Optimized Rate-Adaptive Protograph-Based LDPC Codes for Source Coding with Side Information

Fangping Ye[1], Elsa Dupraz[1], Zeina Mheich[2], Karine Amis[1]

[1] IMT Atlantique, Lab-STICC, UBL, 29238 Brest, France
[2] University of Surrey, United Kingdom

Abstract—**This paper considers the problem of source coding with side information at the decoder, also called Slepian-Wolf source coding scheme. In practical applications of this coding scheme, the statistical relation between the source and the side information can vary from one data transmission to another, and there is a need to adapt the coding rate depending on the current statistical relation. In this paper, we propose a novel rate-adaptive code construction based on LDPC codes for the Slepian-Wolf source coding scheme. The proposed code design method allows to optimize the code degree distributions at all the considered rates, while minimizing the amount of short cycles in the parity check matrices at all rates. Simulation results show that the proposed method greatly reduces the source coding rate compared to the standard Low Density Parity Check Accumulated (LDPCA) solution.**

I. INTRODUCTION

This paper considers the problem of lossless source coding with side information at the decoder, also called Slepian-Wolf source coding [3]. In this scheme, the objective is to reduce the coding rate of the source X by exploiting the side information Y available at the decoder. This problem has regained attention recently due to its use in many modern multimedia applications like Distributed Source Coding (DSC) [4], Free-Viewpoint Television (FTV) [5], [6], or Massive Random Access (MRA) [7]. For instance, in DSC, the source X represents the data sent by one sensor, and the side information Y is the data already transmitted by other adjacent sensors. As another example, FTV is a video system in which the user freely switches from one view to another; in a soccer game, he may decide to follow a player or to focus on the goal. In FTV, X is the current requested view, and Y represents the views previously received by the user. A key aspect of the aforementioned applications is that the source X and the side information Y are correlated. This allows to reduce the source coding rate to $H(X|Y)$ bits/source symbols [3] instead of $H(X)$ when no side information is available.

Practical Slepian-Wolf source coding schemes can be constructed from error-correction codes such as Low Density Parity Check (LDPC) codes [8]. LDPC codes were first invented by Gallager [9] in the context of channel coding. They can also be used in lossy source coding with side information [10]. For very long codewords (more than 10000 bits), they are known to approach Shannon capacity in channel coding, and to achieve

a coding rate close to the conditional entropy in Slepian-Wolf source coding [11]. For shorter codewords, carefully designed LDPC codes show a reasonable loss in performance compared to Shannon capacity or conditional entropy [12].

The performance of an LDPC code depends on its degree distribution, that gives the amount of non-zero values in the code parity check matrix. The code degree distribution can be described either in a polynomial form [13] or by use of a small graph called protograph [14]. In this paper, we consider the protograph description as it allows the design of capacity-approaching LDPC codes in a simpler way than polynomial degree distributions [15]. At short to medium length, the code performance can also be lowered by short cycles in the code parity check matrix [16]. Therefore, the LDPC code construction is usually realized in two steps. The first step optimizes the protograph for good decoding performance [13], [17], [18]. The second step constructs the parity check matrix from the selected protograph by applying a PEG algorithm [19]–[21] that lowers the amount of short cycles in the parity check matrix.

This standard LDPC code construction is adapted to Slepian-Wolf source coding problems with a fixed statistical relation $P(Y|X)$ between the source X and the side information Y. However, when this statistical relation varies from one data frame to another, such codes may suffer from either rate loss or decoding failure. For instance, in FTV, this statistical relation often changes since different sets of views can be available at the decoder, depending on the successive requests of the users. This issue can be solved by rate-adaptive LDPC codes that allow to adapt the rate depending on the current statistical relation between X and Y. In channel coding, standard solutions to construct rate-adaptive LDPC codes are puncturing [22] and parity check matrix extension [23], [24]. However, in source coding, puncturing leads to a poor decoding performance [25] and parity check matrix extension cannot be applied, as it would require to artificially increase the source sequence length.

In source coding, standard methods to construct rate-adaptive LDPC codes are Rateless codes [26], [27] and Low Density Parity Check Accumulated (LDPCA) codes [25]. Rateless codes start from a low rate code and construct higher rate codes by transmitting a part of the source bits. However, the main issue of this method is that it is difficult to construct good low rate LDPC codes [28]. LDPCA takes the opposite approach by starting from a high rate code. Lower rates are then obtained by puncturing accumulated syndrome bits rather

than puncturing the syndrome bits directly. Unfortunately, the LDPCA accumulator has a fixed regular structure, and it cannot be optimized in order to obtain good degree distributions or to avoid short cycles that could degrade the code performance at lower rates. In order to improve the LDPCA construction, it was proposed in [29] to consider a non-regular accumulator optimized for any rate of interest. The method of [29] optimizes the non-regular accumulator for codes with large length, but it does propose any finite-length code construction that would allow to reduce the amount of short cycles in the low-rate codes. As an intermediate solution, [30] starts with an initial rate $1/2$ and applies Rateless codes construction to increase the rate and LDPCA codes construction to decrease the rate. This permits to avoid the main issue of Rateless codes, but the shortage of LDPCA codes still exists.

In this paper, we consider the intermediate solution of [30] and we replace the LDPCA part by an alternative rate-adaptive LDPC code construction that was initially introduced in [1], [2]. This construction replaces the LDPCA accumulator by intermediate graphs that combine the syndrome bits in order to obtain lower rate codes. In this construction, the intermediate graphs must be full-rank in order to allow the code to be rate-adaptive.

The code design method initially proposed in [1], [2] only considers unstructured finite-length code constructions, that is without design of the degree distributions of the lower rate codes. Here, we introduce a novel design method that allows to select the protographs of the intermediate graphs so as to optimize the decoding performance of all the codes constructed at all rates of interest. We also propose a new algorithm called Proto-Circle that constructs the intermediate graphs according to their protographs, while minimizing the amount of short cycles in the codes at all the considered rates. In addition, the rate-adaptive construction of [1], [2] only permits to consider a small number of rates, in the order of magnitude of the size of the protograph. The code design method we propose in this paper permits to obtain a much larger number of rates, in the order of magnitude of the size of the parity check matrix. Our simulation results show that the proposed rate-adaptive LDPC code construction provides improved performance compared to LDPCA. This comes from the careful protograph selection and from the fact that our method reduces the number of short cycles in the constructed codes at all rates.

The outline of this paper is as follows. Section II presents the Slepian-Wolf source coding problem. Section III describes existing rate-adaptive LDPC code construction for source coding. Section IV restates the rate-adaptive construction of [1], [2]. Section V introduces our code design method by describing the protograph optimization and the Proto-Circle algorithm. Section VI presents our solution to consider an increased number of rates. To finish, Section VII shows our simulation results.

II. Source coding with side information at the decoder

This section describes the lossless coding of a source X with side information Y available at the decoder. The source X generates independent and identically distributed (i.i.d.) symbols $X_1, \cdots, X_n$. For all $i \in \{1, \cdots, n\}$, we assume that the source symbol X_i takes values in a binary alphabet $\{0, 1\}$. The probability mass function of the source X is denoted by $P(X)$. The source bits X_i may represent the pixels of an image or quantized sensor measurements.

In the original SW source coding scheme [3] depicted in Figure 1 (a), a side information Y is available at the decoder and helps the reconstruction of X. The side information Y generates i.i.d. symbols $Y_1, \cdots, Y_n$ and the symbols Y_i belong to an alphabet $\mathcal{Y}$ that can be either discrete or continuous. With a slight abuse of notation, we denote by $P(Y|X)$ the correlation channel between X and Y. If $\mathcal{Y}$ is discrete, $P(Y|X)$ is a conditional probability mass function. If Y is continuous, $P(Y|X)$ is a conditional density. According to [3], the minimum achievable rate for lossless SW source coding is given by $R = H(X|Y)$ bits/source symbol. The side information Y reduces the coding rate, since $H(X|Y) \leq H(X)$.

