathcal{S}$ of $\mathcal{T}_1$ to the m_2 CNs $\mathcal{U}$ of $\mathcal{T}_2$. With this construction, the matrix H_2 of rate R_2 is then equal to

$$H_2 = H_{1\to 2} H_1. \tag{2}$$

Figure 2 shows an example of the construction of $\mathcal{T}_2$ from $\mathcal{T}_1$ and $\mathcal{T}_{1\to 2}$. Note that LDPCA codes can be seen as a particular case of this construction. The matrix $H_{1\to 2}$ should be chosen not only to give a good decoding performance for H_2, but also to allow H_1 and H_2 to be rate-adaptive in a sense we now define.

B. Rate-adaptive Condition

In our construction, we set the following transmission rule in order to allow H_1 and H_2 to be rate-adaptive. In order to get a rate R_2, we simply transmit all the parity check values defined by the set $\mathcal{U}$, which corresponds to m_2 equations. The decoding is then realized with the matrix H_2. In order to get a rate R_1, we transmit all the parity check values in $\mathcal{U}$ but also a subset $\mathcal{S}' \subseteq \mathcal{S}$ of size $m_1 - m_2$ of the values in $\mathcal{S}$. In order to use the matrix H_1 for decoding, the receiver must be able to recover the values in $\mathcal{S}$ from $\mathcal{U}$ and $\mathcal{S}'$. The sets $\mathcal{U}$ and $\mathcal{S}'$ define a system of m_1 equations with m_1 unknown variables $\mathcal{S}$. The code that results from the choice of (H_1, $H_{1\to 2}$, $\mathcal{S}'$) is then said to be rate-adaptive if this system has a unique solution. It was shown in [10] that if the matrix $H_{1\to 2}$ is full-rank, then there always exists a set $\mathcal{S}' \subseteq \mathcal{S}$ such that (H_1, $H_{1\to 2}$, $\mathcal{S}'$) is a rate-adaptive code.

As a consequence, as long as $H_{1\to 2}$ is full-rank, the decoding performance does not depend on the choice of the set $\mathcal{S}'$, since at rate R_1, the decoder uses H_1 and at rate R_2, the decoder uses H_2. On the contrary, according to equation (2), the decoding performance of the matrix H_2 still heavily depends on the matrix $H_{1\to 2}$. In [10] the matrix $H_{1\to 2}$ is constructed from an exhaustive search. In the following, we propose a more efficient method that permits to avoid as much as possible short cycles in the matrix H_2.

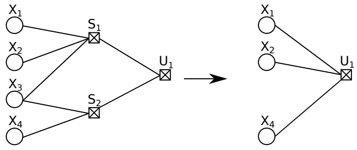

Fig. 3. Since the VN x_3 appears in both s_1 and s_2, combining these CNs into u_1 makes x_3 disappear in H_2.

C. Generalization to Several Rates

The above construction permits to obtain the matrix H_2 of rate $R_2 < R_1$ from the matrix H_1. In order to obtain lower rates $R_I < R_{I-1} < \cdots < R_2 < R_1$, we need to construct the corresponding matrices H_i, $i = 2, \cdots, I$. The matrix H_i is obtained from H_{i-1} by constructing an intermediate matrix $H_{i-1\to i}$ such that $H_i = H_{i-1\to i} H_{i-1}$. In the following, we only describe the construction of the matrix $H_{1\to 2}$. The other matrices $H_{i-1\to i}$ are obtained with exactly the same method.

IV. Construction of the Intermediate Matrices

As the decoding performance of the matrix H_2 heavily depends on the matrix $H_{1\to 2}$, the careful choice of the connections in $\mathcal{T}_{1\to 2}$ is very important. Constructing $H_{1\to 2}$ can be seen as combining the CNs $\mathcal{S}$ of H_1 in order to create the CNs $\mathcal{U}$ of the new parity check matrix H_2. Combining the CNs $\mathcal{S}$ can however cause three issues that could degrade the decoding performance of H_2.

First, combining some of the CNs of H_1 could degrade the connectivity of some VNs in H_2, see Figure 3 for an example. In the worst case, some VNs may not be connected anymore to any CNs in H_2. Second, combining H_1 and $H_{1\to 2}$ may introduce short cycles in H_2 which could severely degrade the decoding performance. As a third issue, the matrix $H_{1\to 2}$ has to be full rank in order to satisfy the rate-adaptive condition described in Section III-B. The construction method we now propose for $H_{1\to 2}$ addresses these three issues through the choice of the degree distribution and of the connections in $H_{1\to 2}$.

