Source Coding with Side Information at the Decoder and Uncertain Knowledge of the Correlation

Elsa DUPRAZ[1], Aline ROUMY[2] and Michel KIEFFER[1,3,4]

[1] L2S - CNRS - SUPELEC - Univ Paris-Sud, 91192 Gif-sur-Yvette, France

[2] INRIA, Campus de Beaulieu, 35042 Rennes, France

[3] Partly on leave at LTCI – CNRS Télécom ParisTech, 75013 Paris, France

[4] Institut Universitaire de France

Abstract

This paper considers the problem of lossless source coding with side information at the decoder, when the correlation model between the source and the side information is uncertain. Four parametrized models representing the correlation between the source and the side information are introduced. The uncertainty on the correlation appears through the lack of knowledge on the value of the parameters.

For each model, we propose a practical coding scheme based on non-binary Low Density Parity Check Codes and able to deal with the parameter uncertainty. At the encoder, the choice of the coding rate results from an information theoretical analysis. Then we propose decoding algorithms that jointly estimate the source vector and the parameters. As the proposed decoder is based on the Expectation-Maximization algorithm, which is very sensitive to initialization, we also propose a method to produce first a coarse estimate of the parameters.

Part of this paper will be presented at the Data Compression Conference (DCC) 2013.

Part of this work was supported by the ANR-09-VERS-019-02 grant (ARSSO project).

I. Introduction

The problem of lossless source coding with side information at the decoder has been well investigated when the correlation model between the source X and the side information (SI) Y is perfectly known. Slepian and Wolf showed that this case induces no loss in performance compared to the conditional setup, *i.e.*, the setup where the side information is also known at the encoder [31]. Following this principle, several works, see, *e.g.*, [26], [32], [37], propose practical coding schemes for the Slepian-Wolf (SW) problem. Most of them are based on channel codes [33], and particularly Low Density Parity Check (LDPC) codes [22], [23]. This approach allows to leverage on many results on LDPC codes for the code construction and optimization [20], [28] even if there is a need to adapt the developed algorithms to the case of SW coding [6].

Nonetheless, most of these works assume perfect knowledge of the joint distribution $P(X,Y)$. In [18], it is shown that the performance of the SW coding scheme remains the same if $P(X)$ is unknown. Here we consider the case where the *correlation channel* $P(Y|X)$ is uncertain because it is in general more difficult to obtain in practical situations. In this way, [30] considers the case where $P(Y|X)$ is given to the decoder but not perfectly known at the encoder. Here we assume that $P(Y|X)$ is uncertain at both the encoder and the decoder. A usual solution to address this problem is to use a feedback channel [1], [11], [36], or to allow interactions between the encoder and the decoder [38]. The advantage of the feedback channel is that the rate is adapted to the true characteristics of the source. However, a feedback channel can be difficult to implement in many practical situations such as sensor networks. Moreover, the feedback channel is in general used by the decoder to ask for additional packets to the encoder or to stop the transmission. Each time a new packet is received, the decoder processes again all the received packets to try to reconstruct the source. This can result in huge decoding delays.

When no feedback is allowed, several practical solutions based on LDPC codes and proposed for channel coding may be adapted to the SW problem. When hard decoding is performed, as proposed in [21], [27] for channel coding, only symbol values are used, at the price of an important loss in performance. An alternative solution is the min-sum decoding algorithm proposed in [5], [29] for channel coding, respectively for binary and non-binary sources. The min-sum algorithm uses soft information for decoding, but does not require the knowledge of the correlation channel. However, for the min-sum to be as efficient as the soft decoding algorithm, a coefficient is required for the initialization of the algorithm and needs to be chosen carefully. Unfortunately, its value depends on the correlation channel.

In many applications, it is possible to restrict the correlation channel model to a given class (*e.g.*, binary symmetric, Gaussian, etc.) due to the nature of the problem. Consequently, in this paper, we introduce four signal models. Each model assumes that the correlation channel belongs to a given class and is parametrized by some unknown parameter vector. For two of the models, the correlation channel between source symbols (X_n, Y_n) is parametrized by an unknown parameter π_n, varying from symbol to symbol. One of these two models assumes the knowledge of a prior distribution $P_{\Pi}(\pi_n)$ for π_n. The case where no prior on π_n is known corresponds to *arbitrarily varying sources* [2], [4]. For the two other models, the correlation channel is parametrized by an unknown parameter θ, fixed for the sequence $\{(X_n, Y_n)\}_{n=1}^{+\infty}$ but allowed to vary from sequence to sequence. This corresponds to universal source coding [14]. The distinction between the two models is also in the knowledge of a prior for θ. The distinction between varying parameters π_n and a fixed parameter θ has been proposed earlier in [24] in the case of channel coding.

The coding scheme we propose is based on non-binary LDPC codes. Hard and min-sum LDPC decoding are not able to exploit the knowledge of the structure of the class. Therefore, the sum-product LDPC decoding algorithm is considered. From an analysis of the performance bounds, we explain for each model how to choose the coding rate and the LDPC coding matrix. Then, we show that the classical sum-product LDPC decoding algorithm can be used for only one model. For the three other models, we propose a decoding algorithm that jointly estimates the source vector and the parameters. As the method is based on the EM (Expectation Maximization) algorithm [17], which is very sensitive to initialization, we also propose a method to obtain first coarse estimates of the values of the parameters.

The paper is organized as follows. In Section II, the four signal models we consider are described formally. Section III explains how to choose the coding rates and to design the LDPC coding matrices. Section IV proposes a decoding method adapted to each model. Finally, Section V presents simulation results.

II. Signal Model

The source X to be compressed and the SI Y available at the decoder produce sequences of symbols $\{X_n\}_{n=1}^{+\infty}$ and $\{Y_n\}_{n=1}^{+\infty}$, respectively. $\mathcal{X}$ and $\mathcal{Y}$ denote the source and SI discrete alphabets. Bold upper case letters, *e.g.*, $\mathbf{X}_1^N = \{X_n\}_{n=1}^N$, denote random vectors, whereas bold lower case letters, $\mathbf{x}_1^N = \{x_n\}_{n=1}^N$, represent their realizations. When it is clear from the context that the distribution of a random variable X_n does not depend on n, the index n is omitted.

The goal of this section is to model the uncertainty on the correlation channel $P(Y|X)$. Each of the four proposed models consists of a family of parametric distributions[1]. In every case, the source distribution $P(X)$ is assumed perfectly known and does not depend on the uncertain parameters. The first two models allow parameter variations from symbol to symbol.

Definition 1. (DP-Source). *A Dynamic with Prior source (X,Y), or DP-Source, produces a sequence of* ***independent*** *symbols $\{(X_n,Y_n)\}_{n=1}^{+\infty}$ drawn $\forall n$ from $P(X_n,Y_n)$ that belongs to a family of distributions $\{P(X,Y|\mathbf{\Pi}=\boldsymbol{\pi}) = P(X)P(Y|X,\mathbf{\Pi}=\boldsymbol{\pi})\}_{\boldsymbol{\pi}\in\mathcal{P}_{\boldsymbol{\pi}}}$* ***parametrized by a***

[1] The four models defined in this section were also introduced with different names in two papers [9], [10], of the same authors. M-Source was for DP-Source, WPM-Source for DwP-Source, P-Source for SP-Source, WP-Source for SwP-Source. The names were changed for the sake of clarity.

random vector $\mathbf{\Pi}_n$*. The* $\{\mathbf{\Pi}_n\}_{n=1}^{+\infty}$ *are i.i.d. with distribution* $P(\mathbf{\Pi})$ *and take their values in a* ***discrete set*** $\mathcal{P}_{\boldsymbol{\pi}}$*. The source symbols* X_n *and* Y_n *take their values in the discrete sets* $\mathcal{X}$ *and* $\mathcal{Y}$*, respectively.*

The DP-Source, completely determined by $\mathcal{P}_{\boldsymbol{\pi}}$, $P(\mathbf{\Pi})$, and $\{P(X,Y|\mathbf{\Pi}=\boldsymbol{\pi})\}_{\boldsymbol{\pi}\in\mathcal{P}_{\boldsymbol{\pi}}}$, is stationary and ergodic, see [13, Section 3.5].

Definition 2. (DwP-Source). *A Dynamic without Prior source* (X,Y)*, or DwP-Source, produces a sequence of* ***independent*** *symbols* $\{(X_n,Y_n)\}_{n=1}^{+\infty}$ *drawn* $\forall n$ *from* $P(X_n,Y_n)$ *that belongs to a family of distributions* $\{P(X,Y|\boldsymbol{\pi})=P(X)P(Y|X,\boldsymbol{\pi})\}_{\boldsymbol{\pi}\in\mathcal{P}_{\boldsymbol{\pi}}}$ ***parametrized by a vector*** $\boldsymbol{\pi}_n$*. Each* $\boldsymbol{\pi}_n$ *takes its values in a* ***discrete set*** $\mathcal{P}_{\boldsymbol{\pi}}$*. The source symbols* X_n *and* Y_n *take their values in the discrete sets* $\mathcal{X}$ *and* $\mathcal{Y}$*, respectively.*

The DwP-Source, determined by $\mathcal{P}_{\boldsymbol{\pi}}$ and $\{P(X,Y|\boldsymbol{\pi})\}_{\boldsymbol{\pi}\in\mathcal{P}_{\boldsymbol{\pi}}}$, is non-stationary and non-ergodic [13, Section 3.5]. The only difference between the DP- and DwP-Sources lies in the definition of the parameters $\boldsymbol{\pi}_n$. In the DwP-Source, no distribution for $\boldsymbol{\pi}_n$ is specified, either because its distribution is not known or because $\boldsymbol{\pi}_n$ is not modeled as a random variable.