Alternatively, in this paper, we consider the problem of source coding with several possible side informations at the decoder [31], see Figure 1 (b). In this setup, the decoder has access to one side information $Y^{(t)}$ that belongs to a set of T possible side informations $\{Y^{(1)}, \cdots Y^{(T)}\}$. Each source $Y^{(t)} \in \{Y^{(1)}, \cdots Y^{(T)}\}$ generates a sequence of n i.i.d symbols $Y_1^{(t)}, \cdots, Y_n^{(t)}$, but the decoder only observes one of these length-n sequences. Whatever $t \in \{1, \cdots, T\}$, the side information symbols $Y_i^{(t)}$ belong to the same alphabet $\mathcal{Y}$. However, each of the side information $Y^{(t)}$ corresponds to a different correlation channel described by the conditional probability distribution $P(Y^{(t)}|X)$. For instance, each correlation channel $P(Y^{(t)}|X)$ may correspond to a BSC with a different crossover probability p_t.

For the setup of Figure 1 (b), if the encoder has access to index t by means of e.g. a feedback channel, the minimum achievable rate depends on t and it is given by $R_t = H(X|Y^{(t)})$ [7], [32], [33]. In order to achieve these rates, [7], [34] propose information-theoretic code constructions that allows to construct one single incremental codeword from which we can extract T subcodewords with rates $R_t = H(X|Y^{(t)})$ depending on the side information $Y^{(t)}$ available at the decoder. In this paper, we propose a practical rate-adaptive coding scheme based on LDPC codes which provides such incremental codeword construction.

III. LDPC codes for source coding with side information

For the two source coding problems described in Section II, LDPC codes provide efficient practical coding schemes that perform close to the conditional entropies $H(X|Y)$ or $H(X|Y^{(t)})$ when the codeword length n is large [11]. In this section, we first consider the case with one possible side information Y available at the decoder, and we describe the standard construction of LDPC codes with good performance for the Slepian-Wolf source coding scheme. We then review existing rate-adaptive LDPC-based solutions that provide practical source coding schemes when side information $Y^{(t)}$ available at the decoder comes from a set $\{Y^{(1)}, \cdots Y^{(T)}\}$.
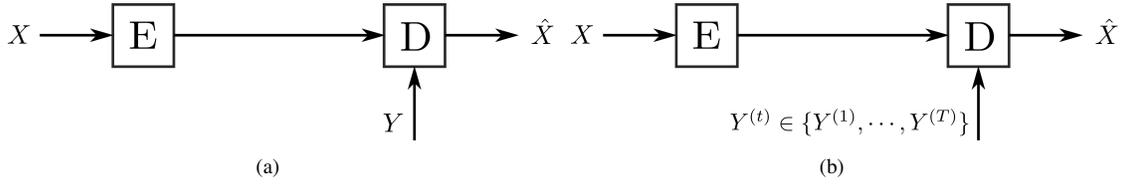
Fig. 1: (a) Slepian-Wolf source coding, (b) Source coding with several possible side informations at the decoder

A. LDPC codes for Slepian-Wolf source coding

In this section, we consider the Slepian-Wolf source coding setup in which one single side information Y is available at the decoder. 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. Consider an LDPC parity check matrix H of size $m \times n$ ($m < n$) and coding rate $R = m/n$. The matrix H is sparse and its non-zero components are all equal to 1. The codeword or syndrome $\underline{c}^m = (c_1, c_2, \cdots c_m)^T$ that is transmitted to the decoder is calculated from $\underline{x}^n$ and H as [8]

$$\underline{c}^m = H\underline{x}^n. \tag{1}$$

Once it receives syndrome $\underline{c}^m$, the decoder produces an estimate $\hat{\underline{x}}^n$ of $\underline{x}^n$ by applying the Belief Propagation algorithm (BP) to $\underline{c}^m$ and the side information vector $\underline{y}^n$ [8], [17].

The parity check matrix H can be represented by a Tanner Graph that connects n Variable Nodes (VN) $X_1, \cdots, X_n$ with m Check Nodes (CN) $C_1, \cdots, C_m$. The decoding performance of LDPC codes highly depends on the choice of the parity check matrix H, as we now describe.

B. LDPC code construction

In this section, for the sake of completeness, we describe standard LDPC code construction from protographs [14], [35]. A protograph $\mathcal{S}$ is a small Tanner Graph of size $S_m \times S_n$ with $S_m/S_n = m/n = R$. A parity check matrix H can be generated from a protograph $\mathcal{S}$ by repeating the protograph structure Z times such that $n = ZS_n$, and by interleaving the connections between the VNs and the CNs. The interleaving can be done by a PEG algorithm [19] that not only permits to satisfy the protograph constraints, but also to lower the number of short cycles that could severely degrade the decoding performance of the matrix H.

The performance of a given parity check matrix H highly depends on its underlying protograph $\mathcal{S}$. Density Evolution [36] permits to evaluate the theoretical threshold of a protograph under asymptotic conditions. In our design, the protograph threshold is evaluated from the method described in [37] that is based on Monte-Carlo simulations. This method was initially proposed for irregular LDPC codes, but it can be easily adapted to protographs. For a given coding rate $R = m/n$, Differential Evolution [18] is an optimization method that permits to find a protograph with a very good threshold.

In order to optimize the protograph, we fix the size $S_m \times S_n$ of the protograph and impose a maximum degree value $d_{\max}$ for the protograph coefficients. Differential Evolution is a

genetic algorithm that starts with an initial population of V protographs and recombines the elements of the population in order to get new protographs. The protographs with the best thresholds (evaluated with density evolution) are then retained in order to form a new population. This process is repeated over L iterations. The recombination operations (mutation step and crossover step) proposed in [18] stand for real vectors, while the protograph coefficients are discrete. In our optimization, we keep the recombination operations of [18] and simply round each obtained real value to the closest integer.

C. Rate-adaptive LDPC codes

In the above LDPC code construction, the coding rate R is fixed once for all. But if the available side information $Y^{(t)}$ comes from a set of possible side informations $\{Y^{(1)}, \cdots, Y^{(T)}\}$, sending the data at fixed rate R will cause either a rate loss or a decoding failure. This is why we now describe rate-adaptive LDPC code constructions that allow to adapt the coding rate depending on the side information $Y^{(t)}$ available at the decoder.

The rate-adaptive Rateless scheme [26], [27] starts by constructing an initial low-rate LDPC code. If a higher rate is needed, a part of the source bits $\underline{x}^n$ will be sent in addition to the syndrome $\underline{c}^m$. However, the major drawback of the rateless scheme is that it difficult to construct good low rate LDPC codes [28], [38][1], and a bad initial low rate code will cause poor performance at any considered higher rate. Therefore, it is not desirable to apply the Rateless construction from very low rates.

On the opposite, the LDPCA scheme [25] starts from a high-rate LDPC code. It then computes new accumulated symbols $\underline{a}^m = [a_1, a_2, \cdots, a_m]^T$ from the syndrome $\underline{c}^m$ (1) as

$$\begin{aligned} a_1 &= c_1, \\ a_i &= a_{i-1} + c_i, \quad \forall i = \{2, \cdots, m\}, \end{aligned} \tag{2}$$

where the binary sum in (2) corresponds to XOR operations. If a lower rate is demanded, only a part of the symbols $(a_1, a_2, \cdots, a_m)$ will be sent. For instance, if the original rate is R and a rate $R/2$ is demanded, only the even symbols $a_2, a_4, a_6, \cdots$ will be transmitted. The decoder will then compute all the differences $a_i - a_{i-2} = c_i + c_{i-1}$, before applying a BP decoder in order to estimate the source vector $\underline{x}^n$ from all the obtained XOR sums $c_i + c_{i-1}$. In this construction, puncturing the source symbols a_i rather than the syndrome

---

[1]Low rate LDPC codes for source coding correspond to high rate LDPC codes for channel coding

bits c_i was shown to better preserve the code structure and to greatly improve the decoding performance [25]. However, in the LDPCA construction, the accumulator structure (2) is fixed and does not allow for an optimization of the combinations of syndrome symbols c_i that are used by the decoder. The accumulator structure may in particular induce short cycles in the lowest rates and eliminate some source bits from the CN constraints. In [29], the LDPCA structure is improved by considering a non-regular accumulator. The non-regular accumulator is designed for any rate of interest by optimizing its polynomial degree distribution under asymptotic conditions. Unfortunately, [29] does not propose any finite-length code construction that could solve the short cycles and VN elimination issues.

