An Iterative Algorithm for Compression of Correlated Sources at Rates Approaching the Slepian-Wolf Bound: Theory and Analysis

F. Daneshgaran, M. Laddomada, and M. Mondin

***Abstract*—This paper proposes a novel iterative algorithm based on Low Density Parity Check codes for compression of correlated sources at rates approaching the Slepian-Wolf bound. The setup considered in the paper looks at the problem of compressing one source without employing the source correlation, and employing the other correlated source as side information at the decoder which decompresses the first source. We demonstrate that depending on the extent of the source correlation estimated through an iterative paradigm, significant compression can be obtained relative to the case the decoder does not use the implicit knowledge of the existence of correlation. Two stages of iterative decoding are employed. During *global iterations* updated estimates of the source correlation is obtained and passed on to the belief-propagation decoder that performs *local iterations* with a pre-defined stopping criterion and/or a maximum number of local decoding iterations. Detailed description of the iterative decoding algorithm with embedded cross-correlation estimation are provided in the paper, in addition to simulation results confirming the potential gains of the approach.**

***Keywords*-Correlated sources; compression; iterative decoding; low density parity check codes; Slepian-Wolf.**

I. INTRODUCTION

Consider two independent identically distributed (i.i.d.) discrete binary memoryless sequences of length k, $X = [x_1, x_2, \ldots, x_k]$ and $Y = [y_1, y_2, \ldots, y_k]$, where pairs of components (x_i, y_i) have joint probability mass function $p(x, y)$. Assume that the two sequences are generated by two transmitters which do not communicate with each other, and that both sequences have to be jointly decoded at a common receiver. Slepian and Wolf demonstrated that the achievable rate region for this problem (i.e., for perfect recovery of both sequences at a joint decoder), is the one identified by the following set of equations imposing constraints on the rates R_X and R_Y by which both correlated sequences are transmitted:

$$\begin{cases} R_X \geq H(X|Y), \\ R_Y \geq H(Y|X), \\ R_X + R_Y \geq H(X,Y) \end{cases} \quad (1)$$

whereby, $H(X|Y)$ is the conditional entropy of source X given source Y, $H(Y|X)$ is the conditional entropy of source Y given source X, and $H(X,Y)$ is the joint entropy of the sources. A pictorial representation of this achievable region is given in Fig. 1.

Fred Daneshgaran is with the ECE Dept., Calif. State Univ., Los Angeles, USA. E-mail: fdanesh@calstatela.edu

Massimiliano Laddomada is with the Electrical Engineering Department of Texas A&M University-Texarkana, USA. E-mail: mladdomada@tamut.edu.

Marina Mondin is with the Dipartimento di Elettronica, Politecnico di Torino, Italy. E-mail: mondin@polito.it

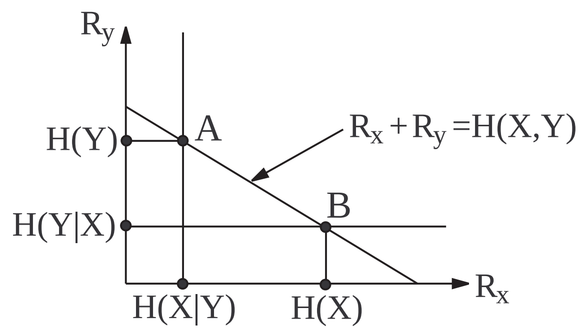

Fig. 1. Rate region for Slepian-Wolf encoding.

In this paper, which is an extended and thorough version of [1], we focus on trying to achieve the corner points A and B in Fig. 1, since any other point between these can be achieved with a time-sharing approach [2]. In particular, we focus on the architecture shown in Fig. 2 in which we assume that one of the two sequences, namely X in our framework, is independently encoded with a source encoder that does not employ the correlation between the sources X and Y. We assume that sequence Y is compressed up to its source entropy $H(Y)$ and is known at the joint decoder as side information, and our aim is at compressing sequence X with a rate $R_1 \leq R_X$ as close as possible to its conditional entropy $R_1 \geq H(X|Y)$ in order to achieve the corner point A in Fig. 1. The decoder tries to decompress the sequence X, in order to obtain an estimate $\widehat{X}$, by employing Y as side information and without the prior knowledge of the amount of correlation between the sequences. Indeed, it estimates this correlation through an iterative algorithm which improves the decoding reliability of X.

Obviously, our solution to joint source coding at point A is directly applicable to point B by symmetry. The overall rate of transmission of both sequences is greater than $H(Y) + H(X|Y) = H(X,Y)$.

With this background, let us provide a quick survey of the recent literature related to the problem addressed in this paper. This survey is by no means exhaustive and is meant to simply provide a sampling of the literature in this area.

In [3], [4], [5], the authors deal with applications of sensor networks for tracking and processing of information to be sent to a common destination. In particular, in [4], the authors inspired by information theory concepts and in particular, the

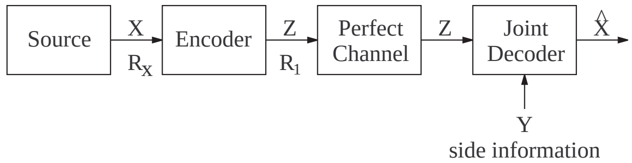

Fig. 2. Architecture of the encoder and joint decoder for the Slepian-Wolf problem.

Slepian-Wolf theorem [2], present the problem of distributed source coding using syndromes. The approach is based on the use of coset codes and significant compression gains are achieved using the proposed technique.

In [6], the authors show that turbo codes can allow to come close to the Slepian-Wolf bound in lossless distributed source coding. In [7], [8], [9], the authors propose a practical coding scheme for separate encoding of the correlated sources for the Slepian-Wolf problem. In [10], the authors propose the use of punctured turbo codes for compression of correlated binary sources whereby compression has been achieved via puncturing. The proposed source decoder utilizes an iterative scheme to estimate the correlation between two different sources. In [11], punctured turbo codes have been applied to the compression of non-binary sources.

The article [12] focuses on the problem of reducing the transmission rate in a distributed environment of correlated sources and the authors propose a source coding scheme for correlated images exploiting modulo encoding of pixel values and compression of the resulting symbols with binary and non-binary turbo codes.

Paper [13] deals with the use of irregular repeat accumulate codes as source-channel codes for the transmission of a memoryless binary sequence with side information at the decoder, while in [14] parallel and serial concatenated convolutional codes are considered for the same problem. In [15], [16], the authors propose a practical coding scheme based on Low Density Parity Check (LDPC) codes for separate encoding of the correlated sources for the Slepian-Wolf problem.

The dual problem of Slepian-Wolf correlated source coding over noisy channels has been dealt with in [17]-[21].

In [22] the authors consider the problem of encoding distributed correlated sources, demonstrating that separate source-channel coding is optimal in any network in which correlated sources do not interfere with each other, and that the information on a network behaves as a flow. Work [23] proposes a distributed joint source-channel coding scheme for multiple correlated sources.

Finally, in [24] the authors consider the design for joint distributed source and network coding.

Almost all proposed works in the literature use soft-metrics for improving the decoding performance. An excellent work describing the "mathematics of soft-decoding" is the paper by Hagenauer *et al.* [25].