The following models consider a time-invariant parameter vector.

Definition 3. (SP-Source) *A Static with Prior source* (X,Y) *(SP-Source) produces a sequence of* ***independent*** *symbols* $\{(X_n,Y_n)\}_{n=1}^{+\infty}$ *drawn from a distribution belonging to a family* $\{P(X,Y|\mathbf{\Theta}=\boldsymbol{\theta})=P(X)P(Y|X,\mathbf{\Theta}=\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}_{\boldsymbol{\theta}}}$ *parametrized by a* ***random vector*** $\mathbf{\Theta}$*. The random vector* $\mathbf{\Theta}$*, with distribution* $P_{\mathbf{\Theta}}(\boldsymbol{\theta})$*, takes its value in a set* $\mathcal{P}_{\boldsymbol{\theta}}$ *that is either* ***discrete or continuous****. The source symbols* X *and* Y *take their values in the discrete sets* $\mathcal{X}$ *and* $\mathcal{Y}$*, respectively. Moreover, the* ***realization of the parameter*** $\boldsymbol{\theta}$ ***is fixed*** *for the sequence* $\{(X_n,Y_n)\}_{n=1}^{+\infty}$*.*

The SP-source, determined by $\mathcal{P}_{\boldsymbol{\theta}}$, $P_{\mathbf{\Theta}}(\boldsymbol{\theta})$, and $\{P(X,Y|\mathbf{\Theta}=\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}_{\boldsymbol{\theta}}}$, is stationary but non-ergodic [13, Section 3.5].

Definition 4. (SwP-Source). *A Static without Prior source (X, Y) (SwP-Source) produces a sequence of* ***independent*** *symbols $\{(X_n, Y_n)\}_{n=1}^{+\infty}$ drawn from a distribution belonging to a family $\{P(X, Y|\boldsymbol{\theta}) = P(X)P(Y|X, \boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}_{\boldsymbol{\theta}}}$ parametrized by a vector $\boldsymbol{\theta}$. The vector $\boldsymbol{\theta}$ takes its value in a set $\mathcal{P}_{\boldsymbol{\theta}}$ that is either* ***discrete or continuous****. The source symbols X and Y take their values in the discrete sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. Moreover, the* ***parameter $\boldsymbol{\theta}$ is fixed*** *for the sequence $\{(X_n, Y_n)\}_{n=1}^{+\infty}$.*

The SwP-source, completely determined by $\mathcal{P}_{\boldsymbol{\theta}}$ and $\{P(X, Y|\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}_{\boldsymbol{\theta}}}$, is stationary but non-ergodic [13, Section 3.5]. The only difference between the SP- and SwP-Sources lies in the definition of $\boldsymbol{\theta}$ (no distribution for $\boldsymbol{\theta}$ is specified in the SwP-Model). Note that both the encoder and the decoder are aware of the model characteristics given in Definitions 1 to 4.

In the SW setup, the infimum of achievable rates for our models are given by

1) for the DP-Source [31],

$$R = H(X|Y) \tag{1}$$

where $H(X|Y)$ is calculated from $P(X = x|Y = y) = \sum_{\pi\in\mathcal{P}_\pi} P(\pi), P(X = x|, Y = y, \pi)$.

2) for the DwP-Source [2],

$$R = \sup_{P(X,Y)\in\text{Conv}(\{P_\pi(X,Y)\}_{\pi\in\mathcal{P}_\pi})} H(X|Y) \tag{2}$$

where $\text{Conv}(\{P_\pi(X, Y)\}_{\pi\in\mathcal{P}_\pi})$ is the convex hull of the elements of $\{P_\pi(X, Y)\}_{\pi\in\mathcal{P}_\pi}$,

3) for the SP-Source [16, Theorem 7.3.4],

$$R = P_\Theta\text{-ess.}\sup H(X|Y, \Theta = \theta), \tag{3}$$

where P_Θ-ess. $\sup$ is the essential sup (the sup on the support of the distribution) with respect to the prior distribution P_Θ,

4) for the SwP-Source [7],

$$R = \sup_{\theta\in\mathcal{P}_\theta} H(X|Y, \theta)\ . \tag{4}$$

We see that for the DwP-Model, the SP-Model, and the SwP-Model, the infimum of achievable rates are given by worst cases defined on the set of values the parameters may take (SP- and SwP-Models), or on the convex hull of this set of values (DwP-Model).

III. Encoding

The coding schemes we propose are based on LDPC codes for SW coding. As suggested by [22], [23], LDPC codes initially introduced for channel coding can also be used for SW coding, after adaptation of the coding process and the decoding algorithm. In channel coding, LDPC codes were proposed for binary-input channels [12] and generalized to non-binary input channels in [8]. The adaptation to the SW setup is described in [22] for the binary case. In this paper, we propose a generalization of this adaptation to the non-binary case. This section describes the encoding part and introduces the involved notations. Note that the encoding part is as in the binary case, except that, now, the encoding operations are performed in GF(q). There are more differences in the decoding part.

We assume that the source symbols X are discrete and belong to GF(q). The SW coding of a source vector $\mathbf{x}$ of length N is performed by producing a vector $\mathbf{s} = H^T\mathbf{x}$ of length $M < N$. The matrix H is sparse, with non-zero coefficients uniformly distributed in GF(q)$\backslash\{0\}$. In the following, $\oplus$, $\ominus$, $\otimes$, $\oslash$ are the usual operators in GF(q). In the bipartite graph representing the dependencies between the random variables of $\mathbf{X}$ and $\mathbf{S}$, the entries of $\mathbf{X}$ are represented by Variable Nodes (VN) and the entries of $\mathbf{S}$ are represented by Check Nodes (CN). The set of CN connected to a VN n is denoted $\mathcal{N}(n)$ and the set of VN connected to a CN m is denoted $\mathcal{N}(m)$. The sparsity of H is determined by the VN degree distribution $\lambda(x) = \sum_{i\geq 2} \lambda_i x^{i-1}$ and the CN degree distribution $\rho(x) = \sum_{i\geq 2} \rho_i x^{i-1}$ with $\sum_{i\geq 2} \lambda_i = 1$ and $\sum_{i\geq 2} \rho_i = 1$. In SW coding, the rate $r(\lambda, \rho)$ of a code is given by $r(\lambda, \rho) = \frac{M}{N} = \frac{\sum_{i\geq 2} \rho_i/i}{\sum_{i\geq 2} \lambda_i/i}$.

In order to perform the encoding of a source vector $\mathbf{X}$, one needs to choose properly the coding rate and to design the LDPC coding matrix, *i.e.*, to impose good degree distributions

$(\lambda(x), \rho(x))$ [20], [27]. The performance analysis of Section II suggests the following approach. For the DP-Source, the LDPC coding matrix is designed for the known distribution $P(X|Y)$. For the three other models, the LDPC coding matrix is designed for the worst cases defined by (2)-(4).

IV. Decoding algorithm

This section introduces LDPC-based decoding algorithms capable of dealing with the uncertainty on the value of the parameters of the models. For the DP-Source, the decoding algorithm is the sum-product LDPC decoder adapted to SW coding. For the other sources, the LDPC decoding algorithm cannot be used directly because of the lack of knowledge on the parameters. We thus propose to jointly estimate the encoded source sequence $\mathbf{X}_1^N$ and the unknown parameters. This joint estimation is performed with an EM algorithm [17]. A method producing a first coarse estimate of the parameters is also presented to properly initialize the EM algorithm.

A. DP-Source: Standard LDPC decoding

In [22] the standard sum-product LDPC decoding algorithm has been adapted to SW coding of binary sources with perfect correlation channel knowledge. This section generalizes the adaptation of the decoding algorithm to non-binary SW coding. Indeed, in the SW case, one needs to take into account both the probability distribution of X and of the received codeword $\mathbf{s}$. For the DP-Source, the conditional distribution is perfectly determined as

$$P(X_n = k|Y_n = y_n) = \sum_{\boldsymbol{\pi} \in \mathcal{P}_{\boldsymbol{\pi}}} P(\boldsymbol{\pi}) P(X_n = k|Y_n = y_n, \boldsymbol{\pi}) \ . \tag{5}$$

The sum-product decoder performs an approximate Maximum *A Posteriori* (MAP) estimation of $\mathbf{x}$ from the received codeword $\mathbf{s}$ and the observed side information $\mathbf{y}$. The messages exchanged in the dependency graph are vectors of length q. The initial messages for each VN n are denoted $\mathbf{m}^{(0)}(n, y_n)$, with components

$$m_k^{(0)}(n, y_n) = \log \frac{P(X_n = 0|Y_n = y_n)}{P(X_n = k|Y_n = y_n)}, \ k = 0 \dots q-1 \ . \tag{6}$$

The messages from CN to VN are computed with the help of a particular Fourier Transform (FT), denoted $\mathcal{F}(\mathbf{m})$. Denoting r the unit root associated to GF(q), the i-th component of the FT is given by [20] as $\mathcal{F}_i(\mathbf{m}) = \sum_{j=0}^{q-1} r^{i\otimes j} e^{-m_j} / \sum_{j=0}^{q-1} e^{-m_j}$.