Due to the drawbacks of Rateless and LDPCA schemes, an intermediate solution was proposed in [30]. It first 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 construction can be avoided, but the drawbacks of LDPCA remain.

In this paper, we thus propose a novel rate-adaptive construction that replaces the LDPCA part in the solution of [30]. The construction we propose replaces the regular LDPCA accumulator described in (2) by a non-regular structure. In addition, in our construction, this non-regular structure is also described by a protograph. We show that the daughter code protographs can be calculated explicitly from the non-regular accumulator protograph and from the mother code protograph. This allows us to optimize the asymptotic code performance by carefully selecting the code protograph at all rates. We also propose a finite length code construction method that permits to reduce the amount of short cycles in all the considered codes. The proposed method is thus well adapted to the construction of short length LDPC codes.

IV. RATE-ADAPTIVE CODE CONSTRUCTION

The rate-adaptive code design method we propose in this paper is based on a rate-adaptive code structure initially proposed in [1], [2]. For the sake of clarity, this section describes the rate-adaptive code structure of [1], [2]. This construction starts from a mother code of the highest rate and then builds a sequence of daughter codes of lower rates. This section only describes the construction of one code of rate R_2 from a code of rate $R_1 > R_2$. This construction is generalized to more rates later in the paper.

A. Rate-adaptive code construction

In the rate-adaptive construction of [1], [2], the mother code is described by a parity check matrix H_1 of size $m_1 \times n$ with coding rate $R_1 = m_1/n$. The Tanner graph $\mathcal{T}_1$ connects the n VNs $\mathcal{X} = \{X_1, \cdots, X_n\}$ to m_1 CNs $\mathcal{C} = \{C_1, \cdots, C_{m_1}\}$. The matrix H_1 is constructed from a protograph $\mathcal{S}_1$ according to the code design method described in Section III. 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 $R_2 = m_2/n < R_1$.
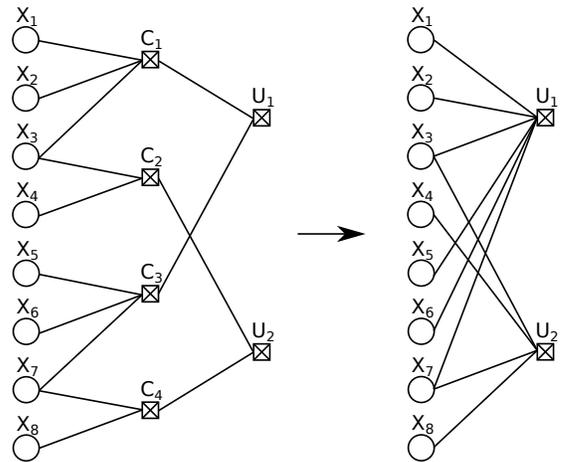

Fig. 2: The left part of the figure shows the combination of $\mathcal{T}_1$ with $\mathcal{T}_{1 \to 2}$. The right part of the figure shows the resulting $\mathcal{T}_2$. Here, the matrix $H_{1 \to 2}$ is full rank, and one may choose between $\mathcal{C}' = \{c_1, c_2\}$, $\mathcal{C}' = \{c_3, c_4\}$, $\mathcal{C}' = \{c_1, c_4\}$, or $\mathcal{C}' = \{c_2, c_3\}$.

The Tanner graph $\mathcal{T}_2$ will connect the n VNs $\mathcal{X}$ to m_2 CNs $\mathcal{U} = \{U_1, \cdots, U_{m_2}\}$.

In the considered construction, the daughter matrix H_2 and the mother matrix H_1 are linked by an intermediate matrix $H_{1 \to 2}$ of size $m_2 \times m_1$ such that

$$H_2 = H_{1 \to 2} H_1. \tag{3}$$

The Tanner graph $\mathcal{T}_{1 \to 2}$ of $H_{1 \to 2}$ connects the m_1 CNs $\mathcal{C}$ of $\mathcal{T}_1$ to the m_2 CNs $\mathcal{U}$ of $\mathcal{T}_2$. 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 intermediate 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 describe.

B. Rate adaptive condition

In the construction of [1], [2], the following transmission rules are set in order to allow H_1 and H_2 related by (3) to be rate-adaptive. In order to get a rate R_2, we simply transmit all the syndrome values $\underline{u}^{m_2}$, which corresponds to m_2 equations defined by the set $\mathcal{U}$. The decoding is then realized with the matrix H_2. In order to get a rate R_1, we transmit all syndrome values in $\underline{u}^{m_2}$ but also a subset $\mathcal{C}' \subseteq \mathcal{C}$ of size $m_1 - m_2$ of the values in $\underline{c}^{m_1}$. This guarantees that the code construction is incremental and that the storage rate is given by $R_1 = \max(R_1, R_2) < R_1 + R_2$. However, in order to use the matrix H_1 for decoding, the receiver must be able to recover the full syndrome $\underline{c}^{m_1}$ from $\underline{u}^{m_2}$ and $\mathcal{C}'$. The code that results from the choice of $(H_1, H_{1 \to 2}, \mathcal{C}')$ is thus said to be rate-adaptive if is satisfies the following condition.

Definition 1 ([1], [2]): The sets $\mathcal{U}$ and $\mathcal{C}'$ define a system of m_1 equations with m_1 unknown variables $\mathcal{C}$. If this system has a unique solution, then the triplet $(H_1, H_{1 \to 2}, \mathcal{C}')$ is said to be a rate-adaptive code.

The following proposition gives a simple condition that permits to verify whether a given intermediate matrix $H_{1\rightarrow2}$ gives a rate-adaptive code.

Proposition 1 ([1], [2]): If the matrix $H_{1\rightarrow2}$ is full rank, then there exists a set $\mathcal{C}' \subseteq \mathcal{C}$ of size $m_1 - m_2$ such that $(H_1, H_{1\rightarrow2}, \mathcal{C}')$ is a rate-adaptive code.

The above proposition shows that if $H_{1\rightarrow2}$ is full rank, it is always possible to find a set $\mathcal{C}'$ that ensures that H_1 and H_2 are rate-adaptive. The decoding performance of H_1 does not depend on the choice of the set $\mathcal{C}'$, since at rate R_1, the decoder uses H_1 and at rate R_2, the decoder uses H_2. On the opposite, according to (3), the decoding performance of the matrix H_2 heavily depends on the matrix $H_{1\rightarrow2}$. In [1], the matrix $H_{1\rightarrow2}$ is constructed from an exhaustive search, which is hardly feasible when the codeword length increases (from 100 bits). In [2], a more efficient method is proposed to construct the intermediate matrix $H_{1\rightarrow2}$ so as to avoid short cycles in H_2. However, the method of [2] does not optimize the theoretical threshold of the degree distribution of H_2, which also influences the code performance. In this paper, we propose a novel method based on protographs for the design of the intermediate matrix $H_{1\rightarrow2}$. This novel method not only allows to optimize the threshold of the protograph of H_2, but also to reduce the amount of short cycles in H_2.

V. INTERMEDIATE MATRIX CONSTRUCTION

This section describes our novel method for the construction of the intermediate matrix $H_{1\rightarrow2}$ introduced in Section IV. The proposed construction seeks to minimize the protograph threshold at rate R_2, and also to reduce the amount of short cycles in the parity check matrix H_2.