Relative to the cited articles, the main novelty of the present work may be summarized as follows: 1) in [10] and [16] the encoder and decoder must both know the correlation between the two sources. We assume knowledge of mean correlation at the encoder. The decoder has implicit knowledge of this via observation of the length of the encoded message. It iteratively estimates the *actual* correlation observed and uses it during decoding; 2) our algorithm can be used with any pair of systematic encoder/decoder without modifying the encoding and decoding algorithm; 3) the proposed algorithm is very efficient in terms of the required number of LDPC decoding iterations. We use quantized integer LLR values (LLRQ) and the loss of our algorithm for using integer LLRQ metrics is quite negligible in light of the fact that it is able to guarantee performance better than that reported in [10] and [16] (where, to the best of our knowledge, authors use floating point metrics) as exemplified by the results shown in table II below; 4) we utilize post detection correlation estimates to generate extrinsic information, which can be applied to any already employed decoder without any modification; and 5) we do not use any interleaver between the sources at the transmitter. Using the approach of [10] in a network, information about interleavers used by different nodes must be communicated and managed. This is not trivial in a distributed network such as the internet. Furthermore, there is a penalty in terms of delay that is incurred.

This paper is an extended version of [1] and is organized as follows. Section II deals with the definition of the encoding algorithm for correlated sources, and presents the class of LDPC codes which is the focus of our current work. In section III, we present modification of the belief-propagation algorithm for decoding of LDPC codes, and follow up with details of our algorithm for iterative joint source decoding of correlated sources with side information and iterative correlation estimation. Section IV deals with the issue of sensitivity of the cross-correlation function between two sequences to residual errors after channel decoding, demonstrating the relative robustness of the empirical cross-correlation measure to channel induced errors. In section V, we present simulation results and comparisons confirming the potential gains that can be obtained from the proposed iterative algorithm. Finally, we present the conclusion in section VI.

II. Architecture of the LDPC-based Source Encoder

This section focuses on the source encoder used for source compression. LDPC coding is essential to achieve performance close to the theoretical limit in [2].

The LDPC matrix [26] for encoding each source is considered as a systematic (n, k) code. The codes used need to be systematic for the decoder to exploit the estimated correlation between X and Y directly.

Each codeword C is composed of a systematic part X, and a parity part Z which together form $C = [X, Z]$. With this setup and given the parity check matrix $H^{n-k,n}$ of the LDPC code, it is possible to decompose $H^{n-k,n}$ as follows:

$$H^{n-k,n} = (H^X, H^Z) \quad (2)$$

whereby, H^X is a $(n-k) \times (k)$ matrix specifying the source bits participating in check equations, and H^Z is a $(n-k) \times$

$(n-k)$ matrix of the form:

$$H^Z = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 1 \end{pmatrix}. \quad (3)$$

The choice of this structure for H, also called staircase LDPC (for the double diagonal of ones in H^Z), has been motivated by the fact that aside from being systematic, we obtain a LDPC code which is encodable in linear time in the codeword length n. In particular, with this structure, the encoding operation is as follows:

$$z_i = \begin{cases} \left[\sum_{j=1}^{k} x_j \cdot H^X_{i,j}\right] \ (mod\ 2), & i = 1 \\ \left[z_{i-1} + \sum_{j=1}^{k} x_j \cdot H^X_{i,j}\right] \ (mod\ 2), & i = 2, ., n-k \end{cases} \quad (4)$$

where, $H^X_{i,j}$ represents the element (i,j) of the matrix H^X, and x_j is the j-th bit of the source sequence X.

Source compression takes place as follows: considering the scheme shown in Fig. 2, we encode the length k sequence belonging to the source X and transmit on a perfect channel only the parity sequence Z, whose bits are evaluated as in (4). The rate guaranteed by such an encoder is

$$R_1 = \frac{n-k}{k}.$$

In relation to the setup shown in Fig. 2, the Slepian-Wolf problem reduces to that of encoding the source X with a rate R_1 as close to $H(X|Y)$ as possible (i.e., $R_1 \geq H(X|Y)$). Note that source correlation is not employed at the encoder.

The objective of the joint decoder as explained in the next section, is to recover sequence X by employing the correlated source Y (considered as a side information at the decoder), and the estimates of the correlation between the sources X and Y obtained in an iterative fashion.

Before explaining the iterative decoding algorithm, let us briefly discuss the correlation model adopted in this framework. We consider the following model in order to follow the same framework pursued in the literature [10], [15]:

$$P(x_j \neq y_j) = p, \ \forall j = 1, \dots, k \quad (5)$$

In light of the considered correlation model, and noting that the sequence Y is available losslessly at the joint decoder, the theoretical limit for lossless compression of X is

$$R_1 \geq H(X|Y) = H(p),$$

whereby $H(p)$ is the binary entropy function defined as

$$H(p) = -p\log_2(p) - (1-p)\log_2(1-p).$$

With this setup, the following holds:

$$R_1 + R_Y \geq H(X,Y) \rightarrow R_1 + 1 \geq H(p) + 1. \quad (6)$$

Note that the encoder needs to know the mean correlation so as to choose a rate close to $H(p)$. It does so, by keeping k constant while choosing n appropriately. We use the term mean correlation, because in any actual setting, the exact correlation between the sequences may be varying about the mean value. Hence, it is beneficial if the decoder estimates the actual correlation value from observations itself. While no side information about the rate is communicated to the decoder, the decoder knows the mean correlation implicitly from the knowledge of block length n.

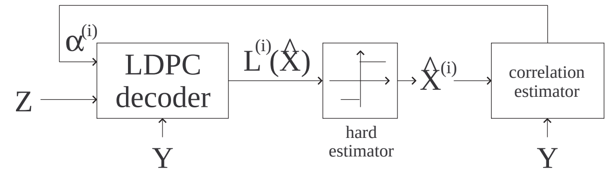

Fig. 3. Architecture of the Iterative Joint decoder of correlated sources.

III. Joint Iterative LDPC-Decoding of Correlated Sources

The architecture of the iterative joint decoder for the Slepian-Wolf problem is depicted in Fig. 3. Its goal is to determine the best estimate $\widehat{X}$ of the source k-sequence X, by starting from the received parity bit $(n-k)$-sequence Z. We note that aside from the fact that the receiver may a-priori presume some correlation between both sequences exist, no side information on this correlation is communicated to the joint decoder. Indeed, this correlation is estimated in an iterative fashion.

With reference to the architecture of the joint decoder depicted in Fig. 3, we note that there are two stages of iterative decoding. Index i denotes a *global iteration* whereby during each global iteration, the updated estimate of the source correlation obtained during the previous global iteration is passed on to the belief-propagation decoder that performs *local iterations* with a pre-defined stopping criterion and/or a maximum number of local decoding iterations.

Based on the notation above, we can now develop the algorithm for exploiting the source correlation in the LDPC decoder. Consider a (n,k)-LDPC identified by the matrix $H^{(n-k,n)}$ as expressed in (2). Note that we only make reference to maximum rank matrix H since the particular structure assumed for H ensures this. In particular, the double diagonal on the parity side of the H matrix always guarantees that the rank of H is equal to the number of its rows, i.e., $n-k$.