At iteration ℓ, the message $\mathbf{m}^{(\ell)}(m,n,s_m)$ from CN m to VN n is

$$\mathbf{m}^{(\ell)}(m,n,s_m) = \mathcal{A}[\bar{s}_m]\mathcal{F}^{-1}\left(\prod_{n'\in\mathcal{N}(m)\backslash n} \mathcal{F}\left(W\left[\overline{H}_{n'm}\right]\mathbf{m}^{(\ell-1)}(n',m,y_{n'})\right)\right) \tag{7}$$

where $\bar{s}_m = \ominus s_m \oslash H_{n,m}$, $\overline{H}_{n'm} = \ominus H_{n',m} \oslash H_{n,m}$ and $W\left[a\right]$ is the $q \times q$ matrix such that $W\left[a\right]_{k,n} = \delta(a \otimes n \ominus k), 0 \leqslant k, n \leqslant q-1$. $\mathcal{A}[k]$ is a $q \times q$ matrix that maps a vector message $\mathbf{m}$ into a vector message $\mathbf{l} = \mathcal{A}[k]\mathbf{m}$ with $l_j = m_{j\oplus k} - m_k$. Note that $\mathcal{A}[k]$ does not appear in the channel coding version of the algorithm and is specific to SW coding. The derivation of (7) is shown in the appendix. At a VN n, a message $\mathbf{m}^{(\ell)}(n,m,y_i)$ is sent to the CN m and an *a posteriori* message $\tilde{\mathbf{m}}^{(\ell)}(n,y_n)$ is computed. They both satisfy

$$\mathbf{m}^{(\ell)}(n,m,y_n) = \sum_{m'\in\mathcal{N}(n)\backslash m} \mathbf{m}^{(\ell)}(m',n,s_{m'}) + \mathbf{m}^{(0)}(n,y_n)\ , \tag{8}$$

$$\tilde{\mathbf{m}}^{(\ell)}(n,y_n) = \sum_{m'\in\mathcal{N}(n)} \mathbf{m}^{(\ell)}(m',n,s_{m'}) + \mathbf{m}^{(0)}(n,y_n)\ . \tag{9}$$

From (9), each VN n produces an estimate $\widehat{x}_n^{(\ell)} = \arg\max_k \tilde{m}_k^{(\ell)}(n,y_n)$ of x_n. The algorithm ends if $\mathbf{s} = H^{\mathrm{T}}\widehat{\mathbf{x}}^{(\ell)}$ or if $\ell = L_{\max}$, the maximum number of iterations.

When the conditional distribution $P(Y|X)$ is uncertain, the previously described decoding algorithm cannot be applied directly, because the initial messages (6) cannot be evaluated accurately.

B. SwP-Source: EM algorithm

We first consider the SwP-Source and then extend the proposed algorithm to the cases of the DwP- and SP-Sources. For the SwP-Source, one needs the actual value of the parameter vector $\boldsymbol{\theta}$ because the sum-product LDPC decoder requires the knowledge of the conditional distribution

$P(X|Y)$. The EM algorithm is thus used to estimate jointly the source sequence $\mathbf{X}$ and the parameter $\boldsymbol{\theta}$. A method to produce coarse estimates of the parameters is also described.

1) Joint estimation of $\boldsymbol{\theta}$ and $\mathbf{x}$*:* The joint estimation of the source vector $\mathbf{x}$ and of the parameter $\boldsymbol{\theta}$ from the observed vectors $\mathbf{y}$ and $\mathbf{s}$ is performed via the EM algorithm [17]. Knowing some estimate $\boldsymbol{\theta}^{(\ell)}$ obtained at iteration ℓ, the EM algorithm maximizes, with respect to $\boldsymbol{\theta}$,

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\ell)}) &= E_{\mathbf{X}|\mathbf{y},\mathbf{s},\boldsymbol{\theta}^{(\ell)}} \left[\log P(\mathbf{y}|\mathbf{X}, \mathbf{s}, \boldsymbol{\theta})\right] && (10) \\ &= \sum_{\mathbf{x} \in \mathrm{GF}(q)^n} P(\mathbf{x}|\mathbf{y}, \mathbf{s}, \boldsymbol{\theta}^{(\ell)}) \log P(\mathbf{y}|\mathbf{x}, \mathbf{s}, \boldsymbol{\theta}) && (11) \\ &= \sum_{n=1}^{N} \sum_{k=0}^{q-1} P(X_n = k|y_n, \mathbf{s}, \boldsymbol{\theta}^{(\ell)}) \log P(y_n|X_n = k, \boldsymbol{\theta}) \ . \end{aligned}$$

Solving this maximization problem gives the update equations detailed in Lemma 1. For simplicity, the correlation model between X and Y is assumed to be additive, *i.e.*, there exists a random variable Z such that $Y = X \oplus Z$ and $\boldsymbol{\theta}$ parametrizes the distribution of Z. The Binary Symmetric correlation Channel (BSC) of unknown transition probability $\theta = P(Y = 1|X = 0) = P(Y = 0|X = 1)$ is a special case, where Z is a binary random variable such that $P(Z = 1) = \theta$.

Lemma 1. *Let (X, Y) be a binary SwP-Source. Let the correlation channel be a Binary Symmetric channel (BSC) with parameter $\theta = P(Y = 0|X = 1) = P(Y = 1|X = 0)$, $\theta \in [0, 1]$. The update equation for the EM algorithm is [35]*

$$\theta^{(\ell+1)} = \frac{1}{N} \sum_{n=1}^{N} |y_n - p_n^{(\ell)}| \qquad (12)$$

where $p_n^{(\ell)} = P(X_n = 1|y_n, \mathbf{s}, \theta^{(\ell)})$.

Let (X, Y) be a SwP-Source that generates symbols in GF(q). Let the correlation channel be such that $Y = X \oplus Z$, where Z is a random variable in GF(q), and $P(Z = k) = \theta_k$. The update equations for the EM algorithm are

$$\forall k \in GF(q), \ \theta_k^{(\ell+1)} = \frac{\sum_{n=1}^{N} P_{y_n \ominus k, n}^{(\ell)}}{\sum_{n=1}^{N} \sum_{k'=0}^{q-1} P_{y_n \ominus k', n}^{(\ell)}} \qquad (13)$$

where $P_{k,n}^{(\ell)} = P(X_n = k | y_n, \mathrm{s}, \boldsymbol{\theta}^{(\ell)})$.

Proof: The binary case is provided by [35]. In the non-binary case, the updated estimate is obtained by maximizing (10) taking into account the constraints $0 \leq \theta_k \leq 1$ and $\sum_{k=0}^{q-1} \theta_k = 1$. ■

Note that $P_{k,n}^{(\ell)} = P(X_n = k | y_n, \mathrm{s}, \boldsymbol{\theta}^{(\ell)})$ in (13) can be estimated with a sum-product algorithm that assumes that the true parameter is $\boldsymbol{\theta}^{(\ell)}$.

2) Initialization of the EM algorithm: We now propose an efficient initialization of the EM algorithm valid for irregular codes and for sources X and Y taking values in GF(q). This generalizes the method proposed in [35] for regular and binary codes. The rationale is to derive a Maximum Likelihood (ML) estimate of $\boldsymbol{\theta}$ from a function $\mathbf{u} = H^{\mathrm{T}}\mathbf{x} \oplus H^{\mathrm{T}}\mathbf{y}$ of the observed data $H^{\mathrm{T}}\mathbf{x}$ and $\mathbf{y}$.

a) The BSC with irregular codes: In this case, each binary random variable U_m is the sum of random variables of $\mathbf{Z}$. Although each Z_n appears in several sums, the following assumption is made in this section.

Assumption 1. *Each U_m is obtained from* i.i.d. *random variables $Z_j^{(m)}$.*

The validity of this assumption depends on the choice of the matrix H and is not true in general. Although it produces an approximate solution, this choice may lead to a reasonable initialization for the EM algorithm. Furthermore, the number of terms in the sum for U_m depends on the degree of the CN m. One can use the CN degree distribution $\rho(x)$ as a probability distribution for these degrees, or decide to take into account the knowledge of the CN degrees. Both cases lead to a probability model for the U_m and enable to obtain an ML estimate for θ, as described in the two following lemmas.

Lemma 2. *Let $\mathbf{U}$ be a binary random vector of length M. Each U_m is the sum of J_m identically distributed binary random variables $Z_j^{(m)}$,* i.e., *$U_m = \sum_{j=1}^{J_m} Z_j^{(m)}$, where the $Z_j^{(m)}$ are independent*

$\forall j, m$. $\{J_m\}_{m=1}^M$ *are* i.i.d. *random variables taking their values in* $\{2, \dots, d_c\}$ *with known probability* $P(J = j) = \rho_j$. *Denote* $\theta = P(Z = 1)$, $\alpha = P(U = 1)$ *and assume that* θ *and* α *are unknown. Then their ML estimates* $\widehat{\theta}$ *and* $\widehat{\alpha}$ *from an observed vector* $\mathbf{u}$ *satisfy* $\widehat{\alpha} = \frac{1}{M}\sum_{m=1}^M u_m$ *and* $\widehat{\theta} = f^{-1}(\widehat{\alpha})$, *where* f *is the invertible function* $f(\theta) = \frac{1}{2} - \frac{1}{2}\sum_{j=2}^{d_c} \rho_j (1-2\theta)^j$, $\forall \theta \in [0, \frac{1}{2}]$.