A. Protograph $\mathcal{S}_2$ of parity check matrix H_2

In order to construct a good parity check matrix H_2 from the initial matrix H_1, we first want to select a protograph $\mathcal{S}_2$ with a good theoretical threshold. In this section, we consider the following notation. Generally speaking, consider the protograph $\mathcal{S}_g$ of size $S_{m_g} \times S_{n_g}$ associated with the matrix H_g, where $g \in \{1, 2, 1 \rightarrow 2\}$. As a particular case, note that $S_{m_{1\rightarrow2}} = S_{m_2}$ and $S_{n_{1\rightarrow2}} = S_{m_1}$. For all $(i,j) \in \{1, \cdots, S_{m_g}\} \times \{1, \cdots, S_{n_g}\}$, denote by $s_{i,j}^{(g)}$ the coefficient at the i-th row, j-th column of $\mathcal{S}_g$. In the protograph $\mathcal{S}_g$, the CN types are denoted $A_1^{(g)}, \cdots, A_{S_{m_g}}^{(g)}$ and the VN types are denoted $B_1^{(g)}, \cdots, B_{S_{n_g}}^{(g)}$. In the parity check matrix H_g, the set of CNs of type $A_i^{(g)}$ is denoted $\mathcal{A}_i^{(g)}$ and the set of VNs of type $B_j^{(g)}$ is denoted $\mathcal{B}_j^{(g)}$. Finally, denote by $\underline{h}_k^{(g)}$ the k-th row of H_g, and denote by $h_{\ell,k}^{(g)}$ the coefficient at the ℓ-th row, k-th column of H_g.

Based on the above notation, the following proposition gives the relation between the three protographs $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_{1\rightarrow2}$.

Proposition 2: Consider a matrix H_1 with protograph $\mathcal{S}_1$ of size $S_{m_1} \times S_n$, a matrix $H_{1\rightarrow2}$ with protograph $\mathcal{S}_{1\rightarrow2}$ of size $S_{m_2} \times S_{m_1}$, and a matrix $H_2 = H_{1\rightarrow2}H_1$. Also consider the following two assumptions:

1) Type structure: for all $j \in \{1, \cdots, S_{m_1}\}$, $\mathcal{B}_j^{(1\rightarrow2)} = \mathcal{A}_j^{(1)}$.

2) No VN elimination: For all $\ell \in \{1, \cdots, m_2\}$, denote by $\mathcal{N}_\ell^{(1\rightarrow2)}$ the positions of the non-zero components in $\underline{h}_\ell^{(1\rightarrow2)}$. Then, $\forall k_1, k_2 \in \mathcal{N}_\ell^{(1\rightarrow2)}$ such that $k_1 \neq k_2$, and $\forall i \in \{1, \cdots, n\}$, $h_{k_1,i}^{(1)} \neq h_{k_2,i}^{(1)}$.

If these two assumptions are fulfilled, then the matrix

$$\mathcal{S}_2 = \mathcal{S}_{1\rightarrow2}\mathcal{S}_1 \qquad (4)$$

is of size $S_{m_2} \times S_n$ and it is a protograph of the matrix H_2. The operation in (4) corresponds to standard matrix multiplication over the field of real numbers.

Proof: In this proof, for clarity, we denote by $\bigoplus$ the modulo two sums and by $\sum$ the standard sums over the field of real numbers. With the above notation, relation (3) can be restated row-wise as

$$\underline{h}_\ell^{(2)} = \bigoplus_{k=1}^{m_1} h_{\ell,k}^{(1\rightarrow2)} \underline{h}_k^{(1)} = \bigoplus_{j=1}^{S_{m_1}} \bigoplus_{\substack{k \in \mathcal{B}_j^{(1\rightarrow2)} \\ \text{s.t. } h_{\ell,k}^{(1\rightarrow2)} \neq 0}} \underline{h}_k^{(1)}. \qquad (5)$$

Relation (5) depends on index ℓ only through $h_{\ell,k}^{(1\rightarrow2)}$. This implies that, in (5) the type combination is the same for every $\ell \in \mathcal{A}_i^{(1\rightarrow2)}$. As a result, for all $i \in \{1, \cdots, S_{m_2}\}$, $\mathcal{A}_i^{(2)} = \mathcal{A}_i^{(1\rightarrow2)}$. In the same way, deriving relation (3) column-wise permits to show that $\forall j \in \{1, \cdots, S_n\}$, $\mathcal{B}_j^{(2)} = \mathcal{B}_j^{(1)}$.

Now consider $i \in \{1, \cdots, S_{m_2}\}$, $v \in \{1, \cdots, S_n\}$, and $\ell \in \mathcal{A}_i^{(2)}$. Then, from (5),

$$s_{i,v}^{(2)} = \sum_{u \in \mathcal{B}_v^{(2)}} h_{l,u}^{(2)} = \sum_{u \in \mathcal{B}_v^{(2)}} \left(\bigoplus_{j=1}^{S_{m_1}} \bigoplus_{\substack{k \in \mathcal{B}_j^{(1\rightarrow2)} \\ \text{s.t. } h_{\ell,k}^{(1\rightarrow2)} \neq 0}} h_{k,u}^{(1)} \right). \qquad (6)$$

In the vector $\underline{h}_k^{(1)}$ with $k \in \mathcal{B}_j^{(1\rightarrow2)}$, there are $s_{j,v}^{(1)}$ non-zero values over the components $h_{k,u}$ such that $u \in \mathcal{B}_v^{(2)}$. In addition, for $k \in \mathcal{B}_j^{(1\rightarrow2)}$, there are $s_{i,j}^{(1\rightarrow2)}$ non-zero values over the components $h_{\ell,k}^{(1\rightarrow2)}$. As a result, and since there is not VN elimination,

$$s_{i,v}^{(2)} = \sum_{j=1}^{S_{m_1}} s_{i,j}^{(1\rightarrow2)} s_{j,v}^{(2)}, \qquad (7)$$

which implies (4). ∎

In Proposition 2, assumption 1) is required because various interleaving structures may be used to construct *e.g.* a matrix H_1 from a given protograph $\mathcal{S}_1$. This assumption guarantees that the same interleaving structure is used for the CNs of $\mathcal{S}_1$ and the VNs of $\mathcal{S}_{1\rightarrow2}$. Further, assumption 2 guarantees that relation (3) does not eliminate any VN from the parity check equations in H_2. This permits to preserve the code structure that will be characterized by protograph $\mathcal{S}_2$. Then, by comparing (3) and (4), we observe that there is the same relation between the protographs $\mathcal{S}_1$, $\mathcal{S}_2$, and between the parity check matrices H_1, H_2. Further, according to (4), the problem of finding a good protograph $\mathcal{S}_2$ for H_2 can be reduced to finding the intermediate protograph $\mathcal{S}_{1\rightarrow2}$ that maximizes the threshold of $\mathcal{S}_2$.

B. Optimization of the intermediate protograph $\mathcal{S}_{1\to 2}$

The protograph $\mathcal{S}_{1\to 2}$ of size $S_{m_2} \times S_{m_1}$ must be full rank in order to satisfy the rate-adaptive condition defined in Section IV-B. However, even if S_{m_1} and S_{m_2} are small, there is still a lot of possible protographs $\mathcal{S}_{1\to 2}$. This is why, here, we impose that each row of $\mathcal{S}_{1\to 2}$ has either 1 or 2 non-zero components, that each column has exactly 1 non-zero component, and that all the non-zero components are equal to 1. These constraints are equivalent to considering that each row of $\mathcal{S}_2$ is either equal to a row of $\mathcal{S}_1$ or equal to the sum of two rows of $\mathcal{S}_1$. They limit the number of possible $\mathcal{S}_{1\to 2}$ without being too restrictive. They will also make the intermediate matrix $H_{1\to 2}$ quite sparse, which will help limiting the amount of short cycles in the matrix H_2. Finally, we observe that these constraints provide satisfactory rate-adaptive code constructions in our simulations. The design algorithms described in the remaining of the paper can also be easily generalized to other constraints on the intermediate protograph.