It is well known that the parity check matrix H can be described by a bipartite graph with two types of nodes as shown in Fig. 4; n bit-nodes corresponding to the LDPC code bits, and $n-k$ check-nodes corresponding to the parity checks as expressed by the rows of the matrix H. Following the notation employed in [29], let $B(m)$ denote the set of bit-nodes connected to the m-th check-node, and $C(n)$ denote the set of check-nodes adjacent to the n-th bit-node. With this setup, $B(m)$ corresponds to the set of positions of the 1's in the m-th row of H, while $C(n)$ is the set of positions of the 1's in the n-th column of H. In addition, let us use the notation $C(n) \setminus m$ and $B(m) \setminus n$ to mean the sets $C(n)$ and $B(m)$ in which the m-th check-node and the n-th bit-node respectively, are excluded. Furthermore, let us identify with $\lambda_{n,m}(u_n)$ the log-likelihood of the message that the n-th bit-node sends to

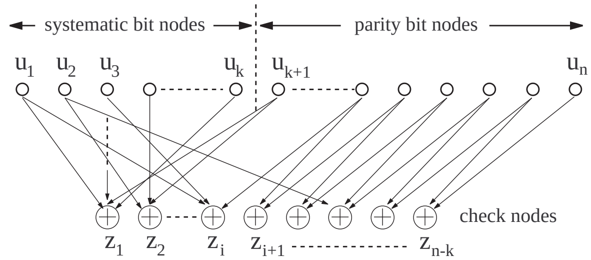

Fig. 4. Pictorial representation of the Tanner graph of a staircase LDPC code.

the m-th check-node, that is, the LLR of the probability that n-th bit-node is 1 or 0 based on all checks involving the n-th bit except the m-th check, and with $\Lambda_{m,n}(u_n)$ the log-likelihood of the message that the m-th check-node sends to the n-th bit-node, that is, the LLR of the probability that the n-th bit-node is 1 or 0 based on all the bit-nodes checked by the m-th check except the information coming from the n-th bit-node. With this setup, we have the following steps of the belief-propagation algorithm:

Initialization Step: each bit-node is assigned an a-posteriori LLR $L(u_j)$ as follows:

$$\begin{cases} \log\left(\frac{P(x_j=1|y_j)}{P(x_j=0|y_j)}\right) = (2y_j-1)\,\alpha^{(i)}, & j=1,\ldots,k \\ (2z_j-1), & j=k+1,\ldots,n \end{cases} \tag{7}$$

whereby,

$$\alpha^{(i)} = \log\left(\frac{p^{(i)}}{1-p^{(i)}}\right)$$

is the correction factor taking into account the estimated correlation between sequences X and Y at global iteration i. Note that this term derives from the correlation model adopted in this paper as expressed in (5), in which the correlation between any bit in the same position in the two sequences X and Y is seen as produced by an equivalent binary symmetric channel with transition probability p.

In our setup, before the first global iteration, $\alpha^{(0)} = 1$ since the degree of correlation is not known a priori.

In summary, for any position (m,n) such that $H_{m,n} = 1$, set:

$$\lambda_{n,m}(u_n) = L(u_n), \tag{8}$$

and

$$\Lambda_{m,n}(u_n) = 0. \tag{9}$$

(1) Check-node update: for each $m = 1,\ldots,n-k$, and for each $n \in B(m)$, compute:

$$\Lambda_{m,n}(u_n) = 2\tanh^{-1}\left(\prod_{p\in B(m)\backslash n} \tanh\left(\frac{\lambda_{p,m}(u_p)}{2}\right)\right) \tag{10}$$

(2) Bit-node update: for each $t = 1,\ldots,n$, and for each $m \in C(t)$, compute:

$$\lambda_{t,m}(u_t) = L(u_t) + \sum_{p\in C(t)\backslash m} \Lambda_{p,n}(u_t). \tag{11}$$

(3) Decision: for each bit node u_t with $t = 1,\ldots,n$, compute:

$$\lambda_t(u_t) = L(u_t) + \sum_{p\in C(t)} \Lambda_{p,n}(u_t), \tag{12}$$

and quantize the results such that $u_t = 0$ if $\lambda_t(u_t) < 0$, and $u_t = 1$ otherwise.

If $H \cdot U^T = \mathbf{0}$ then halt the algorithm and output $\widehat{x}_t = u_t,\ t = 1,\ldots,k$ as the estimate of the decompressed source bits, X, corresponding to the first source. Otherwise, if the number of iterations is less than a predefined maximum number, iterate the process starting from step (1).

The architecture of the iterative joint decoder is depicted in Fig. 3. Let us elaborate on the signal processing involved. In particular, as before let x and y be two correlated binary random variables which can take on the values $\{0,1\}$ and let $r = x \oplus y$. Let us assume that random variable r takes on the values $\{0,1\}$ with probabilities $P(r=1) = p_r$ and $P(r=0) = 1-p_r$.

The correction factor $\alpha^{(i)}$ at global iteration (i) is evaluated as follows,

$$\alpha^{(i)} = \log\left(\frac{p_{\hat{r}}}{1-p_{\hat{r}}}\right), \tag{13}$$

by counting the number of places in which $\widehat{X}^{(i)}$ and Y differ, or equivalently by evaluating the Hamming weight $w_H(.)$ of the sequence $\widehat{R}^{(i)} = \widehat{X}^{(i)} \oplus Y$ whereby, in the previous equation,

$$p_{\hat{r}} = \frac{w_H(\widehat{R}^{(i)})}{k}.$$

In the latter case, by assuming that the sequence $\widehat{R} = \widehat{X} \oplus Y$ is i.i.d., we have:

$$\alpha^{(i)} = \log\left(\frac{w_H(\widehat{R}^{(i)})}{k - w_H(\widehat{R}^{(i)})}\right) \tag{14}$$

where k is the source block size. Above, letters highlighted with $\widehat{.}$ are used to mean that the respective parameters have been estimated.

Formally, the iterative decoding algorithm can be stated as follows:

1) Set the log-likelihood ratios $\alpha^{(0)}$ to one (see Fig. 3). Compute the log-likelihood ratios for any bit node using (7).
2) For each global iteration $i = 1,\ldots,q$, do the following:
   a) perform a belief-propagation decoding on the parity bit sequence Z by using a predefined maximum number of local iterations, and the side information represented by the correlated sequence Y along with the correction factor $\alpha^{(i-1)}$;
   b) Evaluate $\alpha^{(i)}$ using (14);
   c) If $\left|\alpha^{(i)} - \alpha^{(i-1)}\right| \geq 10^{-4}$ go back to (a) and continue iterating, else exit.

Point c) in the previous code fragment is used in order to speed-up the overall iterative algorithm. Extensive tests we conducted suggested that the threshold value of 10^{-4} may be used for this purpose. Obviously, one can keep iterating until the last global iteration as well.