Proof: The random variables U_m are independent (sums of independent variables). They are identically distributed because the J_m and the $Z_j^{(m)}$ are identically distributed. $\alpha = P(U = 1) = \sum_{j=2}^{d_c} \rho_j P(U = 1|J = j)$. Then, from [35], $P(U = 1|J = j) = \sum_{i=1, i \text{ odd}}^{j} \binom{j}{i} \theta^i (1-\theta)^{j-i}$ and from [12, Section 3.8], $P(U = 1|J = j) = \frac{1}{2} - \frac{1}{2}(1-2\theta)^j$. Thus $\alpha = f(\theta)$. The ML estimate $\widehat{\alpha}$ of α given $\mathbf{u}$ is $\widehat{\alpha} = \frac{1}{M}\sum_{m=1}^M u_m$. Finally, as f is invertible for $\theta \in \left[0, \frac{1}{2}\right]$, then from [19, Theorem 7.2], the ML estimate of θ is given by $\widehat{\theta} = f^{-1}(\widehat{\alpha})$. ■

Lemma 3. *Let* $\mathbf{U}$ *be a binary random vector of length* M. *Each* U_m *is the sum of* j_m *identically distributed binary random variables* $Z_j^{(m)}$, i.e., $U_m = \sum_{j=1}^{j_m} Z_j^{(m)}$, *where* $Z_j^{(m)}$ *are independent* $\forall j, m$. *The values of* j_m *are known and belong to* $\{2, \dots, d_c\}$. *Denote* $\theta = P(Z = 1)$ *and assume that* θ *is unknown. Then the entries of* $\mathbf{U}$ *are independent and the ML estimate* $\widehat{\theta}$ *from an observed vector* $\mathbf{u}$ *is the argument of the maximum of*

$$L(\theta) = \sum_{j=2}^{d_c} \mathbb{N}_{1,j}(\mathbf{u}) \log\left(\frac{1}{2} - \frac{1}{2}(1-2\theta)^j\right) + \sum_{j=2}^{d_c} \mathbb{N}_{0,j}(\mathbf{u}) \log\left(\frac{1}{2} + \frac{1}{2}(1-2\theta)^j\right) \tag{14}$$

where $\mathbb{N}_{1,j}(\mathbf{u})$ *and* $\mathbb{N}_{0,j}(\mathbf{u})$ *are the number of symbols in* $\mathbf{u}$ *obtained from the sum of* j *elements and respectively equal to* 1 *and* 0.

Proof: The random variables U_m are independent (sums of independent variables). Therefore, the likelihood function satisfy $L(\theta) = \log P(\mathbf{u}|\theta) = \sum_{m=1}^M \log P(u_m|j_m, \theta)$. Then, as in the proof of Lemma 2, we obtain (14). ■

The method of Lemma 2 is simpler to implement than the one of Lemma 3 but does not take into account the actual matrix H, at the price of a small loss in performance.

b) The non-binary discrete case: Only the case of regular codes is presented here, but the method can be generalized to irregular codes (see the previous section). Assumption 1 also holds in this case. Now, the probability mass function of Z is given by $\boldsymbol{\theta} = [\theta_0 \dots \theta_{q-1}]$ with $\theta_k = P(Z = k)\ \forall k \in \text{GF}(q)$. Now, each U_m is the sum of symbols of $\mathbf{Z}$, weighted by the coefficients contained in H. A first solution does not exploit the knowledge of these coefficients, but uses the fact that the non-zero coefficients of H are distributed uniformly in $\text{GF}(q)\backslash\{0\}$ (Lemma 4). A second solution takes into account the knowledge of the coefficients (Lemma 5).

Lemma 4. *Let* $\mathbf{U}$ *be a length* M *random vector with entries in GF(q) such that each* U_m *is the sum of* d_c i.i.d. *products of random variables,* i.e., $U_m = \sum_{j=1}^{d_c} H_j^{(m)} Z_j^{(m)}$. *The* $Z_j^{(m)}$ *and* $H_j^{(m)}$ *are identically distributed random variables, mutually and individually independent* $\forall j, m$. *The* $H_j^{(m)}$ *are uniformly distributed in GF(q)*$\backslash\{0\}$. *The* $Z_j^{(m)}$ *take their values in GF(q). Denote* $\theta_k = P(Z = k)$, $\alpha_k = P(U = k)$ *and assume that* $\boldsymbol{\theta} = [\theta_0 \dots \theta_{q-1}]$ *and* $\boldsymbol{\alpha} = [\alpha_0 \dots \alpha_{q-1}]$ *are unknown. Then the random variables of* $\mathbf{U}$ *are independent and the parameters satisfy* $\boldsymbol{\alpha} = f(\boldsymbol{\theta})$, *with*

$$f(\boldsymbol{\theta}) = \sum_{n_1,\dots,n_{q-1}} \binom{d_c}{n_1,\dots,n_{q-1}} \left(\frac{1}{q-1}\right)^{d_c} \mathcal{F}^{-1}\left(\prod_{j=0}^{q-1} \left(\mathcal{F}\left(W[j]\boldsymbol{\theta}\right)\right)^{n_j}\right) \tag{15}$$

where the sum is over all the possible combinations of integers $n_1, \dots, n_{q-1}$ *such that* $0 \leq n_k \leq d_c$ *and* $\sum_{k=1}^{q-1} n_k = d_c$ *and* $\binom{d_c}{n_1,\dots,n_{q-1}}$ *is a multinomial coefficient.*

Denote $\widehat{\boldsymbol{\theta}}$ *and* $\widehat{\boldsymbol{\alpha}}$ *the ML estimates of* $\boldsymbol{\theta}$ *and* $\boldsymbol{\alpha}$, *obtained from an observed vector* $\mathbf{u}$, *with* $\widehat{\alpha}_k = \frac{\mathbb{N}_k(\mathbf{u})}{M}$ *where* $\mathbb{N}_k(\mathbf{u})$ *is the number of occurrences of* k *in the vector* $\mathbf{u}$. *Then, if* f *is invertible,* $\widehat{\boldsymbol{\theta}} = f^{-1}(\widehat{\boldsymbol{\alpha}})$.

Proof: The random variables U_m are independent (sums of independent variables). Then, $\alpha_k = P(U = k) = \sum_{\{h_j\}_{j=1}^{d_c}} P(\{h_j\}_{j=1}^{d_c}) P(U = k | \{h_j\}_{j=1}^{d_c})$ in which the sum is on all the possible combinations of coefficients $\{h_j\}_{j=1}^{d_c}$. This can be simplified as $\alpha_k = \sum_{n_1,\dots,n_{q-1}} P(N_1 = n_1, \dots, N_{q-1} = n_{q-1}) P(U = k | n_1, \dots, n_{q-1})$ where n_k is the number of occurrences of k in

$\{h_j\}_{j=1}^{d_c}$. One has $P(N_1 = n_1, \dots, N_{q-1} = n_{q-1}) = \binom{d_c}{n_1,\dots,n_{q-1}} \left(\frac{1}{q-1}\right)^{d_c}$. Then, the vector denoted

$$P_{\mathbf{U}|n_1,\dots,n_{q-1}} = [P(U = 0|n_1,\dots,n_{q-1}) \dots P(U = q-1|n_1,\dots,n_{q-1})] \tag{16}$$

can be expressed as $P_{\mathbf{U}|n_1,\dots,n_{q-1}} = \mathcal{F}^{-1}\left(\prod_{j=1}^{q-1} \left(\mathcal{F}\left(W[j]\boldsymbol{\theta}\right)\right)^{n_j}\right)$. Therefore,

$$\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_{q-1}] = \sum_{n_1,\dots,n_{q-1}} \binom{d_c}{n_1,\dots,n_{q-1}} \left(\frac{1}{q-1}\right)^{d_c} \mathcal{F}^{-1}\left(\prod_{j=1}^{q-1} \left(\mathcal{F}\left(W[j]\boldsymbol{\theta}\right)\right)^{n_j}\right) . \tag{17}$$

The ML estimates $\widehat{\alpha}_k$ of α_k are $\widehat{\alpha}_k = \frac{\mathbb{N}_k(\mathbf{u})}{M}$. Finally, if f is invertible, then from [19, Theorem 7.2], the ML estimate of $\boldsymbol{\theta}$ is given by $\widehat{\boldsymbol{\theta}} = f^{-1}(\widehat{\boldsymbol{\alpha}})$. ■

Lemma 5. *Let* $\mathbf{U}$ *be a length* M *random vector with entries in GF(q) such that each* U_m *is the sum of* d_c i.i.d. *random variables,* i.e., $U_m = \sum_{j=1}^{d_c} h_j^{(m)} Z_j^{(m)}$. *The* $Z_j^{(m)}$ *are independent* $\forall j, m$, *and identically distributed random variables taking their values in GF(q). The values of the coefficients* $h_j^{(m)}$ *are known and belong to GF(q)*$\backslash\{0\}$. *Denote* $\theta_k = P(Z = k)$, $\alpha_k = P(U = k)$ *and assume that* $\boldsymbol{\theta} = [\theta_0, \dots, \theta_{q-1}]$ *and* $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_{q-1}]$ *are unknown. Then the random variables of* $\mathbf{U}$ *are independent and the ML estimate* $\widehat{\boldsymbol{\theta}}$ *from an observed vector* $\mathbf{u}$ *maximizes*

$$L(\boldsymbol{\theta}) = \sum_{m=1}^{M} \log \mathcal{F}_{u_m}^{-1}\left(\prod_{j=1}^{dc} \mathcal{F}(W[h_j^{(m)}]\boldsymbol{\theta})\right) \tag{18}$$

under the constraints $0 \le \theta_k \le 1$ *and* $\sum_{k=0}^{q-1} \theta_k = 1$.