For the optimization, we then generate all the possible intermediate protographs $\mathcal{S}_{1\to 2}$ that satisfy the above two conditions ($\mathcal{S}_{1\to 2}$ is full rank and each of its rows has either 1 or 2 non-zero components), and select the intermediate protograph that maximizes the threshold of the protograph $\mathcal{S}_2$ calculated from (4).

The intermediate protograph $\mathcal{S}_{1\to 2}$ defines the degree distribution of the intermediate matrix $H_{1\to 2}$. It also indicates the rows of H_1 that can be combined in order to construct the daughter matrix H_2. We would like those rows to be combined in the best possible way in order to produce H_2. In particular, we would like to avoid both short circles and VN elimination during the construction of H_2. In the following, we propose an algorithm that constructs H_2 from these conditions.

C. Algorithm Proto-Circle: connections in $H_{1\to 2}$

In Section V-B, we selected the intermediate protograph $\mathcal{S}_{1\to 2}$ that gives the protograph $\mathcal{S}_2$ with highest threshold. We now explain how to construct $H_{1\to 2}$ in order to follow the degree distribution defined by protograph $\mathcal{S}_{1\to 2}$, but also to limit the amount of short cycles in H_2 and to avoid VN elimination. Applying the PEG algorithm directly on $H_{1\to 2}$ would reduce the amount of short cycles on $H_{1\to 2}$ but would not guarantee that the number of cycles in H_2 is reduced as well. As an alternative, the algorithm Proto-Circle we propose is described in Algorithm 1. It constructs one row of H_2 at a time by combining rows of H_1, which can be regarded as defining the coefficients of the intermediate matrix $H_{1\to 2}$. For each new row of H_2, we want to limit the number of short cycles that are added to the parity check matrix H_2.

According to section V-B, each row of the protograph $\mathcal{S}_{1\to 2}$ has either 1 or 2 non-zero components. The rows of $\mathcal{S}_{1\to 2}$ that have 2 non-zero components indicate that two rows of H_1 of some given types should be combined in order to obtain one row of H_2. More formally, assume that the i-th row of $\mathcal{S}_{1\to 2}$ is such that $s_{i,j_1}^{(1\to 2)} = 1$ and $s_{i,j_2}^{(1\to 2)} = 1$ ($j_1 \neq j_2$). This means that two rows of H_1 of types $A_{j_1}^{(1)}$ and $A_{j_2}^{(1)}$ should be combined in order to obtain one row of H_2 of type $A_i^{(2)}$. For

---

Algorithm 1 Proto-Circle: construction of the low-rate matrix H_2

Inputs: H_1, $\mathcal{S}_1$, $\mathcal{S}_{1\to 2}$, K, $H_2 = \{\phi\}$
for $i = 1$ to S_{m_2} **do**
  if i-th row of $\mathcal{S}_{1\to 2}$ has two non-zero components $s_{i,j_1}^{(1\to 2)}$, $s_{i,j_2}^{(1\to 2)}$ **then**
    for $\ell = 1$ to m_1/S_{m_1} **do**
      Pick u at random in $\mathcal{A}_{j_1}^{(1)}$ and $v_1, \cdots, v_K$ at random in $\mathcal{A}_{j_2}^{(1)}$ such that $\forall k \in \{1, \cdots, K\}$, $\forall w \in \{1, \cdots, m_1\}$, $h_{v_1,w}^{(1)}.h_{v_2,w}^{(1)} = 0$
      For all $k \in \{1, \cdots, K\}$, count the number $N_{4,k}$ of length-4 cycles in $H_2 \cup \{\underline{h}_u^{(1)} + \underline{h}_{v_k}^{(1)}\}$
      For the index $k^\star$ that minimizes $N_{4,k}$, do $H_2 \leftarrow H_2 \cup \{\underline{h}_u^{(1)} + \underline{h}_{v_{k^\star}}^{(1)}\}$
      Remove u from $\mathcal{A}_{j_1}^{(1)}$ and $v_{k^\star}$ from $\mathcal{A}_{j_2}^{(1)}$
    end for
  else
    for $\ell = 1$ to m_1/S_{m_1} **do**
      Pick u at random in $\mathcal{A}_{j_1}^{(1)}$ ($s_{i,j_1}^{(1\to 2)} \neq 0$) and do $H_2 \leftarrow H_2 \cup \{\underline{h}_u^{(1)}\}$, remove u from $\mathcal{A}_{j_1}^{(1)}$
    end for
  end if
end for
outputs: H_2, N_4 (number of length-4 cycles in H_2)

---

this, we select at random one row $\underline{h}_u^{(1)}$ of H_1 of type $A_{j_1}^{(1)}$ and K rows $\underline{h}_{v_1}^{(1)}, \cdots \underline{h}_{v_K}^{(1)}$ of type $A_{j_2}^{(1)}$ such that $\forall k \in \{1, \cdots, K\}$, $\forall w \in \{1, \cdots, m_1\}$, $h_{v_1,w}^{(1)}.h_{v_2,w}^{(1)} = 0$ (binary AND operation). This condition avoids VN elimination. The algorithm counts the number $N_{4,k}$ of length-4 cycles that would be added if a new row $\underline{h}_u^{(1)} + \underline{h}_{v_k}^{(1)}$ was added to H_2. The number of length-4 cycles in H_2 is computed with the algorithm proposed in [39]. Note that the algorithm can be easily modified to also consider larger cycles. The algorithm then chooses the row combination that adds least cycles in H_2.

Once all the lines of types $A_{j_1}^{(1)}$ and $A_{j_2}^{(1)}$ have been combined, the algorithm passes to the next row of $\mathcal{S}_{1\to 2}$ with two non-zero components and repeats the same process. It then processes the rows of $\mathcal{S}_{1\to 2}$ with one non-zero component. For instance, assume that row i' of $\mathcal{S}_{1\to 2}$ has one non-zero component $s_{i',j'}^{(1\to 2)} = 1$. Then, all the lines of H_1 of type $A_{j'}^{(1)}$ are placed into H_2. The placement order does not have any influence on the amount of cycles in the matrix H_2.

After constructing all the rows of H_2, the algorithm counts the total number of length-4 cycles in the newly created H_2. At the end, repeating the algorithm Proto-Circle several times allows us to choose the matrix H_2 with least short cycles.

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

The intermediate matrix $H_{1\to 2}$ follows the structure of the protograph $\mathcal{S}_{1\to 2}$. As a result, according to Section V-B, each of its lines has either 1 or 2 non-zero components. Further, the algorithm Proto-Circle introduced in Section V-C imposes that each row of H_1 participates to exactly one

combination for the constructions of the rows of H_2. These two conditions guarantee that $H_{1\rightarrow2}$ is full-rank so that the rate-adaptive condition presented in Section IV-B is satisfied. However, in order to completely define the rate-adaptive code $(H_1, H_{1\rightarrow2}, \mathcal{C}')$, we need to define a set $\mathcal{C}'$ of symbols of $\mathcal{C}$ that will be sent together with the set $\mathcal{U}$ in order to obtain the rate R_1.

The set $\mathcal{C}'$ will serve to solve a system of m_1 equations $\mathcal{U}$ with m_1 unknowns $\mathcal{C} \setminus \mathcal{C}'$. For each syndrome symbol $u_i \in \mathcal{U}$ of degree d_k in $H_{1\rightarrow2}$, we hence decide to put $d_k - 1$ of the d_k CNs connected to u_i into $\mathcal{C}'$. For example, if $u_1 = c_1 \oplus c_2 \oplus c_3$, c_1 and c_2 may be placed into $\mathcal{C}'$. This strategy guarantees that it is always possible to reconstruct the set $\mathcal{C}$ from $\mathcal{U}$ and $\mathcal{C}'$. In the above example, it indeed suffices to recover c_3 as $c_3 = u_1 \oplus c_1 \oplus c_2$.