A. Overview of Integer-Metrics Belief-Propagation Decoder

In this section, we briefly describe the LDPC decoder working with integer LLRs. This approach leads to efficient belief-propagation decoding. Following the rationales described in [27], we begin by quantizing any real LLR (denoted LLRQ after quantization) employed in the initialization phase of the belief-propagation decoder in (7), using the following transformation:

$$LLRQ = \begin{cases} \lfloor 2^q L(u_j) + 0.5 \rfloor, & j = 1, \dots, k \\ \lfloor 2z_j - 1 \rfloor \cdot S, & j = k+1, \dots, n \end{cases} \quad (15)$$

whereby $\lfloor \cdot \rfloor$ stands for rounding to the smaller integer in the unit interval in which the real number falls, $L(u_j)$ is the real LLR, S is a suitable scaling factor, and q is the precision chosen to represent the LLR with integer metrics. In our belief-propagation decoder, we use $q = 3$, which guarantees a good trade-off between BER performance and complexity of the decoder implementation [27]. The scaling factor S is the greatest integer metric processed by the iterative decoder. In our set-up, we use $S = 10000$. Note that such a scaling factor depends on the practical implementations of the belief-propagation decoder. Suffice it to say that in our setup, S gives high likelihood to the parity bits z_j, $\forall j = k+1, \dots, n$, since they are transmitted through a perfect channel to the decoder.

All the summations shown in equations (11) and (12) are performed on integer LLRs. Finally, (10) has to be modified for working with integer metrics. To this end, we follow the schedule proposed in [29] in order to avoid real calculations and only use look-up tables and integer additions. This way, the basic operation $L(x \oplus y)$ in the check node update is approximated as follows:

$$L(x \oplus y) \approx \text{sign}(L(x)) \cdot \text{sign}(L(y)) \cdot \min(|L(x)|, |L(y)|) + \log\left(1 + e^{L(x)+L(y)}\right) - \log\left(e^{L(x)} + e^{L(y)}\right) \quad (16)$$

whereby, sign$(\cdot)$ is the function which evaluates the sign of the enclosed LLR. The latter two log functions are evaluated through look-up tables containing samples of the log functions approximated with piecewise linear approximation as suggested in [29].

IV. Sensitivity of the Cross-Correlation to Residual Errors after Channel Decoding

In this section, we wish to demonstrate the relative robustness of the empirical cross-correlation between sequences $\widehat{X}$ and Y to residual errors after channel decoding. To this end, let X and Y be two binary vectors of length k. Correlation between two binary streams can be measured in two ways. One technique is to map a logic-0 to the integer $+1$, logic-1 to -1 and to take the inner product of the two vectors defining our data packets and divide by the vector length (assumed to be the same for both packets). It is well known that the resulting empirical correlation estimate lies in the range $-1 \leq \rho \leq +1$. Positive value of ρ signifies a similarity between X and Y, while a negative ρ signifies a similarity between X and $\bar{Y}$, where $\bar{Y}$ is the complement of vector Y. This traditional measure of correlation is not of direct interest to us since we wish to relate a correlation measure to an empirical probability which is strictly positive.

To this end, let us define $r_n = x_n \oplus y_n$ as the XOR of the n-th component of the vectors X and Y. Similarly, we define $R = X \oplus Y$ whereby R is obtained via componentwise XOR of the components of the vectors X and Y. Let the number of places in which X and Y agree be γ and define the empirical cross-correlation between these two vectors as $\rho = \gamma/k$. Note that with this definition, we may take ρ as the empirical probability that $r_n = 0$ and a value of for instance $\rho = 0.7$ and $\rho = 0.3$ would have the same *information content* as measured by the value of the binary entropy function. Notice that, since intrinsic soft information is generated from probability estimates, this measure of correlation is more useful to us than the traditional definition.

Let us consider the Slepian-Wolf problem discussed above, and suppose that what is available at the receiver is a noisy version of X denoted $\widehat{X}$. For instance, $\widehat{X}$ could be an erroneous version of X obtained after transmission through a noisy channel modelled as a Binary Symmetric Channel (BSC) with transition probability p_e. We assume that the error events inflicting the sequence X are independent identically distributed (i.i.d.). The joint decoder generates an empirical estimate of the cross-correlation based on the use of the sequences $\widehat{X}$ and Y by forming the vector $\widehat{R} = \widehat{X} \oplus Y$ and counting the number of places where $\widehat{R}$ is zero. Let us denote this count as $\hat{\gamma}$. Clearly, $\hat{\gamma}$ is a random variable. The question is, what is the Probability Mass Function (PMF) of $\hat{\gamma}$? Knowledge of this PMF allows us to assess the sensitivity of our estimate of the cross-correlation to errors in the original sequence X.

It is relatively straightforward to express the probability that $(\hat{r}_n = r_n)$ and $(\hat{r}_n \neq r_n)$ as $P(\hat{r}_n = r_n) = P(\hat{x}_n = x_n) = 1 - p_e,\ P(\hat{r}_n \neq r_n) = P(\hat{x}_n \neq x_n) = p_e$. Consider applying a permutation to the sequences X and Y so that the permuted sequences agree in the *first* γ locations, and disagree in the remaining $(k-\gamma)$ locations. The permutation is applied to simplify the explanation of how we may go about obtaining the PMF of $\hat{\gamma}$ and by no means impacts the results. It is evident that the permuted sequence $\pi(R)$ contains γ zeros in the first γ locations and $(k-\gamma)$ ones in the remaining locations. Now consider evaluation of the probability $P(\hat{\gamma} = \gamma + \nu)$ for $\nu = 0, 1, .., (k-\gamma)$. We define $\pi(R)_\gamma$ to represent the first γ bits of $\pi(R)$ and $\pi(R)_{k-\gamma}$ the remaining $(k-\gamma)$ bits. Similarly we define $\pi(\widehat{R})_\gamma$ and $\pi(\widehat{R})_{k-\gamma}$. For a fixed ν, the event $\{\hat{\gamma} = \gamma + \nu\}$ corresponds to the union of the events of the type: $\pi(\widehat{R})_{k-\gamma}$ differs from $\pi(R)_{k-\gamma}$ in $(\nu + l)$ positions for some $l \in \{0, 1, .., \gamma\}$, $\pi(\widehat{R})_\gamma$ differs from $\pi(R)_\gamma$ in l positions, and the remaining bits of $\pi(\widehat{R})$ and $\pi(R)$ are identical. The probability of such elementary events are given by:

$$\binom{\gamma}{l}\binom{k-\gamma}{\nu+l}(1-p_e)^{k-\nu-2l}(p_e)^{\nu+2l}. \quad (17)$$

It can be shown that the probability of the event $\{\hat{\gamma} = \gamma + \nu\}$ for $\nu = 0, 1, .., (k-\gamma)$ is given by:

$$P(\hat{\gamma} \;=\; \gamma + \nu) = \sum_{l=0}^{\gamma} \left[\binom{\gamma}{l}\binom{k-\gamma}{\nu+l} \cdot \right.$$

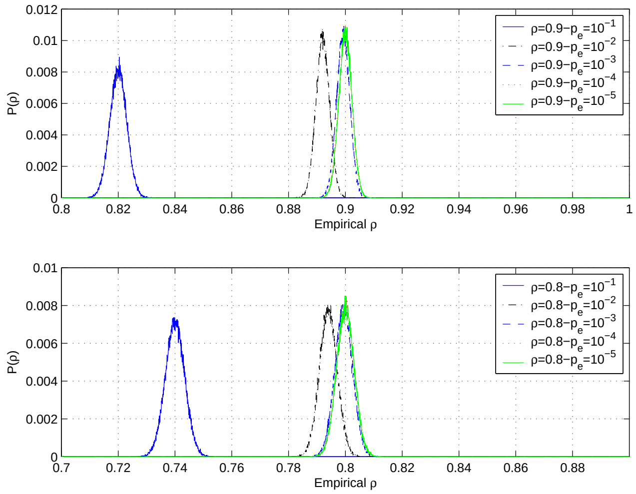

Fig. 5. Probability mass function (PMF) of $\hat{\rho}$ for three different values of raw error rate p when the true cross-correlations between the data packets are $\rho = 0.8$ and 0.9 at data block length of $k = 16400$.