Proof: The random variables U_m are independent (sums of independent variables). The ML estimate $\widehat{\boldsymbol{\theta}}$ is the value that maximizes the likelihood function given by

$$\begin{aligned} L(\boldsymbol{\theta}) &= \log P(\mathbf{u}|\boldsymbol{\theta}, \{h_j^{(m)}\}_{j=1,m=1}^{d_c,M}) && (19) \\ &= \sum_{m=1}^{M} \log P(u_m|\boldsymbol{\theta}, \{h_j^{(m)}\}_{j=1}^{d_c}) && (20) \end{aligned}$$

under the constraint that $0 \le \theta_k \le 1$ and $\sum_{k=0}^{q-1} \theta_k = 1$. The second equality (20) comes from the independence assumption. Following the steps of Lemma 4, we show that (20) becomes $L(\boldsymbol{\theta}) = \sum_{m=1}^{M} \log \mathcal{F}_{u_m}^{-1}\left(\prod_{j=1}^{d_c} \mathcal{F}(W[h_j^{(m)}]\boldsymbol{\theta})\right)$. ■

C. DwP-Source

The DwP-Source is non-stationary. Consequently, if one assumes a stationary model such that

$$P(X_n = k|Y_n = m) = \alpha_{k,m} \tag{21}$$

and tries to produce an estimate $\hat{\alpha}_{k,m}^{(n)}$ from observed sequences $(\mathbf{x}, \mathbf{y})$ of length n, then the sequence of estimates $\hat{\alpha}_{k,m}^{(n)}$ does not necessarily converge as n goes to infinity. However, such an estimate is well defined for a fixed length n. Thus, we apply the procedure defined for the SwP-Source to get $\hat{\alpha}_{k,m}^{(n)}$ from $\mathbf{y}$ and $\mathbf{u}$.

D. SP-Source: MAP with EM

For the SP-Source, the distribution $P_{\mathbf{\Theta}}(\boldsymbol{\theta})$ is available and one can perform the MAP estimation of $\mathbf{\Theta}$. Then, the EM equation (10) for the MAP estimation becomes [3]

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\ell)}) = E_{\mathbf{X}|\mathbf{y},\mathbf{s},\boldsymbol{\theta}^{(\ell)}} \left[\log P(\mathbf{X}|\mathbf{y}, \mathbf{s}, \boldsymbol{\theta})\right] + \log P_{\mathbf{\Theta}}(\boldsymbol{\theta}) \ . \tag{22}$$

Knowing some estimate $\boldsymbol{\theta}^{(\ell)}$ of $\boldsymbol{\theta}$ at iteration ℓ, one has to maximize (22) with respect to $\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta}^{(\ell+1)}$. As for the SwP-Source, the LDPC decoding algorithm initialized with $\boldsymbol{\theta}^{(l)}$ provides an approximate version of $P(\mathbf{X}|\mathbf{y}, \mathbf{s}, \boldsymbol{\theta}^{(\ell)})$, required to perform the MAP estimation of $\boldsymbol{\theta}^{(l+1)}$.

A coarse estimation of $\boldsymbol{\theta}$ can be obtained from $\mathbf{u} = H^{\mathrm{T}}\mathbf{x} + H^{\mathrm{T}}\mathbf{y}$ as

$$\boldsymbol{\theta}^{(0)} = \arg\max_{\boldsymbol{\theta} \in \mathcal{P}_{\boldsymbol{\theta}}} \log P_{\mathbf{\Theta}}(\boldsymbol{\theta}) + \log P(\mathbf{u}|H, \boldsymbol{\theta}) \tag{23}$$

in order to initialize the EM algorithm. In the binary case and from the assumptions of Lemma 3 this corresponds to maximizing

$$L_{\mathrm{MAP}}(\theta) = \log P_{\Theta}(\theta) + \sum_{j=2}^{d_c} \mathbb{N}_{1,j}(\mathbf{u}) \log\left(\frac{1}{2} - \frac{1}{2}(1-2\theta)^j\right) + \sum_{j=2}^{d_c} \mathbb{N}_{0,j}(\mathbf{u}) \log\left(\frac{1}{2} + \frac{1}{2}(1-2\theta)^j\right) \tag{24}$$

with respect to θ. In the non-binary case and from the assumptions of Lemma 5 this corresponds to maximizing

$$L_{\mathrm{MAP}}(\boldsymbol{\theta}) = \log P_{\mathbf{\Theta}}(\boldsymbol{\theta}) + \sum_{m=1}^{M} \log \mathcal{F}_{u_m}^{-1}\left(\prod_{j=1}^{dc} \mathcal{F}(W[h_{s_{zm},j}]\boldsymbol{\theta})\right) \tag{25}$$

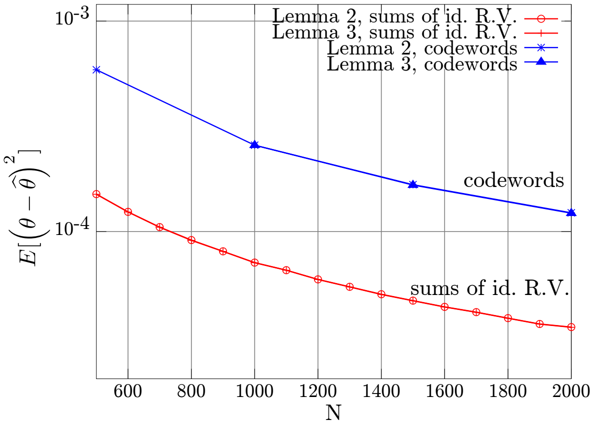

Fig. 1. MSE of the estimators for the binary case. (SwP-Source)

under the constraints $0 \leq \theta_k \leq 1$ and $\sum_{k=0}^{q-1} \theta_k = 1$.

However, this approach does not fully exploit the density over $\boldsymbol{\theta}$ but only its mode, because a hard value of $\boldsymbol{\theta}$ is estimated at each iteration and used for the following iterations. To deal with this problem, one could think of using Variational Bayesian Expectation Maximization (VBEM) [3]. Unfortunately, the VBEM equations are intractable for most of the distributions, particularly in the discrete case. The discrete additive model considered here is not a conjugate exponential model, for which a tractable implementation exists.

V. Simulations

The performance of the initialization techniques obtained in Lemmas 2 to 5 are first compared. Then, we evaluate the joint estimation methods proposed for the various models introduced in Section II. In the following, the random variables are either binary or non-binary with values in GF(4). The correlation model is such that there exists a random variable Z with $Y = X \oplus Z$, and X is distributed uniformly. The Matlab codes are available on demand.

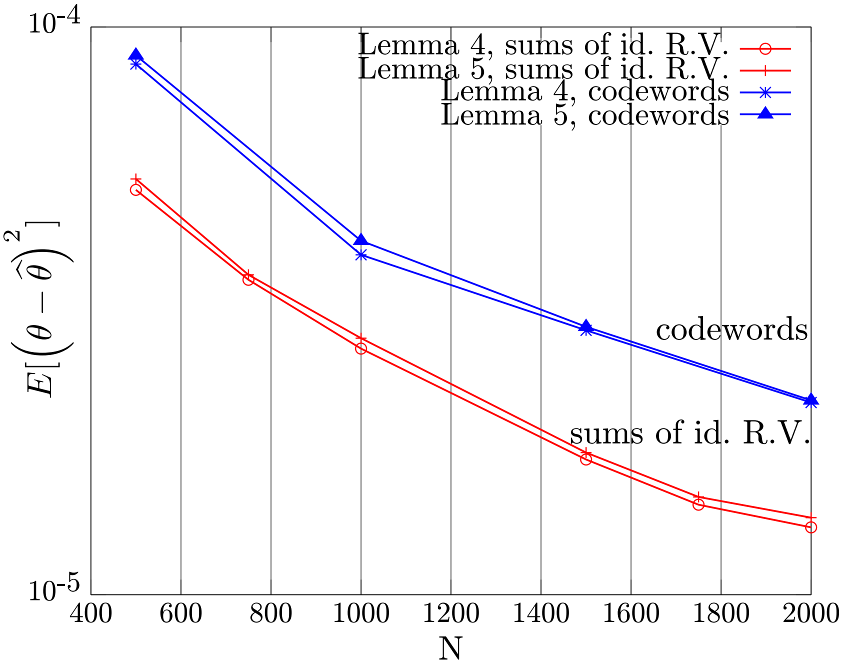

Fig. 2. MSE of the estimators for the non- binary case. (SwP-Source)

A. Performance of the initialization techniques (SwP-Model)

The binary case is considered first. Z is such that $P(Z=1)=\theta$, $\theta \in \mathcal{P}_\theta = [0, 0.18]$. We choose a code $\lambda(x) = 0.4295x + 0.2750x^2 + 0.0745x^8 + 0.1150x^9 + 0.0035x^{11} + 0.0930x^{14} + 0.0095x^{15}$ and $\rho(x) = 0.2187x^5 + 0.7760x^6 + 0.0053x^7$, designed for the worst possible parameter $\theta = 0.18$ and obtained from a code optimization realized with a differential evolution algorithm [34]. The rate of this code is $R = 0.75$ bit/symbol. The initialization methods of Lemmas 2 and 3 are evaluated and compared through two experiments. Indeed, the models defined in the formulations of the lemmas are supposed to represent the behavior of the LDPC encoding using Assumption 1. In this section, we want to determine whether this assumption is meaningful.

First, we wish to evaluate the performance of the estimation methods on simulated codewords, *i.e.*, generated at random from the models as they are defined in the formulations of the lemmas. For that purpose, 50000 vectors $\mathbf{U}$ of length M are generated according to the models introduced in Lemmas 2 and 3, for $\theta = 0.1$. Assumption 1 is taken into account and the symbols U_m are drawn as sums of independent random variables. Then, the two proposed estimation methods are applied and the Mean Squared Error (MSE) $E\left[(\theta - \widehat{\theta})^2\right]$ is evaluated as a function of $N = \frac{M}{R}$. The estimated parameters are obtained numerically from a gradient descent initialized at random in $\mathcal{P}_\theta$. This gives the two superposed lower curves of Figure 1, showing that the methods of the two lemmas provide similar performance.