We now count the number of symbols c_i that are placed into $\mathcal{C}'$ with this strategy. Since each line of $H_{1\rightarrow2}$ has either 1 or 2 non-zero components, we have $d_k = 1$ or $d_k = 2$. Denote by α the proportion of values u_k of degree 1. We have the following relation between m_1, m_2 and α:

$$m_1 = \alpha m_2 + 2(1-\alpha)m_2.$$

This gives that $\alpha = 2 - \frac{m_1}{m_2}$. Further, according to the code construction proposed in Section V-C, each c_i participates to exactly one equation u_j. As a result, in the above strategy, the set $\mathcal{C}'$ is composed by $(1-\alpha)m_2 = m_1 - m_2$ different values c_i, which is exactly what is required by the rate-adaptive construction.

VI. GENERALIZATION TO SEVERAL RATES

The above method constructs the matrix H_2 of rate $R_2 < R_1$ from the matrix H_1. In order to obtain lower rates $R_T < R_{T-1} < \cdots < R_2 < R_1$, we need to construct the successive matrices H_t, $t \in \{2, \cdots, T\}$. As initially proposed in [1], the matrices H_t can be constructed recursively from intermediate matrices $H_{t-1\rightarrow t}$ such that $H_t = H_{t-1\rightarrow t}H_{t-1}$. The intermediate matrices $H_{t-1\rightarrow t}$ are constructed by from the method described in Section V.

However, with the method of Section V, the rate values $R_2, \cdots, R_T$ are constrained by the size of the initial protograph $\mathcal{S}_1$. For a protograph $\mathcal{S}_1$ of size $S_{m_1} \times S_n$, the rate granularity is given by

$$r_g = \frac{R_1}{S_{m_1}}. \tag{8}$$

For instance, if $R_1 = 1/2$ and $\mathcal{S}_1$ is of size 4×8, only rates $R_2 = 3/8$, $R_3 = 1/4$, $R_4 = 1/8$ can be achieved. This is why, in this section, we propose two alternatives methods that allow to decrease the rate granularity r_g.

A. Protograph extension

The first method called "protograph extension" consists of lifting the mother protograph $\mathcal{S}_1$ by a factor Z_e, in the same way as for producing a parity check matrix from a given protograph (see Section III-B). This extension produces

a protograph $\mathcal{S}_1'$ of size $Z_e S_{m_1} \times Z_e S_n$. For instance, the protograph

$$\mathcal{S}_1 = \begin{bmatrix} 1 & 2 & 1 & 3 \\ 1 & 0 & 2 & 5 \end{bmatrix} \tag{9}$$

can be extended as

$$\mathcal{S}_1' = \begin{bmatrix} 1 & 1 & 1 & 2 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 4 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 4 \end{bmatrix}. \tag{10}$$

The protograph $\mathcal{S}_1$ permits to generate an ensemble $\mathcal{H}_1$ of parity check matrices with asymptotic codeword length. According to [13, Theorem 2], all the asymptotic parity check matrices in $\mathcal{H}_1$ have the same decoding performance given by the threshold of $\mathcal{S}_1$. The extended protograph $\mathcal{S}_1'$ generates a code ensemble $\mathcal{H}_1' \subseteq \mathcal{H}_1$. As a result, the asymptotic matrices in $\mathcal{H}_1'$ have the same decoding performance as the matrices in $\mathcal{H}_1$, and $\mathcal{S}_1$ and $\mathcal{S}_1'$ have the same theoretical threshold.

The above protograph extension allows to consider more rates, since the rate granularity r_g' of $\mathcal{S}_1'$ is given by $r_g' = r_g/Z_e \leq r_g$. However, it is not desirable neither to end up with an extended protograph $\mathcal{S}_1'$ of large size, $e.g.$ in the order of magnitude of m_1. Indeed, in this case, the number of possibilities for intermediate protographs $\mathcal{S}_{t-1\rightarrow t}$ would also become very large. In addition, it becomes computationally difficult to compute the theoretical thresholds for large protographs. As a result, if the size of $\mathcal{S}_1'$ is large, it will be very difficult to optimize the successive protographs $\mathcal{S}_t$ according to the method described in Section V-B. This is why we now propose a second method that allows to push further the rate granularity improvement.

B. Anchor rates

In this second method, consider a protograph $\mathcal{S}_1$ of size $S_{m_1} \times S_n$. As a first step, we do the protograph optimization of Section V-B for all the possible rates

$$R_t = R_1 - \frac{(t-1)R_1}{S_{m_1}}, \tag{11}$$

where $t \in \{1, \cdots, S_{m_1}\}$, and $R_{t-1} - R_t = R_1/S_{m_1}$. This produces a sequence of protographs $\mathcal{S}_t$, and the rates R_t are called the anchor rates. We now want to construct all the possible intermediate rates between any R_{t-1} and R_t, with a rate granularity $r_g = R_1/m_1$.

According to Section V-B, the rows of the intermediate protographs $\mathcal{S}_{t-1\rightarrow t}$ have either one or two non-zero components. In addition, in order to obtain all the rates R_t defined in (11), exactly one row of $\mathcal{S}_{t-1\rightarrow t}$ has two non-zero components. This is why, in order to obtain a rate $R_{t-1} - \frac{1}{m_1}$, we propose to combine two rows of the corresponding type in H_{t-1}. The resulting matrix contains the considered row combination, as well as all the non-combined rows of H_{t-1}. As in the algorithm Proto-Circle described in Section V-C, we choose the row combination that minimizes the amount of short cycles that will be added in the resulting matrix. Applying this process recursively allows to obtain all rates $R_{t-1} - kR_1/m_1$, with $k \in \{1, \cdots, m_1/S_{m_1}\}$, and $m_1/S_{m_1} = Z_1$, where Z_1
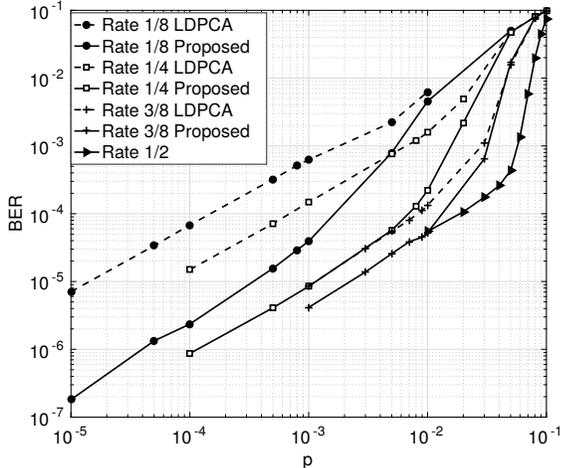
Fig. 3: BER performance of code $\mathcal{C}_1$ with dimension 248×496 using proposed construction compared with LDPCA
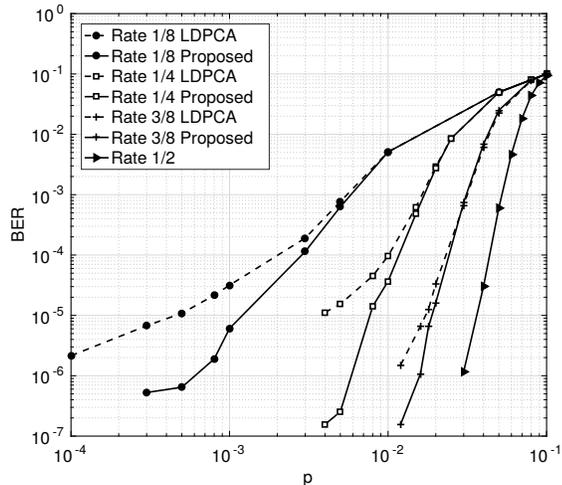

Fig. 4: BER performance of code $\mathcal{C}_2$ with dimension 256×512 using proposed construction compared with LDPCA

is the lifting factor. This approach also guarantees that at rate R_t, the resulting matrix follows the structure of protograph $\mathcal{S}_t$.