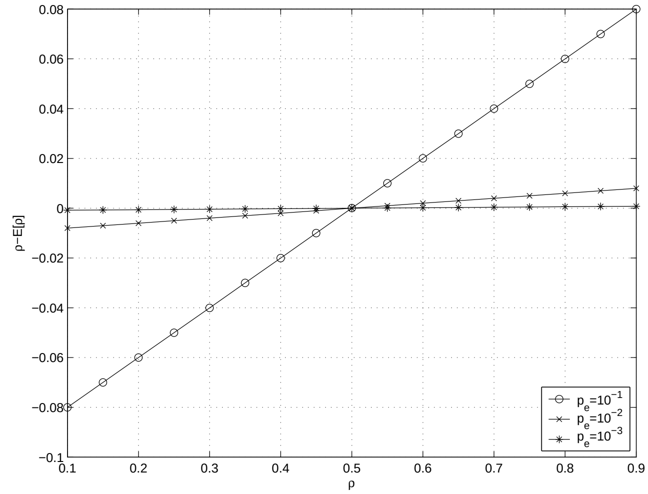

Fig. 6. The difference $(\rho - M(\hat{\rho}))$ versus ρ whereby $M(\hat{\rho})$ denotes the most probable value of $\hat{\rho}$ obtained from empirical evaluation of the cross-correlation from noisy received vectors (block length $k = 16400$).

$$\cdot (1-p_e)^{k-\nu-2l}(p_e)^{\nu+2l}\Big], \quad (18)$$

while, for $m = 1, 2, .., \gamma$ we have:

$$P(\hat{\gamma} \;=\; \gamma - m) = \sum_{l=m}^{\gamma}\left[\left(\begin{array}{c}\gamma\\ l\end{array}\right)\left(\begin{array}{c}k-\gamma\\ l-m\end{array}\right)\cdot\right.$$
$$\left.\cdot(1-p_e)^{k-2l+m}(p_e)^{2l-m}\right]. \quad (19)$$

In the next section, we shall present simulation results for iterative joint decoding of correlated sources for data packet size of $k = 16400$. Hence, let us look at representative results associated with the estimation of ρ for this block length:

1) Fig. 5 depicts the PMF of $\hat{\rho}$ for five different values of raw error rate p_e when the true cross-correlations between the data packets are $\rho = 0.8$ and 0.9 at data block length of $k = 16400$. The following observation is in order: there is a bias in the empirical estimate of ρ as measured by the most probable value of $\hat{\rho}$ and the true value of ρ. This bias is a strong function of p_e.
2) As noted above, the most likely value of $\hat{\rho}$, denoted $M(\hat{\rho})$ (i.e., the Mode), obtained from evaluation of the empirical cross-correlation from noisy received vectors is not necessarily the true value ρ. This is particularly so at larger values of p_e and for small and large values of ρ. Fig. 6 captures this behavior for three values of $p_e = 10^{-1}$, 10^{-2} and $p_e = 10^{-3}$ as a function of ρ for block length $k = 16400$. In particular, this figure shows the difference $(\rho - M(\hat{\rho}))$ versus ρ obtained from empirical evaluation of the cross-correlation from noisy received vectors.
3) The standard deviation of $\hat{\rho}$ is a strong function of p_e itself. Fig. 7 depicts the standard deviation of $\hat{\rho}$ as a function of p_e for $k = 16400$ for three different values of ρ. This figure shows that the standard deviation increases slowly with increasing p_e. Note that even at values of p_e as large as $p = 0.1$ this standard deviation is still relatively small for ρ in the range $\rho = 0.1$ to $\rho = 0.9$.

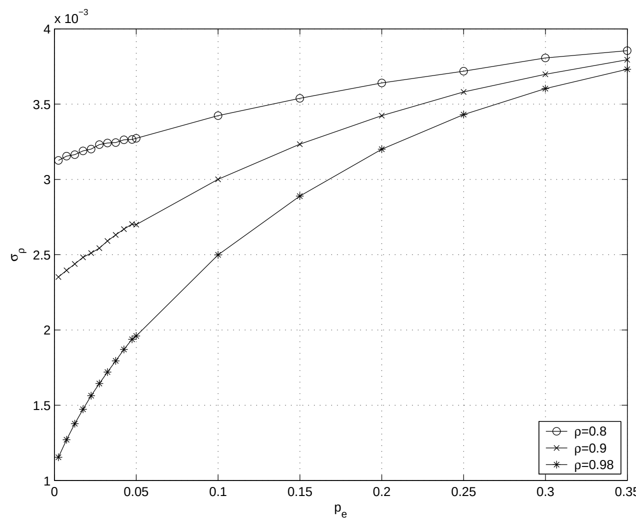

Fig. 7. Standard deviation of $\hat{\rho}$ as a function of p (block length $k = 16400$).

V. Simulation Results and Comparisons

We have simulated the performance of our proposed iterative joint source decoder. In order to compare our results with others proposed in the literature, we follow the same framework as in [10], [15], [16].

In the following, we provide sample simulation results associated with various (n, k) LDPC codes designed with the technique proposed in [28]. In particular, for a fair comparison with the results provided in [16], we designed various LDCP codes with $k = 16400$ and $k = 16000$. The details and the parameters of the designed LDPCs are given in Tables I-II.

Parameters given in Table I are the source block length k, the codeword length n, the rate R_1 of the source, expressed as $\frac{n-k}{k}$ (i.e., inverse of the compression ratio), the average degree d_v of the bit nodes, and the average degree d_c of the check nodes of the designed LDPCs. Note that, the encoding procedure adopted in our approach is different from the one proposed in [16] in that we source encode k bits at a time

TABLE I
PARAMETERS OF THE DESIGNED LDPCS.

LDPC	k	n	R_1	d_v	d_c
L_1	16400	30750	0.875	2.8	6
L_2	16400	26200	0.597	3	8
L_3	16400	22400	0.365	3.21	12
L_4	16000	20000	0.25	3.59	18
L_5	16400	20300	0.237	3.45	18
L_6	16400	20000	0.219	3.23	18
L_7	16400	19300	0.176	3.0	20
L_8	16400	19500	0.189	3.0	19

TABLE II
DEGREE DISTRIBUTIONS OF THE DESIGNED LDPCS.