Second, as the models introduced in the lemmas are supposed to represent the effects of the LDPC encoding, we also evaluate the performance of the estimators on actual codewords, *i.e.*, obtained from LDPC coding. Consequently, 10000 vectors $\mathbf{z}$ of length N are generated considering $\theta = 0.1$. Note that the estimation method requires the knowledge of $\mathbf{u} = H^{\mathrm{T}}\mathbf{y} \ominus H^{\mathrm{T}}\mathbf{x} = H^{\mathrm{T}}\mathbf{z}$ and thus vectors $\mathbf{z}$ are generated directly. The vectors $\mathbf{u}$ are then obtained by multiplying $\mathbf{z}$ by a matrix H of the considered code. The two proposed estimation methods are then applied to each realization to evaluate the MSE. This gives the two superposed upper curves of Figure 1. As before the two methods give the same performance. However, we observe a loss of a factor 10 in MSE compared to the ideal case, due to the fact that the entries of $\mathbf{U}$ are not independent. Nevertheless, the performance seems sufficient for the initialization of the EM algorithm.

For the non-binary case, the probability distribution of Z is given by $\boldsymbol{\theta} = [\theta_0, \dots, \theta_3]$ where $Pr(Z = k) = \theta_k$. The set $\mathcal{P}_{\boldsymbol{\theta}}$ is such that $\forall \boldsymbol{\theta} \in \mathcal{P}_{\boldsymbol{\theta}}$, $\theta_0 \geq 0.76$. We choose a code $\lambda(x) = 0.413x + 0.375x^2 + 0.012x^4$ and $\rho(x) = x$, giving $R = 1.6$ bit/symbol. In this case, the code was tuned for the worst case $\boldsymbol{\theta} = [0.76, 0.08, 0.08, 0.08]$. We do not perform code optimization here because it is not defined, neither for the non-binary SW setup (see [6] for the binary SW setup), neither for the models we consider (see [20] for density evolution for non-binary codes with Gaussian approximation). However, a code with variable node degree 2 exhibit good performance in the non-binary case [25] and the check node degree distribution is chosen to obtain the rate $R = 1.6$ bit/symbol. The experiments described in the binary case are repeated for the methods proposed in Lemmas 4 and 5. The parameter estimates are now obtained from a projected gradient descent. Figure 2 shows the MSE of the two cases obtained by averaging over 1000 vectors of length 10000, generated from $\boldsymbol{\theta} = [0.82\ 0.06\ 0.06\ 0.06]$ The conclusions of the binary case hold also in this setup and in the following, the method of Lemma 4 is used since it is less complex.

B. Complete coding scheme for the SwP- and SP-Sources

The performance of the complete scheme is now evaluated, in the non-binary case. The parameter $\boldsymbol{\theta}$ of the distribution of Z can take the following forms: $\boldsymbol{\theta}^{(0)} = [1-3\alpha, \alpha, \alpha, \alpha]$, $\boldsymbol{\theta}^{(1)} = [\alpha, 1-3\alpha, \alpha, \alpha]$, $\boldsymbol{\theta}^{(2)} = [\alpha, \alpha, 1-3\alpha, \alpha]$, or $\boldsymbol{\theta}^{(3)} = [\alpha, \alpha, \alpha, 1-3\alpha]$, where $\alpha \in [0, 0.08]$. The worst case in terms of rate is given by $\alpha = 0.08$ and corresponds to four distinct elements of $\mathcal{P}_{\boldsymbol{\theta}}$.

The case of the SwP-Source is treated first, and four setups are compared. In each setup, 100 source vectors of length 1000 are generated. For each vector, a $\boldsymbol{\theta}^{(i)}$ is selected uniformly at random and α is sampled uniformly at random in $[0, 0.08]$. The considered LDPC code is the one defined in Section V-A for the non-binary case. We set 20 iterations for the LDPC decoder and 3 iterations for the EM algorithm (when required). Three performance criteria are considered: the coding rate, given by the choice of the code, the mean error calculated for each vector as $\frac{\sum_{n=1}^{1000} \mathbf{1}_{X_n \neq \hat{X}_n}}{1000}$ where $\mathbf{1}$ is the indicator function, and the mean decoding time. All the results are presented in Table I.

For the genie-aided setup, $\boldsymbol{\theta}$ is given to the decoder.

For the second setup, a learning sequence of 200 source symbols is first transmitted by the encoder, to enable the decoder to produce a ML estimate of $\boldsymbol{\theta}$. This results in a rate increase because the learning sequence is transmitted without the help of the SI, at $H(X) = 2$ bits/symbol. Note that the mean decoding time is smaller than in the genie-aided setup, because the LDPC code processes blocs of only 800 symbols.

Setup	Err	Time (s)	Rate (bit/symb.)
Genie-aided	$< 10^{-5}$	5.4	1.6
Learn. Seq.	$< 10^{-5}$	4.2	1.7
EM	$< 10^{-5}$	9.1	1.6
EM random	7.2×10^{-3}	47.0	1.6

TABLE I

SETUP COMPARISON FOR THE SWP-SOURCE

The third setup corresponds to the method presented in the paper. Coarse estimate of $\boldsymbol{\theta}$ obtained from Lemma 4 initializes the EM algorithm. Since $\mathcal{P}_{\boldsymbol{\theta}}$ consists of disjoint sets, the first idea is to try to estimate $\boldsymbol{\theta}$ successively with the four possible $\boldsymbol{\theta}^{(j)}$, and to keep the $\hat{\boldsymbol{\theta}}^{(j)}$ that maximizes $L(\hat{\boldsymbol{\theta}}^{(j)})$, introduced in (18). Unfortunately, we see that this solution only works when $j = 0$. This suggests that the method performs well only if $\mathbf{z}$ has a significant number of null entries. For this reason, the following approach is applied. For $j = 0, \dots, 3$, we compute $\tilde{\mathbf{u}}^{(j)} = H^{\mathrm{T}}\mathbf{x} \ominus H^{\mathrm{T}}\mathbf{y} \ominus H^{T}\mathbf{j}$ where $\mathbf{j}$ is a column vector full of j. Then an estimate $\hat{\boldsymbol{\theta}}^{(j)}$ is produced from each $\tilde{\mathbf{u}}^{(j)}$. The corresponding log-likelihood value $L(\hat{\boldsymbol{\theta}}^{(j)})$ is calculated and we keep the $\hat{\boldsymbol{\theta}}^{(j)}$ that maximizes $L(\hat{\boldsymbol{\theta}}^{(j)})$. In this case, the method performs well.

In the fourth setup, the EM algorithm is initialized at random. We first assume that the $\boldsymbol{\theta}^{(0)}$ is the true distribution, and initialize α at random in $[0, 0.08]$. If the EM algorithm does not converge with this assumption, the procedure is repeated, assuming that the distribution is $\boldsymbol{\theta}^{(1)}$, and so on until convergence or until $\boldsymbol{\theta}^{(3)}$ has been tested. We see that this method increases the decoding time and produces poor performance.

For the SP-Source, the same model is considered and the prior distribution on the $\boldsymbol{\theta}^{(j)}$ is such that $P(\boldsymbol{\theta}^{(j)}) = p_j$, $\mathbf{p}_{\boldsymbol{\theta}} = [p_0 \dots p_3] = [0.5, 0.25, 0.125, 0.125]$ and α is distributed uniformly on $[0, 0.08]$. The four setups: genie-aided, learning sequence, method described in the paper, EM initialized at random, are tested again over 100 source vectors of length 1000. In the genie-

Setup	Err	Time (s)	Rate (bit/symb.)
Genie-aided	$< 10^{-5}$	5.4	1.6
Learn. Seq.	$< 10^{-5}$	4.2	1.7
EM	$< 10^{-5}$	9.1	1.6
EM random	8.4×10^{-3}	41.0	1.6

TABLE II

SETUP COMPARISON FOR THE SP-SOURCE

aided setup, $\boldsymbol{\theta}$ is given to the decoder. In the learning-sequence case, a MAP estimate of $\boldsymbol{\theta}$ is produced from a learning sequence of length 200. For the method described in the paper, the initialization method is performed as before by trying to estimate $\boldsymbol{\theta}$ successively with the four possible $\boldsymbol{\theta}^{(j)}$. But now, the selection is performed by keeping the $\hat{\boldsymbol{\theta}}^{(j)}$ that maximizes $L_{\text{MAP}}(\hat{\boldsymbol{\theta}}^{(j)}) = L(\hat{\boldsymbol{\theta}}^{(j)}) + \log p_j$. For the EM initialized at random, the successive assumptions $\boldsymbol{\theta}^{(j)}$ are tested as before, beginning with the $\boldsymbol{\theta}^{(j)}$ of highest *a priori* probability. The results are presented in Table II. We see that there is no change compared to the SwP-setup, except that the EM random is slightly faster.

C. Comparison to a solution with feedback

In this section we compare our no-feedback coding approach with a 1-bit feedback transmission for a source generated from the SwP-Model of Section V-B.. The 1-bit feedback is sent by the receiver to the encoder to ask for additional packets or stop the transmission. The goal is to save rate by avoiding sending data at the worst rate as in the no-feedback method. However, it results in multiple decodings and thus potentially large delays. It is therefore of interest to study the rate/decoding delay tradeoff.

Only an evaluation of the achievable rate and estimated mean-time decoding are provided. It is sufficient to determine the advantages and the drawbacks of the solution with feedback.