A. Degree distribution for $H_{1\to 2}$

In order to be able to construct the intermediate matrix $H_{1\to 2}$ in a systematic way, we first need to choose a degree distribution for $H_{1\to 2}$. As a first constraint, we impose that each CN $s_i \in \mathcal{S}$ is connected to exactly one CN of $\mathcal{U}$. We also impose that each $u_j \in \mathcal{U}$ is connected to one or more CN of $\mathcal{S}$. These two conditions ensure that $H_{1\to 2}$ will be full rank. In addition, the CNs in $\mathcal{S}$ are all of degree 1, and we only need to describe the degree distribution of the $u_j \in \mathcal{U}$.

We denote the degree distribution of the CNs in $\mathcal{U}$ in $H_{1\to 2}$ as $(\underline{\alpha}, \underline{d})$, where $\underline{\alpha} = [\alpha_1, \cdots, \alpha_K]$, $\underline{d} = [d_1, \cdots, d_K]$, and K represents the number of possible degrees. The value α_k denotes the proportion of CNs of $\mathcal{U}$ connected to exactly d_k symbols of $\mathcal{S}$. The degree distribution $(\underline{\alpha}, \underline{d})$ satisfies

$$\frac{m_1}{m_2} = \sum_{k=1}^{K} \alpha_k d_k. \quad (3)$$

Note that the degree distribution $(\underline{\alpha}, \underline{d})$ of the CNs in $\mathcal{U}$ in $H_{1\to 2}$ is not the same as the degree distribution of the CNs $\mathcal{U}$ in H_2. We could think of optimizing the degree distribution $(\underline{\alpha}, \underline{d})$ in $H_{1\to 2}$ by applying density evolution on the resulting degree distribution in H_2. However, here, in order to focus on the finite length code construction, we do not consider optimization from density evolution and we simply choose the degrees d_k as small as possible. For example, if $R_2 = 3/8$, we set $\underline{d} = [1, 2]$ and the proportions α_1 and α_2 are set to $\alpha_1 = 1/2$, $\alpha_2 = 1/2$. Setting low degrees in $H_{1\to 2}$ increases the chances of avoiding short cycles in the resulting H_2.

B. Connections in $H_{1\to 2}$

We now explain how to choose the connections between $\mathcal{S}$ and $\mathcal{U}$ according to the degree distribution $(\underline{\alpha}, \underline{d})$. In our method, the degree of each CN $u_j \in \mathcal{U}$ is selected at random according to the degree distribution $(\underline{\alpha}, \underline{d})$. Then, whatever the degree d_k of a given u_j, we impose the following two conditions in order to choose the CNs of $\mathcal{S}$ that will be connected to u_j:

1) We choose d_k CNs in $\mathcal{S}$ that are not connected to any common VN. This permits to avoid eliminating VN connections in the resulting H_2.
2) We choose the d_k CNs in $\mathcal{S}$ in order to minimize the number of resulting cycles in H_2.

Condition 1) is very easy to verify while condition 2) requires to count the number of cycles in H_2. There exists several methods to calculate the number of shorts cycles in the parity check matrix of an LDPC codes. Here, since we are mainly concerned with short cycles, we choose the method proposed in [18] which is very efficient for the counting of short cycles of length 4, 6, and 8.

Then, in order to construct u_j, we need to select d_k CNs of $\mathcal{S}$. The first CN s_i is selected at random from the set of CNs that have not yet been used in any already constructed u'_j. The next $d_k - 1$ CNs s_i are chosen so as to minimize the number of length-4, length-6, and length-8 cycles introduced in H_2 by the newly created u_j. In order to select the best $d_k - 1$ CNs s_i, we try T possible combinations of $d_k - 1$ CNs selected at random from the set of remaining CNs. As an example, Algorithm 1 shows the algorithm that is used in a particular case $(\underline{\alpha}, \underline{d}) = (1, 2)$ when we only want to minimize the number of length-4 cycles.