The anchor rates method allows to obtain a rate granularity $r_g = R_1/m_1$. In the simulation section, we combine both approaches (protograph extension and anchor rates) in order to obtain an incremental code construction that permits to handle a wide range of statistical relations between the source and the side information.

VII. SIMULATION RESULTS

Rate	LDPCA	Our method
$R = 3/8$	453	455
$R = 1/4$	1216	737
$R = 1/8$	5361	3477

TABLE I: Number of length-4 cycles for code $\mathcal{C}_1$

Fig. 5: BER performance of code $\mathcal{C}_3$ with dimension 512×1024 using proposed construction compared with LDPCA

This section evaluates from Monte Carlo simulations the performance of the proposed rate-adaptive construction. We assume a BSC of parameter p and we consider three binary LDPC codes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$ constructed from protographs. These codes are set as mother codes for the initial rate $R = 1/2$. The algorithm introduced in Section V-C then produces the corresponding daughter codes for lower rates $3/8$, $1/4$, $1/8$. In the following, we compare the performance of the obtained rate-adaptive codes with LDPCA.

The first code $\mathcal{C}_1$ is of size 248x496. In order to construct $\mathcal{C}_1$, we first obtained the protograph $\mathcal{S}_1$ of size 2×4 in (9) from the Differential Evolution optimization method described in Section III-B. Differential Evolution was applied by considering $V = 60$ elements in the population. This follows [18] which suggests to choose $5D < V < 10D$, where in our case, $D = S_n S_m = 8$. In addition, the number of iterations was set as $L = 100$, and the maximum degree was set as $d_{\max} = 10$. The theoretical threshold of $\mathcal{S}_1$ is equal to $p = 0.094$, which is very close to the maximum value $p = 0.11$ that can be considered at rate $1/2$. Protograph $\mathcal{S}_1$ was then extended to the protograph $\mathcal{S}_1'$ of size 4×8 in (10) according to the method described in Section VI-A. This extension allows ton consider more anchor rates $3/8$, $1/4$ and $1/8$, rather than $1/4$ only with $\mathcal{S}_1$.

The parity check matrix of $\mathcal{C}_1$ was constructed from the protograph $\mathcal{S}_1'$ by the PEG algorithm [19]. We then applied our construction method introduced in Section V in order to obtain lower rates $3/8$, $1/4$ and $1/8$. For this, we first needed to decide which rows of the protograph $\mathcal{S}_{opt1}$ should be combined (see Section V-B) by checking the thresholds of all the possible combinations using Density Evolution. From Density Evolution, we chose row combinations $A_1^{(1)} + A_3^{(1)}$ for rate $3/8$ and $A_1^{(1)} + A_3^{(1)}$, $A_2^{(1)} + A_4^{(1)}$ for rate $1/4$, where the $A_i^{(1)}, i = 1, 2, \cdots, S_m$, denotes the rows of $\mathcal{S}_1'$. From the selected row combinations, we then constructed the corresponding matrices of rate $3/8$, $1/4$, $1/8$ from the algorithm

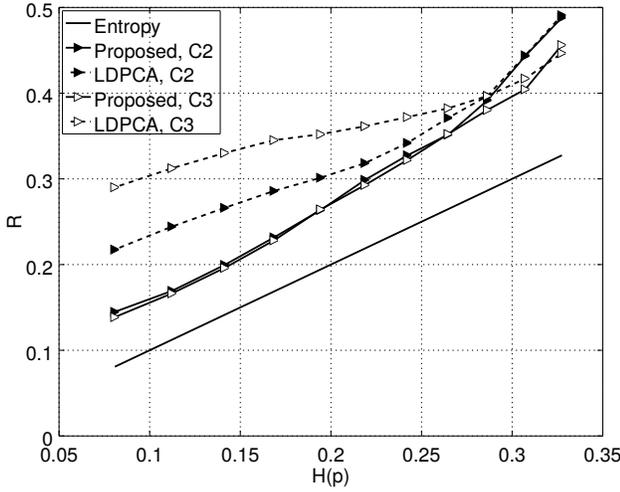Fig. 6: Required rate R with respect to $H(p)$ for LDPCA and for our method, for codes $\mathcal{C}_2$ and $\mathcal{C}_3$

Fig. 7: Performance of five code constructions with different protographs at rate 3/8

Proto-Circle described in Section V-C. This algorithm was applied with $K = 20$ and repeated 10 times in order to choose the low-rate matrices with the least short cycles.

Figure 3 shows the Bit Error Rate (BER) performance with respect to the BSC parameter p for the four considered rates for $\mathcal{C}_1$. We observe that our code construction performs better than LDPCA at all the considered rates. Table I indeed shows that there are less length-4 cycles at rates $1/4$ and $1/8$ in our construction than in the LDPCA matrices.

The second code $\mathcal{C}_2$ is of size 256×512 and it was generated from another protograph

$$\mathcal{S}_{opt2} = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 2 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (12)$$

obtained from Differential Evolution and protograph extension. The codes of lower rates $3/8$, $1/4$, and $1/8$ were constructed by following the same steps as for $\mathcal{C}_1$, according to the construction of Section V. The BER performance of these codes are shown in Figure 4 and compared to LDPCA. For this case as well, our construction shows better performance than LDPCA. Finally, the code $\mathcal{C}_3$ is of size 512×1024 and it was generated from the same protograph $\mathcal{S}_{opt2}$ as $\mathcal{C}_2$. Figure 5 shows that for $\mathcal{C}_3$ as well, our algorithm perform better than LDPCA at all the considered rates, with a larger code size.

The curves of Figures 3, 4, 5, considered the code performance for the anchor rates given in Section VI-B. We then applied the method described in Section VI-B to codes $\mathcal{C}_2$ and $\mathcal{C}_3$ in order to obtain rate granularities of $R_1/m_1 = 9.8 \times 10^{-4}$ for $\mathcal{C}_2$ and $R_1/m_1 = 4.9 \times 10^{-4}$ for $\mathcal{C}_3$, rather than $R_1/S_{m_1} = 0.125$. For this, we considered different values of p, and for every considered value, we generated 1000 couples $(\underline{x}^n, \underline{y}^n)$ from a BSC or parameter p. For every generated couple, we found the minimum rate that permits to decode $\underline{x}^n$ from $\underline{y}^n$ without any error. The same kind of analysis was performed in [25] and [29], with different criterion to measure the rate needed for a given couple $(\underline{x}^n, \underline{y}^n)$. In [25], this rate
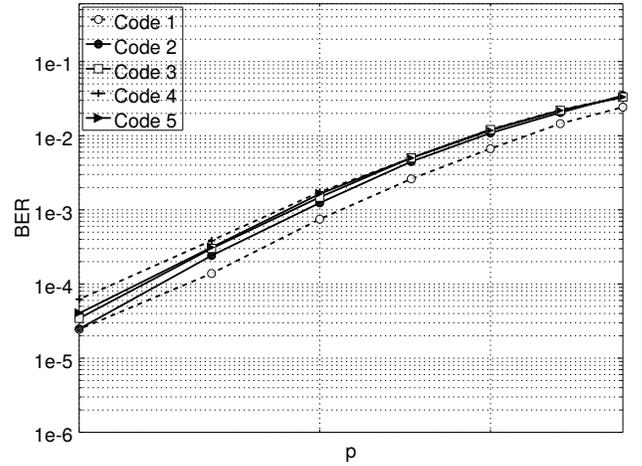
was determined as the minimum rate such that the decoded codeword $\underline{\hat{x}}^n$ verifies $H^T \underline{\hat{x}}^n = \underline{c}^m$, see (1). However, this criterion does not necessarily means that the codeword was correctly decoded ($\underline{\hat{x}}^n$ can be different from $\underline{x}^n$), and this is why we do not consider it here. In [29], the required rate was determined as the minimum rate that gives a BER lower than 10^{-6}. This is equivalent to our criterion, since one uncorrectly decoded bit gives a BER of 2.0×10^{-3} for $\mathcal{C}_2$, and of 1.0×10^{-3} for $\mathcal{C}_3$.