In the solution with feedback, when the decoder cannot decode with the received codeword, it requests more check equations via the feedback channel. Each time it receives new equations, the decoder tries to reconstruct the source vector with the use of a sum-product LDPC decoder.

Denote n the length of the source vector and $\overline{\alpha} = 0.08$, consider K rate levels $R_1, \dots, R_K$ associated to K intervals $I_1 = [0, \overline{\alpha}/K] \dots I_K = [\overline{\alpha}(K-1)/K, \overline{\alpha}]$. The coding system processes as follows. The encoder first sends nR_1 symbols to the decoder. The decoder tries to reconstruct the source, assuming the true parameter is $\overline{\alpha}/K$ and considering successively the four assumptions $\boldsymbol{\theta}^{(0)}$ to $\boldsymbol{\theta}^{(3)}$. If it fails, it sends a request via the feedback channel and the encoder sends $n(R_2 - R_1)$ new symbols. The decoder then tries to reconstruct the source from the nR_2 received symbols, assuming the true parameter is $2 \times \overline{\alpha}/K$. The process continues until the source vector has been decoded. Note that here, it is assumed that the I_k intervals are small enough to allow the decoder to perform well with a parameter that is not exactly the true one.

Five setups are compared, in terms or achievable rate (R) and of estimated mean decoding time (T). The results are shown in Figure 3. Denote t the decoding time of one LDPC decoder iteration and N_{it} the required number of iterations. In the following, we set $K = 8$, $N_{\text{it}} = 20$ and choose $t = 0.2s$ from the previous experiments. Remark that the entropy of a source of parameter α is the same whatever $\boldsymbol{\theta}^{(i)}$ and denote $h(\alpha) = H(X|Y, \alpha)$. In each case, we assume that a code or a sequence of codes reaching the entropy can be constructed.

For the solution with feedback, assume that we can construct a sequence of codes such that $R_1 = h(\overline{\alpha}/K), \dots R_K, = h(\overline{\alpha})$ and achieving small probability of error respectively for $\alpha \in I_1, \dots, \alpha \in I_K$. Thus, $\alpha \in I_k$, $R_k = h(k\overline{\alpha}/K)$. We also assume that the delay induced by the feedback is negligible compared to the decoding time. Then, for $\alpha \in I_k$ the mean decoding time is estimated as $T_k = 2 \times t \times N_{\text{it}} \times k$ where the coefficient 2 comes from the assumption that $\boldsymbol{\theta}^{(i)}$ is selected uniformly at random. In the curve of Figure 3, the circles represent the various (R_k, T_k).

For the genie-aided setup, the rate is dimensioned for the worst case, *i.e.*, $R = h(\overline{\alpha})$ and

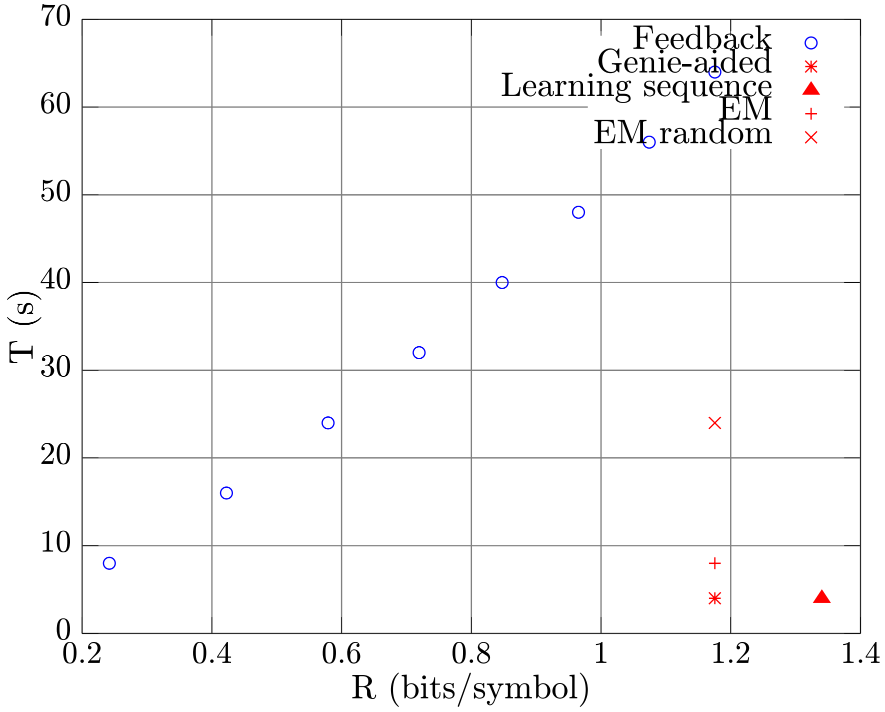

Fig. 3. Rate/Time performance of a solution with feedback

an approximation of the mean decoding time is calculated as $T = N_{\text{it}} \times t$. For the setup with learning sequence, assuming a sequence length representing a fraction $1/5$ of the total length, $R = 4/5h(\overline{\alpha}) + 1/5H(X)$ and $T = N_{\text{it}} \times t$. For the coding scheme described in the paper, $R = h(\overline{\alpha})$ and we approximate $T = 2 \times N_{\text{it}} \times t$, assuming that 2 iterations of the EM algorithm are required, and that the $\boldsymbol{\theta}^{(i)}$ is correctly retrieved at the initialization step. For the coding scheme with EM initialized at random, $R = h(\overline{\alpha})$ and we approximate $T = 3 \times 2 \times N_{\text{it}} \times t$, assuming that 3 iterations of the EM algorithm are required, and that $\boldsymbol{\theta}^{(i)}$ is selected uniformly at random.

When the parameter α is small, the solution with feedback induces a significant rate gain. However, when α increases, the price to pay for adaptated rate is a very large decoding delay. The choice of the parameter K is important: if K decreases, the size of the intervals I_k increases which reduces the mean decoding time. On the other hand, as for $\alpha \in I_k$, the effective coding

m	Err	Time (s)
1	$< 10^{-5}$	6.6
100	$< 10^{-5}$	6.9
500	$< 10^{-5}$	7.2
random	$< 10^{-5}$	10.8

TABLE III

SETUP COMPARISON FOR THE DWP-SOURCE

rate is R_k, the rate needed to decode for α can increase.

D. DwP-Source

The solution proposed for the DwP-Source is now evaluated in the non-binary case. The distribution of Z is given by $\boldsymbol{\pi} = [1 - 3\pi, \pi, \pi, \pi]$, where π can take four different values: $\pi_0 = 0$, $\pi_1 = 0.02$, $\pi_2 = 0.04$ and $\pi_3 = 0.06$, defining four states. Indeed, for the DwP-Model, the worst case is taken on the convex hull of the possible distributions.

We now consider source vectors of length 1000 and fix a block length m. For each block of length m in a vector, a probability distribution for the states is generated uniformly at random. The values $m = 1, 100$ and 500 are tested. A fourth setup is tested for m generated uniformly at random in $\{1, ..., 1000\}$. More precisely, a first value m_0 is generated, a second one m_1 is generated after m_0 symbols and so on. The method proposed for the SwP-Source is then applied with the same code over 100 realizations for each m. The complete decoding technique described for the SwP-Source is used: coarse estimate of the parameter from Lemma 4 followed by EM algorithm. The decoder is provided the information that the set of possible distributions for Z is on the form $[1 - 3\theta, \theta, \theta, \theta]$ and $\theta \in [0, 0.06]$. The results are presented in Table III. Mean error and mean decoding time are measured. We see that smaller m give smaller decoding time, because with a small m, the source behavior is better approximated by the SwP-Model.

Now consider a DwP-Model with four states, giving four possible distributions for Z: $\boldsymbol{\pi}^{(0)} = [0.91, 0.03, 0.03, 0.03]$, $\boldsymbol{\pi}^{(1)} = [0.03, 0.91, 0.03, 0.03]$, $\boldsymbol{\pi}^{(2)} = [0.03, 0.03, 0.91, 0.03]$, $\boldsymbol{\pi}^{(3)} = [0.03, 0.03, 0.03, 0.91]$. In this case, from (2), the infimum of achievable rates is given by $R = 2$ bits/symbol = $H(X)$ bits/symbol, because the infimum of achievable rates is given by the worst cases on the convex hull of the set of possible distributions. Therefore, the SI cannot be exploited.

VI. Conclusion

This paper introduced four signal models modeling the uncertainty on the correlation channel between the source and the SI. Practical coding schemes based on non-binary LDPC codes were proposed for the SW setup and for the four models. Simulation results exhibit good performance in terms of probability of error, rate, or decoding delay, compared to the solution with a learning sequence or the solution with an EM algorithm initialized at random.

Future works will be on the design of good degree distributions for our models with non-binary symbols, and on the extension to the lossy case. We will also investigate correlation model selection, *i.e.*, the choice of one of the four source correlation models and of the structure of the family distribution for the model.

Acknowledgments

The authors would like to thank Valentin Savin for the helpful discussions and advices.