C. Construction of the set $\mathcal{S}'$

The degree distribution defined in Section IV-A as well as the code construction proposed in Section IV-B ensure that the matrix $H_{1\to 2}$ is full rank. This guarantees that the rate-adaptive condition presented in Section III is satisfied. In order to completely define the rate-adaptive code $(H_1, H_{1\to 2}, \mathcal{S}')$, we need to define a set $\mathcal{S}'$ of symbols of $\mathcal{S}$ that will be sent together with the set $\mathcal{U}$ in order to obtain the rate R_1.

The set $\mathcal{S}'$ will serve to solve a system of m_2 equations $\mathcal{U}$ with m_2 unknowns $\mathcal{S} \setminus \mathcal{S}'$. For each equation $u_i \in \mathcal{U}$ of degree d_k, we hence decide to put $d_k - 1$ of the d_k CNs connected to u_i into $\mathcal{S}'$. For example, if $u_1 = s_1 \oplus s_2 \oplus s_3$, s_1 and s_2 can be placed into $\mathcal{S}'$. This strategy gives that the set $\mathcal{S}'$ is, as expected, composed by

$$\sum_{k=1}^{K} \alpha_k (d_k - 1) m_2 = m_1 - m_2 \quad (4)$$

different CNs of $\mathcal{S}$. It also guarantees that it is always possible to reconstruct the set $\mathcal{S}$ from $\mathcal{U}$ and $\mathcal{S}'$. In the above example, it indeed suffices to recover s_3 as $s_3 = u_1 \oplus s_1 \oplus s_2$.

Algorithm 1 Construction of the intermediate matrix $H_{1\to 2}$ in the particular case $(\underline{\alpha}, \underline{d}) = (1, 2)$

```
Fix T, C = {1, 2 ··· , m_1}
for j = 1 to m_2 do
  Fix MinCycle = ∞
  Select a value p at random among the set C
  Remove p from the set C
  for t = 1 to T do
    Select a value q at random among the set C
    if s_p and s_q are connected to a common VN, then
      break
    else
      Count the number NbCycles of length-4 cycles in
      H_2 with j-th line H_{2,j} = H_{1,p} ⊕ H_{1,q}
      if NbCycles < MinCycle then
        MinCycle = NbCycles
        Set qchoosen = q
      end if
    end if
  end for
  Set H_{2,j} = H_{1,p} ⊕ H_{1,qchoosen}
end for
```

	$N4(C_1)$	$N6(C_1)$	$N_4(C_2)$	$N_6(C_2)$
$R = 1/2$	0	1856	0	584
$R = 3/8$ LDPCA	256	5232	204	2256
$R = 3/8$ Proposed	184	6823	83	2232
$R = 1/4$ LDPCA	928	15328	568	5600
$R = 1/4$ Proposed	465	19073	200	6130
$R = 1/8$ LDPCA	2632	67384	2336	42620
$R = 1/8$ Proposed	2425	166227	1193	53101

TABLE I
NUMBER OF LENGTH-4 (N4) AND LENGTH-6 (N6) CYCLES FOR THE TWO CONSIDERED CODES

V. SIMULATIONS

In this section, we evaluate the performance of the proposed solution compared to LDPCA. The two codes which we choose

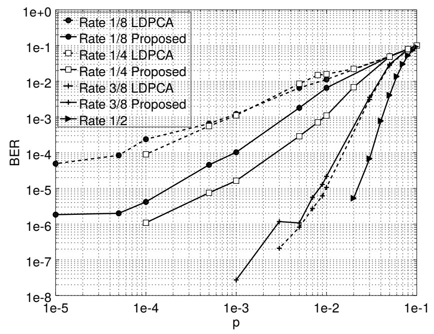

Fig. 4. BER performance on a BSC channel for code C_1 of size 128×256

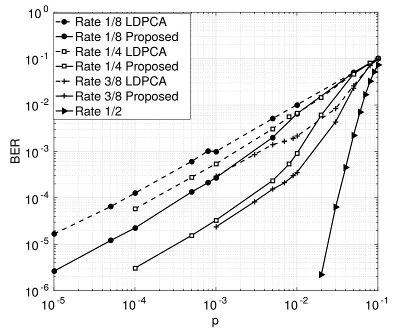

Fig. 5. BER performance on a BSC channel for code C_2 of size 96×192

to evaluate were obtained from [19]. The first code called C_1 code is of size 128×256. The second code called C_2 code is generated from a WiMax protograph. It is of size 96×192. These two codes have rate $R = 1/2$. For each of the two codes, we generated three codes of rates $3/8$, $1/4$, $1/8$, both with the LDPCA method and with our construction. In our algorithm, only short cycles of length 4 were considered.