$\lambda_{L_1}(x)$	$1 + 14349x + 15400x^2 + 200x^3 + 800x^{12}$
$\rho_{L_1}(x)$	$x^4 + 14349x^5$
$\lambda_{L_2}(x)$	$1 + 9799x + 11000x^2 + 4700x^3 + 700x^9$
$\rho_{L_2}(x)$	$x^6 + 9799x^7$
$\lambda_{L_3}(x)$	$1 + 5999x + 13700x^2 + 1800x^3 + 900x^{12}$
$\rho_{L_3}(x)$	$x^{10} + 5999x^{11}$
$\lambda_{L_4}(x)$	$1 + 3999x + 12600x^2 + 1300x^3 + 2100x^9$
$\rho_{L_4}(x)$	$x^{16} + 3999x^{17}$
$\lambda_{L_5}(x)$	$1 + 3899x + 12800x^2 + 2400x^3 + 1200x^{11}$
$\rho_{L_5}(x)$	$x^{16} + 3899x^{17}$
$\lambda_{L_6}(x)$	$1 + 3599x + 14400x^2 + 1200x^3 + 800x^{11}$
$\rho_{L_6}(x)$	$x^{16} + 3599x^{17}$
$\lambda_{L_7}(x)$	$1 + 2899x + 15400x^2 + 750x^3 + 250x^{11}$
$\rho_{L_7}(x)$	$x^{18} + 2899x^{19}$
$\lambda_{L_8}(x)$	$1 + 3099x + 15900x^2 + 500x^9$
$\rho_{L_8}(x)$	$x^{17} + 3099x^{18}$

and transmit only $n - k$ bits. In [16], the authors proposed a source compression which encodes n source bits at a time, and transmits $n - k$ syndrome bits.

The degree distributions of the designed LDPC codes, labelled L_1 to L_8 in Table I, are shown in Table II whereby, the integer number multiplying any power of the indeterminate x^w identifies the number of bit nodes in $\lambda_{L_r}(x)$, and check nodes in $\rho_{L_r}(x)$ having degree $w + 1$ in the r-th designed LDPC with $r = 1, \ldots, 8$.

For local decoding of the LDPC codes, the maximum number of local iterations has been set to 50, while the maximum number of global iterations is 5, even though the stopping criterion discussed in the previous section has been adopted.

The results on the compression achieved with the proposed algorithm are shown in Table III. The first row shows the correlation parameter assumed, that is, the probability $p = P(x_j \neq y_j)$, $\forall j = 1, \ldots, k$. The second row shows the joint entropy limit as expressed in (6) for various values of the correlation parameter p. The third and fourth rows show the results on source compression presented in papers [10], [16], while the last row presents the results on compression achieved with the proposed algorithm employing a maximum of 5 global iterations in conjunction with using the stopping criterion noted in the previous section. As in [16], we assume error free compression for a target Bit Error Rate (BER) 10^{-6}. Note that statistic of the results shown has been obtained by counting 30 erroneous frames.

From Table III it is evident that significant compression gains with respect to the theoretical limits can be achieved as the correlation between sequences X and Y increases. Notice that as opposed to the algorithm proposed in [16], we do not assume p is known at the decoder. Indeed, this parameter is iteratively estimated as discussed in the previous section. A similar approach has been pursued in [10], whereby an iterative approach is employed for the estimation of the correlation between the two correlated sequences, but employing turbo codes.

Finally, note that we employ integer soft-metrics as explained in the previous section, while in [10], [16], to the best our knowledge, the authors employ real metrics. The algorithm working on integer metrics is very fast and reduces considerably the complexity burden required by the two-stage iterative algorithm (i.e., the local-global combination).

Fig. 8 shows the BER performance of the proposed iterative decoding algorithm for various global iterations and as a function of the joint entropy between sources X and Y, when the stopping criterion for global iterations is applied and the maximum number of global iterations specified in the legend are used. The LDPC used for encoding is the one labelled L_8 in Tables I-II which guarantees a compression rate of $R_1 = 0.189$. At BER$= 10^{-6}$, the overall rate $R = R_1 + R_Y = 1.189$ is within 0.075 of the theoretical limit of 1.1175.

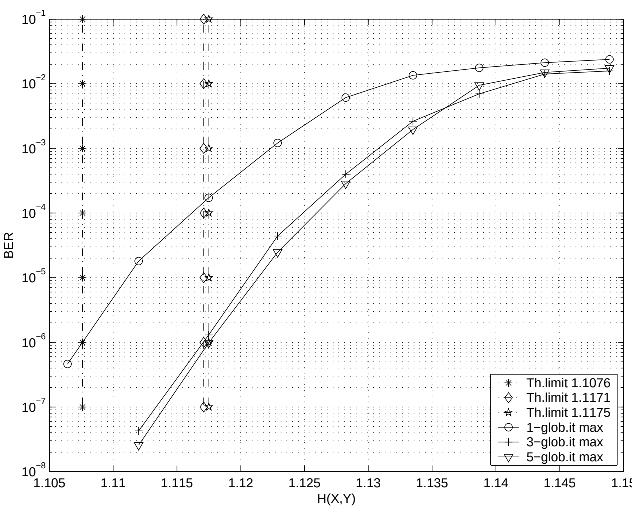

Fig. 8. BER performance of the proposed iterative decoding algorithm for various global iterations as a function of the joint entropy between sources X and Y, when the stopping criterion for global iterations is applied. Results refer to the LDPC labelled L_8 in Tables I-II, guaranteeing a compression rate of $R_1 = 0.189$. For a target BER$= 10^{-6}$, we also show the theoretical limit as expressed by the Slepian-Wolf bound achieved by the iterative algorithm applied to the LDPC L_8 for increasing values of the maximum number of global iterations.

In order to assess the effects of the global iterations on BER performances, we simulated the same LDPC L_8 without stopping criterion on the global iterations. The resulting BER performance are shown in Fig. 9. Notice that 3- or 4- global iterations suffice to get almost all that can be gained from the knowledge of the cross-correlation.

In Table IV, we show the average number of local iterations

TABLE III
COMPRESSION RATE PERFORMANCE OF THE ITERATIVE ALGORITHM FOR VARIOUS JOINT ENTROPIES.

p	0.01	0.02	0.025	0.02875	0.05	0.1	0.2
$H(p)+1$	1.08	1.141	1.169	1.188	1.286	1.469	1.722
R [10]	-	-	1.31	-	1.435	1.63	1.89
R [16]	-	-	1.276	-	1.402	1.60	1.875
$R = R_1 + R_Y$	1.176-L_7	1.219-L_6	1.237 -L_5	1.25-L_4	1.365-L_3	1.597-L_2	1.875-L_1

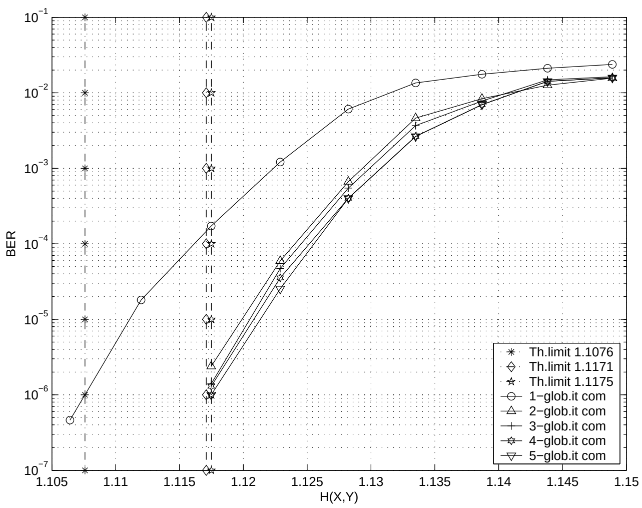

Fig. 9. BER performance of the proposed iterative decoding algorithm for various global iterations as a function of the joint entropy between sources X and Y, without stopping criterion on global iterations. Results refer to the LDPC labelled L_8 in Tables I-II, guaranteeing a compression rate of $R_1 = 0.189$. For a target BER= 10^{-6}, we also show the theoretical limit as expressed by the Slepian-Wolf bound achieved by the iterative algorithm applied to the LDPC L_8 for increasing values of the maximum number of global iterations.