Appendix

In this appendix, we detail the derivation of the update rule (7) at a CN for the SW problem, when the LDPC code is non binary and the decoder is the sum-product algorithm. This update rule derives from the parity check equation at CN m, given by $\sum_{n' \in \mathcal{N}(m)} H_{m,n'} \otimes x_{n'} = s_m$, that can be restated as

$$x_n = s_m \oslash H_{m,n} \ominus \sum_{n' \in \mathcal{N}(m) \backslash n} (H_{m,n'} \oslash H_{m,n}) \otimes x_{n'} \ . \tag{26}$$

The update rule at a CN, and for the sum-product algorithm, consists in computing the reliability information on the variable x_n as a function of the reliability information on the variables $x_{n'}$, denoted $\mathbf{m}^{(\ell-1)}(n', m, y_{n'})$. Thus, the k-th component of the CN message m to VN n (7) is

$$\log \frac{P\left(X_n = 0 | s_m, \{\mathbf{m}^{(\ell-1)}(n', m, y_{n'})\}_{n' \in \mathcal{N}(m) \backslash n}\right)}{P\left(X_n = k | s_m, \{\mathbf{m}^{(\ell-1)}(n', m, y_{n'})\}_{n' \in \mathcal{N}(m) \backslash n}\right)} \tag{27}$$

We first detail the impact of the operator $\otimes$ on a message, to study the term $\ominus(H_{m,n'} \oslash H_{m,n}) \otimes x_{n'}$ in (26). Given a random variable Z taking its values in GF(q), and with a probability vector $\mathbf{p} = [P(Z = 0), \ldots, P(Z = q-1)]^{\mathrm{T}}$, the probability vector of $a \otimes Z$ satisfies $\mathbf{q} = [P(a \otimes Z = 0), \ldots, P(a \otimes Z = q-1)]^{\mathrm{T}} = W[a]\mathbf{p}$, where the matrix $W[a]$ has been defined just after (7). Similarly, $\mathbf{l} = \left[\log \frac{P(a \otimes Z = 0)}{P(a \otimes Z = 0)}, \ldots, \log \frac{P(a \otimes Z = 0)}{P(a \otimes Z = q-1)}\right]$ is obtained from $\mathbf{m} = \left[\log \frac{P(Z=0)}{P(Z=0)}, \ldots, \log \frac{P(Z=0)}{P(Z=q-1)}\right]$ as $\mathbf{l} = W[a]\mathbf{m}$. Therefore, in (27), we need $W\left[\overline{H}_{n'm}\right] \mathbf{m}^{(\ell-1)}(n', m, y_{n'})$, $\forall n' \in \mathcal{N}(m) \backslash n$, where $\overline{H}_{n'm} = \ominus H_{n',m} \oslash H_{n,m}$.

We now detail the impact of the operator $\ominus \sum$ on a message to deal with $\ominus \sum_{n' \in \mathcal{N}(m) \backslash n} (H_{m,n'} \oslash H_{m,n}) \otimes x_{n'}$. The probabilities of a sum of random variables in GF(q) can be evaluated with the help of a particular Fourier transform [15]. From [20], the the i-th component of the Fourier transform applied on a message vector $\mathbf{m}$ is $\mathcal{F}_i(\mathbf{m}) = \sum_{j=0}^{q-1} r^{i \otimes j} e^{-m_j} / \sum_{j=0}^{q-1} e^{-m_j}$ and the k-th component of its inverse is $\mathcal{F}_k^{-1}(\mathbf{f}) = \log\left(\sum_{i=0}^{q-1} f_i / \sum_{i=0}^{q-1} r^{-i \otimes k} f_i\right)$.

Finally, the term $s_m \oslash H_{m,n}$, specific to SW coding, is taken into account. Denote Γ a random variable taking its values in GF(q) and $\mathbf{m} = \left[\log \frac{P(\Gamma=0)}{P(\Gamma=0)}, \ldots, \log \frac{P(\Gamma=0)}{P(\Gamma=q-1)}\right]$. The message vector $\mathbf{l} = \left[\log \frac{P(a \oplus \Gamma = 0)}{P(a \oplus \Gamma = 0)}, \ldots, \log \frac{P(a \oplus \Gamma = 0)}{P(a \oplus \Gamma = q-1)}\right]$ corresponding to $a \oplus \Gamma$ is obtained as $\mathbf{l} = \mathcal{A}[a]\mathbf{m}$. Setting $a = s_n \oslash H_{m,n}$ gives the final message vector $\mathbf{m}^{(\ell)}(n, m, y_n)$.

REFERENCES

[1] A. Aaron, R. Zhang, and B. Girod. Wyner-Ziv coding of motion video. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 240–244, 2002.

[2] R. Ahlswede. Coloring hypergraphs: A new approach to multi-user source coding-1. *Journal of Combinatorics*, 4(1):76–115, 1979.

[3] M.J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[4] T. Berger. The source coding game. *IEEE Transactions on Information Theory*, 17(1):71–76, Jan 1971.

[5] J. Chen and M.P.C. Fossorier. Near optimum universal belief propagation based decoding of Low-Density Parity Check codes. *IEEE Transactions on Communications*, 50(3):406–414, 2002.

[6] J. Chen, D.K. He, and A. Jagmohan. The equivalence between Slepian-Wolf coding and channel coding under density evolution. *IEEE Transactions on Communications*, 57(9):2534–2540, 2009.

[7] I. Csiszar. Linear codes for sources and source networks: Error exponents, universal coding. *IEEE Transactions on Information Theory*, 28(4):585–592, 1982.

[8] M.C. Davey and D.J.C. MacKay. Low Density Parity Check codes over GF(q). In *Information Theory Workshop, Proceedings.*, pages 70–71, 1998.

[9] E. Dupraz, A. Roumy, and M. Kieffer. Source coding with side information at the decoder: Models with uncertainty, performance Bounds, and practical coding schemes. In *Proceedings of the International Symposium on Information Theory and its Applications 2012*, pages 1–5, October 2012.

[10] E. Dupraz, A. Roumy, and M. Kieffer. Practical coding scheme for universal source coding with side information at the decoder. In *Accepted to the Data Compression Conference, Snowbird*, March 2013.

[11] A.W. Eckford and W. Yu. Rateless Slepian-Wolf Codes. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, pages 1757 – 1761, 2005.

[12] R.G. Gallager. *Low-Density Parity Check Codes*. PhD thesis, Massachusetts Institute of Technology, 1963.

[13] R.G. Gallager. *Information theory and reliable communication*. Wiley, 1968.

[14] R.G. Gallager. Source coding with side information and universal coding. 1979.

[15] A. Goupil, M. Colas, G. Gelle, and D. Declercq. FFT-based BP decoding of general LDPC codes over Abelian groups. *IEEE Transactions on Communications*, 55(4):644–649, 2007.

[16] T.S. Han. *Information-spectrum methods in information theory*. Springer, 2003.

[17] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

[18] S. Jalali, S. Verdú, and T. Weissman. A universal scheme for Wyner-Ziv coding of discrete sources. *IEEE Transactions on Information Theory*, 56(4):1737–1750, 2010.

[19] S.M. Kay. *Fundamentals of Statistical Signal Processing, Estimation theory*. Prentice Hall PTR, 1993.

[20] G. Li, I.J. Fair, and W.A. Krzymien. Density evolution for nonbinary LDPC codes under Gaussian approximation. *IEEE Transactions on Information Theory*, 55(3):997–1015, 2009.

[21] B. Liu, J. Gao, W. Tao, and G. Dou. Weighted symbol-flipping decoding algorithm for nonbinary LDPC codes with flipping patterns. *Journal of Systems Engineering and Electronics*, 22(5):848–855, 2011.

[22] A. Liveris, Z. Xiong, and C. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6:440–442, 2002.

[23] T. Matsuta, T. Uyematsu, and R. Matsumoto. Universal Slepian-Wolf source codes using Low-Density Parity-Check matrices. In *IEEE International Symposium on Information Theory, Proceedings.*, pages 186–190, june 2010.

[24] P. Piantanida, G. Matz, and P. Duhamel. Outage behavior of discrete memoryless channels under channel estimation errors. *IEEE Transactions on Information Theory*, 55(9):4221–4239, Sep 2009.

[25] C. Poulliat, M. Fossorier, and D. Declercq. Design of regular $(2, d_c)$-LDPC codes over GF(q) using their binary images. *IEEE Transactions on Communications*, 56(10):1626–1635, october 2008.

[26] R. Puri and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Annual Allerton Conference on Communications Control and Computing, Proceedings.*, volume 40, pages 586–595, 2002.

[27] 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*, 47(2):619–637, 2001.

[28] T.J. Richardson and R.L. Urbanke. The capacity of Low-Density Parity-Check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001.

[29] V. Savin. Min-Max decoding for non binary LDPC codes. In *IEEE International Symposium on Information Theory, Proceedings.*, pages 960–964, 2008.

[30] A. Sgarro. Source coding with side information at several decoders. *IEEE Transactions on Information Theory*, 23(2):179–182, 1977.

[31] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, July 1973.

[32] V. Stankovic, A.D.Liveris, Z. Xiong, and C.N. Georghiades. On code design for the Slepian-Wolf problem and lossless multiterminal networks. *IEEE Transactions on Information Theory*, 52(4):1495 –1507, april 2006.

[33] V. Stankovic, A.D. Liveris, Z. Xiong, and C.N. Georghiades. Design of Slepian-Wolf codes by channel code partitioning. In *Data Compression Conference, Proceedings.*, pages 302 – 311, march 2004.

[34] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[35] V. Toto-Zarasoa, A. Roumy, and C. Guillemot. Maximum Likelihood BSC Parameter Estimation for the Slepian-Wolf Problem. *IEEE Communications Letters*, 15(2):232 –234, February 2011.

[36] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive codes for distributed source coding. *EURASIP Signal Processing*, 86(11):3123–3130, 2006.

[37] Z. Xiong, A.D. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, 21(5):80–94, Sep 2004.

[38] E.H. Yang and D.K. He. Interactive encoding and decoding for one way learning: Near lossless recovery with side information at the decoder. *IEEE Transactions on Information Theory*, 56(4):1808–1824, 2010.