In our simulations, we assumed a binary symmetric channel of parameter p and we evaluated the Bit Error Rate (BER) performance of both LDPCA and our construction for the two considered codes. The results are shown in Figure 4 for the code C_1 and in Figure 5 for code C_2. In both cases and for almost all rates, we observe that our rate-adaptive construction gives a better performance than the LDPCA. It even outperforms LDPCA by almost one order of magnitude. The only particular case is the rate $3/8$ for code C_1. In this case, LDPCA shows slightly better performance than our method. After a cycle analysis given in Table I from the method of [18], we observe that at all rates, our code construction contains less length-4 cycles than LDPCA, which explains its improved performance. On the opposite, our code construction contains more length-6 cycles than LDPCA. This probably explains why LDPCA works slightly better than our method for the rate $3/8$ for C_1. This can probably be improved in the future. We can however conclude that our method works better than LDPCA in almost all the considered cases.

VI. Conclusion

In this paper, we proposed a novel rate adaptive LDPC code construction method for short-length binary Slepian-Wolf codes. The proposed method permits to avoid VN elimination from the code constraints and minimizes the number of short cycles at all the considered rates. From numerical simulations, it has been demonstrated to be superior to LDPCA code in almost all the considered rates.

VII. Acknowledgement

This work has received a French government support granted to the Cominlabs excellence laboratory and managed by the National Research Agency in the Investing for the Future program under reference ANR-10-LABX-07-01.

References

[1] R. Gallager, “Low-density parity-check codes,” *IEEE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.
[2] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct 2002.
[3] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE signal processing magazine*, vol. 21, no. 5, pp. 80–94, 2004.
[4] A. Roumy and T. Maugey, “Universal lossless coding with random user access: The cost of interactivity,” in *IEEE International Conference on Image Processing*, Sept 2015, pp. 1870–1874.
[5] E. Dupraz, T. Maugey, A. Roumy, and M. Kieffer, “Rate-storage regions for massive random access,” *arXiv preprint arXiv:1612.07163*, 2016.
[6] A. W. Eckford and W. Yu, “Rateless Slepian-Wolf codes,” in *in Asilomar conference on signals, systems and computers*, 2005, pp. 1757–1761.
[7] D. Varodayan, A. Aaron, and B. Girod, “Rate-adaptive codes for distributed source coding,” *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.
[8] S.-Y. Shin, M. Jang, S.-H. Kim, and B. Jeon, “Design of binary LDPCA codes with source revealing rate-adaptation,” *IEEE Communications Letters*, vol. 16, no. 11, pp. 1836–1839, 2012.
[9] K. Kasai, T. Tsujimoto, R. Matsumoto, and K. Sakaniwa, “Rate-compatible Slepian-Wolf coding with short non-binary LDPC codes,” in *Data Compression Conference*, 2010, pp. 288–296.
[10] Z. Mheich and E. Dupraz, “Short length non-binary rate-adaptive ldpc codes for slepian-wolf source coding,” in *IEEE Wireless Communications and Networking Conference*, 2018.
[11] F. Cen, “Design of degree distributions for LDPCA codes,” *IEEE Communications Letters*, vol. 13, no. 7, pp. 525–527, 2009.
[12] C. Yu and G. Sharma, “Improved low-density parity check accumulate (LDPCA) codes,” *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3590–3599, 2013.
[13] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE transactions on information theory*, vol. 47, no. 2, pp. 619–637, 2001.
[14] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *IPN progress report*, vol. 42, no. 154, pp. 42–154, 2003.
[15] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb 2001.
[16] R. Storn and K. Price, “Minimizing the real functions of the icec’96 contest by differential evolution,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, May 1996, pp. 842–844.
[17] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, “Regular and irregular progressive edge-growth tanner graphs,” *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, Jan 2005.
[18] Y. Mao and A. H. Banihashemi, “A heuristic search for good low-density parity-check codes at short block lengths,” in *IEEE International Conference on Communications*, vol. 1, 2001, pp. 41–44.
[19] M. Helmling, S. Scholl, F. Gensheimer, T. Dietz, K. Kraft, S. Ruzika, and N. Wehn, “Database of Channel Codes and ML Simulation Results,” www.uni-kl.de/channel-codes, 2017.