TABLE IV
AVERAGE NUMBER OF LOCAL ITERATIONS FOR LDPC L_8 DECODING WITH MAXIMUM LOCAL ITERATIONS PARAMETER SET TO 50, FOR DIFFERENT GLOBAL ITERATION INDICES.

p	$H(X,Y)$	1-st it.	2-nd it.	3-rd it.	4-th it.
0.0213	1.1487	50.00	50.00	50.00	50.00
0.0204	1.1437	50.00	50.00	50.00	50.00
0.0195	1.1386	50.00	46.16	46.12	46.11
0.0186	1.1335	50.00	39.51	38.45	34.58
0.0177	1.1283	42.47	23.65	23.35	22.31
0.0168	1.1231	34.73	16.16	16.09	15.17
0.0159	1.1178	22.97	12.59	12.58	12.57

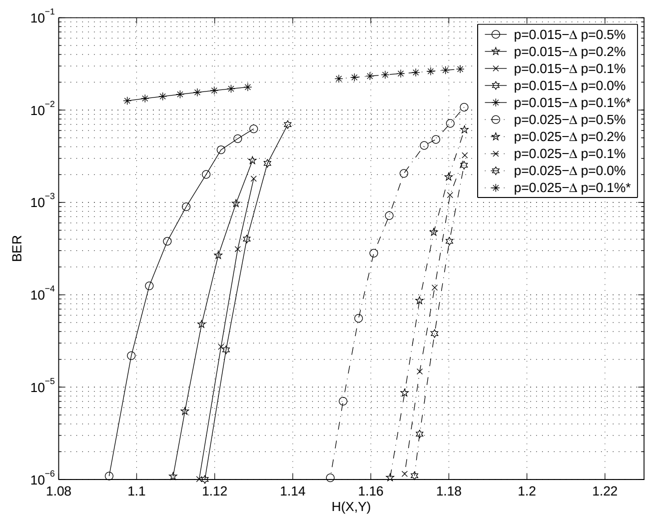

Fig. 10. BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations as a function of the joint entropy between sources X and Y, when the stopping criterion for global iterations is applied. Results refer to the LDPCs labelled L_8 and L_5 in Table I. The legend shows the mean correlation value p and the maximum value of the correlation variation with respect to the mean value. Curves labelled with $*$ refer to the ones obtained without the iterative paradigm, using the mean correlation value.

performed by the joint decoder at the end of a given global iteration, for various values of correlation between the sources and when the stopping criterion on the global iterations is not applied. Finally, we evaluated the average number of global iterations performed by the iterative algorithm when the stopping criterion on global iterations is employed during decoding. Simulation results show that when the LDPC decoder works at BER levels less than or equal to 10^{-5}, the average number of global iterations equals 1.2, thus guaranteeing a very efficient iterative approach to the co-decompression problem. In other words, an overall average number of 80 LDPC decoding iterations suffices to obtain good BER performance.

In the following set of simulation results, we adopted another approach in order to test the proposed algorithm for varying actual correlation levels. In particular, for any given value of mean correlation p, we generate a uniform random variable having mean value equal to the mean correlation itself and with a maximum variation of Δp around this mean value. We used the following maximum variations: $\Delta p = 0.5, 0.2, 0.1\%$, and $\Delta p = 0.0\%$ which refers to the case in which the correlation value is not variable, but fixed.

For each data block, we set the actual correlation equal to the mean correlation plus this perturbation. The decoder iterates to estimate the actual correlation value which varies around its mean value from one block to the next. In effect, the parameter p is iteratively estimated as discussed in the previous section. A similar approach has been pursued in [10] for fixed correlation level, whereby an iterative approach is used for the estimation of the correlation between the two correlated sequences, but employing turbo codes.

Fig. 10 shows the BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations and as a function of the joint entropy between sources X and Y, when the stopping criterion for global iterations is applied. LDPCs used for encoding are the one labelled L_5 and L_8 in

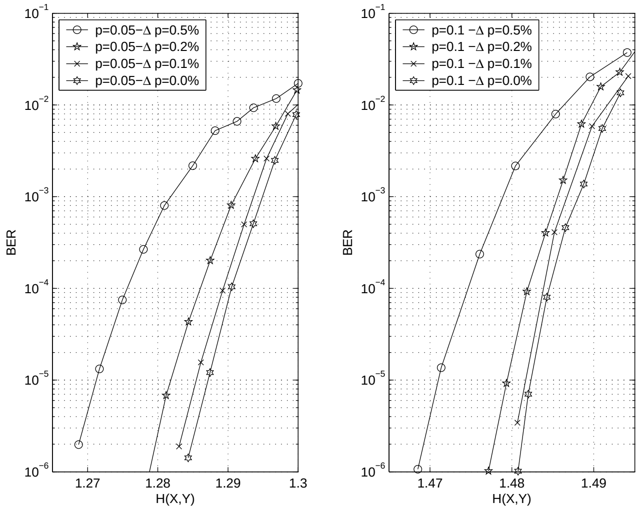

Fig. 11. BER performance of the proposed iterative decoding algorithm for a maximum of 5 global iterations as a function of the joint entropy between sources X and Y, when the stopping criterion for global iterations is applied. Results refer to the LDPCs labelled L_3 (left subplot) and L_2 (right subplot) in Table I. The legend shows the mean correlation value p and the maximum value of the correlation variation with respect to the mean value.

Table I which guarantee compression rates of $R_X = 0.237$ and $R_X = 0.189$, respectively. LDPC labelled L_5 is used at mean values of p equal to 0.025, while LDPC L_8 is adopted for a mean correlation of 0.015. From Fig. 10 one clearly sees that LDPC decoding does not converge when the decoder does not iterate for estimating the actual value of p, but uses only its mean value for setting the extrinsic information. Notice also that the performances of the iterative decoder when the correlation value is fixed (curves labelled $\Delta p = 0.0$ in Fig. 10), are very close to the case in which the actual correlation value varies within $\Delta p = 0.1\%$ from the mean value.

Similar considerations can be deduced from Fig. 11 which shows the BER performance of the proposed iterative decoding algorithm when using LDPCs labelled L_2 and L_3 in Table I which guarantee compression rates of $R_X = 0.597$ and $R_X = 0.365$, respectively. LDPC labelled L_2 is used at mean correlation equal to 0.1, while LDPC L_3 is used with a mean correlation of 0.05. Note that the performance degrades as Δp increases since the encoder works further away from its optimal operating point.