At the end, Figure **??** represents the average rates needed for the considered values of p with respect to $H(p)$. We first observe that our method shows a loss compared to the optimal rate $H(p)$. This rate loss is expected since we consider relatively short codeword length 512 for $\mathcal{C}_2$ and 1024 for $\mathcal{C}_3$. In addition, for the same codes $\mathcal{C}_2$ and $\mathcal{C}_3$, LDPCA shows a much more significant rate loss compared to our method, which was also expected from the results of Figures 4 and 5. This shows that our construction combined with the anchor rates method is valid and outperforms LDPCA at all the considered values of p.

Name	Protograph	Theoretical threshold
Code 1	$A_1^{(2)} + A_2^{(2)}, A_3^{(2)}, A_4^{(2)}$	0.051
Code 2	$A_1^{(2)} + A_2^{(2)}, A_3^{(2)} + A_4^{(2)}, A_4^{(2)}$	0.049
Code 3	$A_1^{(2)} + A_2^{(2)}, A_3^{(2)} + A_4^{(2)}, A_2^{(2)}$	0.050
Code 4	$2A_1^{(2)} + A_2^{(2)}, A_3^{(2)}, A_4^{(2)}$	0.049
Code 5	$A_1^{(2)} + A_2^{(2)}, 2A_3^{(2)}, A_4^{(2)}$	0.049

TABLE II: Protographs with different constraints for rate 3/8. The protographs are described in the form of combination of the lines $A_i^{(2)}$ of the original protograph $\mathcal{S}_{opt2}$.

To finish, we discuss the influence of the constraints expressed in Section V-B for the code construction. These constraints were that in the intermediate protograph $\mathcal{S}_{1 \to 2}$, each column has only 1 non-zero components, and that all the non-zero components are equal to 1. In order to discuss the influence of these constraints, we again consider the extended protograph $\mathcal{S}_{opt2}$ of rate $1/2$ given in (12) and we

consider the construction of a daughter code of rate $3/8$. If the above constraints are not satisfied, there are many possibilities to construct such daughter codes. We consider 5 different protographs that can be obtained from $\mathcal{S}_{opt2}$ for the daughter codes. These protographs are given in Table II, with their theoretical thresholds. Code 1 corresponds to a protograph that satisfies the constraints of Section V-B. We see that the other protographs have either more than 1 non-zero component per column (Codes 2 and 3), or a non-zero component equal to 2 (Codes 4 and 5). Note that we selected give protographs with close theoretical thresholds, in order to compare the finite-length performance of the considered codes.

The performance of the five considered codes are compared in Figure 7. We see that despite the fact that the five codes have very close theoretical thresholds, their BER performance vary from one code to another. As expected, Code 1 shows the best performance. This can be explained by the fact that its parity check matrix is more sparse than the other ones. We also note that Codes 4 and 5 for which the protographs contain non-zero components equal to 2 show the worst performance. This shows that, in the considered case, the above constraints are reasonable.

VIII. Conclusion

This paper introduced a novel rate-adaptive construction based on LDPC codes for the Slepian-Wolf source coding problem. The introduced construction is based on an optimization of the protographs of the incremental codes constructed at different rates. It not only allows to optimize the code thresholds at these rates, but also to reduce the amount of short cycles in the obtained parity check matrices. The proposed method shows improved performance compared to the standard LDPCA for all the considered codes and at all the considered rates. This method may be easily generalized to to channel coding and to non-binary LDPC codes. Extension to lossy source coding with side information (Wyner-Ziv problem) will be considered in future works.

IX. Acknowledgement

References

[1] 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.

[2] F. Ye, Z. Mheich, E. Dupraz, and K. Amis, "Optimized short-length rate-adaptive LDPC Codes for Slepian-Wolf source coding," in *International Conference on Telecommunication (ICT)*, 2018.

[3] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[4] 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.

[5] 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.

[6] M. Tanimoto, "Free-viewpoint television," in *Image and Geometry Processing for 3-D Cinematography*. Springer, 2010, pp. 53–76.

[7] E. Dupraz, T. Maugey, A. Roumy, and M. Kieffer, "Rate-storage regions for massive random access," *arXiv preprint arXiv:1612.07163*, 2016.

[8] 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.

[9] R. Gallager, "Low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[10] Y.-P. Wei, S.-C. Lin, S.-J. Lin, H.-J. Su, and H. V. Poor, "Residual-quantization based code design for compressing noisy sources with arbitrary decoder side information," *IEEE Transactions on Communications*, vol. 64, no. 4, pp. 1711–1725, 2016.

[11] J. Chen, D.-k. He, and A. Jagmohan, "On the duality between Slepian–Wolf coding and channel coding under mismatched decoding," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4006–4018, 2009.

[12] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[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] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, 2009.

[16] G. Han, Y. L. Guan, and L. Kong, "Construction of irregular QC-LDPC codes via masking with ACE optimization," *IEEE Communications Letters*, vol. 18, no. 2, pp. 348–351, 2014.

[17] E. Dupraz, V. Savin, and M. Kieffer, "Density evolution for the design of non-binary low density parity check codes for Slepian-Wolf coding," *IEEE Transactions on Communications*, vol. 63, no. 1, pp. 25–36, 2015.

[18] 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*, 1996, pp. 842–844.

[19] 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, 2005.

[20] X. Jiang, X.-G. Xia, and M. H. Lee, "Efficient progressive edge-growth algorithm based on chinese remainder theorem," *IEEE Transactions on Communications*, vol. 62, no. 2, pp. 442–451, 2014.

[21] C. T. Healy and R. C. de Lamare, "Design of LDPC codes based on multipath EMD strategies for progressive edge growth," *IEEE Transactions on Communications*, vol. 64, no. 8, pp. 3208–3219, 2016.

[22] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.

[23] M. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," in *IEEE International Conference on Communications*, vol. 1. IEEE, 2004, pp. 430–434.

[24] T. Van Nguyen, A. Nosratinia, and D. Divsalar, "The design of rate-compatible protograph LDPC codes," *IEEE Transactions on Communications*, vol. 60, no. 10, pp. 2841–2850, 2012.

[25] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.

[26] A. W. Eckford and W. Yu, "Rateless Slepian-Wolf codes," in *in Asilomar Conference on Signals, Systems and Computers*, 2005, pp. 1757–1761.

[27] 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.

[28] M. Yang, W. E. Ryan, and Y. Li, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Transactions on Communications*, vol. 52, no. 4, pp. 564–571, 2004.

[29] F. Cen, "Design of degree distributions for LDPCA codes," *IEEE Communications Letters*, vol. 13, no. 7, pp. 525–527, 2009.

[30] 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.

[31] A. Sgarro, "Source coding with side information at several decoders," *IEEE Transactions on Information Theory*, vol. 23, no. 2, pp. 179–182, 1977.

[32] S. C. Draper, "Universal Incremental Slepian-Wolf Coding," in *Allerton Conference on Communication, Control and Computing*, 2004, pp. 1757–1761.

[33] E. Yang and D. He, "Interactive encoding and decoding for one way learning: near lossless recovery with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1808–1824, 2010.

[34] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.

[35] D. G. Mitchell, R. Smarandache, and D. J. Costello, "Constructing good QC-LDPC codes by pre-lifting protographs," in *Information Theory Workshop (ITW), 2012*, 2012, pp. 202–206.

[36] T. Richardson, R. Urbanke *et al.*, "Multi-edge type LDPC codes," in *Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California*, 2002, pp. 24–25.

[37] M. Gorgoglione, V. Savin, and D. Declercq, "Optimized puncturing distributions for irregular non-binary LDPC codes," in *International Symposium on Information Theory and its Applications (ISITA)*, 2010, pp. 400–405.

[38] 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.

[39] 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.