The results on the compression achieved with the proposed algorithm are shown in Table V for the case in which the correlation value is fixed. The first row shows the fixed correlation parameter assumed, namely, $p = P(x_j \neq y_j),\ \forall j = 1, \ldots, k$ in our model. The second row shows the joint entropy limit for various values of the fixed correlation parameter p. The third row presents the results on compression achieved with the proposed algorithm employing a maximum of 5 global iterations in conjunction with using the stopping criterion noted in the previous section. As in [16], we assume error free compression for a target Bit Error Rate (BER) 10^{-6}. Note that statistic of the results shown has been obtained by counting 30 erroneous frames.

From Table V it is evident that significant compression gains

TABLE V
COMPRESSION RATE PERFORMANCE OF THE ITERATIVE ALGORITHM FOR $H(p) = 0.112$.

p	0.015
$H(p) + 1$	1.112
$R = R_X + R_Y$	1.189 -L_8

with respect to the theoretical limits can be achieved as the correlation between sequences X and Y increases.

VI. Conclusion

In this paper we have presented a novel iterative joint decoding algorithm based on LDPC codes for the Slepian-Wolf problem of compression of correlated information sources.

In the considered scenario, two correlated sources transmit toward a common receiver. The first source is compressed by transmitting the parity check bits of a systematic LDPC encoded codeword, without the knowledge of the correlation among the sources. The correlated information of the second source is employed as side information at the receiver and used for decompressing and decoding of the first source. The correlation is not known a-priori, but is estimated through an iterative paradigm. Both the iterative decoding algorithm and the cross-correlation estimation procedure have been described in detail.

Simulation results suggest that relatively large compression gains are achievable at relatively small number of global iterations specially when the sources are highly correlated.

VII. ACKNOWLEDGEMENTS

This work was partially supported by the PRIN 2007 project *Feasibility study of an optical Earth-satellite quantum communication channel* and by the CAPANINA project (FP6-IST-2003-506745) as part of the EU VI Framework Programme.

References

[1] F. Daneshgaran, M. Laddomada, and M. Mondin, "LDPC-based iterative algorithm for compression of correlated sources at rates approaching the Slepian-Wolf bound," *In Proc. of the Int. Conf. SPACOMM 2009,* pp.74-79, Colmar, France, July 2009.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory,* vol.IT-19, no. 4, pp.471-480, July 1973.

[3] L. J. Guibas, "Sensing, tracking, and reasoning with relations," *IEEE Signal Processing Magazine,* no.3, pp.73-85, March 2002.

[4] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine,* no.3, pp.51-60, March 2002.

[5] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration," *IEEE Signal Processing Magazine,* no.3, pp.61-72, March 2002.

[6] A. Aaron and B. Girod, "Compression with side information using turbo-codes," *In Proc. of Data Compress. Conf., DCC'02,* pp.252-261, April 2002.

[7] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," *In Proc. of Global Telecomm. Conf., GLOBECOM'01,* vol.2, pp.1400-1404, Nov. 2001.

[8] J. Bajcsy and P. Mitran, "Coding for the Wyner-Ziv problem with turbo-like codes", *In Proc. of Inter. Symposium on Infor. Theory, ISIT'02,* pp.91, 2002.

[9] J. Bajcsy and I. Deslauriers, "Serial turbo coding for data compression and the Slepian-Wolf problem",*In Proc. of Inform. Theory Workshop,* pp.296-299, March 31-April 4 2003.

[10] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo-codes," *IEEE Communications Letters,* vol.5, no.10, pp.417-419, October 2001.

[11] Y. Zhao and J. Garcia-Frias, "Joint estimation and compression of correlated nonbinary sources using punctured turbo codes," *IEEE Transactions on Communications,* vol.53, no.3, pp.385-390, March 2005.

[12] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "A distributed source coding technique for correlated images using turbo-codes," *IEEE Communications Letters,* vol.6, no.9, pp.379-381, September 2002.

[13] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Joint source-channel coding of binary sources with side information at the decoder using IRA codes," *In Proc. of IEEE 2002 Multimedia Signal Processing Workshop,* December 2002.

[14] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," *In Proc. of IEEE Data Compression Conference,* 2003.

[15] 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, October 2002.

[16] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information using low-density parity-check codes," *In Proc. of IEEE GLOBECOM 2002,* vol. 2, pp.1300-1304, 17-21 Nov. 2002.

[17] J. Garcia-Frias, "Joint source-channel decoding of correlated sources over noisy channels," *In Proc. of DCC 2001,* pp.283-292, March 2001.

[18] J. Garcia-Frias, W. Zhong, and Y. Zhao, "Iterative decoding schemes for source and joint source-channel coding of correlated sources," *In Proc. of Asilomar Conference on Signals, Systems, and Computers,* November 2002.

[19] F. Daneshgaran, M. Laddomada, and M. Mondin, "Iterative joint channel decoding of correlated sources employing serially concatenated convolutional codes," *IEEE Transactions on Information Theory,* vol.51, no.7, pp.2721-2731, July 2005.

[20] F. Daneshgaran, M. Laddomada, and M. Mondin, "Iterative joint channel decoding of correlated sources," *IEEE Transactions on Wireless Communications,* Vol. 5, No. 10, pp.2659-2663, October 2006.

[21] F. Daneshgaran, M. Laddomada, and M. Mondin, "LDPC-based channel coding of correlated sources with iterative joint decoding," *IEEE Transactions on Communications,* Vol. 54, No. 4, pp.577-582, April 2006.

[22] J. Barros and S.D. Servetto, "The sensor reachback problem," *Submitted to IEEE Transactions on Information Theory*, available online at http://cn.ece.cornell.edu/publications/papers/20031112, November 2003.

[23] X. Zhu, L. Zhang, and Y. Liu, " A distributed joint source-channel coding scheme for multiple correlated sources," *Fourth International Conference on Digital Communications and Networking in China*, 2009, pp. 1 - 6.

[24] W. Yunnan, V. Stankovic, Z. Xiong, and S. Kung, " On practical design for joint distributed source and network coding," *IEEE Transactions on Information Theory*, Vol. 55, No. 4, pp. 1709 - 1720, April 2009.

[25] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. on Inform. Theory,* vol.42, no.2, pp.429-445, March 1996.

[26] T.J. Richardson and R.L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol.47, no.2, pp.638 - 656, Feb. 2001.

[27] G. Montorsi and S. Benedetto, "Design of fixed-point iterative decoders for concatenated codes with interleavers," *IEEE Journal on Selected Areas in Communications,* vol.19, No. 5, pp.871-882, May 2001.

[28] Xiao-Yu Hu, E. Eleftheriou, and D.M. Arnold, "Progressive edge-growth Tanner graphs," *IEEE Global Telecommunications Conference*, vol.2, pp.995-1001, 25-29 Nov. 2001.

[29] Xiao-Yu Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," *IEEE Global Telecommunications Conference*, vol.2, pp.25-29, 25-29 Nov. 2001.